Atmel Corporation
Microcontroller
Data Book
October 1995

is the registered trademark of Atmel Corporation
2125 O'Nel Drive, San Jose, CA 95131

Important Notice

Atmel guarantees that its circuits will be free from defects of material and workmanship under normal use and service, and that these circuits will perform to current specifications in accordance with, and subject to, the Company's standard warranty which is detailed in Atmel's Purchasing Order Acknowledgment.

Atmel reserves the right to change devices or specifications detailed in this data book at any time without notice, and assumes no responsibility for any errors within this document. Atmel does not make any commitment to update this information. Atmel assumes no responsibility for the use of any circuits described in this data book, nor does the Company assume responsibility for the functioning of undescribed features or parameters.

In the absence of a written agreement to the contrary, Atmel assumes no liability with respect to the use of semiconductor devices described in this data book for applications assistance, customers' product design or infringement of patents or copyrights of third parties.

Atmel's products are not authorized for use as critical components in life support devices or systems and the use as such implies that user bears all risk of such use.

If Atmel is an approved vendor on a Standard Military Drawing (SMD), the Atmel similar part number specification is compliant with the SMD.

Trademarks or registered trademarks used in this document may be the property of others.

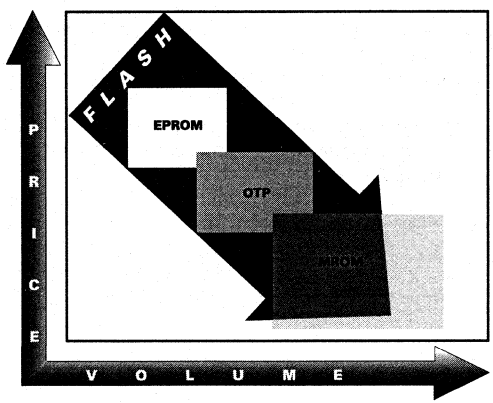
Printed on recycled paper.

Atmel's AT89 Series of Flash Microcontrollers

With the Flash memory-based microcontrollers from Atmel, you can achieve safe, easy reconfigurability in any 80C51-based product. With reconfigurability, you can make every product on your line exactly what your customers want.

Benefits

- Flash memory-
  change operating code in
  seconds and shorten development cycle
- 80C51 socket compatible-
  direct replacement (use existing code) with 40/44-pin devices
- Static state clock mode-
  saves power
- Stock just one part-
  make many model options
  in a flash, JIT
- Zero scrap due to misprogramming-
  every device is reprogrammable
- Accelerate final test-
  substitute test vectors for
  operating code in assembly
- Make changes remotely-
  increase customer satisfaction



Each device on the Flash microcontroller family consists of all the core features plus some additional features. A feature comparison of all the Atmel microcontrollers is shown in the table below.

The Atmel Family of Flash Microcontrollers

	AT89C51	AT89LV51	AT89C52	AT89LV52	AT89C2051	AT89C1051	AT89S8252
Bytes Flash Program Memory	4K	4K	8K	8K	2K	1K	8K
Bytes Data Memory	128 RAM	128 RAM	256 RAM	256 RAM	128 RAM	64 RAM	256 RAM
Bytes On Board EEPROM							2K EEPROM
I/O Pins	32	32	32	32	15	15	32
16-Bit Timer/Counters	2	2	3	3	2	1	3
UART	X	X	X	X	X		X
Interrupt Sources	6	6	8	8	6	3	9
Power Down and Idle Mode	X	X	X	X	X	X	X
Low Voltage Operation		X		X	X	X	X
Security Lock Bits	3	3	3	3	2	2	3
SPI Serial Interface							X
Watchdog Timer							X
Dual Data Pointer							X
Interupt Recovery from Power Down							X

Atmel Corporation designs, manufactures, and markets high quality and high performance CMOS memory, logic and analog integrated circuits. Founded in 1984, the Company serves the manufacturers of computation, communications and instrumentation equipment in commercial, industrial and military environments.

Atmel's broad line of products provide customers with a variety of solutions to their memory and logic applications. Atmel offers high-density, high-speed memory and logic standard products as well as custom gate arrays.

Atmel guarantees quality and reliability by fabricating all products— no matter what their intended application— to meet or exceed the specifications of Military Standard 883.

Whether you are new to programmable logic or an experienced user, Atmel is committed to your success. If you have any questions or would like to place an order, please contact your local Atmel sales office as listed in the back of this data book, or contact Atmel's corporate headquarters.

Atmel Corporation
2125 O'Nel Drive
San Jose, CA 95131
PHONE: (408) 441-0311
FAX: (408) 436-4300

FAX-ON-DEMAND:
U.S. (1-800) 29-ATMEL (292-8635)
International (1-408) 441-0732
Atmel BBS: (408) 436-4309

We thank you for considering Atmel semiconductors.

Table of Contents

Contents

Microcontroller

Part Number	Memory Size	Description	Availability
AT89C1051	1K x 8	2.7-Volt, 80C31 Microcontroller with 1 Kbyte Flash, 20-Pin Package	Now
AT89C2051	2K x 8	2.7-Volt, 80C31 Microcontroller with 2 Kbytes Flash, 20-Pin Package	Now
AT89C51	4K x 8	80C31 Microcontroller with 4 Kbytes Flash	Now
AT89LV51	4K x 8	2.7-Volt, 80C31 Microcontroller with 4 Kbytes Flash	Now
AT89C52	8K x 8	80C32 Microcontroller with 8 Kbytes Flash	Now
AT89LV52	8K x 8	2.7-Volt, 80C32 Microcontroller with 8 Kbytes Flash	Now
AT89S8252	8K x 8	80C32, Downloadable Microcontroller with 8 Kbytes Flash, 2K EEPROM	Now

Explanation of Atmel's Part Number Code

All Atmel part numbers begin with the prefix "AT". The next four to nine digits are the part number. In addition, Atmel parts can be ordered in particular speeds, in specific packages, for particular temperature ranges and with the option of 883C level B military compliance.

All Atmel Microcontrollers use 12 volt programming voltage if ordered as a standard part. However, the Atmel AT89C51 and AT89C52 can be special ordered as 5-volt programmable devices. If this option is desired the part must be ordered with a -5 at the end of the ordering code (AT89C5X-XXXX-5).

The available options for each part are listed at the back of its data sheet in its "Ordering Information" table. These options are designated by the following suffixes placed at the end of the Atmel part number, in the order given:

Prefix AT

Device 89CXXX

Suffix X X X X

Processing
- Blank = Standard
- /883 = MIL-STD-883, Class B Fully Compliant

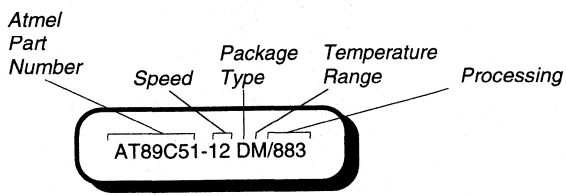
Temperature Range
- C = Commercial
- I = Industrial
- A = Automotive
- M = Military

Package
- D = Cerdip
- J = Plastic J-Lead Chip Carrier
- L = Leadless Chip Carrier
- P = Plastic DIP
- S = SOIC
- Q = PQFP
- A = TQFP
- W = Die

Speed
- 12 = 12 MHz
- 16 = 16 MHz
- 20 = 20 MHz
- 24 = 24 MHz

9 = FLASH

Here is an example Atmel part number:



0512A

Product Index

Part Number		Description					
MCUs	Pins	Package	VCC	Speed	Temperature	Flash	Page
AT89C1051-12PC	20	PDIP	3 V	12 MHz	Commercial	1 K	3-3
AT89C1051-12SC	20	SOIC	3 V	12 MHz	Commercial	1 K	3-3
AT89C1051-12PI	20	PDIP	3 V	12 MHz	Industrial	1 K	3-3
AT89C1051-12SI	20	SOIC	3 V	12 MHz	Industrial	1 K	3-3
AT89C1051-24PC	20	PDIP	5 V	24 MHz	Commercial	1 K	3-3
AT89C1051-24SC	20	SOIC	5 V	24 MHz	Commercial	1 K	3-3
AT89C1051-24PI	20	PDIP	5 V	24 MHz	Industrial	1 K	3-3
AT89C1051-24SI	20	SOIC	5 V	24 MHz	Industrial	1 K	3-3
AT89C2051-12PC	20	PDIP	3 V	12 MHz	Commercial	2 K	3-17
AT89C2051-12SC	20	SOIC	3 V	12 MHz	Commercial	2 K	3-17
AT89C2051-12PI	20	PDIP	3 V	12 MHz	Industrial	2 K	3-17
AT89C2051-12SI	20	SOIC	3 V	12 MHz	Industrial	2 K	3-17
AT89C2051-24PC	20	PDIP	5 V	24 MHz	Commercial	2 K	3-17
AT89C2051-24SC	20	SOIC	5 V	24 MHz	Commercial	2 K	3-17
AT89C2051-24PI	20	PDIP	5 V	24 MHz	Industrial	2 K	3-17
AT89C2051-24SI	20	SOIC	5 V	24 MHz	Industrial	2 K	3-17
AT89C51-12AC	44	TQFP	5 V	12 MHz	Commercial	4 K	3-33
AT89C51-12JC	44	PLCC	5 V	12 MHz	Commercial	4 K	3-33
AT89C51-12PC	40	PDIP	5 V	12 MHz	Commercial	4 K	3-33
AT89C51-12QC	44	PQFP	5 V	12 MHz	Commercial	4 K	3-33
AT89C51-12AI	44	TQFP	5 V	12 MHz	Industrial	4 K	3-33
AT89C51-12JI	44	PLCC	5 V	12 MHz	Industrial	4 K	3-33
AT89C51-12PI	40	PDIP	5 V	12 MHz	Industrial	4 K	3-33
AT89C51-12QI	44	PQFP	5 V	12 MHz	Industrial	4 K	3-33
AT89C51-12AA	44	TQFP	5 V	12 MHz	Automotive	4 K	3-33
AT89C51-12JA	44	PLCC	5 V	12 MHz	Automotive	4 K	3-33
AT89C51-12PA	40	PDIP	5 V	12 MHz	Automotive	4 K	3-33
AT89C51-12QA	44	PQFP	5 V	12 MHz	Automotive	4 K	3-33
AT89C51-12DM	40	CERDIP	5 V	12 MHz	Military	4 K	3-33
AT89C51-12LM	44	LCC	5 V	12 MHz	Military	4 K	3-33
AT89C51-12DM/883	40	CERDIP	5 V	12 MHz	Military	4 K	3-33
AT89C51-12LM/883	44	LCC	5 V	12 MHz	Military	4 K	3-33
AT89C51-16AC	44	TQFP	5 V	16 MHz	Commercial	4 K	3-33
AT89C51-16JC	44	PLCC	5 V	16 MHz	Commercial	4 K	3-33
AT89C51-16PC	40	PDIP	5 V	16 MHz	Commercial	4 K	3-33
AT89C51-16QC	44	PQFP	5 V	16 MHz	Commercial	4 K	3-33
AT89C51-16AI	44	TQFP	5 V	16 MHz	Industrial	4 K	3-33
AT89C51-16JI	44	PLCC	5 V	16 MHz	Industrial	4 K	3-33
AT89C51-16PI	40	PDIP	5 V	16 MHz	Industrial	4 K	3-33
AT89C51-16QI	44	PQFP	5 V	16 MHz	Industrial	4 K	3-33
AT89C51-16AA	44	TQFP	5 V	16 MHz	Automotive	4 K	3-33
AT89C51-16JA	44	PLCC	5 V	16 MHz	Automotive	4 K	3-33
AT89C51-16PA	40	PDIP	5 V	16 MHz	Automotive	4 K	3-33
AT89C51-16QA	44	PQFP	5 V	16 MHz	Automotive	4 K	3-33
AT89C51-20AC	44	TQFP	5 V	20 MHz	Commercial	4 K	3-33
AT89C51-20JC	44	PLCC	5 V	20 MHz	Commercial	4 K	3-33
AT89C51-20PC	40	PDIP	5 V	20 MHz	Commercial	4 K	3-33
AT89C51-20QC	44	PQFP	5 V	20 MHz	Commercial	4 K	3-33

Product Index (continued)

Part Number	Description						
MCUs	Pins	Package	VCC	Speed	Temperature	Flash	Page
AT89C51-20AI	44	TQFP	5 V	20 MHz	Industrial	4 K	3-33
AT89C51-20JI	44	PLCC	5 V	20 MHz	Industrial	4 K	3-33
AT89C51-20PI	40	PDIP	5 V	20 MHz	Industrial	4 K	3-33
AT89C51-20QI	44	PQFP	5 V	20 MHz	Industrial	4 K	3-33
AT89C51-24AC	44	TQFP	5 V	24 MHz	Commercial	4 K	3-33
AT89C51-24JC	44	PLCC	5 V	24 MHz	Commercial	4 K	3-33
AT89C51-24PC	40	PDIP	5 V	24 MHz	Commercial	4 K	3-33
AT89C51-24QC	44	PQFP	5 V	24 MHz	Commercial	4 K	3-33
AT89C51-24AI	44	TQFP	5 V	24 MHz	Industrial	4 K	3-33
AT89C51-24JI	44	PLCC	5 V	24 MHz	Industrial	4 K	3-33
AT89C51-24PI	40	PDIP	5 V	24 MHz	Industrial	4 K	3-33
AT89C51-24QI	44	PQFP	5 V	24 MHz	Industrial	4 K	3-33
AT89LV51-12AC	44	TQFP	3 V	12 MHz	Commercial	4 K	3-49
AT89LV51-12JC	44	PLCC	3 V	12 MHz	Commercial	4 K	3-49
AT89LV51-12PC	40	PDIP	3 V	12 MHz	Commercial	4 K	3-49
AT89LV51-12QC	44	PQFP	3 V	12 MHz	Commercial	4 K	3-49
AT89C52-12AC	44	TQFP	5 V	12 MHz	Commercial	8 K	3-65
AT89C52-12JC	44	PLCC	5 V	12 MHz	Commercial	8 K	3-65
AT89C52-12PC	40	PDIP	5 V	12 MHz	Commercial	8 K	3-65
AT89C52-12QC	44	PQFP	5 V	12 MHz	Commercial	8 K	3-65
AT89C52-12AI	44	TQFP	5 V	12 MHz	Industrial	8 K	3-65
AT89C52-12JI	44	PLCC	5 V	12 MHz	Industrial	8 K	3-65
AT89C52-12PI	40	PDIP	5 V	12 MHz	Industrial	8 K	3-65
AT89C52-12QI	44	PQFP	5 V	12 MHz	Industrial	8 K	3-65
AT89C52-12AA	44	TQFP	5 V	12 MHz	Automotive	8 K	3-65
AT89C52-12JA	44	PLCC	5 V	12 MHz	Automotive	8 K	3-65
AT89C52-12PA	40	PDIP	5 V	12 MHz	Automotive	8 K	3-65
AT89C52-12QA	44	PQFP	5 V	12 MHz	Automotive	8 K	3-65
AT89C52-12DM	40	CERDIP	5 V	12 MHz	Military	8 K	3-65
AT89C52-12LM	44	LCC	5 V	12 MHz	Military	8 K	3-65
AT89C52-12DM/883	40	CERDIP	5 V	12 MHz	Military	8 K	3-65
AT89C52-12LM/883	44	LCC	5 V	12 MHz	Military	8 K	3-65
AT89C52-16AC	44	TQFP	5 V	16 MHz	Commercial	8 K	3-65
AT89C52-16JC	44	PLCC	5 V	16 MHz	Commercial	8 K	3-65
AT89C52-16PC	40	PDIP	5 V	16 MHz	Commercial	8 K	3-65
AT89C52-16QC	44	PQFP	5 V	16 MHz	Commercial	8 K	3-65
AT89C52-16AI	44	TQFP	5 V	16 MHz	Industrial	8 K	3-65
AT89C52-16JI	44	PLCC	5 V	16 MHz	Industrial	8 K	3-65
AT89C52-16PI	40	PDIP	5 V	16 MHz	Industrial	8 K	3-65
AT89C52-16QI	44	PQFP	5 V	16 MHz	Industrial	8 K	3-65
AT89C52-16AA	44	TQFP	5 V	16 MHz	Automotive	8 K	3-65
AT89C52-16JA	44	PLCC	5 V	16 MHz	Automotive	8 K	3-65
AT89C52-16PA	40	PDIP	5 V	16 MHz	Automotive	8 K	3-65
AT89C52-16QA	44	PQFP	5 V	16 MHz	Automotive	8 K	3-65
AT89C52-20AC	44	TQFP	5 V	20 MHz	Commercial	8 K	3-65
AT89C52-20JC	44	PLCC	5 V	20 MHz	Commercial	8 K	3-65
AT89C52-20PC	40	PDIP	5 V	20 MHz	Commercial	8 K	3-65
AT89C52-20QC	44	PQFP	5 V	20 MHz	Commercial	8 K	3-65

Product Index (continued)

Part Number	Description						
MCUs	Pins	Package	VCC	Speed	Temperature	Flash	Page
AT89C52-20AI	44	TQFP	5 V	20 MHz	Industrial	8 K	3-65
AT89C52-20JI	44	PLCC	5 V	20 MHz	Industrial	8 K	3-65
AT89C52-20PI	40	PDIP	5 V	20 MHz	Industrial	8 K	3-65
AT89C52-20QI	44	PQFP	5 V	20 MHz	Industrial	8 K	3-65
AT89C52-24AC	44	TQFP	5 V	24 MHz	Commercial	8 K	3-65
AT89C52-24JC	44	PLCC	5 V	24 MHz	Commercial	8 K	3-65
AT89C52-24PC	40	PDIP	5 V	24 MHz	Commercial	8 K	3-65
AT89C52-24QC	44	PQFP	5 V	24 MHz	Commercial	8 K	3-65
AT89C52-24AI	44	TQFP	5 V	24 MHz	Industrial	8 K	3-65
AT89C52-24JI	44	PLCC	5 V	24 MHz	Industrial	8 K	3-65
AT89C52-24PI	40	PDIP	5 V	24 MHz	Industrial	8 K	3-65
AT89C52-24QI	44	PQFP	5 V	24 MHz	Industrial	8 K	3-65
AT89LV52-12AC	44	TQFP	3 V	12 MHz	Commercial	8 K	3-87
AT89LV52-12JC	44	PLCC	3 V	12 MHz	Commercial	8 K	3-87
AT89LV52-12PC	40	PDIP	3 V	12 MHz	Commercial	8 K	3-87
AT89LV52-12QC	44	PQFP	3 V	12 MHz	Commercial	8 K	3-87
AT89S8252-12AC	44	TQFP	3 V	12 MHz	Commercial	8 K	3-109
AT89S8252-12JC	44	PLCC	3 V	12 MHz	Commercial	8 K	3-109
AT89S8252-12PC	40	PDIP	3 V	12 MHz	Commercial	8 K	3-109
AT89S8252-12QC	44	PQFP	3 V	12 MHz	Commercial	8 K	3-109
AT89S8252-12AI	44	TQFP	3 V	12 MHz	Industrial	8 K	3-109
AT89S8252-12JI	44	PLCC	3 V	12 MHz	Industrial	8 K	3-109
AT89S8252-12PI	40	PDIP	3 V	12 MHz	Industrial	8 K	3-109
AT89S8252-12QI	44	PQFP	3 V	12 MHz	Industrial	8 K	3-109
AT89S8252-24AC	44	TQFP	5 V	24 MHz	Commercial	8 K	3-109
AT89S8252-24JC	44	PLCC	5 V	24 MHz	Commercial	8 K	3-109
AT89S8252-24PC	40	PDIP	5 V	24 MHz	Commercial	8 K	3-109
AT89S8252-24QC	44	PQFP	5 V	24 MHz	Commercial	8 K	3-109
AT89S8252-24AI	44	TQFP	5 V	24 MHz	Industrial	8 K	3-109
AT89S8252-24JI	44	PLCC	5 V	24 MHz	Industrial	8 K	3-109
AT89S8252-24PI	40	PDIP	5 V	24 MHz	Industrial	8 K	3-109
AT89S8252-24QI	44	PQFP	5 V	24 MHz	Industrial	8 K	3-109

Contents

2

Flash Microcontroller Architectural Overview

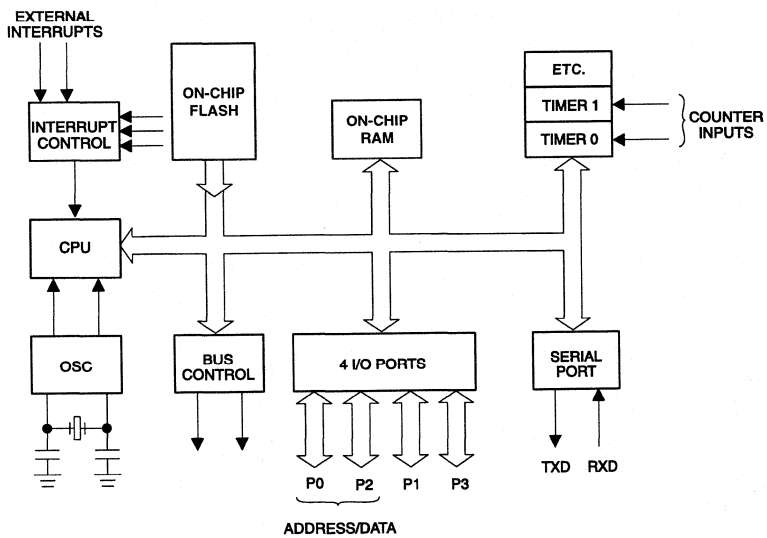
Features

- 8-Bit CPU Optimized for Control Applications
- Extensive Boolean Processing Capabilities (Single-Bit Logic)
- On-Chip Flash Program Memory
- On-Chip Data RAM
- Bidirectional and Individually Addressable I/O Lines
- Multiple 16-Bit Timer/Counters
- Full Duplex UART
- Multiple Source/Vector/Priority Interrupt Structure
- On-Chip Clock Oscillator
- On-chip EEPROM (AT89S series)
- SPI Serial Bus Interface (AT89S Series)
- Watchdog Timer (AT89S Series)

The basic architectural structure of this AT89C51 core is shown in Figure 1.

Block Diagram

Figure 1. Block Diagram of the AT89C core



For more information on the individual devices and features, refer to the Hardware Descriptions and Data Sheets of the specific device.

Figure 2. Block Diagram of the AT89S core

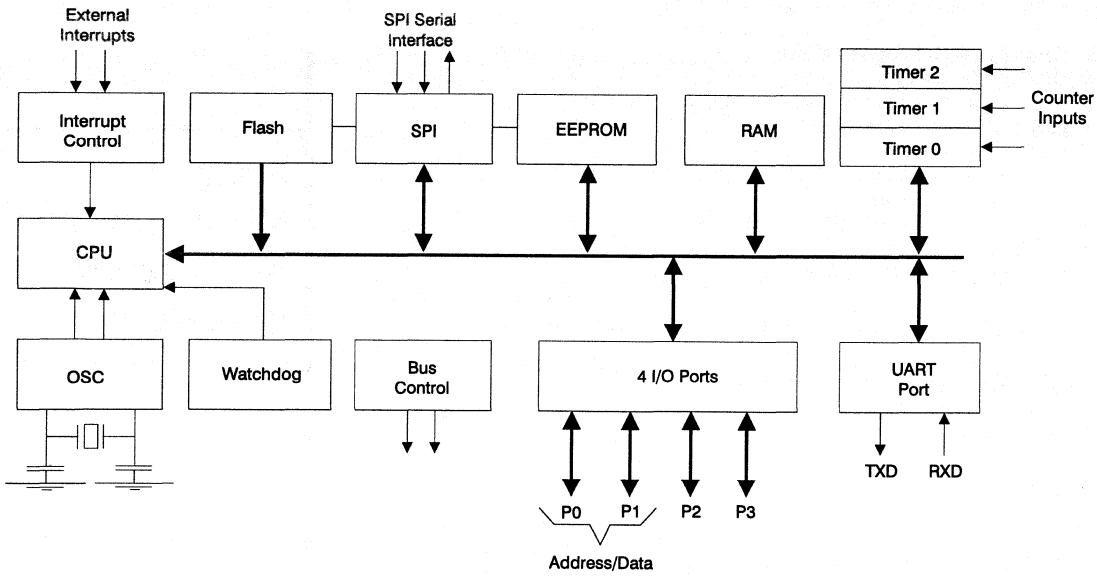
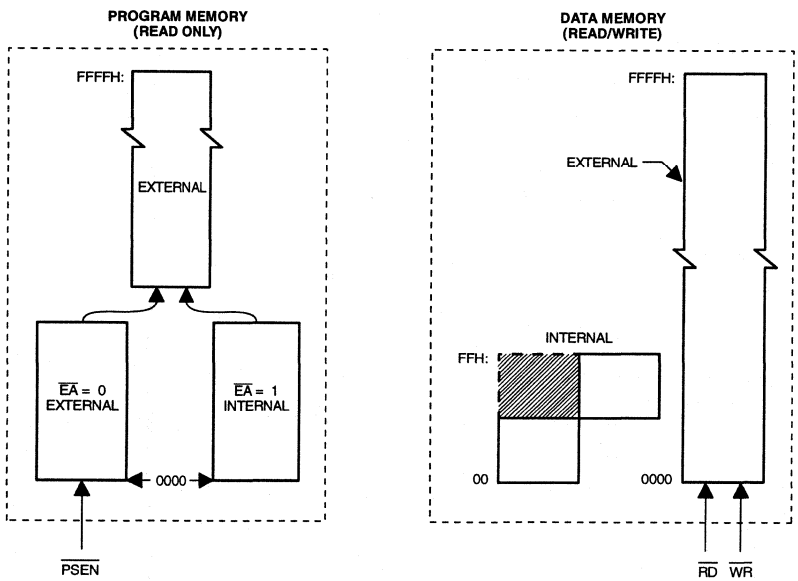


Figure 3. AT89C51/LV51 and AT89C52/LV52 Memory Structure



Reduced Power Modes

To exploit the power savings available in CMOS circuitry, Atmel's Flash microcontrollers have two software-invoked reduced power modes.

- Idle Mode. The CPU is turned off while the RAM and other on-chip peripherals continue operating. In this mode, current draw is reduced to about 15 percent of the current drawn when the device is fully active.
- Power Down Mode. All on-chip activities are suspended, while the on-chip RAM continues to hold its data. In this mode, the device typically draws less than 15 µA, and can be as low as 0.6 µA.

In addition, these devices are designed using static logic, which does not require continuous clocking. That is, the clock frequency can be slowed or even stopped while waiting for an internal event.

Memory Organization

Logical Separation of Program and Data Memory

All Atmel Flash microcontrollers have separate address spaces for program and data memory, as shown in Figure 3. The logical separation of program and data memory allows the data memory to be accessed by 8-bit addresses, which can be more quickly stored and manipulated by an 8-bit CPU. Nevertheless, 16-bit data memory addresses can also be generated through the DPTR register.

Program memory can only be read. There can be up to 64K bytes of directly addressable program memory. The read strobe for external program memory is the Program Store Enable signal ($\overline{PSEN}$).

Data memory occupies a separate address space from program memory. Up to 64K bytes of external memory can be directly addressed in the external data memory space. The CPU generates read and write signals, $\overline{RD}$ and $\overline{WR}$, during external data memory accesses.

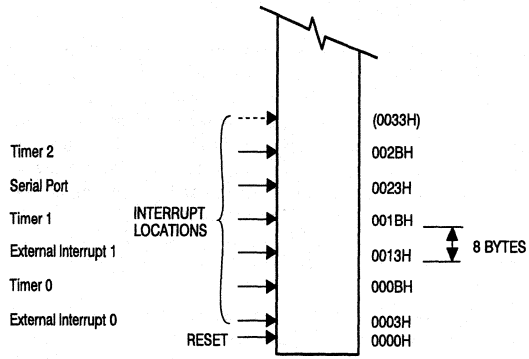
External program memory and external data memory can be combined by applying the $\overline{RD}$ and $\overline{PSEN}$ signals to the inputs of an AND gate and using the output of the gate as the read strobe to the external program/data memory.

Program Memory

Figure 4 shows a map of the lower part of the program memory. After reset, the CPU begins execution from location 0000H.

As shown in Figure 4, each interrupt is assigned a fixed location in program memory. The interrupt causes the CPU to jump to that location, where it executes the service routine. External Interrupt 0, for example, is assigned to location 0003H. If External Interrupt 0 is used, its service routine must begin at location 0003H. If the interrupt is not used, its service location is available as general purpose program memory.

Figure 4. Program Memory



The interrupt service locations are spaced at 8-byte intervals: 0003H for External Interrupt 0, 000BH for Timer 0, 0013H for External Interrupt 1, 001BH for Timer 1, and so on. If an interrupt service routine is short enough (as is often the case in control applications), it can reside entirely within that 8-byte interval. Longer service routines can use a jump instruction to skip over subsequent interrupt locations, if other interrupts are in use.

The lowest addresses of program memory can be either in the on-chip Flash or in an external memory. To make this selection, strap the External Access ($\overline{EA}$)pin to either V_{CC} or GND.

For example, in the AT89C51 with 4K bytes of on-chip Flash, if the $\overline{EA}$ pin is strapped to V_{CC}, program fetches to addresses 0000H through 0FFFH are directed to the internal Flash. Program fetches to addresses 1000H through FFFFH are directed to external memory.

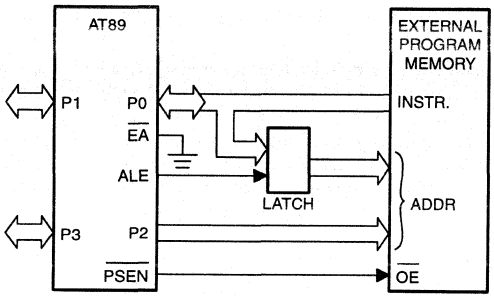
In the AT89C52 (8K bytes Flash), $\overline{EA} = V_{CC}$ selects addresses 0000H through 1FFFH to be internal and addresses 2000H through FFFFH to be external.

If the $\overline{EA}$ pin is strapped to GND, all program fetches are directed to external memory.

The read strobe to external memory, $\overline{PSEN}$, is used for all external program fetches. Internal program fetches do not activate $\overline{PSEN}$.

The hardware configuration for external program execution is shown in Figure 5. Note that 16 I/O lines (Ports 0 and 2) are

Figure 5. Executing from External Program Memory



dedicated to bus functions during external program memory fetches. Port 0 (P0 in Figure 5) serves as a multiplexed address/data bus. It emits the low byte of the Program Counter (PCL) as an address and then goes into a float state while waiting for the arrival of the code byte from the program memory. During the time that the low byte of the Program Counter is valid on P0, the signal ALE (Address Latch Enable) clocks this byte into an address latch. Meanwhile, Port 2 (P2 in Figure 5) emits the high byte of the Program Counter (PCH). Then $\overline{PSEN}$ strobes the external memory, and the microcontroller reads the code byte.

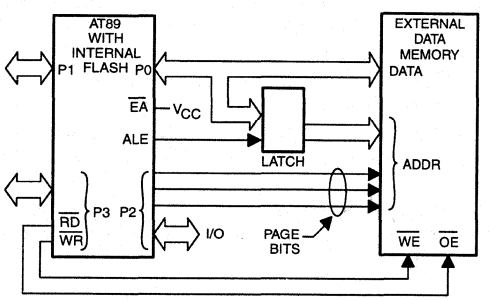
Program memory addresses are always 16 bits wide, even though the actual amount of program memory used may be less than 64K bytes. External program execution sacrifices two of the 8-bit ports, P0 and P2, to the function of addressing the program memory.

Data Memory

The right half of Figure 3 shows the internal and external data memory spaces available on Atmel's Flash microcontrollers.

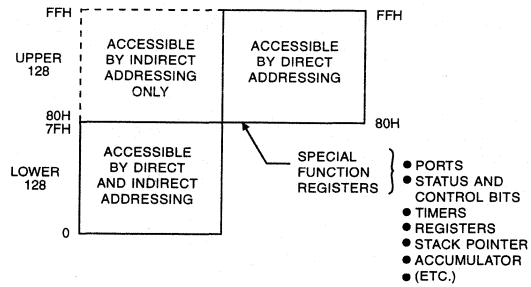
Figure 6 shows a hardware configuration for accessing up to 2K bytes of external RAM. In this case, the CPU executes from internal Flash. Port 0 serves as a multiplexed address/data bus to the RAM, and 3 lines of Port 2 are used to page the RAM. The CPU generates $\overline{RD}$ and $\overline{WR}$ signals as needed during external RAM accesses.

Figure 6. Accessing external data memory. If the program memory is internal, the other bits of P2 are available as I/O.



You can assign up to 64K bytes of external data memory. External data memory addresses can be either 1 or 2 bytes wide. One-byte addresses are often used in conjunction with one or more other I/O lines to page the RAM, as shown in Figure 6. Two-byte addresses can also be used, in which case the high address byte is emitted at Port 2.

Figure 7. Internal Data Memory



Internal data memory is shown in Figure 7. The memory space is divided into three blocks, which are generally referred to as the Lower 128, the Upper 128, and SFR space.

Internal data memory addresses are always 1 byte wide, which implies an address space of only 256 bytes. However, the addressing modes for internal RAM can in fact accommodate 384 bytes. Direct addresses higher than 7FH access one memory space, and indirect addresses higher than 7FH access a different memory space. Thus, Figure 7 shows the Upper 128 and SFR space occupying the same block of addresses, 80H through FFH, although they are physically separate entities.

Figure 8. The Lower 128 Bytes of Internal RAM

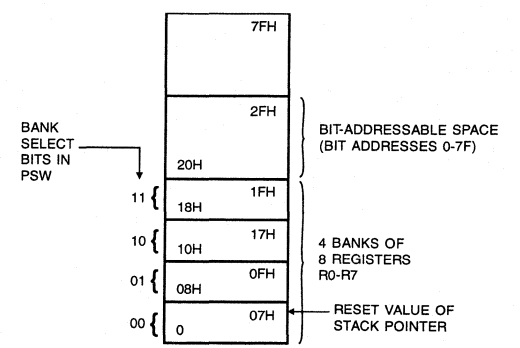


Figure 8 shows how the lower 128 bytes of RAM are mapped. The lowest 32 bytes are grouped into 4 banks of 8 registers. Program instructions call out these registers as R0 through R7. Two bits in the Program Status Word (PSW) select which register bank is in use. This architecture allows more efficient use of code space, since register instructions are shorter than instructions that use direct addressing.

The next 16 bytes above the register banks form a block of bit-addressable memory space. The microcontroller instruction set includes a wide selection of single-bit instructions, and these instructions can directly address the 128 bits in this area. These bit addresses are 00H through 7FH.

All of the bytes in the Lower 128 can be accessed by either direct or indirect addressing. The Upper 128 (Figure 8) can only be accessed by indirect addressing. The Upper 128 bytes of RAM are only in the devices with 256 bytes of RAM.

Figure 10 gives a brief look at the Special Function Register (SFR) space. SFRs include Port latches, timers, peripheral controls, etc. These registers can only be accessed by direct addressing. In general, all Atmel microcontrollers have the same SFRs at the same addresses in SFR space as the AT89C51 and other compatible microcontrollers. However, upgrades to the AT89C51 have additional SFRs.

Sixteen addresses in SFR space are both byte- and bit-addressable. The bit-addressable SFRs are those whose address ends in 000B. The bit addresses in this area are 80H through FFH.

Figure 9. The Upper 128 Bytes of Internal RAM

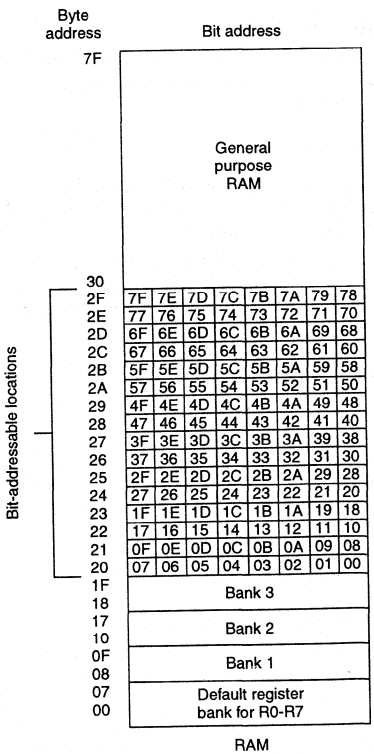


Figure 10. SFR Space

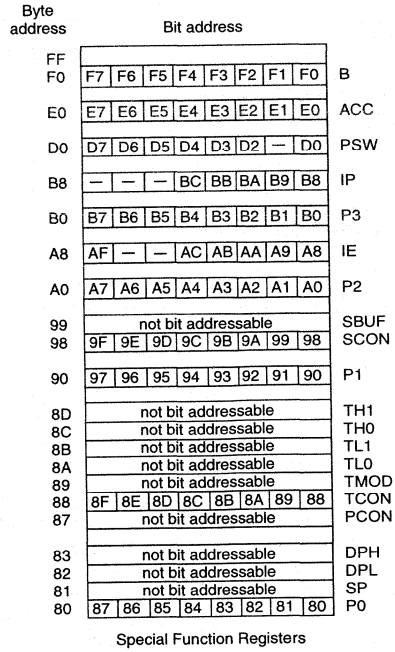
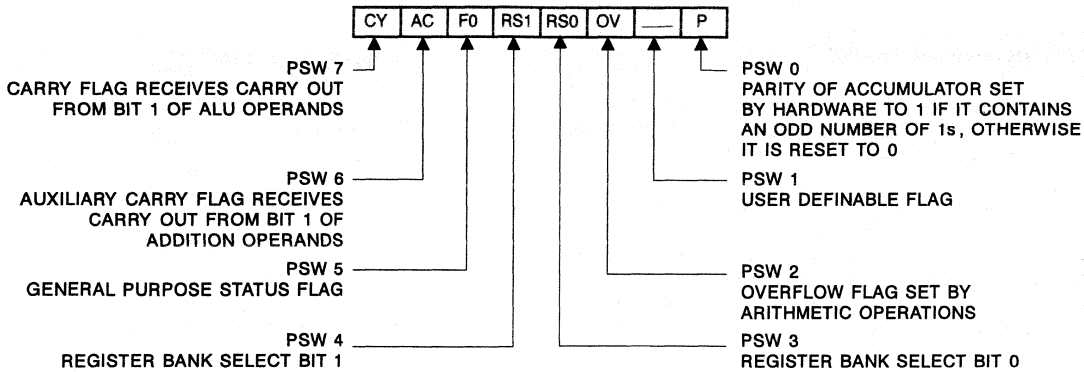


Figure 11. PSW (Program Status Word) Register in Atmel Flash Microcontrollers



The Instruction Set

All members of the Atmel microcontroller family execute the same instruction set. This instruction set is optimized for 8-bit control applications and it provides a variety of fast addressing modes for accessing the internal RAM to facilitate byte operations on small data structures. The instruction set provides extensive support for 1-bit variables as a separate data type, allowing direct bit manipulation in control and logic systems that require Boolean processing.

The following overview of the instruction set gives a brief description of how certain instructions can be used.

Program Status Word

The Program Status Word (PSW) contains status bits that reflect the current state of the CPU. The PSW, shown in Figure 11, resides in SFR space. The PSW contains the Carry bit, the Auxiliary Carry (for BCD operations), the two-register bank select bits, the Overflow flag, a Parity bit, and two user-definable status flags.

The Carry bit, in addition to serving as a Carry bit in arithmetic operations, also serves as the "Accumulator" for a number of Boolean operations.

The bits RS0 and RS1 select one of the four register banks shown in Figure 8. A number of instructions refer to these RAM locations as R0 through R7. The status of the RS0 and RS1 bits at execution time determines which of the four banks is selected.

The Parity bit reflects the number of 1s in the Accumulator: P = 1 if the Accumulator contains an odd number of 1s, and P = 0 if the Accumulator contains an even number of 1s. Thus, the number of 1s in the Accumulator plus P is always even.

Two bits in the PSW are uncommitted and can be used as general purpose status flags.

Addressing Modes

The addressing modes in the Flash microcontroller instruction set are as follows.

Direct Addressing

In direct addressing, the operand is specified by an 8-bit address field in the instruction. Only internal data RAM and SFRs can be directly addressed.

Indirect Addressing

In indirect addressing, the instruction specifies a register that contains the address of the operand. Both internal and external RAM can be indirectly addressed.

The address register for 8-bit addresses can be either the Stack Pointer or R0 or R1 of the selected register bank. The address register for 16-bit addresses can be only the 16-bit data pointer register, DPTR.

Register Instructions

The register banks, which contain registers R0 through R7, can be accessed by instructions whose opcodes carry a 3-bit register specification. Instructions that access the registers this way make efficient use of code, since this mode eliminates an address byte. When the instruction is executed, one of the eight registers in the selected bank is accessed. One of four banks is selected at execution time by the two bank select bits in the PSW.

Register-Specific Instructions

Some instructions are specific to a certain register. For example, some instructions always operate on the Accumulator, so no address byte is needed to point to it. In these cases, the opcode itself points to the correct register. Instructions that refer to the Accumulator as A assemble as Accumulator-specific opcodes.

Immediate Constants

The value of a constant can follow the opcode in program memory. For example,

MOV A, #100

loads the Accumulator with the decimal number 100. The same number could be specified in hex digits as 64H.

Indexed Addressing

Program memory can only be accessed via indexed addressing. This addressing mode is intended for reading look-up tables in program memory. A 16-bit base register (either DPTR or the Program Counter) points to the base of the table, and the Accumulator is set up with the table entry number. The address of the table entry in program memory is formed by adding the Accumulator data to the base pointer.

Another type of indexed addressing is used in the "case jump" instruction. In this case the destination address of a jump instruction is computed as the sum of the base pointer and the Accumulator data.

Arithmetic Instructions

The menu of arithmetic instructions is listed in Table 1. The table indicates the addressing modes that can be used with each instruction to access the <byte> operand. For example, the ADD A, <byte> instruction can be written as follows.

```
ADD   A,7FH     (direct addressing)
ADD   A,@R0     (indirect addressing)
ADD   A,R7      (register addressing)
ADD   A, #127   (immediate constant)
```

The execution times listed in Table 1 assume a 12 MHz clock frequency. All of the arithmetic instructions execute in 1 µs except the INC DPTR instruction, which takes 2 µs, and the Multiply and Divide instructions, which take 4 µs.

Note that any byte in the internal data memory space can be incremented or decremented without using the Accumulator.

The INC DPTR instruction operates on the 16-bit Data Pointer. The Data Pointer generates 16-bit addresses for external memory, so the ability to be incremented in one 16-bit operation is a useful feature.

The MUL AB instruction multiplies the Accumulator by the data in the B register and puts the 16-bit product into the concatenated B and Accumulator registers.

The DIV AB instruction divides the Accumulator by the data in the B register and leaves the 8-bit quotient in the Accumulator and the 8-bit remainder in the B register.

Note: DIV AB is less useful in arithmetic "divide" routines than in radix conversions and programmable shift operations. In shift operations, dividing a number by 2^n shifts its n bits to the right. Using DIV AB to perform the division completes the shift in 4 µs and leaves the B register holding the bits that were shifted out.

The DA A instruction is for BCD arithmetic operations. In BCD arithmetic, ADD and ADDC instructions should always be followed by a DA A operation, to ensure that the result is also in BCD. Note that DA A will not convert a binary number to BCD. The DA A operation produces a meaningful result only as the second step in the addition of two BCD bytes.

Table 1. A List of Atmel Microcontroller Arithmetic Instructions

Mnemonic	Operation	Addressing Modes				Execution Time (µs)
		Dir	Ind	Reg	Imm	
ADD A, <byte>	A = A + <byte>	X	X	X	X	1
ADDCA, <byte>	A = A + <byte> + C	X	X	X	X	1
SUBB A, <byte>	A = A - <byte> - C	X	X	X	X	1
INC A	A = A + 1	Accumulator only				1
INC <byte>	<byte> = <byte> + 1	X	X	X		1
INC DPTR	DPTR = DPTR + 1	Data Pointer only				2
DEC A	A = A - 1	Accumulator only				1
DEC <byte>	<byte> = <byte> - 1	X	X	X		1
MUL AB	B:A = B x A	ACC and B only				4
DIV AB	A = Int [A/B] B = Mod [A/B]	ACC and B only				4
DA A	Decimal Adjust	Accumulator only				1

Table 2. Logical Instructions

Mnemonic	Operation	Addressing Modes				Execution Time (µs)
		Dir	Ind	Reg	Imm	
ANL A, <byte>	A = A .AND. <byte>	X	X	X	X	1
ANL <byte> ,A	<byte> = <byte> .AND. A	X				1
ANL <byte> , #data	<byte> = <byte> .AND. #data	X				2
ORL A, <byte>	A = A .OR. <byte>	X	X	X	X	1
ORL <byte> ,A	<byte> = <byte> .OR. A	X				1
ORL <byte> , #data	<byte> = <byte> .OR. #data	X				2
XRL A, <byte>	A = A .XOR. <byte>	X	X	X	X	1
XRL <byte> ,A	<byte> = <byte> .XOR. A	X				1
XRL <byte>, #data	<byte> = <byte> .XOR. #data	X				2
CRL A	A = 00H	Accumulator only				1
CPL A	A = .NOT. A	Accumulator only				1
RL A	Rotate ACC Left 1 bit	Accumulator only				1
RLC A	Rotate Left through Carry	Accumulator only				1
RR A	Rotate ACC Right 1 bit	Accumulator only				1
RRC A	Rotate Right through Carry	Accumulator only				1
SWAPA	Swap Nibbles in A	Accumulator only				1

Logical Instructions

Table 2 shows the Atmel Flash microcontroller logical instructions. The instructions that perform Boolean operations (AND, OR, Exclusive OR, NOT) on bytes operate on a bit-by-bit basis. That is, if the Accumulator contains 00110101B and <byte> contains 01010011B, then

```
ANL A, <byte>
```

leaves the Accumulator holding 00010001B.

Table 2 also lists the addressing modes that can be used to access the <byte> operand. Thus, the ANL A, <byte> instruction may take any of the following forms.

```
ANL   A,7FH     (direct addressing)
ANL   A,@R1     (indirect addressing)
ANL   A,R6      (register addressing)
ANL   A, # 53H  (immediate constant)
```

All of the logical instructions that are Accumulator-specific execute in 1 µs (using a 12 MHz clock). The others take 2 µs.

Note that Boolean operations can be performed on any byte in the lower 128 internal data memory space or the SFR space using direct addressing, without using the Accumulator. The XRL <byte>, #data instruction, for example, offers a quick and easy way to invert port bits, as in the following example.

```
XRL   P1,#0FFH
```

If the operation is in response to an interrupt, not using the Accumulator saves the time required to stack it in the service routine.

The Rotate instructions (RL A, RLC A, etc.) shift the Accumulator 1 bit to the left or right. For a left rotation, the MSB rolls into the LSB position. For a right rotation, the Least Significant Bit (LSB) rolls into the Most Significant Bit (MSB) position.

The SWAP A instruction interchanges the high and low nibbles within the Accumulator. This exchange is useful in BCD manipulations. For example, if the Accumulator contains a binary number that is known to be less than 100, the following code can quickly convert it to BCD.

```
MOV   B, # 10
DIV   AB
SWAP  A
ADD   A,B
```

Dividing the number by 10 leaves the tens digit in the low nibble of the Accumulator, and the ones digit in the B register. The

Data Transfers

Internal Ram

Table 3 shows the menu of instructions and associated addressing modes that are available for moving data within the internal memory spaces. With a 12 MHz clock, all of these instructions execute in either 1 or 2 µs.

The MOV <dest> , <src> instruction allows data to be transferred between any two internal RAM or SFR locations without going through the Accumulator.

Table 3. Data Transfer Instructions that Access Internal Data Memory Space

Mnemonic	Operation	Addressing Modes				Execution Time (µs)
		Dir	Ind	Reg	Imm	
MOV A, <src>	A = <src>	X	X	X	X	1
MOV <dest> ,A	<dest> = A	X	X	X		1
MOV <dest>, <src>	<dest> = <src>	X	X	X	X	2
MOV DPTR, #data16	DPTR = 16-bit immediate constant.				X	2
PUSH <src>	INC SP : MOV "@SP",<src>	X				2
POP <dest>	MOV <dest> , "@SP" ; DEC SP	X				2
XCH A, <byte>	ACC and <byte> exchange data	X	X	X		1
XCHDA,@Ri	ACC and @Ri exchange low nibbles		X			1

Note that in all Atmel Flash microcontroller devices, the stack resides in on-chip RAM and grows upwards. The PUSH instruction first increments the Stack Pointer (SP), then copies the byte into the stack. PUSH and POP use only direct addressing to identify the byte being saved or restored, but the stack itself is accessed by indirect addressing using the SP register. This means the stack can go into the Upper 128, if they are implemented, but not into SFR space.

In devices that do not implement the Upper 128, if the SP points to the Upper 128, PUSHed bytes are lost, and POPped bytes are indeterminate.

The Data Transfer instructions include a 16-bit MOV that can initialize the Data Pointer (DPTR) for look-up tables in program memory or for 16-bit external data memory accesses.

The XCH A, <byte> instruction exchanges the data in the Accumulator and the addressed byte. The XCHD A,@Ri instruction is similar, but only the low nibbles are exchanged.

Figure 12. Shifting a BCD Number Two Digits to the Right

	2A	2B	2C	2D	2E	ACC
MOV A,2EH	00	12	34	56	78	78
MOV 2EH,2DH	00	12	34	56	56	78
MOV 2DH,2CH	00	12	34	34	56	78
MOV 2CH,2BH	00	12	12	34	56	78
MOV 2BH,#0	00	00	12	34	56	78

(a) Using direct MOVs: 14 bytes, 9 µs

	2A	2B	2C	2D	2E	ACC
CLR A	00	12	34	56	78	00
XCH A,2BH	00	00	34	56	78	12
XCH A,2CH	00	00	12	56	78	34
XCH A,2DH	00	00	12	34	78	56
XCH A,2EH	00	00	12	34	56	78

(b) Using XCHs: 9 bytes, 5 µs

To see how XCH and XCHD can facilitate data manipulations, consider the problem of shifting an 8-digit BCD number two digits to the right. Figure 12 compares how direct MOVs and XCH instructions can do this operation. The contents of the registers that hold the BCD number and the content of the Accumulator are shown along side each instruction to indicate their status after the instruction executes.

After the routine executes, the Accumulator contains the two digits that were shifted to the right. Using direct MOVs requires 14 code bytes and 9µs of execution time (under a 12 MHz clock). Using XCHs for the same operation requires less code and executes almost twice as fast.

To right-shift by an odd number of digits, a one-digit shift must be executed. Figure 13 shows a sample of code that right-shifts a BCD number one digit, using the XCHD instruction.

In this example, pointers R1 and R0 point to the two bytes containing the last four BCD digits. Then a loop leaves the last byte, location 2EH, holding the last two digits of the shifted number. The pointers are decremented, and the loop is repeated for location 2DH.

Note: The CJNE instruction (Compare and Jump if Not Equal) is a loop control that will be described later.

The loop is executed from LOOP to CJNE for R1 = 2EH, 2DH, 2CH and 2BH. At that point, the digit that was originally shifted out on the right has propagated to location 2AH. Since that location should be left with 0s, the lost digit is moved to the Accumulator.

Figure 13. Shifting a BCD Number One Digit to the Right

	2A	2B	2C	2D	2E	ACC
MOV R1,#2EH	00	12	34	56	78	XX
MOV R0,#2DH	00	12	34	56	78	XX
loop for R1 = 2EH:						
LOOP:MOV A,@R1	00	12	34	56	78	78
XCHD A,@R0	00	12	34	58	78	76
SWAP A	00	12	34	58	78	67
MOV @R1,A	00	12	34	58	67	67
DEC R1	00	12	34	58	67	67
DEC R0	00	12	34	58	67	67
CJNE R1,#2AH,LOOP						
loop for R1 = 2DH:	00	12	38	45	67	45
loop for R1 = 2CH:	00	18	23	45	67	23
loop for R1 = 2BH:	08	01	23	45	67	01
CLR A	08	01	23	45	67	00
XCH A,2AH	00	01	23	45	67	08

External Ram

Table 4 lists the Data Transfer instructions that access external data memory. Only indirect addressing can be used. Either a one-byte address, @Ri, where Ri can be either R0 or R1 of the selected register bank, or a two-byte address, @DPTR, can be used. The disadvantage of using 16-bit addresses when only a few Kbytes of external RAM are involved is that 16-bit addresses use all 8 bits of Port 2 as address bus. On the other hand, 8-bit addresses allow a few Kbytes of RAM to be used without sacrificing all of Port 2, as shown in Figure 6.

All of these instructions execute in 2 µs with a 12 MHz clock.

Table 4. Data Transfer Instructions that Access External Data Memory

Address Width	Mnemonic	Operation	Execution Time (µs)
8 bits	MOVX A,@Ri	Read external RAM @Ri	2
8 bits	MOVX @Ri,A	Write external RAM @Ri	2
16 bits	MOVX A, @DPTR	Read external RAM @DPTR	2
16 bits	MOVX @DPTR,A	Write external RAM @DPTR	2

Table 5. Lookup Table Read Instructions

Mnemonic	Operation	Execution Time (µs)
MOVC A,@A + DPTR	Read Pgm Memory at (A + DPTR)	2
MOVC A,@A + PC	Read Pgm Memory at (A + PC)	2

Note that in all external Data RAM accesses, the Accumulator is always either the destination or source of the data.

The read and write strobes to external RAM are activated only during the execution of a MOVX instruction. Normally these signals are inactive, and if they are not going to be used at all, their pins are available as extra I/O lines.

Lookup Tables

Table 5 shows the two instructions that are available for reading lookup tables in program memory. Since these instructions access only program memory, the lookup tables can only be read, not updated. The mnemonic for "move constant" is MOVC.

If the table access is to external program memory, then the read strobe is $\overline{PSEN}$.

The first MOVC instruction in Table 5 can accommodate a table of up to 256 entries, numbered 0 through 255. The number of the desired entry is loaded into the Accumulator, and the Data Pointer is set up to point to beginning of the table. Then the following instruction copies the desired table entry into the Accumulator.

```
MOVC A, @A+ DPTR
```

The other MOVC instruction works the same way, except the Program Counter (PC) is the table base, and the table is accessed through a subroutine. First, the number of the desired entry is loaded into the Accumulator, and the following subroutine is called.

```
MOV A,ENTRY_NUMBER
CALL TABLE
```

The subroutine TABLE would look like the following example.

```
TABLE:  MOVC A,@A + PC
        RET
```

The table itself immediately follows the RET (return) instruction in program memory. This type of table can have up to 255 entries, numbered 1 through 255. Number 0 can not be used, because at the time the MOVC instruction is executed, the PC contains the address of the RET instruction. An entry numbered 0 would be the RET opcode itself.

Boolean Instructions

Atmel's Flash microcontrollers contain a complete Boolean (single-bit) processor. The internal RAM contains 128 addressable bits, and the SFR space can support up to 128 other addressable bits. All of the port lines are bit-addressable, and each one can be treated as a separate single-bit port. The instructions that access these bits are not just conditional branches, but a complete menu of move, set, clear, complement, OR, and AND instructions. These kinds of bit operations are not easily obtained in other architectures with any amount of byte-oriented software.

Table 6. Boolean Instructions

Mnemonic	Operation	Execution Time (µs)
ANL C,bit	C = C .AND. bit	2
ANL C,/bit	C = C .AND. .NOT.bit	2
ORL C,bit	C = C .OR. bit	2
ORL C,/bit	C = C .OR. .NOT. bit	2
MOV C,bit	C = bit	1
MOV bit,C	bit = C	2
CLR C	C = 0	1
CLR bit	bit = 0	1
SETB C	C = 1	1
SETB bit	bit = 1	1
CPL C	C = .NOT. C	1
CPL bit	bit = .NOT. bit	1
JC rel	Jump if C = 1	2
JNC rel	Jump if C = 0	2
JB bit,rel	Jump if bit = 1	2
JNB bit,rel	Jump if bit = 0	2
JBC bit,rel	Jump if bit = 1; CLR bit	2

The instruction set for the Boolean processor is shown in Table 6. All bit accesses are by direct addressing. Bit addresses 00H through 7FH are in the Lower 128, and bit addresses 80H through FFH are in SFR space.

The following example shows how easily an internal flag can be moved to a port pin.

```
MOV   C,FLAG
MOV   P1.0,C
```

In this example, FLAG is the name of any addressable bit in the Lower 128 or SFR space. An I/O line (the LSB of Port 1, in this case) is set or cleared depending on whether the flag bit is 1 or 0.

The Carry bit in the PSW is used as the single-bit Accumulator of the Boolean processor. Bit instructions that refer to the Carry bit as C assemble as Carry-specific instructions (CLR C, etc). The Carry bit also has a direct address, since it resides in the PSW register, which is bit-addressable.

The Boolean instruction set includes ANL and ORL, but not the XRL (Exclusive OR) operation. Implementing XRL in software is simple. Suppose, for example, that an application requires the Exclusive OR of two bits.

C = bit1 .XRL. bit2

The software to do this operation could be as follows.

```
      MOV C,bit1
      JNB bit2,OVER
      CPL C
OVER  (continue)
```

First, bitl is moved to the Carry. If bit2 = 0, then C now contains the correct result. That is, bit1 .XRL. bit2 = bit1 if bit2 = 0. On the other hand, if bit2 = 1, C now contains the complement of the correct result. C CARRY need only be inverted (CPL C) to complete the operation.

This code uses the JNB instruction, one of a series of bit-test instructions which execute a jump if the addressed bit is set (JC, JB, JBC) or if the addressed bit is not set (JNC, JNB). In the above case, bit2 is being tested, and if bit2 = 0, the CPL C instruction is jumped over.

If the addressed bit is set, JBC executes the jump and also clears the bit. Thus, a flag can be tested and cleared in one operation.

All the PSW bits are directly addressable, so the Parity bit, or the general purpose flags, for example, are also available to the bit-test instructions.

Relative Offset

The destination address for these jumps is specified to the assembler by a label or by an actual address in program memory. However, the destination address assembles to a relative offset byte. This is a signed (two's complement) offset byte that is added to the PC in two's complement arithmetic if the jump is executed.

The range of the jump is therefore -128 to +127 program memory bytes relative to the first byte following the instruction.

Jump Instructions

Table 7 shows the list of unconditional jumps.

Table 7. Unconditional Jumps in Flash Microcontrollers

Mnemonic	Operation	Execution Time (µs)
JMP addr	Jump to addr	2
JMP @A+DPTR	Jump to A+DPTR	2
CALL addr	Call subroutine at addr	2
RET	Return from subroutine	2
RETI	Return from interrupt	2
NOP	No operation	1

Table 8. Conditional Jumps in Flash Microcontrollers

Mnemonic	Operation	Addressing Modes				Execution Time (µs)
		Dir	Ind	Reg	Imm	
JZ rel	Jump if A = 0	Accumulator only				2
JNZ rel	Jump if A ≠ 0	Accumulator only				2
DJNZ <byte> , rel	Decrement and jump if not zero	X		X		2
CJNE A, <byte> ,rel	Jump if A ≠ <byte>	X			X	2
CJNE <byte> ,#data,rel	Jump if <byte> ≠ #data		X	X		2

Table 7 lists a single JMP addr instruction, but in fact there are three—SJMP, LJMP and AJMP—which differ in the format of the destination address. JMP is a generic mnemonic that can be used if the programmer does not care which way the jump is encoded.

The SJMP instruction encodes the destination address as a relative offset, as described above. The instruction is 2 bytes long, consisting of the opcode and the relative offset byte. The jump distance is limited to a range of -128 to +127 bytes, relative to the instruction following the SJMP.

The LJMP instruction encodes the destination address as a 16-bit constant. The instruction is 3 bytes long, consisting of the opcode and two address bytes. The destination address can be anywhere in the 64K program memory space.

The AJMP instruction encodes the destination address as an 11-bit constant. The instruction is 2 bytes long, consisting of the opcode, which itself contains 3 of the 11 address bits, followed by another byte containing the low 8 bits of the destination address. When the instruction is executed, these 11 bits are simply substituted for the low 11 bits in the PC. The high 5 bits stay the same. Hence, the destination has to be within the same 2K block as the instruction following the AJMP.

In all cases, the programmer specifies the destination address to the assembler the same way: as a label or as a 16-bit constant. The assembler puts the destination address into the correct format for the given instruction. If the format required by the instruction does not support the distance to the specified destination address, a "Destination out of range" message is written into the List file.

The JMP @A+DPTR instruction supports case jumps. The destination address is computed at execution time as the sum of the 16-bit DPTR register and the Accumulator. Typically, DPTR is set up with the address of a jump table, and the Accumulator is given an index to the table. In a 5-way branch, for example, an integer 0 through 4 is loaded into the Accumulator. The code to be executed might be as follows.

```
MOV     DPTR, # JUMP__TABLE
MOV     A,INDEX__NUMBER
RL      A
JMP     @A+ DPTR
```

The RL A instruction converts the index number (0 through 4) to an even number in the range 0 through 8, because each entry in the jump table is 2 bytes long, as shown in the following example.

```
JUMP__TABLE:
        AJMP    CASE__0
        AJMP    CASE__1
        AJMP    CASE__2
        AJMP    CASE__3
        AJMP    CASE__4
```

Table 8 shows a single CALL addr instruction, but there are two CALL instructions—LCALL and ACALL—which differ in the format in which the subroutine address is given to the CPU. CALL is a generic mnemonic that can be used if the programmer does not care which way the address is encoded.

The LCALL instruction uses the 16-bit address format, and the subroutine can be anywhere in the 64K program memory space. The ACALL instruction uses the 11-bit format, and the subroutine must be in the same 2K block as the instruction following the ACALL.

In any case, the programmer specifies the subroutine address to the assembler the same way: as a label or as a 16-bit constant. The assembler puts the address into the correct format for the given instructions.

Subroutines should end with a RET instruction, which returns execution to the instruction following the CALL.

RETI is used to return from an interrupt service routine. The only difference between RET and RETI is that RETI tells the interrupt control system that the interrupt in progress is finished. If no interrupt is in progress at the time RETI is executed, then the RETI is functionally identical to RET.

Table 8 shows the list of conditional jumps available. All of these jumps specify the destination address by the relative offset method and so are limited to a jump distance of -128 to +127 bytes from the instruction following the conditional jump instruction. However, the user specifies to the assembler the actual destination address the same way as the other jumps: as a label or a 16-bit constant.

There is no 0 bit in the PSW. The JZ and JNZ instructions test the Accumulator data for that condition.

The DJNZ instruction (Decrement and Jump if Not Zero) is for loop control. To execute a loop N times, load a counter byte with N and terminate the loop with a DJNZ to the beginning of the loop, as shown below for N = 10.

```
        MOV     COUNTER,#10
LOOP:   (begin loop)
          *
          *
          *
        (end loop)
        DJNZ    COUNTER,LOOP
        (continue)
```

The CJNE instruction (Compare and Jump if Not Equal) can also be used for loop control, as shown in Figure 13. Two bytes are specified in the operand field of the instruction. The jump is executed only if the two bytes are not equal. In the example of Figure 13, the two bytes were the data in R1 and the constant 2AH. The initial data in R1 was 2EH. Every time the loop was executed, R1 was decremented, and the looping continued until the R1 data reached 2AH.

Another application of this instruction is in "greater than, less than" comparisons. The two bytes in the operand field are taken as unsigned integers. If the first is less than the second, then the Carry bit is set (1). If the first is greater than or equal to the second, then the Carry bit is cleared.

CPU Timing

All Atmel Flash microcontrollers have an on-chip oscillator, which can be used as the clock source for the CPU. To use the on-chip oscillator, connect a crystal or ceramic resonator between the XTAL1 and XTAL2 pins of the microcontroller, and connect the capacitors to ground as shown in Figure 14.

Examples of how to drive the clock with an external oscillator are shown in Figure 15b.

The internal clock generator defines the sequence of states that make up the microcontroller machine cycle.

Figure 14. Using the On-Chip Oscillator

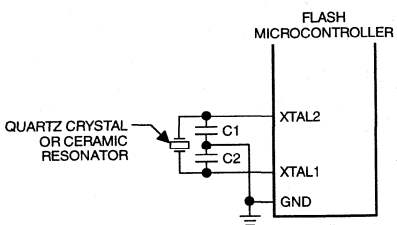


Figure 15a. Oscillator Connections

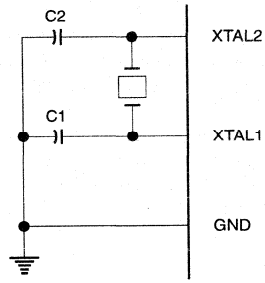
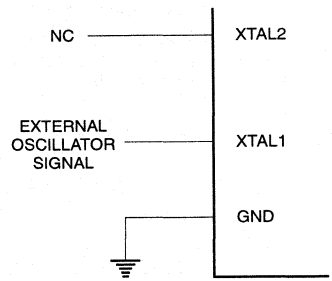


Figure 15b. External Clock Drive Configuration



Machine Cycles

A machine cycle consists of a sequence of 6 states, numbered S1 through S6. Each state time lasts for two oscillator periods. Thus, a machine cycle lasts 12 oscillator periods or 1 µs if the oscillator frequency is 12 MHz.

Each state is divided into a Phase 1 half and a Phase 2 half. Figure 16 shows the fetch/execute sequences in states and phases for various kinds of instructions. Normally two program fetches are generated during each machine cycle, even if the instruction being executed does not require it. If the instruction being executed does not need more code bytes, the CPU ignores the extra fetch, and the Program Counter is not incremented.

Execution of a one-cycle instruction (Figure 16A and B) begins during State 1 of the machine cycle, when the opcode is latched into the Instruction Register. A second fetch occurs during S4 of the same machine cycle. Execution is complete at the end of State 6 of this machine cycle.

The MOVX instructions take two machine cycles to execute. No program fetch is generated during the second cycle of a MOVX

instruction. This is the only time program fetches are skipped. The fetch/execute sequence for MOVX instructions is shown in Figure 16(D).

The fetch/execute sequences are the same whether the program memory is internal or external to the chip. Execution times do not depend on whether the program memory is internal or external.

Figure 17 shows the signals and timing involved in program fetches when the program memory is external. If program memory is external, the program memory read strobe $\overline{PSEN}$ is normally activated twice per machine cycle, as shown in Figure 17(A).

If an access to external data memory occurs, as shown in Figure 17(B), two $\overline{PSEN}$s are skipped, because the address and data bus are being used for the data memory access.

A data memory bus cycle takes twice as much time as a program memory bus cycle. Figure 17 shows the relative timing of the addresses being emitted at Ports 0 and 2 and of ALE and PSEN. ALE latches the low address byte from P0 into the address latch.

When the CPU is executing from internal program memory, $\overline{PSEN}$ is not activated, and program addresses are not emitted. However, ALE continues to be activated twice per machine cycle and is therefore available as a clock output signal. Note, however, that one ALE is skipped during the execution of the MOVX instruction.

Figure 16. State Sequences in Atmel Flash Microcontrollers

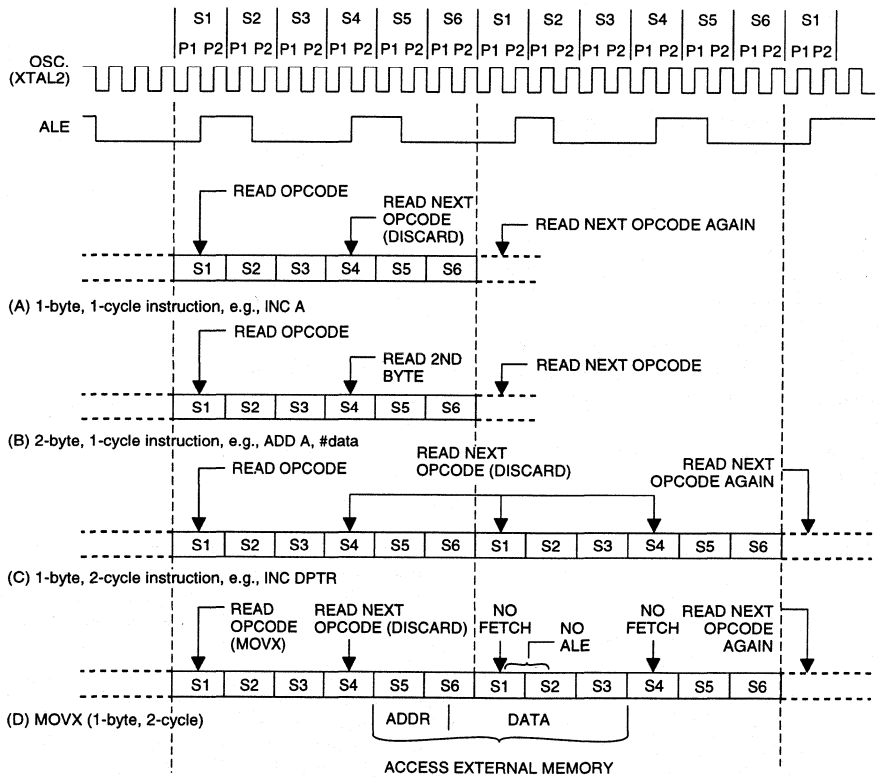
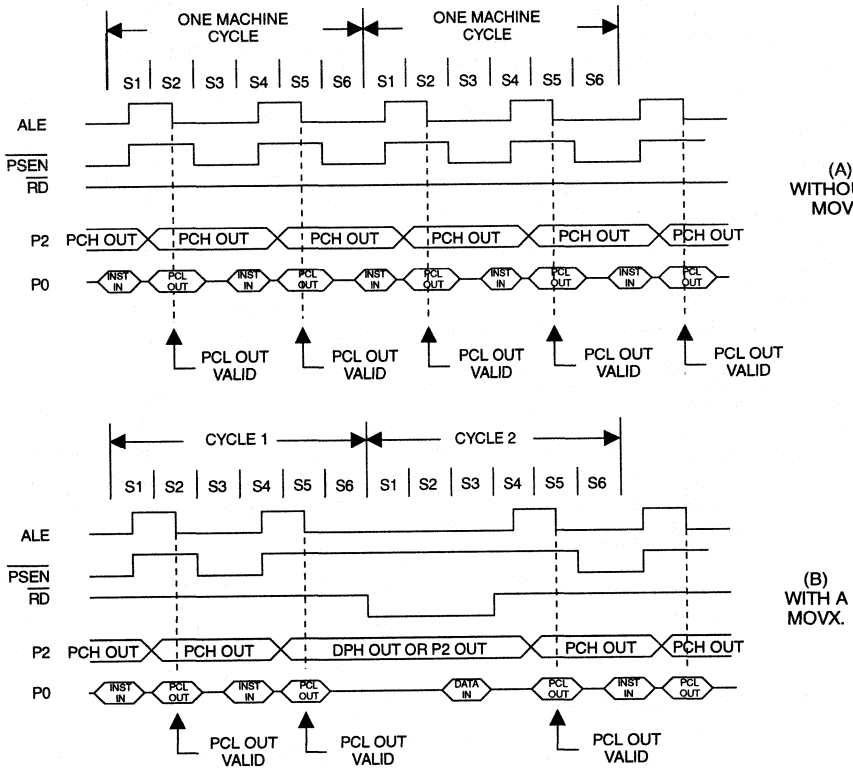


Figure 17. Bus Cycles Executing from External Program Memory



Interrupt Structure

The AT89C51 core provides 5 interrupt sources: 2 external interrupts, 2 timer interrupts, and the serial port interrupt. What follows is an overview of the interrupt structure for the AT89C51. Other Atmel Flash microcontrollers have additional interrupt sources and vectors. Refer to the data sheets on other devices for further information on their interrupts.

Interrupt Enables

Each of the interrupt sources can be individually enabled or disabled by setting or clearing the Interrupt Enable (IE) bit in the SFR. This register also contains a global disable bit, which can be cleared to disable all interrupts at once. Figure 18 shows the IE register for the AT89C51.

Figure 18. Interrupt Enable (IE) Register in the AT89C51

(MSB)							(LSB)
EA	—	—	ES	ET1	EX1	ET0	EX0

Enable bit = 1 enables the interrupt.

Enable bit = 0 disables it.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	reserved.*
—	IE.5	reserved.*
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 overflow interrupt enable bit.
EX1	IE.2	External Interrupt 1 enable bit.
ET0	IE.1	Timer 0 Overflow Interrupt enable bit.
EX0	IE.0	External Interrupt 0 enable bit.

*These reserved bits are used in other Atmel microcontrollers.

Interrupt Priorities

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing the Interrupt Priority (IP) bit in the SFR. Figure 19 shows the IP register in the AT89C51.

A low-priority interrupt can be interrupted by a high-priority interrupt but not by another low-priority interrupt. A high-priority interrupt can not be interrupted by any other interrupt source.

If two interrupt requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If interrupt requests of the same priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus, within each priority level the polling sequence determines a second priority structure.

Figure 20 shows how the IE and IP registers and the polling sequence work to determine which (if any) interrupt will be serviced.

In operation, all the interrupt flags are latched into the interrupt control system during State 5 of every machine cycle. The samples are polled during the following machine cycle. If the flag for an enabled interrupt is found to be set (1), the interrupt system generates an LCALL to the appropriate location in program memory, unless some other condition blocks the interrupt. Several conditions can block an interrupt, including an interrupt of equal or higher priority level already in progress.

Figure 19. IP (Interrupt Priority) Register in the AT89C51

(MSB)							(LSB)
—	—	—	PS	PT1	PX1	PT0	PX0

Priority bit = 1 assigns high priority.

Priority bit = 0 assigns low priority.

Symbol	Position	Function
—	IP.7	reserved*
—	IP.6	reserved*
—	IP.5	reserved*
PS	IP.4	Serial Port interrupt priority bit.
PT1	IP.3	Timer 1 interrupt priority bit.
PX1	IP.2	External interrupt 1 priority bit.
PT0	IP.1	Timer 0 interrupt priority bit.
PX0	IP.0	External interrupt 0 priority bit.

*These reserved bits are used in other Atmel microcontrollers.

The hardware-generated LCALL pushes the contents of the Program Counter onto the stack and reloads the PC with the beginning address of the service routine. As previously noted (Figure 4), the service routine for each interrupt begins at a fixed location.

Only the Program Counter is automatically pushed onto the stack, not the PSW or any other register. Because only the PC is automatically saved, the programmer can decide how much time to spend saving other registers. This enhances the interrupt response time, albeit at the expense of increasing the programmer's burden of responsibility. As a result, many interrupt functions that are typical in control applications—toggling a port pin, reloading a timer, or unloading a serial buffer, for example—can often be completed in less time than it takes other architectures to begin them.

Simulating a Third Priority Level in Software

Some applications require more than the two priority levels that are provided by on-chip hardware in Atmel Flash microcontrollers. In these cases, relatively simple software can be written to produce the same effect as a third priority level.

First, interrupts that require higher priority than 1 are assigned to priority 1 in the IP register. The service routines for priority 1

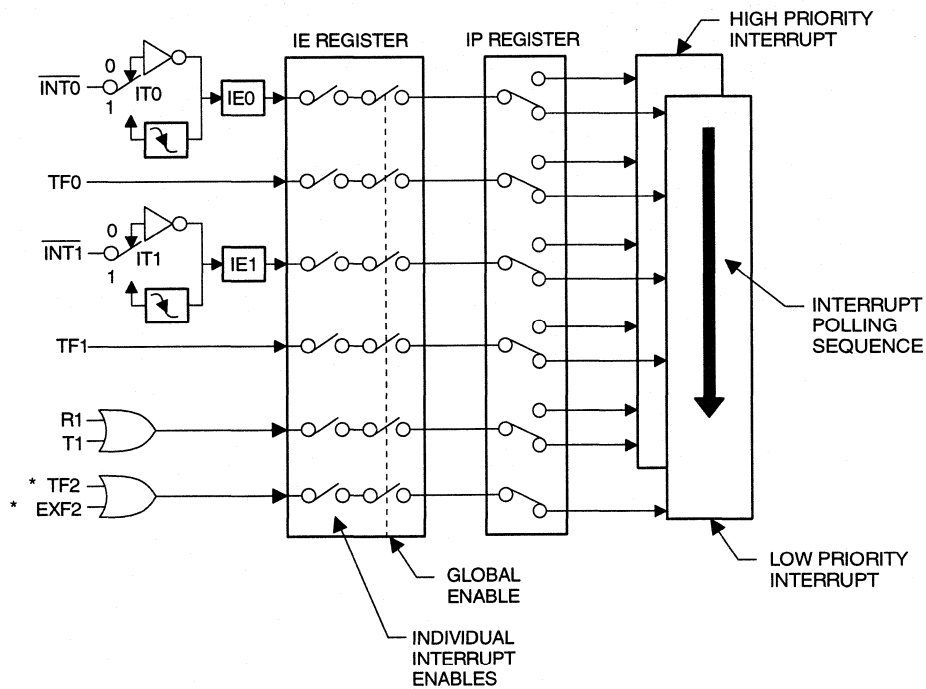
interrupts that are supposed to be interruptible by priority 2 interrupts are written to include the following code.

```
        PUSH    IE
        MOV     IE, # MASK
        CALL    LABEL
         * * * * * * *
   (execute service routine)
         * * * * * * *
        POP     IE
        RET
LABEL:  RETI
```

As soon as any priority 1 interrupt is acknowledged, the IE register is redefined to disable all but priority 2 interrupts. Then, a CALL to LABEL executes the RETI instruction, which clears the priority 1 interrupt-in-progress flip-flop. At this point, any enabled priority 1 interrupt can be serviced, but only priority 2 interrupts are enabled.

POPping IE restores the original enable byte. Then, a normal RET (rather than another RETI) is used to terminate the service routine. The additional software adds 10 µs (at 12 MHz) to priority 1 interrupts.

Figure 20. AT89 Interrupt Control System



* Only on AT89C52/AT89LV52/AT89S8252

The information presented in this chapter is collected from the Microcontroller Architectural Overview, AT89C51, AT89LV51, AT89C52, AT89LV52, AT89C2051, and AT89C1051 data sheets of this book. The material has been selected and rearranged to form a quick and convenient reference for the programmers of Atmel's microcontroller family of devices. This guide pertains specifically to the AT89C51, AT89LV51, AT89C52, and AT89LV52.

Memory Organization

Program Memory

The AT89C Microcontroller has separate address spaces for program memory and data memory. The program memory can be up to 64 Kbytes long. The lower addresses may reside on-chip.

Figure 1 shows a map of the AT89C51 program memory, and Figure 2 shows a map of the AT89C52 program memory. The AT89C1051/2051 do not have off-board memory expansion.

Figure 1. AT89C51 Program Memory

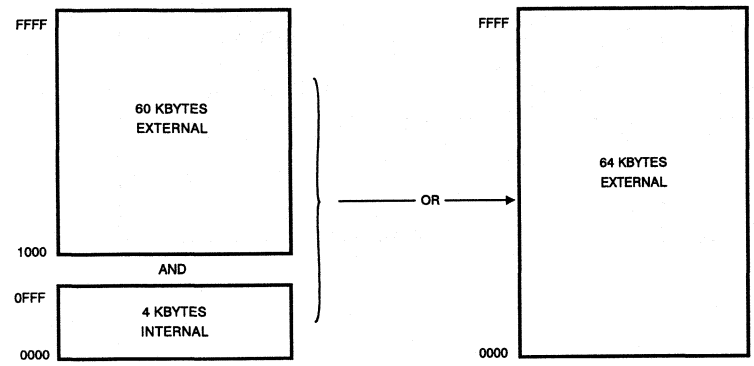
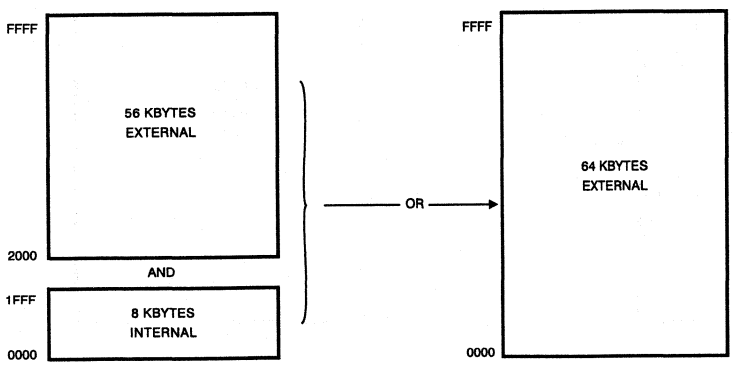


Figure 2. AT89C52 Program Memory



Flash Microcontroller Memory Organization

2

Data Memory

The AT89C can directly address up to 64 Kbytes of data memory external to the chip. The MOVX instruction accesses the external data memory. (Refer to the Instruction Set section in this chapter for a detailed description of instructions).

The AT89C51 has 128 bytes of on-chip RAM (256 bytes in the AT89C52) plus a number of Special Function Registers (SFRs). The lower 128 bytes of RAM can be accessed either by direct addressing (MOV data addr) or by indirect addressing (MOV @Ri). Figure 3 shows the AT89C51 and the AT89C52 data memory organization.

Figure 3a. The AT89C51 Data Memory

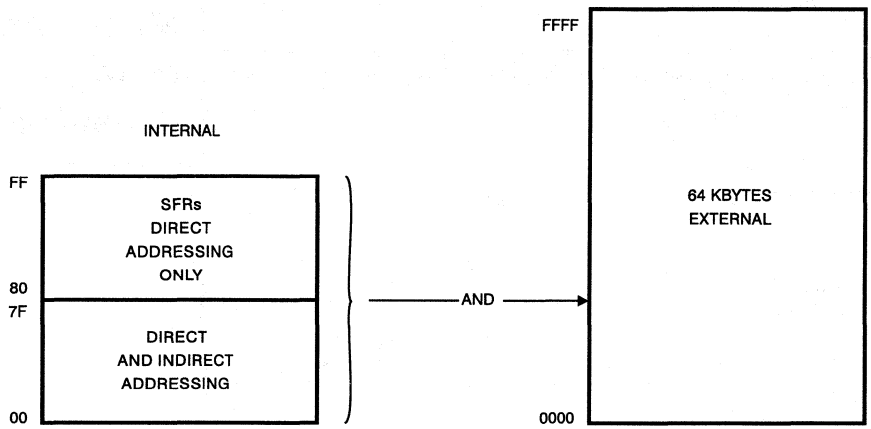
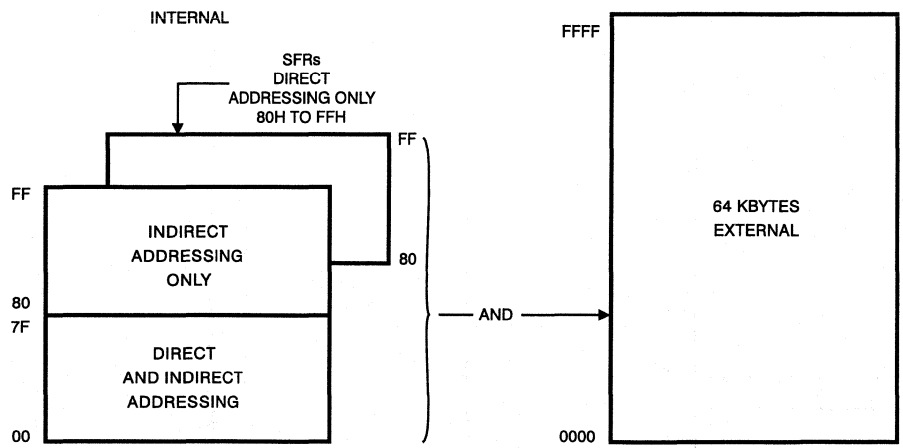


Figure 3b. The AT89C52 Data Memory



Indirect Address Area

In Figure 3b, the SFRs and the indirect address RAM have the same addresses (80H through 0FFH). Nevertheless, they are two separate areas and are accessed in two different ways.

For example, the following instruction writes 0AAH to Port 0, which is one of the SFRs.

```
MOV  80H, # 0AAH
```

The following instruction writes 0BBH in location 80H of the data RAM.

```
MOV  R0, # 80H
MOV  @ R0, # 0BBH
```

Thus, after executing both of these instructions, Port 0 contains 0AAH, and location 80H of the RAM contains 0BBH.

The stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space in devices that implement 256 bytes of internal RAM.

Direct and Indirect Address Area

The 128 bytes of RAM that can be accessed by both direct and indirect addressing can be divided into 3 segments as described in this section and as shown in Figure 4.

1. Register Banks 0-3: Locations 0 through 1FH (32 bytes). Reset default is to register bank 0. To use the other register banks, the user must select them in the software. Each register bank contains eight 1-byte registers, 0 through 7.

Reset initializes the Stack Pointer to location 07H. The Stack Pointer is then incremented once to start from location 08H, which is the first register (R0) of the second register bank. Thus, in order to use more than one register bank, the SP should be initialized to a different location of the RAM that is not used for data storage (that is, a higher part of the RAM).

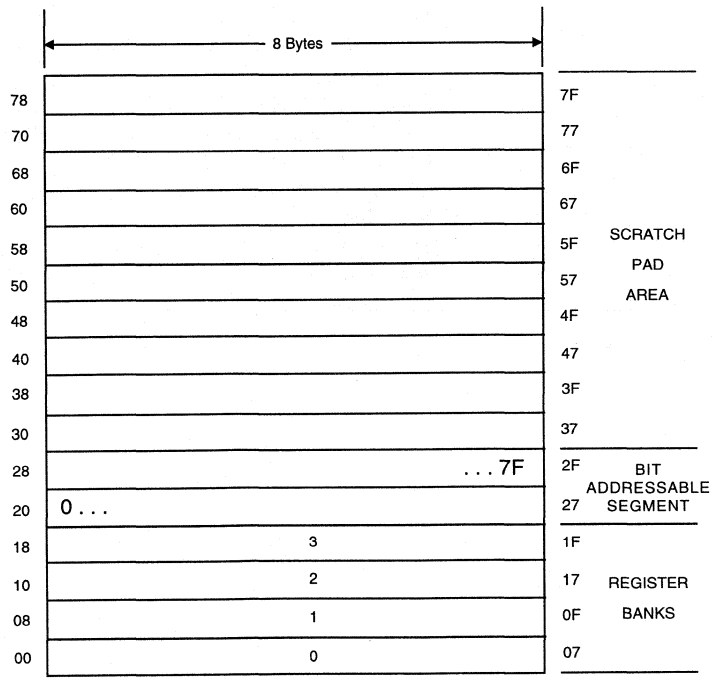
2. Bit Addressable Area: 16 bytes have been assigned for this segment, 20H through 2FH. Each of the 128 bits of this segment can be directly addressed (0 through 7FH).

These bits can be referred to in two ways. One way is to refer to their addresses, that is, 0 to 7FH. The other way is with reference to bytes 20H to 2FH. Thus, bits 0 through 7 can also be referred to as bits 20.0 through 20.7, and bits 8 through FH are the same as 21.0 through 21.7, and so on.

Each of the 16 bytes in this segment can also be addressed as a byte.

3. Scratch Pad Area: Bytes 30H through 7FH are available to the user as data RAM. However, if the stack pointer has been initialized to this area, enough bytes should be left aside to prevent SP data destruction.

Figure 4. 128 Bytes of Directly and Indirectly Addressable RAM



Special Function Registers

Table 1 contains a list of all the SFRs and their addresses.

All of the SFRs that are byte- and bit-addressable are located on the first column of the diagram in Figure 5.

Table 1. Special Function Registers

Symbol	Name	Address
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 Bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port 1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*+T2CON	Timer/Counter 2 Control	0C8H
+T2MOD	Timer/Counter 2 Mode Control	0C9H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
+TH2	Timer/Counter 2 High Byte	0CDH
+TL2	Timer/Counter 2 Low Byte	0CCH
+RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
+RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

* = Bit addressable
+ = AT89C52 only

Contents of the SFRs Just After Power-On or a Reset

Table 2. Contents of the SFRs after power-on or a hardware reset

Register	Value in Binary
*ACC	00000000
*B	00000000
*PSW	00000000
SP	00000111
DPTR	
DPH	00000000
DPL	00000000
*PO	11111111
*P1	11111111
*P2	11111111
*P3	11111111
*IP	80C51 XXX00000, 80C52 XX000000
*IE	80C51 0XX00000, 80C52 0X000000
TMOD	00000000
+T2MOD	XXXXXX00
*TCON	00000000
*+T2CON	00000000
TH0	00000000
TL0	00000000
TH1	00000000
TL1	00000000
+TH2	00000000
+TL2	00000000
+RCAP2H	00000000
+RRAP2L	00000000
*SCON	00000000
SBUF	Indeterminate
PCON	CMOS 0XXX0000

X = Undefined
* = Bit Addressable
+ = AT89C52 only

Special Function Register Map

Figure 5. SFR Memory Map

	8 Bytes								
F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW*								D7
C8	T2CON*+	T2MOD+	RCAP2L+	RCAP2H+	TL2+	TH2+			CF
C0									C7
B8	IP*								BF
B0	P3								B7
A8	IE*								AF
A0	P2								A7
98	SCON*	SBUF							9F
90	P1								97
88	TCON*	TMOD*	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON*	87

↑
Bit
Addressable

* SFRs converting mode or control bits
\+ AT89C52 only

SFRs whose bits are assigned for various functions are listed in this section. For more detailed information, refer to the Microcontroller Architectural Overview chapter of this book.

PSW: Program Status Word (Bit Addressable)

CY	AC	F0	RS1	RS0	OV	—	P

CY	PSW.7	Carry flag.
AC	PSW.6	Auxiliary carry flag.
F0	PSW.5	Flag 0 available to the user for general purpose.
RS1	PSW.4	Register Bank selector bit 1 (see note 1).
RS0	PSW.3	Register Bank selector bit 0 (see note 1).
OV	PSW.2	Overflow flag.
—	PSW.1	User definable flag.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an odd/even number of 1 bits in the accumulator.

Note: 1. The values of RS0 and RS1 select the corresponding register bank.

RS1	RS0	Register Bank	Address
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

PCON: Power Control Register (Not Bit Addressable)

SMOD	—	—	—	GF1	GF0	PD	IDL

SMOD	Double baud rate bit. If Timer 1 is used to generate baud rate and SMOD = 1, the baud rate is doubled when the Serial Port is used in modes 1, 2, or 3.
—	Not implemented, reserved for future use. *
—	Not implemented, reserved for future use.*
—	Not implemented, reserved for future use. *
GF1	General purpose flag bit.
GF0	General purpose flag bit.
PD	Power Down bit. Setting this bit activates Power Down operation in the AT89C51.
IDL	Idle Mode bit. Setting this bit activates Idle Mode operation in the AT89C51.

If 1s are written to PD and IDL at the same time, PD takes precedence.

* User software should not write 1s to reserved bits. These bits may be used in future microcontrollers to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

Interrupts

In order to use any of the interrupts in the Flash microcontroller, take the following three steps.

1. Set the EA (enable all) bit in the IE register to 1.
2. Set the corresponding individual interrupt enable bit in the IE register to 1.
3. Begin the interrupt service routine at the corresponding Vector Address of that interrupt. See the following table.

Interrupt Source	Vector Address
IE0	0003H
TF0	000BH
IE1	0013H
TF1	001BH
R1 & T1	0023H
TF2 & EXF2*	002BH

* AT89C52 only.

In addition, for external interrupts, pins $\overline{INT0}$ and $\overline{INT1}$ (P3.2 and P3.3) must be set to 1, and depending on whether the interrupt is to be level or transition activated, bits IT0 or IT1 in the TCON register may need to be set to 1.

ITx = 0 level activated

ITx = 1 transition activated

IE: Interrupt Enable Register (Bit Addressable)

If the bit is 0, the corresponding interrupt is disabled. If the bit is 1, the corresponding interrupt is enabled.

EA	—	ET2	ES	ET1	EX1	ET0	EX0

EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Not implemented, reserved for future use.*
ET2	IE.5	Enables or disables the Timer 2 overflow or capture interrupt (AT89C52 only).
ES	IE.4	Enables or disables the serial port interrupt.
ET1	IE.3	Enables or disables the Timer 1 overflow interrupt.
EX1	IE.2	Enables or disables External Interrupt 1.
ET0	IE. 1	Enables or disables the Timer 0 overflow interrupt.
EX0	IE.0	Enables or disables External Interrupt 0 .

* User software should not write 1s to reserved bits. These bits may be used in future Flash microcontrollers to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

Assigning Higher Priority to One or More Interrupts

In order to assign higher priority to an interrupt the corresponding bit in the IP register must be set to 1.

While an interrupt service is in progress, it cannot be interrupted by an interrupt of the same or lower priority.

Priority Within Level

The only purpose of priority within a level is to resolve simultaneous requests of the same priority level.

From high to low, interrupt sources are listed below.

IE0
TF0
IE1
TF1
RI or TI
TF2 or EXF2

IP: Interrupt Priority Register (Bit Addressable)

If the bit is 0, the corresponding interrupt has a lower priority. If the bit is 1, the corresponding interrupt has a higher priority.

—	—	PT2	PS	PT1	PX1	PT0	PX0

—	IP. 7	Not implemented, reserved for future use.*
—	IP. 6	Not implemented, reserved for future use.*
PT2	IP. 5	Defines the Timer 2 interrupt priority level (AT89C52 only).
PS	IP. 4	Defines the Serial Port interrupt priority level.
PT1	IP. 3	Defines the Timer 1 interrupt priority level.
PX1	IP. 2	Defines External Interrupt 1 priority level.
PT0	IP. 1	Defines the Timer 0 interrupt priority level.
PX0	IP. 0	Defines the External Interrupt 0 priority level.

* User software should not write 1s to reserved bits. These bits may be used in future Flash microcontrollers to invoke new features. In that case, the reset or inactive value of the new bit will be 0, and its active value will be 1.

TCON: Timer/Counter Control Register (Bit Addressable)

TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

TF1 TCON. 7 Timer 1 overflow flag. Set by hardware when the Timer/Counter 1 overflows. Cleared by hardware as the processor vectors to the interrupt service routine.

TR1 TCON. 6 Timer 1 run control bit. Set/cleared by software to turn Timer/Counter 1 ON/OFF.

TF0 TCON. 5 Timer 0 overflow flag. Set by hardware when the Timer/Counter 0 overflows. Cleared by hardware as the processor vectors to the service routine.

TR0 TCON. 4 Timer 0 run control bit. Set/cleared by software to turn Timer/Counter 0 ON/OFF.

IE1 TCON. 3 External Interrupt 1 edge flag. Set by hardware when the External Interrupt edge is detected. Cleared by hardware when the interrupt is processed.

IT1 TCON. 2 Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

IE0 TCON. 1 External Interrupt 0 edge flag. Set by hardware when External Interrupt edge detected. Cleared by hardware when interrupt is processed.

IT0 TCON. 0 Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered External Interrupt.

TMOD: Timer/Counter Mode Control Register (Not Bit Addressable)

Timer 1				Timer 0			
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0

GATE When TRx (in TCON) is set and GATE = 1, TIMER/COUNTERx runs only while the INTx pin is high (hardware control). When GATE = 0, TIMER/COUNTERx will run only while TRx = 1 (software control).

C/$\overline{T}$ Timer or Counter selector. Cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).

M1 Mode selector bit (note 1).

M0 Mode selector bit (note 1).

NOTE 1:

M1	M0	Operating Mode	
0	0	0	13-bit Timer
0	1	1	16-bit Timer/Counter
1	0	2	8-bit Auto-Reload Timer/Counter
1	1	3	Split Timer Mode: (Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits, TH0 is an 8-bit Timer and is controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

Timer Set-Up

Tables 3 through 6 give TMOD values that can be used to set up Timer 0 in different modes.

It is assumed that only one timer is used at a time. If Timers 0 and 1 must run simultaneously in any mode, the value in TMOD for Timer 0 must be ORed with the value shown for Timer 1 (Tables 5 and 6).

For example, if Timer 0 must run in mode 1 GATE (external control), and Timer 1 must run in mode 2 COUNTER, then the value that must be loaded into TMOD is 69H (09H from Table 3 ORed with 60H from Table 6).

Moreover, it is assumed that the user is not ready at this point to turn the timers on and will do so at another point in the program by setting bit TRx (in TCON) to 1.

Timer/Counter 0

Table 3. Timer/Counter 0 Used as a Timer

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	08H
1	16-bit Timer	01H	09H
2	8-bit Auto-Reload	02H	0AH
3	two 8-bit Timers	03H	0BH

Table 4. Timer/Counter 0 Used as a Counter

MODE	TIMER 0 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	04H	0CH
1	16-bit Timer	05H	0DH
2	8-bit Auto-Reload	06H	0EH
3	one 8-bit Counter	07H	0FH

NOTES: 1. The Timer is turned ON/OFF by setting/clearing bit TR0 in the software.
2. The Timer is turned ON/OFF by the 1 to 0 transition on $\overline{INT0}$ (P3.2) when TR0 = 1 (hardware control).

Timer/Counter 1

Table 5. Timer/Counter 1 Used as a Timer

MODE	TIMER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	00H	80H
1	16-bit Timer	10H	90H
2	8-bit Auto-Reload	20H	A0H
3	does not run	30H	B0H

Table 6. Timer/Counter 1 Used as a Counter

MODE	COUNTER 1 FUNCTION	TMOD	
		INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
0	13-bit Timer	40H	C0H
1	16-bit Timer	50H	D0H
2	8-bit Auto-Reload	60H	E0H
3	not available	—	—

NOTES: 1. The Timer is turned ON/OFF by setting/clearing bit TR1 in the software.
2. The Timer is turned ON/OFF by the 1 to 0 transition on $\overline{INT1}$ (P3.3) wen TR1 = 1 (hardware control).

T2CON: Timer/Counter 2 Control Register (Bit Addressable)

AT89C52 Only

TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

TF2 T2CON. 7 Timer 2 overflow flag set by hardware and cleared by software. TF2 cannot be set when either RCLK = 1 or CLK = 1

EXF2 T2CON. 6 Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX, and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 causes the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.

RCLK T2CON. 5 Receive clock flag. When set, causes the Serial Port to use Timer 2 overflow pulses for its receive clock in modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.

TLCK T2CON. 4 Transmit clock flag. When set, causes the Serial Port to use Timer 2 overflow pulses for its transmit clock in modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.

EXEN2 T2CON. 3 Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of negative transition on T2EX if Timer 2 is not being used to clock the Serial Port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.

TR2 T2CON. 2 Software START/STOP control for Timer 2. A logic 1 starts the Timer.

C/$\overline{T2}$ T2CON. 1 Timer or Counter select.

0 = Internal Timer. 1 = External Event Counter (triggered by falling edge).

CP/$\overline{RL2}$ T2CON. 0 Capture/Reload flag. When set, captures occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto-reloads occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the Timer is forced to auto-reload on Timer 2 overflow.

T2MOD: Timer 2 Mode Control Register

T2MOD Address = 0C9H **Reset Value = XXXX XX00B**

Not Bit Addressable

-	-	-	-	-	-	T2OE	DCEN
Bit 7	6	5	4	3	2	1	0

Symbol	Function
-	Not implemented, reserved for future use
T2OE	Timer 2 Output Enable bit
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.

Timer/Counter 2 Set-Up

Except for the baud rate generator mode, the values given for T2CON do not include the setting of the TR2 bit. Therefore, bit TR2 must be set separately to turn the Timer on.

Table 7. Timer/Counter 2 Used as a Timer

MODE	T2CON	
	INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
16-bit Auto-Reload	00H	08H
16-bit Capture	01H	09H
Baud rate generator receive and transmit same baud rate	34H	36H
receive only	24H	26H
transmit only	14H	16H

Table 8. Timer/Counter 2 Used as a Counter

MODE	TMOD	
	INTERNAL CONTROL (NOTE 1)	EXTERNAL CONTROL (NOTE 2)
16-bit Auto Reload	02H	0AH
16-bit Capture	03H	0BH

NOTES: 1.Capture/Reload occurs only on Timer/Counter overflow.
2.Capture/Reload occurs on Timer/Counter overflow and a 1 to 0 transition on T2EX (P1.1) pin except when Timer 2 is used in the baud rate generating mode.

SCON: Serial Port Control Register (Bit Addressable)

SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON. 7	Serial Port mode specifer. (NOTE 1).
SM1	SCON. 6	Serial Port mode specifer. (NOTE 1).
SM2	SCON. 5	Enables the multiprocessor communication feature in modes 2 and 3. In mode 2 or 3, if SM2 is set to 1, then RI is not activated if the received ninth data bit (RB8) is 0. In mode 1, if SM2 = 1, then RI is not activated if a valid stop bit was not received. In mode 0, SM2 should be 0. (See Table 9).
REN	SCON. 4	Set/Cleared by software to Enable/Disable reception.
TB8	SCON. 3	The ninth bit that is transmitted in modes 2 and 3. Set/Cleared by software.
RB8	SCON. 2	In modes 2 and 3, is the ninth data bit that was received. In mode 1, if SM2 = 0, RB8 is the stop bit that was received. In mode 0, RB8 is not used.
TI	SCON. 1	Transmit interrupt flag. Set by hardware at the end of the eighth bit time in mode 0 or at the beginning of the stop bit in the other modes. Must be cleared by software.
RI	SCON. 0	Receive interrupt flag. Set by hardware at the end of the eighth bit time in mode 0 or halfway through the stop bit time in the other modes (except see SM2). Must be cleared by software.

NOTE 1:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	SHIFT REGISTER	Fosc./12
0	1	1	8-Bit UART	Variable
1	0	2	9-Bit UART	Fosc./64 OR Fosc./32
1	1	3	9-Bit UART	Variable

Table 9. Serial Port Set-Up

MODE	SCON	SM2 VARIATION
0	10H	Single Processor
1	50H	Environment
2	90H	(SM2 = 0)
3	D0H	
0	NA	Multiprocessor
1	70H	Environment
2	B0H	(SM2 = 1)
3	F0H	

Generating Baud Rates

Serial Port in Mode 0

Mode 0 has a fixed baud rate, which is 1/12 of the oscillator frequency. To run the serial port in this mode, none of the Timer/Counters need to be set up. Only the SCON register needs to be defined.

$$\text{Baud Rate} = \frac{\text{Osc Freq}}{12}$$

Serial Port in Mode 1

Mode 1 has a variable baud rate. The baud rate can be generated by either Timer 1 or Timer 2 (AT89C52 only).

Using Timer/Counter 1 to Generate Baud Rates

For this purpose, Timer l is used in mode 2 (Auto-Reload). Refer to the Timer Setup section of this chapter.

$$\text{Baud Rate} = \frac{\text{K x Oscillator Freq.}}{32 \text{ x } 12 \text{ x } [256 - (\text{TH1})]}$$

If SMOD = 0, then K = 1.

If SMOD = 1, then K = 2. (SMOD is the PCON register).

The user usually knows the baud rate but needs to know the reload value for TH1. Therefore, the equation to calculate TH1 can be written as follows.

$$\text{TH1} = 256 - \frac{\text{K x Osc Freq.}}{384 \text{ x baud rate}}$$

TH1 must be an integer value. Rounding off TH1 to the nearest integer may not produce the desired baud rate. In this case, the user may have to choose another crystal frequency. See Baud Rate table.

Since the PCON register is not bit addressable, one way to set the bit is logical ORing the PCON register (that is, ORL PCON, # 80H). The address of PCON is 87H.

Using Timer/Counter 2 to Generate Baud Rates

For this purpose, Timer 2 must be used in the baud rate generating mode. Refer to Timer 2 Setup Table in this chapter. If Timer 2 is clocked through pin T2 (P1.0) the baud rate given by the following equation.

$$\text{Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

If it is being clocked internally the baud rate is given by the following equation.

$$\text{Baud Rate} = \frac{\text{Osc Freq.}}{32 \text{ x } [65536 - (\text{RCAP2H, RCAP2L})]}$$

To obtain the reload value for RCAP2H and RCAP2L the previous equation can be rewritten as follows.

$$\text{RCAP2H, RCAP2L} = 65536 - \frac{\text{Osc Freq.}}{32 \text{ x Baud Rate}}$$

Serial Port in Mode 2

The baud rate is fixed in this mode and is 1/32 or 1/64 of the oscillator frequency, depending on the value of the SMOD bit in the PCON register.

In this mode, none of the Timers is used, and the clock comes from the internal phase 2 clock.

SMOD = 1, Baud Rate = 1/32 Osc Freq.

SMOD = O, Baud Rate = 1/64 Osc Freq.

To set the SMOD bit, use ORL PCON, # 80H. The address of PCON is 87H.

Serial Port in Mode 3

The baud rate in mode 3 is variable and sets up exactly the same as in mode 1.

Baud Rate Table

Crystal Frequency	7.3728 MHz	8 MHz	11.0592 MHz	12.00 MHz	14.75156 MHz	16.00 MHz
TH1						
E0	600	651	900	976	1,200	1,302
F0	1,200	1.302	1,800	1,953	2,400	2,604
F8	2,400	2,604	3,600	3,906	4,800	5,208
F9	2,743	2,976	8,229	4,464	5,486	5,952
FA	3,200	3,472	9,600	5,208	6,400	6,944
FF	19,200	20,833	57,600	62,500		41,666

Table 10. Baud Rate Summary

Baud Rate	Crystal Frequency	SMOD	TH1 Reload Value	Actual Baud Rate	Error
9600	12.000 MHz	1	-7 (F9H)	8923	7%
2400	12.000 MHz	0	-13 (F3H)	2404	0.16%
1200	12.000 MHz	0	-26 (E6H)	1202	0.16%
19200	11.059 MHz	1	-3 (FDH)	19200	0
9600	11.059 MHz	0	-3 (FDH)	9600	0
2400	11.059 MHz	0	-12 (F4H)	2400	0
1200	11.059 MHz	0	-24 (E8H)	1200	0

NOTE: Due to rounding, there is a slight error in the resulting baud rate. Generally, a 5% error is tolerable using asynchronous (start/stop) communications. Exact baud rates are possible using an 11.059 MHz crystal. The table above summarizes the TH1 reload values for the most common baud rates, using a 12.000 MHz or 11.059 MHz crystal.

Introduction

This chapter presents a comprehensive description of the on-chip hardware features of Atmel's Flash-based microcontrollers. Included in this description are the following items.

- The port drivers and how they function both as ports and, for Ports 0 and 2, in bus operations
- The Timer/Counters
- The Serial Interface
- The Interrupt System
- Reset
- The Reduced Power Modes and Low Power Idle

The devices under consideration are listed in Table 1.

Figure 1 shows a functional block diagram of the AT89C51 and AT89C52.

Table 1. Atmel's Flash Microcontrollers

Device Name	Program Memory	Data Memory Bytes	16-bit Timers	Technology
AT89C1051	1K Flash	64 RAM	1	CMOS
AT89C2051	2K Flash	128 RAM	2	CMOS
AT89C51	4K Flash	128 RAM	2	CMOS
AT89C52	8K Flash	256 RAM	3	CMOS
AT89S8252	8K Flash	256 RAM 2K EEPROM	3	CMOS

Special Function Registers

A map of the on-chip memory area called Special Function Register (SFR) space is shown in Figure 2. SFRs marked by parentheses are resident in the AT89C52 but not in the AT89C51.

Figure 1. AT89C51 and AT89C52 Flash-Based Microcontroller Architectural Block Diagram

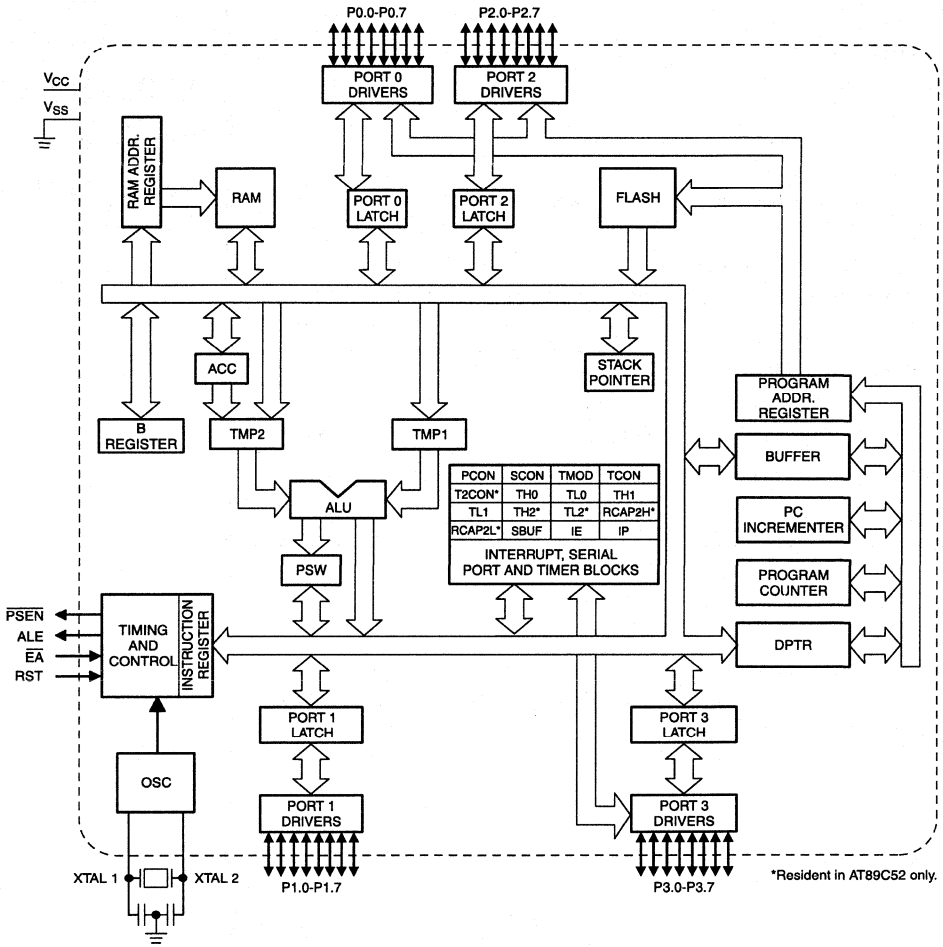


Figure 2. SFR Map. (...) Indicates Resident in AT89C52, not in AT89C51.

	8 Bytes								
F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8	(T2CON)	(T2MOD)	(RCAP2L)	(RCAP2H)	(TL2)	(TH2)			CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

Not all of the addresses are occupied. Unoccupied addresses are not implemented on the chip. Read accesses to these addresses in general return random data, and write accesses have no effect.

User software should not write 1s to these unimplemented locations, since they may be used in future microcontrollers to invoke new features. In that case, the reset or inactive values of the new bits will always be 0, and their active values will be 1.

The functions of the SFRs are outlined in the following sections.

Accumulator

ACC is the Accumulator register. The mnemonics for Accumulator-specific instructions, however, refer to the Accumulator simply as A.

B Register

The B register is used during multiply and divide operations. For other instructions it can be treated as another scratch pad register.

Program Status Word

The PSW register contains program status information, as detailed in Figure 3.

Stack Pointer

The Stack Pointer Register is 8 bits wide. It is incremented before data is stored during PUSH and CALL executions. While the stack may reside anywhere in on-chip RAM, the Stack Pointer is initialized to 07H after a reset. This causes the stack to begin at location 08H.

Data Pointer

The Data Pointer (DPTR) consists of a high byte (DPH) and a low byte (DPL). Its function is to hold a 16-bit address. It may be manipulated as a 16-bit register or as two independent 8-bit registers.

Ports 0 To 3

P0, P1, P2, and P3 are the SFR latches of Ports 0, 1, 2, and 3, respectively.

Serial Data Buffer

The Serial Data Buffer is actually two separate registers, a transmit buffer and a receive buffer register. When data is moved to SBUF, it goes to the transmit buffer, where it is held for serial transmission. (Moving a byte to SBUF initiates the transmission.) When data is moved from SBUF, it comes from the receive buffer.

Timer Registers

Register pairs (TH0, TL0), (TH1, TL1), and (TH2, TL2) are the 16-bit Counter registers for Timer/Counters 0, 1, and 2, respectively.

Capture Registers

The register pair (RCAP2H, RCAP2L) are the Capture registers for the Timer 2 Capture Mode. In this mode, in response to a transition at the AT89C52's T2EX pin, TH2 and TL2 are copied into RCAP2H and RCAP2L. Timer 2 also has a 16-bit auto-reload mode, and RCAP2H and RCAP2L hold the reload value for this mode.

Control Registers

Special Function Registers IP, IE, TMOD, TCON, T2CON, T2MOD, SCON, and PCON contain control and status bits for the interrupt system, the Timer/Counters, and the serial port. They are described in later sections of this chapter.

Figure 3. PSW: Program Status Word Register

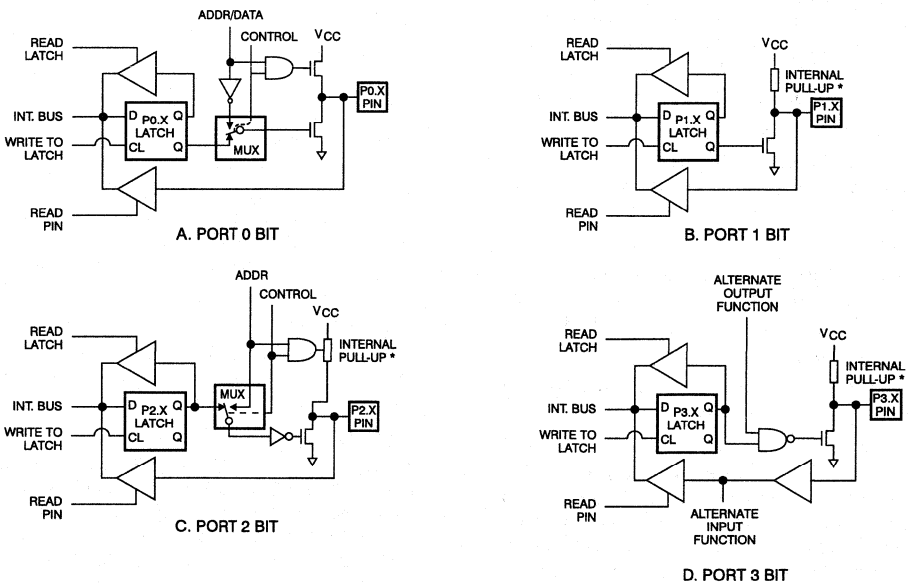
(MSB)							(LSB)
CY	AC	F0	RS1	RS0	OV	—	P

Symbol	Position	Name and Significance
CY	PSW.7	Carry flag.
AC	PSW.6	Auxillary Carry flag. (For BCD operations.)
F0	PSW.5	Flag 0 (Available to the user for general purposes.)
RS1	PSW.4	Register bank select control bits 1 and 0. Set/cleared by software to determine working register bank (see Note).
RS0	PSW.3	
OV	PSW.2	Overflow flag.
—	PSW.1	User definable flag.
P	PSW.0	Parity flag. Set/cleared by hardware each instruction cycle to indicate an add/even number of 1 bits in the Accumulator, that is, even parity.

NOTE:
The contents of (RS1, RS0) enable the working register banks as follows:
(0.0)—Bank 0(00H-07H)
(0.1)—Bank 1(08H-0FH)
(1.0)—Bank 2(10H-17H)
(1.1)—Bank 3(18H-1FH)

Figure 4. AT89C51 and AT89C52 Port Bit Latches and I/O Buffers

*See Figure 5 for details of the internal pullup.



Port Structures and Operation

All four ports in the AT89C51 and AT89C52 are bidirectional. Each consists of a latch (Special Function Registers P0 through P3), an output driver, and an input buffer.

The output drivers of Ports 0 and 2, and the input buffers of Port 0, are used in accesses to external memory. In this application, Port 0 outputs the low byte of the external memory address, time-multiplexed with the byte being written or read. Port 2 outputs the high byte of the external memory address when the address is 16 bits wide. Otherwise the Port 2 pins continue to emit the P2 SFR content.

All the Port 3 pins, and two Port 1 pins (in the AT89C52) are multifunctional. They are not only port pins, but also provide the special features listed in the following table.

Port Pin	Alternate Function
*P1.0	T2 (Timer/Counter 2 external input)
*P1.1	T2EX (Timer/Counter 2 Capture/Reload trigger)
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt)
P3.3	$\overline{INT1}$ (external interrupt)
P3.4	T0 (Timer/Counter 0 external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

*P1.0 and P1.1 serve these alternate functions only on the AT89C52.

The alternate functions can only be activated if the corresponding bit latch in the port SFR contains a 1. Otherwise the port pin is stuck at 0.

I/O Configurations

Figure 4 shows a functional diagram of a typical bit latch and I/O buffer in each of the four ports. The bit latch (one bit in the port's SFR) is represented as a Type D flip-flop, which clocks a value from the internal bus in response to a "write to latch" signal from the CPU. The Q output of the flip-flop is placed on the internal bus in response to a "read latch" signal from the CPU. The level of the port pin itself is placed on the internal bus in response to a "read pin" signal from the CPU. Some instructions that read a port activate the "read latch" signal, and others activate the "read pin" signal.

As shown in Figure 4, the output drivers of Ports 0 and 2 can be switched to an internal ADDR and ADDR/DATA bus by an internal CONTROL signal for use in external memory accesses. During external memory accesses, the P2 SFR remains unchanged, but 1s are written to the P0 SFR.

If a P3 bit latch contains a 1, then the output level is controlled by the alternate output function signal, as shown in Figure 4. The actual P3.X pin level is always available to the pin's alternate input function.

Ports 1, 2, and 3 have internal pullups. Port 0 has open drain outputs. Each I/O line can be used independently as an input or an output. (Ports 0 and 2 may not be used as general purpose I/O when being used as the ADDR/DATA BUS). To be used as an input, the port bit latch must contain a 1, which turns off the output driver FET. Then, for Ports 1, 2, and 3, the pin is pulled high by the internal pullup but can be pulled low by an external source.

Port 0 has no internal pullups. The FET pullup in the P0 output driver (see Figure 4) is used only when the Port emits 1s during external memory accesses. Otherwise, the FET pullup is off. Consequently, P0 lines that are used as output port lines are open drain. Writing a 1 to the bit latch leaves both FET outputs off, so the pin floats. In this condition, it can be used a high-impedance input.

Because Ports 1, 2, and 3 have fixed internal pullups, they are sometimes called quasi-bidirectional ports. When configured as inputs, they pull high and source current (I_{IL}) when externally pulled low. Port 0, on the other hand, is considered truly bidirectional, because it floats when configured as an input.

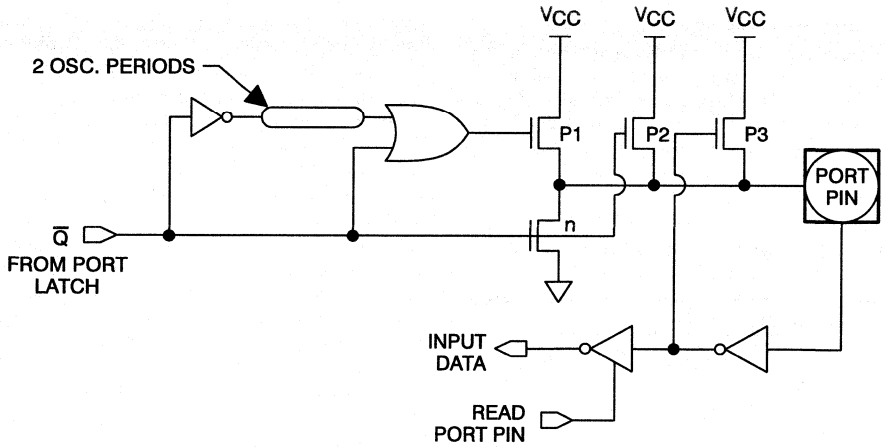
The reset function writes 1s to all the port latches in the AT89C51and AT89C52. If a 0 is subsequently written to a port latch, the latch can be reconfigured as an input if a 1 is written to it.

Writing to a Port

When an instruction changes a port latch value, the new value arrives at the latch during S6P2 of the final cycle of the instruction. However, port latches are sampled by their output buffers only during Phase 1 of any clock period. (During Phase 2, the output buffer holds the value sampled during the previous Phase 1). Consequently, the new value in the port latch does not actually appear at the output pin until the next Phase 1, which is at S1P1 of the next machine cycle. See Figure 39 in the Internal Timing section.

If the change requires a 0-to-1 transition in Port 1, 2, or 3, an additional pullup is turned on during S1P1 and S1P2 of the cycle in which the transition occurs to increase the transition speed. The extra pullup can source about 100 times the current that the normal pullup can. The internal pullups are field-effect transistors, not linear resistors. The pullup arrangements are shown in Figure 5.

Figure 5. Ports 1 and 3 Internal Pullup Configurations. Port 2 is similar except that it holds the strong pullup on while emitting 1s that are address bits. (See text, "Accessing External Memory".)



pFET1 is turned on for 2 osc. periods after $\overline{Q}$ makes a 0-to-1 transition. During this time, pFET1 also turns on pFET3 through the inverter to form a latch which holds the 1. pFET2 is also on.

The pullup consists of three pFETs. An n-channel FET (nFET) turns on when a logical 1 is applied to its gate, and turns off when a logical 0 is applied to its gate. A p-channel FET (pFET) is the opposite: it is on when its gate sees a 0 and off when its gate sees a 1.

The pFET1 transistor in Figure 5 is turned on for 2 oscillator periods after a 0-to-l transition in the port latch. While pFET1 is on, it turns on pFET3 (a weak pullup) through the inverter. This inverter and pFET3 form a latch that holds the 1.

If the pin emits a 1, a negative glitch on the pin from some external source can turn off pFET3, causing the pin to go into a float state. pFET2 is a very weak pullup which is on whenever the nFET is off, in traditional CMOS style. pFET2 is only about 1/10 the strength of pFET3. Its function is to restore a 1 to the pin in the event the pin lost a 1 in a glitch.

Port Loading and Interfacing

The output buffers of Ports 1, 2, and 3 can each drive 4 LS TTL inputs. CMOS pins can be driven by open-collector and open-drain outputs, but 0-to-l transitions will not be fast. An input 0 turns off pullup pFET3, leaving only the very weak pullup pFET2 to drive the transition.

In external bus mode, Port 0 output buffers can drive 8 LS TTL inputs. As port pins, they require external pullups to drive any inputs.

Read-Modify-Write Feature

Some instructions that read a port read the latch and others read the pin. Read-modify-write instructions read the latch rather than the pin, and these instructions read a value, possibly change it, and then rewrite it to the latch. When the destination operand is a port, or a port bit, the read-modify-write instructions given in the following table read the latch rather than the pin.

Mnemonic	Instruction	Example
ANL	Logical AND	ANL P1, A
ORL	Logical OR	ORL P2, A
XRL	Logical EX-OR	XRL P3, A
JBC	Jump if bit = 1 and clear bit	JBC P1.1, LABEL
CPL	Complement bit, CPL P3.0	
INC	Increment	INC P2
DEC	Decrement	DEC P2
DJNZ	Decrement and jump if not zero	DJNZ P3, LABEL
MOV, PX.Y, C	Move carry bit to bit Y of Port X	
CLR PX.Y	Clear bit Y of Port X	
SETB PX.Y	Set bit Y of Port X	

The last three instructions in this list are read-modify-write instructions, because they read all 8 bits of the port byte, modify the addressed bit, then write the new byte back to the latch.

Read-modify-write instructions are directed to the latch rather than the pin in order to avoid misinterpreting the voltage level at the pin. For example, a port bit might be used to drive the base of a transistor. When a 1 is written to the bit, the transistor is

turned on. If the CPU then reads the same port bit at the pin rather than the latch, it will read the base voltage of the transistor and interpret it as a 0. Reading the latch rather than the pin will return the correct value of 1.

Accessing External Memory

Accesses to external memory are either to program memory or to data memory. Accesses to external program memory use the $\overline{PSEN}$ (program store enable) signal as the read strobe. Accesses to external data memory use $\overline{RD}$ or $\overline{WR}$ (alternate functions of P3.7 and P3.6) to strobe the memory. Refer to Figures 36 through 38 in the Internal Timing section for more information.

Fetches from external progam memory always use a 16-bit address. Accesses to external data memory can use either a 16-bit address (MOVX @DPTR) or an 8-bit address (MOVX @Ri).

Whenever a 16-bit address is used, the high byte of the address comes out on Port 2, where it is held for the duration of the read or write cycle. Note that the Port 2 drivers use the strong pullups during the entire time that they emit address bits that are 1s (during the execution of a MOVX @DPTR instruction.) During this time, the Port 2 latch (the Special Function Register) does not have to contain 1s, and the contents of the Port 2 SFR are not modified. If the external memory cycle is not immediately followed by another external memory cycle, the undisturbed contents of the Port 2 SFR reappear in the next cycle.

If an 8-bit address is used (MOVX @Ri), the contents of the Port 2 SFR remain at the Port 2 pins throughout the external memory cycle, which facilitates paging.

In any case, the low byte of the address is time-multiplexed with the data byte on Port 0. The ADDR/DATA signal drives both FETs in the Port 0 output buffers. Thus, in this application the Port 0 pins are not open-drain outputs and do not require external pullups. The Address Latche Enable (ALE) signal should be used to capture the address byte into an external latch. The address byte is valid at the negative transition of ALE. Then, in a write cycle, the data byte to be written appears on Port 0 just before $\overline{WR}$ is activated and remains there until after $\overline{WR}$ is deactivated. In a read cycle, the incoming byte is accepted at Port 0 just before the read strobe is deactivated.

During any access to external memory, the CPU writes 0FFH to the Port 0 latch (the Special Function Register), thus obliterating any information in the Port 0 SFR. If the user writes to Port 0 during an external memory fetch, the incoming code byte is corrupted. Therefore, do not write to Port 0 if external program memory is used.

External program memory is accessed under the following two conditions.

1) When the $\overline{EA}$ signal is active; or

2) When the program counter (PC) contains a number larger than 0FFFH (1FFFH for the AT89C52).

When the CPU is executing out of external program memory, all 8 bits of Port 2 are dedicated to an output function and may not be used for general purpose I/O. During external program fetches, they output the high byte of the PC. During this time, the Port 2 drivers use the strong pullups to emit PC bits that are 1s.

Timer/Counters

The AT89C51 has two 16-bit Timer/Counter registers: Timer 0 and Timer 1. The AT89C52 has these two Timer/Counters, and in addition Timer 2. All three can be configured to operate either as Timers or event Counters.

As a Timer, the register is incremented every machine cycle. Thus, the register counts machine cycles. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

As a Counter, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T0, T1, or (in the AT89C52) T2. The external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since 2 machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. There are no restrictions on the duty cycle of the external input signal, but it should be held for at least one full machine cycle to ensure that a given level is sampled at least once before it changes.

In addition to the Timer or Counter functions, Timer 0 and Timer 1 have four operating modes: (13 bit timer, 16 bit timer, 8 bit auto-reload, split timer). Timer 2 in the AT89C52 has three modes of operation: Capture, Auto-Reload, and baud rate generator.

Timer 0 and Timer 1

Timer/Counters 1 and 0 are present in both the AT89C51 and AT89C52. The Timer or Counter function is selected by control bits $C/\overline{T}$ in the Special Function Register TMOD (Figure 6). These two Timer/Counters have four operating modes, which are selected by bit pairs (M1, M0) in TMOD. Modes 0, 1, and 2 are the same for both Timer/Counters, but Mode 3 is different. The four modes are described in the following sections.

Mode 0

Both Timers in Mode 0 are 8-bit Counters with a divide-by-32 prescaler. Figure 7 shows the Mode 0 operation as it applies to Timer 1.

In this mode, the Timer register is configured as a 13-bit register. As the count rolls over from all 1s to all 0s, it sets the Timer interrupt flag TF1. The counted input is enabled to the Timer when TR1 = 1 and either GATE = 0 or $\overline{INT1}$ = 1. Setting GATE = 1 allows the Timer to be controlled by external input $\overline{INT1}$, to facilitate pulse width measurements. TR1 is a control bit in the Special Function Register TCON (Figure 8). GATE is in TMOD.

The 13-bit register consists of all 8 bits of TH1 and the lower 5 bits of TL1. The upper 3 bits of TL1 are indeterminate and should be ignored. Setting the run flag (TR1) does not clear the registers.

Mode 0 operation is the same for Timer 0 as for Timer 1, except that TR0, TF0 and $\overline{INT0}$ replace the corresponding Timer 1 signals in Figure 7. There are two different GATE bits, one for Timer 1 (TMOD.7) and one for Timer 0 (TMOD.3).

Figure 6. TMOD: Timer/Counter Mode Control Register

(MSB)							(LSB)
GATE	C/$\overline{T}$	M1	M0	GATE	C/$\overline{T}$	M1	M0
Timer 1				**Timer 0**			

GATE	Gating control when set. Timer/Counter x is enabled only while INTx pin is high and TRx control pin is set. When cleared, Timer x is enabled whenever TRx control bit is set.	Timer 0 gate bit
C/$\overline{T}$	Timer or Counter Selector cleared for Timer operation (input from internal system clock). Set for Counter operation (input from Tx input pin).	Timer 0 counter/timer select bit
M1	Mode bit 1	Timer 0 M1 bit
M0	Mode bit 0	Timer 0 M0 bit

M1	M0	Mode	Operating Mode
0	0	0	13-bit Timer Mode. 8-bit Timer/Counter THz with TLx as 5-bit prescaler.
0	1	1	16-bit Timer Mode. 16-bit Timer/Counters THx and TLx are cascaded; there is no prescaler.
1	0	2	8-bit Auto Reload. 8-bit auto-reload Timer/Counter THx holds a value which is to be reloaded into TLx each time it overflows.
1	1	3	Split Timer Mode. (Timer 0) TL0 is an 8-bit Timer/Counter controlled by the standard Timer 0 control bits. TH0 is an 8-bit timer only controlled by Timer 1 control bits.
1	1	3	(Timer 1) Timer/Counter 1 stopped.

Timer SFR	Purpose	Address	Bit-Addressable
TCON	Control	88H	Yes
TMOD	Mode	89H	No
TL0	Timer 0 low-byte	8AH	No
TL1	Timer 1 low-byte	8BH	No
TH0	Timer 0 high-byte	8CH	No
TH1	Timer 1 high-byte	8DH	No
T2CON*	Timer 2 control	C8H	Yes
T2MOD*	Timer 2 Mode	C9H	No
RCAP2L*	Timer 2 low-byte capture	CAH	No
RCAP2H*	Timer 2 high-byte capture	CBH	No
TL2*	Timer 2 low-byte	CCH	No
TH2*	Timer 2 high-byte	CDH	No

*AT8952 only

Figure 7. Timer/Counter 1 Mode 0: 13-Bit Counter

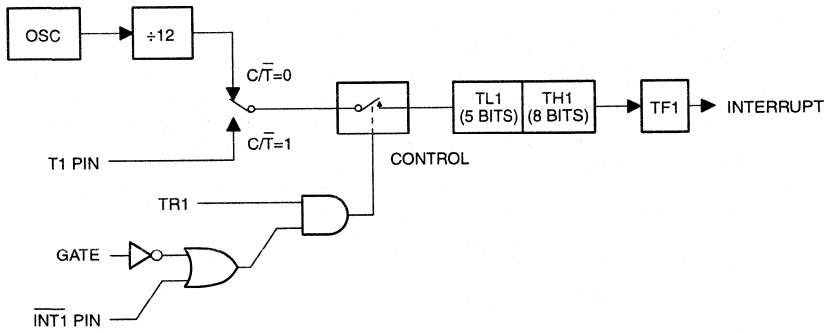
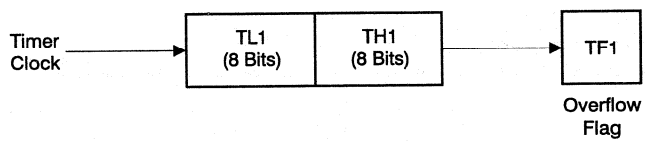


Figure 8. Timer/Counter 1 Mode 1: 16-Bit Counter



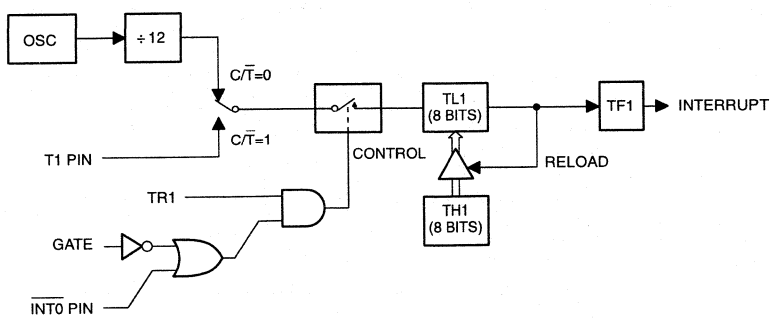
Mode 1

Mode 1 is the same as Mode 0, except that the Timer register is run with all 16 bits. The clock is applied to the combined high and low timer registers (TL1/TH1). As clock pulses are received, the timer counts up: 0000H, 0001H, 0002H, etc. An overflow occurs on the FFFFH-to-0000H overflow flag. The timer continues to count. The overflow flag is the TF1 bit in TCON that is read or written by software. See Figure 8.

Mode 2

Mode 2 configures the Timer register as an 8-bit Counter (TL1) with automatic reload, as shown in Figure 9. Overflow from TL1 not only sets TF1, but also reloads TL1 with the contents of TH1, which is preset by software. The reload leaves TH1 unchanged. Mode 2 operation is the same for Timer/Counter 0.

Figure 9. Timer/Counter 1 Mode 2: 8-Bit Auto-Reload



Mode 3

Timer 1 in Mode 3 simply holds its count. The effect is the same as setting TR1 = 0.

Timer 0 in Mode 3 establishes TL0 and TH0 as two separate counters. The logic for Mode 3 on Timer 0 is shown in Figure 10. TL0 uses the Timer 0 control bits: C/T, GATE, TR0, INT0, and TF0. TH0 is locked into a timer function (counting machine cycles) and over the use of TR1 and TF1 from Timer 1. Thus, TH0 now controls the Timer 1 interrupt.

Mode 3 is for applications requiring an extra 8-bit timer or counter. With Timer 0 in Mode 3, the AT89C51 can appear to have three Timer/Counters, and an AT89C52, can appear to have four. When Timer 0 is in Mode 3, Timer 1 can be turned on and off by switching it out of and into its own Mode 3. In this case, Timer 1 can still be used by the serial port as a baud rate generator or in any application not requiring an interrupt.

Figure 10. Timer/Counter 0 Mode 3: Two 8-Bit Counters

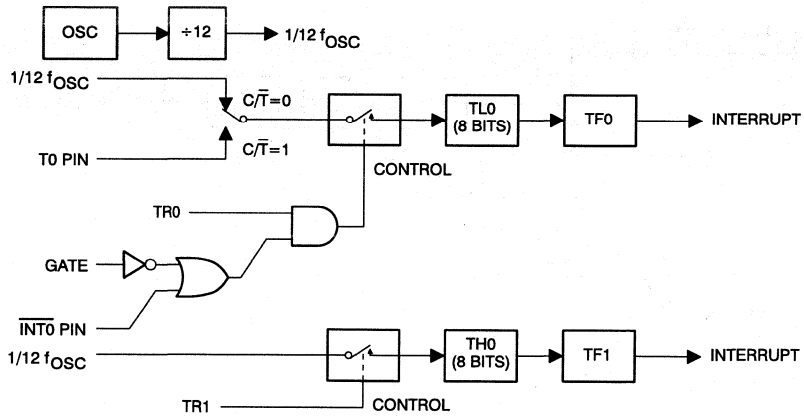


Figure 11. TCON: Timer/Counter Control Register

(MSB)							(LSB)
TF1	TR1	TF0	TR0	IE1	IT1	IE0	IT0

Symbol	Position	Name and Significance
TF1	TCON.7	Timer 1 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR1	TCON.6	Timer 1 run control bit. Set/cleared by software to turn Timer/Counter on/off.
TF0	TCON.5	Timer 0 overflow flag. Set by hardware on Timer/Counter overflow. Cleared by hardware when processor vectors to interrupt routine.
TR0	TCON.4	Timer 0 run control bit. Set/cleared by software to turn Timer/Counter on/off.
IE1	TCON.3	Interrupt 1 edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT1	TCON.2	Interrupt 1 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.
IE0	TCON.1	Interrupt 0 edge flag. Set by hardware when external interrupt edge detected. Cleared when interrupt processed.
IT0	TCON.0	Interrupt 0 type control bit. Set/cleared by software to specify falling edge/low level triggered external interrupts.

Timer 2

Timer 2 is a 16-bit Timer/Counter present only in the AT89C52. This is a powerful addition to the other two just discussed. Five extra special function registers are added to accommodate Timer 2 which are: the timer registers, TL2 and TH2, the timer control register, T2CON, and the capture registers, RCAP2L and RCAP2H. Like Timers 0 and 1, it can operate either as a timer or as an event counter, depending on the value of bit C/$\overline{T2}$ in the Special Function Register T2CON (Figure 12). Timer 2 has three operating modes: capture, auto-reload, and baud rate generator, which are selected by bits in T2CON, as shown in Table 2.

Table 2. Timer 2 Operation Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	Mode
0	0	1	16-bit Auto-Reload
0	1	1	16-bit Capture
1	X	1	Baud Rate Generator
X	X	0	(off)

Figure 12. T2CON Timer/Counter 2 Control Register

(MSB)							(LSB)
TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$

Symbol	Position	Name and Significance
TF2	T2CON.7	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	T2CON.6	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software.
RCLK	T2CON.5	Receive clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in Modes 1, 3 and Timer 1 provides transmit baud rate. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	T2CON.4	Transmit clock flag. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in Modes 1, 3 and Timer 1 provides transmit baud rate. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	T2CON.3	Timer 2 external enable flag. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	T2CON.2	Start/stop control for Timer 2. A logic 1 starts the timer.
C/$\overline{T2}$	T2CON.1	Timer or counter select. (Timer 2) 0 = Internal timer (OSC/12) 1 = External event counter (falling edge triggered).
CP/$\overline{RL2}$	T2CON.0	Capture/Reload flag. When set, captures will occur on negative transitions at T2EX if EXEN2 = 1. When cleared, auto-reloads will occur either with Timer 2 overflows or negative transitions at T2EX when EXEN2 = 1. When either RCLK = 1 or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

In the Capture Mode, the EXEN2 bit in T2CON selects two options. If EXEN2 = 0, then Timer 2 is a 16-bit timer or counter whose overflow sets bit TF2, the Timer 2 overflow bit, which can be used to generate an interrupt. If EXEN2 = 1, then Timer 2 performs the same way, but a l-to-0 transition at external input T2EX also causes the current value in the Timer 2 registers, TL2 and TH2, to be captured into the RCAP2L and RCAP2H registers, respectively. (RCAP2L and RCAP2H are new Special Function Registers in the AT89C52.) In addition, the transition at T2EX sets the EXF2 bit in T2CON, and EXF2, like TF2, can generate an interrupt.

The Capture Mode is illustrated in Figure 13.

In the auto-reload mode, the EXEN2 bit in T2CON also selects two options. If EXEN2 = 0, then when Timer 2 rolls over it sets TF2 and also reloads the Timer 2 registers with the 16-bit value in the RCAP2L and RCAP2H registers, which are preset by software. If EXEN2 = 1, then Timer 2 performs the same way, but a 1-to-0 transition at external input T2EX also triggers the 16-bit reload and sets EXF2.

The auto-reload mode is illustrated in Figure 14.

The baud rate generator mode is selected by RCLK = 1 and/or TCLK = 1. This mode is described in conjunction with the serial port. (Figure 17)

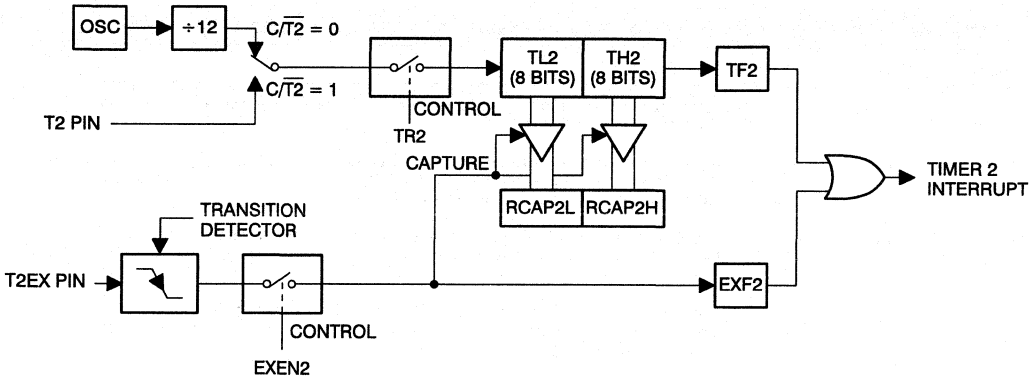
Serial Interface

The serial port is full duplex, which means it can transmit and receive simultaneously. It is also receive-buffered, which means it can begin receiving a second byte before a previously received byte has been read from the receive register. (However, if the first byte still has not been read when reception of the second byte is complete, one of the bytes will be lost.) The serial port receive and transmit registers are both accessed at Special Function Register SBUF. Writing to SBUF loads the transmit register, and reading SBUF accesses a physically separate receive register.

The serial port can operate in the following four modes.

Mode 0: Serial data enters and exits through RXD. TXD outputs the shift clock. Eight data bits are transmitted/received, with the LSB first. The baud rate is fixed at 1/12 the oscillator frequency.

Figure 13. Timer 2 In Capture Mode



Mode 1: 10 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in Special Function Register SCON. The baud rate is variable.

Mode 2: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable ninth data bit, and a stop bit (1). On transmit, the 9th data bit (TB8 in SCON) can be assigned the value of 0 or 1. Or, for example, the parity bit (P, in the PSW) can be moved into TB8. On receive, the 9th data bit goes into RB8 in Special Function Register SCON, while the stop bit is ignored. The baud rate is programmable to either 1/32 or 1/64 the oscillator frequency.

Mode 3: 11 bits are transmitted (through TXD) or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable ninth data bit, and a stop bit (1). In fact, Mode 3 is the same as Mode 2 in all respects except the baud rate, which is variable in Mode 3.

In all four modes, transmission is initiated by any instruction that uses SBUF as a destination register. Reception is initiated in Mode 0 by the condition RI = 0 and REN = 1. Reception is initiated in the other modes by the incoming start bit if REN = 1.

Multiprocessor Communications

Modes 2 and 3 have a special provision for multiprocessor communications. In these modes, 9 data bits are received, followed by a stop bit. The ninth bit goes into RB8. Then comes a stop bit. The port can be programmed such that when the stop bit is received, the serial port interrupt is activated only if RB8 = 1. This feature is enabled by setting bit SM2 in SCON.

The following example shows how to use the serial interrupt for multiprocessor communications. When the master processor must transmit a block of data to one of several slaves, it first sends out an address byte that identifies the target slave. An address byte differs from a data byte in that the 9th bit is 1 in an address byte and 0 in a data byte. With SM2 = 1, no slave is interrupted by a data byte. An address byte, however, interrupts all slaves, so that each slave can examine the received byte and see if it is being addressed. The addressed slave clears its SM2 bit and prepares to receive the data bytes that follows. The slaves that are not addressed set their SM2 bits and ignore the data bytes.

SM2 has no effect in Mode 0 but can be used to check the validity of the stop bit in Mode 1. In a Mode 1 reception, if SM2 = 1, the receive interrupt is not activated unless a valid stop bit is received.

Serial Port Control Register

The serial port control and status register is the Special Function Register SCON, shown in Figure 15. This register contains the mode selection bits, the 9th data bit for transmit and receive (TB8 and RB8), and the serial port interrupt bits (TI and RI).

Figure 14. Timer 2 in Auto-Reload Mode

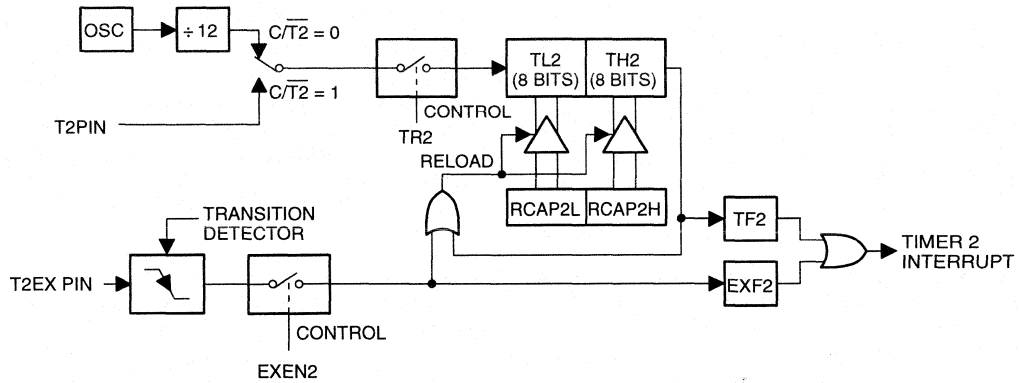


Figure 15. SCON: Serial Port Control Register

(MSB)							(LSB)
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Position	Symbol	Name and Significance
SCON.7	SM0	Serial port mode bit 0 (see table below).
SCON.6	SM1	Serial port mode bit 1 (see table below).
SCON.5	SM2	Enables the multiprocessor communication feature in Modes 2 and 3. In Mode 2 or 3, if SM2 is set to 1, then RI will not be activated if the received 9th data bit (RB8) is 0. In Mode 1, if SM2 = 1, then RI will not be activated if a valid stop bit was not received. In Mode 0, SM2 should be 0.
SCON.4	REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
SCON.3	TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software.
SCON.2	RB8	In Modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
SCON.1	TI	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
SCON.0	RI	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

Where SM0, SM1 specify the serial port mode, as follows:

SM0	SM1	Mode	Description	Baud Rate
0	0	0	shift register	fixed ($f_{osc.}$/12)
0	1	1	8-bit UART	variable (set by timer)
1	0	2	9-bit UART	fixed ($f_{osc.}$/64 or $f_{osc.}$/32)
1	1	3	9-bit UART	variable (set by timer)

Baud Rates

The baud rate in Mode 0 is fixed as shown in the following equation.

$$\text{Mode 0 Baud Rate} = \frac{\text{Oscillator Frequency}}{12}$$

The baud rate in Mode 2 depends on the value of the SMOD bit in Special Function Register PCON. If SMOD = 0 (the value on reset), the baud rate is 1/64 of the oscillator frequency. If SMOD = 1, the baud rate is 1/32 of the oscillator frequency, as shown in the following equation.

$$\text{Mode 2 Baud Rate} = \frac{2^{\text{SMOD}}}{64} \times (\text{Oscillator Frequency})$$

In the AT89C51, the Timer 1 overflow rate determines the baud rates in Modes 1 and 3. In the AT89C52, these baud rates can be determined by Timer 1, by Timer 2, or by both (one for transmit and the other for receive).

Using Timer 1 to Generate Baud Rates

When Timer 1 is the baud rate generator, the baud rates in Modes 1 and 3 are determined by the Timer 1 overflow rate and the value of SMOD according to the following equation.

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{\text{SMOD}}}{32} \times (\text{Timer 1 Overflow Rate})$$

The Timer 1 interrupt should be disabled in this application. The Timer itself can be configured for either timer or counter operation in any of its 3 running modes. In the most typical applications, it is configured for timer operation in auto-reload mode (high nibble of TMOD = 0010B). In this case, the baud rate is given by the the following formula.

$$\text{Modes 1, 3 Baud Rate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{Oscillator Frequency}}{12 \times [256 - (\text{TH1})]}$$

Programmers can achieve very low baud rates with Timer 1 by leaving the Timer 1 interrupt enabled, configuring the Timer to run as a 16-bit timer (high nibble of TMOD = 0001B), and using the Timer 1 interrupt to do a 16-bit software reload.

Figure 16 lists commonly used baud rates and how they can be obtained from Timer 1.

Using Timer 2 to Generate Baud Rates

In the AT89C52, setting TCLK and/or RCLK in T2CON selects Timer 2 as the baud rate generator (Figure 11). Under these conditions, the baud rates for transmit and receive can be simultaneously different. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 17.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 reloads the Timer 2 registers with the 16-bit value in the RCAP2H and RCAP2L registers, which are preset by software.

In this case, the baud rates in Modes 1 and 3 are determined by the Timer 2 overflow rate according to the following equation.

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

Timer 2 can be configured for either timer or counter operation. In the most typical applications, it is configured for timer operation ($C/\overline{T2} = 0$). Normally, a timer increments every machine cycle (thus at 1/12 the oscillator frequency), but timer operation is a different for Timer 2 when it is used as a baud rate generator. As a baud rate generator, Timer 2 increments every state time (thus at 1/2 the oscillator frequency). In this case, the baud rate is given by the following formula.

$$\text{Modes 1, 3 Baud Rate} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Figure 16. Commonly Used Baud Rates Generated by Timer 1

Baud Rate	fosc	SMOD	Timer 1		
			C/$\overline{T}$	Mode	Reload Value
Mode 0 Max: 1 MHz	12 MHz	X	X	X	X
Mode 2 Max: 375K	12 MHz	1	X	X	X
Modes 1, 3: 62.5K	12 MHz	1	0	2	FFH
19.2K	11.059 MHz	1	0	2	FDH
9.6K	11.059 MHz	0	0	2	FDH
4.8K	11.059 MHz	0	0	2	FAH
2.4K	11.059 MHz	0	0	2	F4H
1.2K	11.059 MHz	0	0	2	E8H
137.5	11.986 MHz	0	0	2	1DH
110	6 MHz	0	0	2	72H
110	12 MHz	0	0	1	FEEBH

Figure 17. Timer 2 in Baud Rate Generator Mode

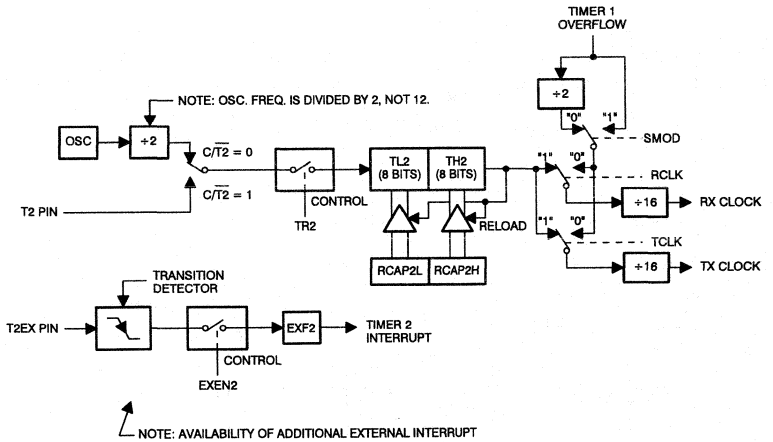


Figure 17 shows Timer 2 as a baud rate generator. This figure is valid only if RCLK + TCLK = 1 in T2CON. A rollover in TH2 does not set TF2 and does not generate an interrupt. Therefore, the Timer 2 interrupt does not have to be disabled when Timer 2 is in the baud rate generator mode. If EXEN2 is set, a l-to-0 transition in T2EX sets EXF2 but does not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus, when Timer 2 is used as a baud rate generator, T2EX can be used as an extra external interrupt.

When Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, programmers should not read from or write to TH2 or TL2. Under these conditions, Timer 2 is incremented every state time, and the results of a read or write may not be accurate. The RCAP registers may be read, but should not be written to, because a write might overlap a reload and cause write and/or reload errors. Turn Timer 2 off (clear TR2) before accessing the Timer 2 or RCAP registers, in this case.

More About Mode 0

Serial data enters and exits through RXD. TXD outputs the shift clock. Eight data bits are transmitted/received, with the LSB first. The baud rate is fixed at 1/12 the oscillator frequency.

Figure 18 shows a simplified functional diagram of the serial port in Mode 0 and associated timing.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF' signal at S6P2 also loads a 1 into the ninth position of the transmit shift register and tells the TX Control block to begin a transmission. The internal timing is such that one full machine cycle will elapse between "write to SBUF" and activation of SEND.

SEND transfers the output of the shift register to the alternate output function line of P3.0, and also transfers SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK is low during S3, S4, and S5 of every machine cycle, and high during S6, S1, and S2. At S6P2 of every machine cycle in which SEND is active, the contents of the transmit shift register are shifted one position to the right.

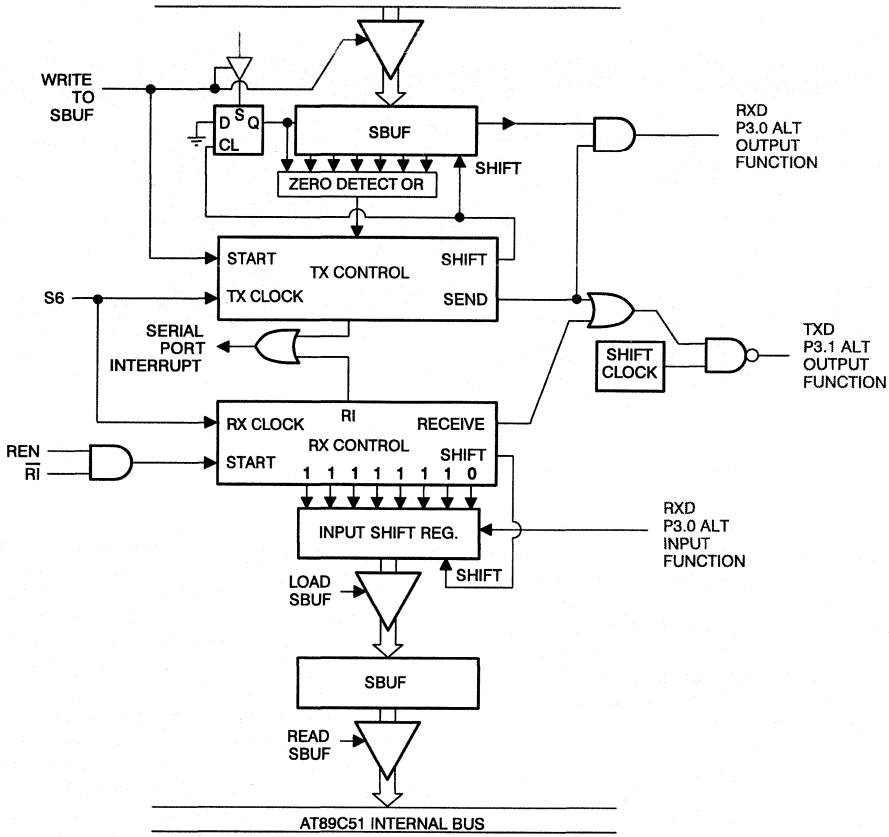
As data bits shift out to the right, 0s come in from the left. When the MSB of the data byte is at the output position of the shift register, the 1 that was initially loaded into the ninth position is just to the left of the MSB, and all positions to the left of that contain 0s. This condition flags the TX Control block to do one last shift, then deactivate SEND and set TI. Both of these actions occur at S1P1 of the tenth machine cycle after "write to SBUF."

Reception is initiated by the condition REN = 1 and R1 = 0. At S6P2 of the next machine cycle, the RX Control unit writes the bits 11111110 to the receive shift register and activates RECEIVE in the next clock phase.

RECEIVE enables SHIFT CLOCK to the alternate output function line of P3.1. SHIFT CLOCK makes transitions at S3P1 and S6P1 of every machine cycle. At S6P2 of every machine cycle in which RECEIVE is active, the contents of the receive shift register are shifted one position to the left. The value that comes in from the right is the value that was sampled at the P3.0 pin at S5P2 of the same machine cycle.

As data bits come in from the right, 1s shift out to the left. When the 0 that was initially loaded into the right-most position arrives at the left-most position in the shift register, it flags the RX Control block to do one last shift and load SBUF. At S1P1 of the l0th machine cycle after the write to SCON that cleared RI, RECEIVE is cleared and RI is set.

Figure 18. Serial Port Mode 0



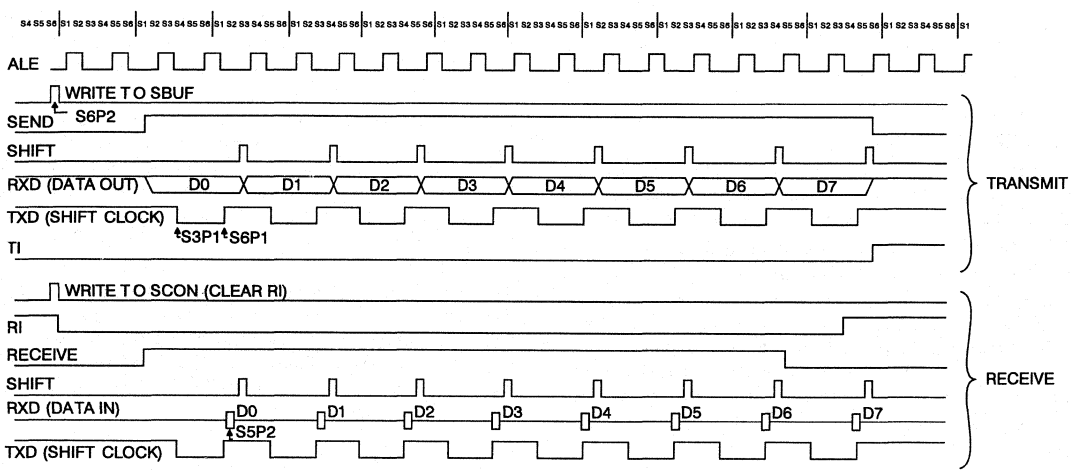
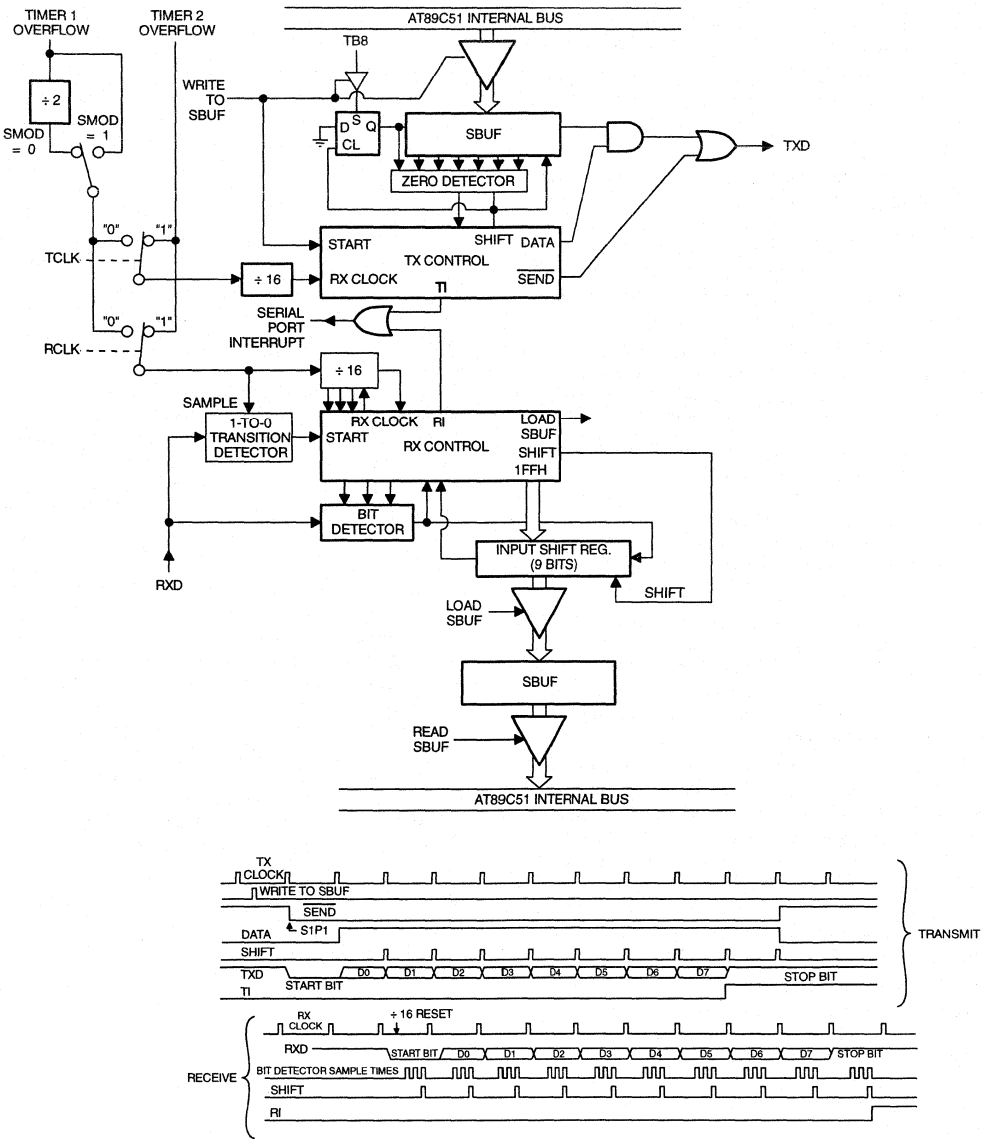


Figure 19. Serial Port Mode 1. TCLK, RCLK and Timer 2 are Present In the AT89C52 Only.



More About Mode 1

Ten bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), and a stop bit (1). On receive, the stop bit goes into RB8 in SCON. In the AT89C51, the baud rate is determined by the Timer 1 overflow rate. In the AT89C52 the baud rate is determined either by the Timer 1 overflow rate, the Timer 2 overflow rate, or both . In this case, one Timer is for transmit, and the other is for receive.

Figure 19 shows a simplified functional diagram of the serial port in Mode 1 and associated timings for transmit and receive.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads a 1 into the ninth bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission actually commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal.

The transmission begins when $\overline{\text{SEND}}$ is activated, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that.

As data bits shift out to the right, 0s are clocked in from the left. When the MSB of the data byte is at the output position of the shift register, the 1 that was initially loaded into the ninth position is just to the left of the MSB, and all positions to the left of that contain 0s. This condition flags the TX Control unit to do one last shift, then deactivate $\overline{\text{SEND}}$ and set TI. This occurs at the tenth divide-by-16 rollover after "write to SBUF."

Reception is initiated by a 1-to-0 transition detected at RXD. For this purpose, RXD is sampled at a rate of 16 times the established baud rate. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written into the input shift register. Resetting the divide-by-16 counter aligns its rollovers with the boundaries of the incoming bit times.

The 16 states of the counter divide each bit time into 16ths. At the seventh, eighth, and ninth counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of the 3 samples. This is done to reject noise. In order to reject false bits, if the value accepted during the first bit time is not 0, the receive circuits are reset and the unit continues looking for another 1-to-0 transition. If the start bit is valid, it is shifted into the input shift register, and reception of the rest of the frame proceeds.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register, (which is a 9-bit register in mode 1), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8 and to set RI is generated if, and only if, the following conditions are met at the time the final shift pulse is generated.

1) RI = 0 and

2) Either SM2 = 0, or the received stop bit = 1

If either of these two conditions is not met, the received frame is irretrievably lost. If both conditions are met, the stop bit goes into RB8, the 8 data bits go into SBUF, and RI is activated. At this time, whether or not the above conditions are met, the unit continues looking for a 1-to-0 transition in RXD.

More About Modes 2 and 3

Eleven bits are transmitted (through TXD), or received (through RXD): a start bit (0), 8 data bits (LSB first), a programmable ninth data bit, and a stop bit (1). On transmit, the ninth data bit (TB8) can be assigned the value of 0 or 1. On receive, the ninth data bit goes into RB8 in SCON. The baud rate is programmable to either 1/32 or 1/64 of the oscillator frequency in Mode 2. Mode 3 may have a variable baud rate generated from either Timer 1 or 2, depending on the state of TCLK and RCLK.

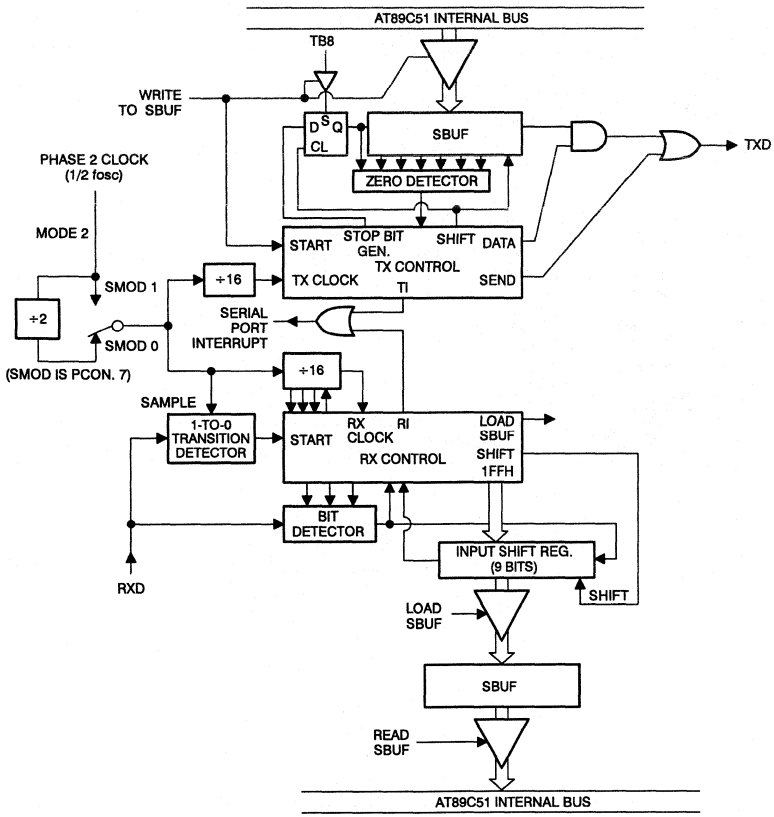
Figures 20 and 21 show a functional diagram of the serial port in Modes 2 and 3. The receive portion is exactly the same as in Mode 1. The transmit portion differs from Mode 1 only in the ninth bit of the transmit shift register.

Transmission is initiated by any instruction that uses SBUF as a destination register. The "write to SBUF" signal also loads TB8 into the ninth bit position of the transmit shift register and flags the TX Control unit that a transmission is requested. Transmission commences at S1P1 of the machine cycle following the next rollover in the divide-by-16 counter. Thus, the bit times are synchronized to the divide-by-16 counter, not to the "write to SBUF" signal.

The transmission begins when $\overline{\text{SEND}}$ is activated, which puts the start bit at TXD. One bit time later, DATA is activated, which enables the output bit of the transmit shift register to TXD. The first shift pulse occurs one bit time after that. The first shift clocks a 1 (the stop bit) into the ninth bit position of the shift register. Thereafter, only 0s are clocked in. Thus, as data bits shift out to the right, 0s are clocked in from the left. When TB8 is at the output position of the shift register, then the stop bit is just to the left of TB8, and all positions to the left of that contain 0s. This condition flags the TX Control unit to do one last shift, then deactivate SEND and set TI. This occurs at the 11th divide-by-16 rollover after "write to SBUF."

Reception is initiated by a 1-to-0 transition detected at RXD. For this purpose, RXD is sampled at a rate of 16 times the established baud rate. When a transition is detected, the divide-by-16 counter is immediately reset, and 1FFH is written to the input shift register.

Figure 20. Serial Port Mode 2



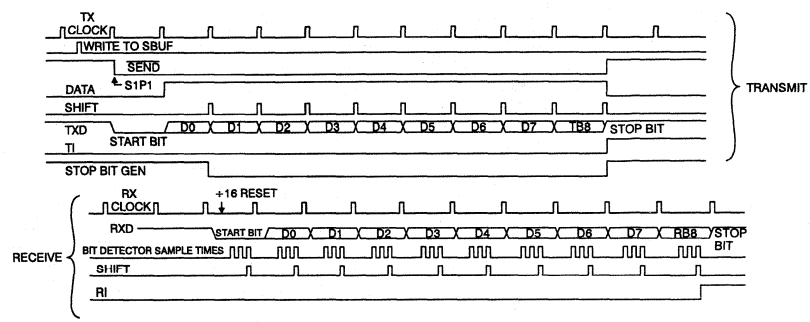
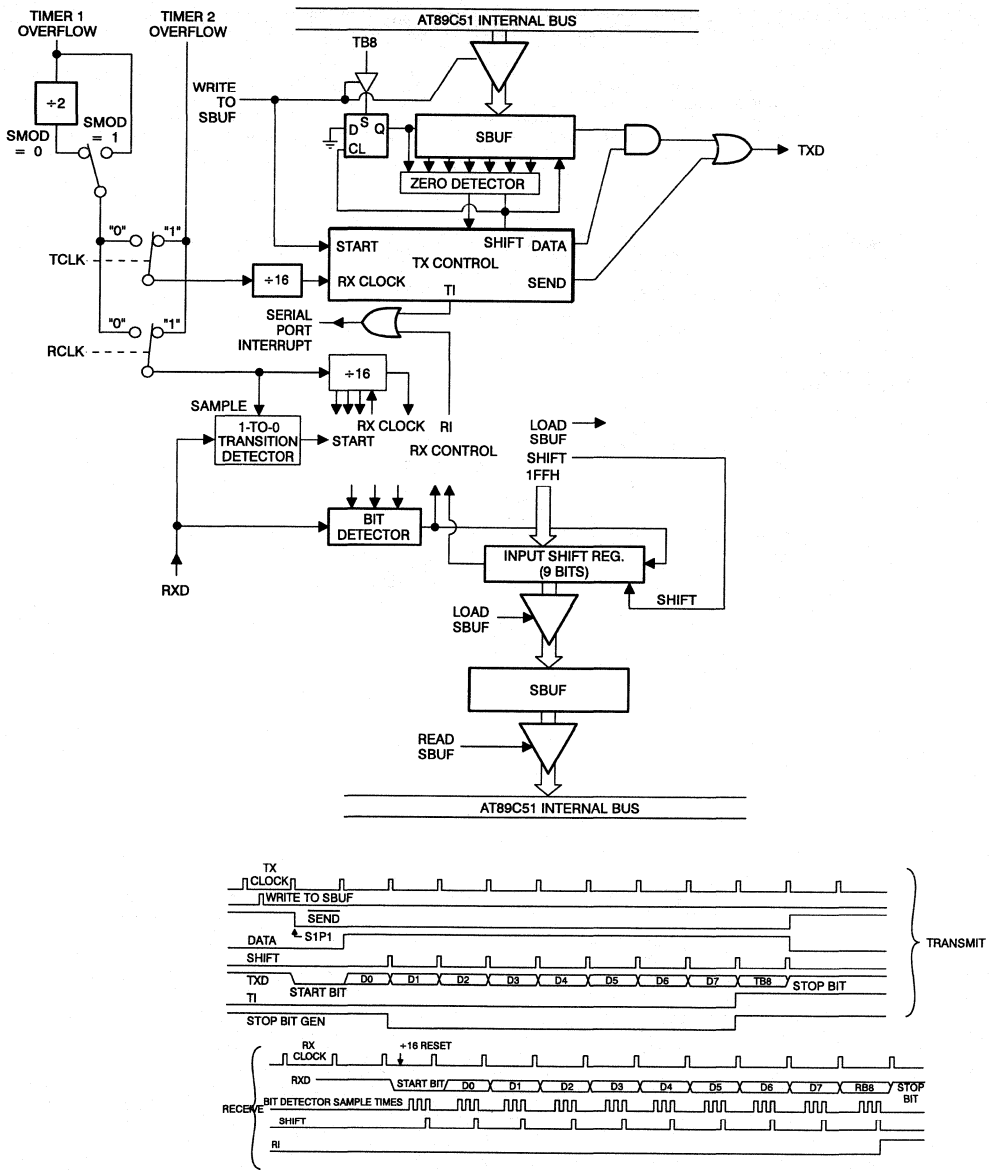


Figure 21. Serial Port Mode 3. TCLK, RCLK, and Timer 2 are Present in AT89C52 only



At the seventh, eighth and ninth counter states of each bit time, the bit detector samples the value of RXD. The value accepted is the value that was seen in at least 2 of the 3 samples. If the value accepted during the first bit time is not 0, the receive circuits are reset and the unit continues looking for another l-to-0 transition. If the start bit proves valid, it is shifted into the input shift register, and reception of the rest of the frame proceeds.

As data bits come in from the right, 1s shift out to the left. When the start bit arrives at the leftmost position in the shift register (which in Modes 2 and 3 is a 9-bit register), it flags the RX Control block to do one last shift, load SBUF and RB8, and set RI. The signal to load SBUF and RB8 and to set RI is generated if, and only if, the following conditions are met at the time the final shift pulse is generated:

1) RI = 0, and

2) Either SM2 = 0 or the received 9th data bit = 1

If either of these conditions is not met, the received frame is irretrievably lost, and RI is not set. If both conditions are met, the received ninth data bit goes into RB8, and the first 8 data bits go into SBUF. One bit time later, whether the above conditions were met or not, the unit continues looking for a 1-to-0 transition at the RXD input.

Note that the value of the received stop bit is irrelevant to SBUF, RB8, or RI.

Interrupts

The AT89C51 provides 5 interrupt sources: two external interrupts, two timer interrupts, and a serial port interrupt. The AT89C52 provides 6 with the extra timer. These are shown in Figure 22.

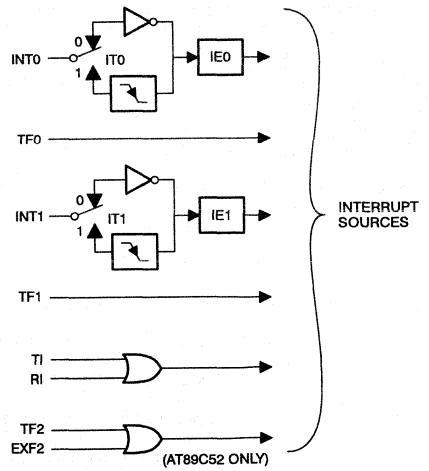
The External Interrupts $\overline{INT0}$ and $\overline{INT1}$ can each be either level-activated or transition-activated, depending on bits IT0 and IT1 in Register TCON. The flags that actually generate these interrupts are the IE0 and IE1 bits in TCON. When the service routine is vectored to, hardware clears the flag that generated an external interrupt *only* if the interrupt was transition-activated. If the interrupt was level-activated, then the external requesting source (rather than the on-chip hardware) controls the request flag.

The Timer 0 and Timer 1 Interrupts are generated by TF0 and TF1, which are set by a rollover in their respective Timer/Counter registers (except for Timer 0 in Mode 3). When a timer interrupt is generated, the on-chip hardware clears the flag that generated it when the service routine is vectored to.

The Serial Port Interrupt is generated by the logical OR of RI and TI. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine normally must determine whether RI or TI generated the interrupt, and the bit must be cleared in software.

In the AT89C52, the Timer 2 Interrupt is generated by the logical OR of TF2 and EXF2. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether TF2 or EXF2 generated the interrupt, and the bit must be cleared in software.

Figure 22. Interrupt Sources



All of the bits that generate interrupts can be set or cleared by software, with the same result as though they had been set or cleared by hardware. That is, interrupts can be generated and pending interrupts can be canceled in software.

Figure 23. IE: Interrupt Enable Register

(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable bit = 1 enables the interrupt.

Enable bit = 0 disables it.

Symbol	Position	Function
EA	IE.7	Global enable/disable. disables all interrupts. If EA = 0, no interrupt will be acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Undefined/reserved.
ET2	IE.5	Timer 2 interrupt enable bit. (AT89C52)
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External Interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.

User software should never write 1s to unimplemented bits, since they may be used in future AT89 Series products.

Figure 24. IP: Interrupt Priority Register

(MSB)							(LSB)
—	—	PT2	PS	PT1	PX1	PT0	PX0

Priority bit = 1 assigns high priority.

Priority bit = 0 assigns low priority.

Symbol	Position	Function
—	IP.7	reserved
—	IP.6	reserved
PT2	IP.5	Timer 2 interrupt priority bit.
PS	IP.4	Serial Port interrupt priority bit.
PT1	IP.3	Timer 1 interrupt priority bit.
PX1	IP.2	External Interrupt 1 priority bit.
PT0	IP.1	Timer 0 interrupt priority bit.
PX0	IP.0	External interrupt 0 priority bit.

User software should never write 1s to unimplemented bits, since they may be used in future products.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE (interrupt enable) at address OA8H. As well as individual enable bits for each interrupt source, there is a global enable/disable bit that is cleared to disable all interrupts or set to turn on interrupts (see Figure 23).

Figure 23 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future microcontrollers.

Priority Level Structure

Each interrupt source can also be individually programmed to one of two priority levels by setting or clearing a bit in Special Function Register IP (interrupt priority) at address 0B8H (Figure 24). IP is cleared after a system reset to place all interrupts at the lower priority level by default. A low-priority interrupt can be interrupted by a high-priority interrupt but not by another low-priority interrupt. A high-priority interrupt can not be interrupted by any other interrupt source.

If two requests of different priority levels are received simultaneously, the request of higher priority level is serviced. If requests of the **same** priority level are received simultaneously, an internal polling sequence determines which request is serviced. Thus within each priority level there is a second priority structure determined by the polling sequence, as follows.

	Source	Priority Within Level
1.	IE0	(highest)
2.	TF0	
3.	IE1	
4.	TF1	
5.	RI + TI	
6.	TF2 + EXF2	(lowest)

Note that the "priority within level" structure is only used to resolve *simultaneous requests of the same priority level.*

The IP register contains a number of unimplemented bits. IP.7 and IP.6 are vacant in the AT89C52, and in the AT89C51 these bits and IP.5 are vacant. User software should not write 1s to these bit positions, since they may be used in future products.

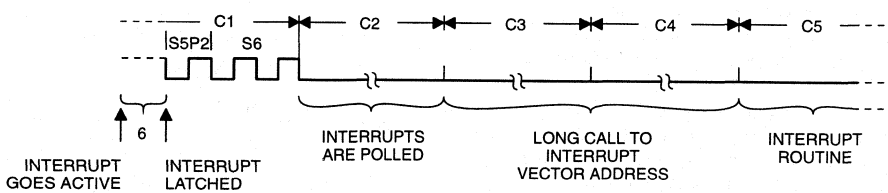
How Interrupts Are Handled

The interrupt flags are sampled at S5P2 of every machine cycle. The samples are polled during the following machine cycle. The AT89C52 Timer 2 interrupt cycle is different, as described in the Response Time Section. If one of the flags was in a set condition at S5P2 of the preceding cycle, the polling cycle will find it and the interrupt system will generate an LCALL to the appropriate service routine, provided this hardware generated LCALL is not blocked by any of the following conditions.

1. An interrupt of equal or higher priority level is already in progress.
2. The current (polling) cycle is not the final cycle in the execution of the instruction in progress.
3. The instruction in progress is RETI or any write to the IE or IP registers.

Any of these three conditions will block the generation of the LCALL to the interrupt service routine. Condition 2 ensures that the instruction in progress will be completed before vectoring to

Figure 25. Interrupt Response Timing Diagram



This is the fastest possible response when C2 is the final cycle of an instruction other than RETI or an access to IE or IP.

any service routine. Condition 3 ensures that if the instruction in progress is RETI or any access to IE or IP, then at least *one more* instruction will be executed before any interrupt is vectored to.

The polling cycle is repeated with each machine cycle, and the values polled are the values that were present at S5P2 of the previous machine cycle. If an active interrupt flag is not being serviced because of one of the above conditions and is not *still* active when the blocking condition is removed, the denied interrupt will not be serviced. In other words, the fact that the interrupt flag was once active but not serviced is not remembered. Every polling cycle is new.

The polling cycle/LCALL sequence is illustrated in Figure 25.

Note that if an interrupt of higher priority level goes active prior to S5P2 of the machine cycle labeled C3 in Figure 25, then in accordance with the above rules it will be serviced during C5 and C6, without any instruction of the lower priority routine having been executed.

Thus, the processor acknowledges an interrupt request by executing a hardware-generated LCALL to the appropriate servicing routine. In some cases it also clears the flag that generated the interrupt, and in other cases it does not. It never clears the Serial Port or Timer 2 flags. This must be done in the user's software. The processor clears an external interrupt flag (IE0 or IE1) only if it was transition-activated. The hardware-generated LCALL pushes the contents of the Program Counter onto the stack (but it does not save the PSW) and reloads the PC with an address that depends on the source of the interrupt being serviced, as shown in the following table.

Interrupt	Source	Vector Address
External 0	IE0	0003H
Timer 0	TF0	000BH
External 1	IE1	0013H
Timer 1	TF1	001BH
Serial Port	RI or TI	0023H
Timer 2	TF2 or EXF2	002BH
System Reset	RST	0000H

NOTE: When vectoring to an interrupt the flag that caused the interrupt is automatically cleared by hardware. The exceptions are RI and TI for serial port interrupts, and TF2 and EXF2 for Timer 2 interrupts. Since there are two possible sources for each of these interrupts, it is not practical for the CPU to clear the interrupt flag. These bits must be tested in the ISR to determine the source of the interrupt, and then the interrupting flag is cleared by software.

Execution proceeds from that location until the RETI instruction is encountered. The RETI instruction informs the processor that this interrupt routine is no longer in progress, then pops the top two bytes from the stack and reloads the Program Counter. Execution of the interrupted program continues from where it left off.

Note that a simple RET instruction would also have returned execution to the interrupted program, but it would have left the interrupt control system thinking an interrupt was still in progress.

Interrupt Flag Bits

Interrupt	Flag	SFR Register and Bit Position
External 0	IE0	TCON.1
External 1	IE1	TCON.3
Timer 1	TF1	TCON.7
Timer 0	TF0	TCON.5
Serial port	TI	SCON.1
Serial port	RI	SCON.0
Timer 2	TF2	T2CON.7 (AT89C52)
Timer 2	EXF2	T2CON.6 (AT89C52)

When an interrupt is accepted the following action occurs:

1. The current instruction completes operation.
2. The PC is saved on the stack.
3. The current interrupt status is saved internally.
4. Interrupts are blocked at the level of the interrupts.
5. The PC is loaded with the vector address of the ISR (interrupt service routine).
6. The ISR executes.

The ISR executes and takes action in response to the interrupt. The ISR finishes with RETI (return from interrupt) instruction. This retrieves the old value of the PC from the stack and restores the old interrupt status. Execution of the main program continues where it left off.

External Interrupts

The external sources can be programmed to be level-activated or transition-activated by setting or clearing bit IT1 or IT0 in Register TCON. If ITx = 0, external interrupt x is triggered by a detected low at the INTx pin. If ITx = 1, external interrupt x is edge-triggered. In this mode if successive samples of the $\overline{\text{INTx}}$ pin show a high in one cycle and a low in the next cycle, interrupt request flag IEx in TCON is set. Flag bit IEx then requests the interrupt.

Since the external interrupt pins are sampled once each machine cycle, an input high or low should hold for at least 12 oscillator periods to ensure sampling. If the external interrupt is transition-activated, the external source has to hold the request pin high for at least one machine cycle, and then hold it low for at least one machine cycle to ensure that the transition is seen so that interrupt request flag IEx will be set. IEx will be automatically cleared by the CPU when the service routine is called.

If the external interrupt is level-activated, the external source has to hold the request active until the requested interrupt is actually generated. Then the external source must deactivate the request before the interrupt service routine is completed, or else another interrupt will be generated.

Response Time

The $\overline{INT0}$ and $\overline{INT1}$ levels are inverted and latched into the interrupt flags IE0 and IE1 at S5P2 of every machine cycle. Similarly, the Timer 2 flag EXF2 and the Serial Port flags RI and TI are set at S5P2. The values are not actually polled by the circuitry until the next machine cycle.

The Timer 0 and Timer 1 flags, TF0 and TF1, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag TF2 is set at S2P2 and is polled in the same cycle in which the timer overflows.

If a request is active and conditions are right for it to be acknowledged, a hardware subroutine call to the requested service routine will be the next instruction executed. The call itself takes two cycles. Thus, a minimum of three complete machine cycles elapsed between activation of an external interrupt request and the beginning of execution of the first instruction of the service routine. Figure 25 shows interrupt response timings.

A longer response time results if the request is blocked by one of the 3 previously listed conditions. If an interrupt of equal or higher priority level is already in progress, the additional wait time depends on the nature of the other interrupt's service routine. If the instruction in progress is not in its final cycle, the additional wait time cannot be more than 3 cycles, since the longest instructions (MUL and DIV) are only 4 cycles long. If the instruction in progress is RETI or an access to IE or IP, the additional wait time cannot be more than 5 cycles (a maximum of one more cycle to complete the instruction in progress, plus 4 cycles to complete the next instruction if the instruction is MUL or DIV).

Thus, in a single-interrupt system, the response time is always more than 3 cycles and less than 9 cycles.

Single-Step Operation

The AT89C51 interrupt structure allows single-step execution with very little software overhead. As previously noted, an interrupt request will not be serviced while an interrupt of equal priority level is still in progress, nor will it be serviced after RETI until at least one other instruction has been executed. Thus, once an interrupt routine has been entered, it cannot be re-entered until at least one instruction of the interrupted program is executed. One way to use this feature for single-stop operation is to program one of the external interrupts (for example, $\overline{INT0}$) to be level-activated. The service routine for the interrupt will terminate with the following code.

```
JNB     P3.2,$      ;Wait Here Till INT0 Goes High
JB      P3.2,$      ;Now Wait Here Till it Goes Low
RETI                ;Go Back and Execute One Instruction
```

If the $\overline{INT0}$ pin, which is also the P3.2 pin, is held normally low, the CPU will go right into the External Interrupt 0 routine and stay there until $\overline{INT0}$ is pulsed (from low to high to low). Then it will execute RETI, go back to the task program, execute one instruction, and immediately reenter the External Interrupt 0 routine to await the next pulsing of P3.2. One step of the task program is executed each time P3.2 is pulsed.

Reset

The reset input is the RST pin, which is the input to a Schmitt Trigger.

A reset is accomplished by holding the RST pin high for at least two machine cycles (24 oscillator periods), *while the oscillator is running*. The CPU responds by generating an internal reset, with the timing shown in Figure 26.

The external reset signal is asynchronous to the internal clock. The RST pin is sampled during State 5 Phase 2 of every machine cycle. The port pins will maintain their current activities for 19 oscillator periods after a logic 1 has been sampled at the RST pin; that is, for 19 to 31 oscillator periods after the external reset signal has been applied to the RST pin.

While the RST pin is high, ALE and $\overline{PSEN}$ are weakly pulled high. After RST is pulled low, it will take 1 to 2 machine cycles for ALE and $\overline{PSEN}$ to start clocking. For this reason, other devices can not be synchronized to the internal timings of the AT89C51.

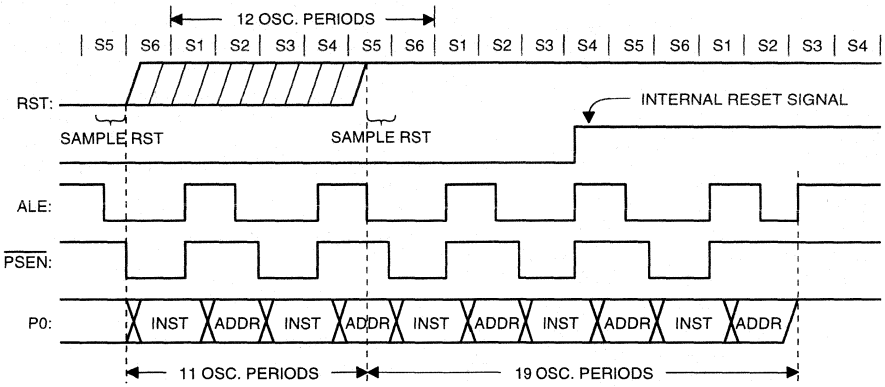
Driving the ALE and $\overline{PSEN}$ pins to 0 while reset is active could cause the device to go into an indeterminate state.

The internal reset algorithm writes 0s to all the SFRs except the port latches, the Stack Pointer, and SBUF. The port latches are initialized to FFH, the Stack Pointer to 07H, and SBUF is indeterminate. Table 3 lists the SFRs and their reset values.

The internal RAM is not affected by reset. On power-up the RAM content is indeterminate.

Note: There is no internal pulldown reset pin on NMOS devices, unlike that of Atmel's CMOS microcontroller devices.

Figure 26. Reset Timing



Power-On Reset

For CMOS devices, the external resistor can be removed because the RST pin has an internal pulldown. The capacitor value can then be reduced to 1 µF in Figure 27.

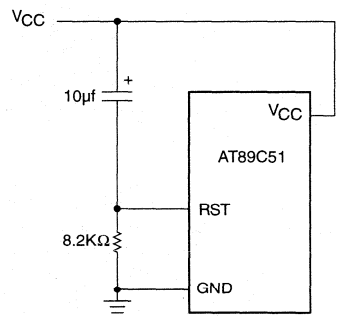
When power is turned on, the circuit holds the RST pin high for an amount of time that depends on the capacitor value and the rate at which it charges. To ensure a valid reset, the RST pin must be held high long enough to allow the oscillator to start up plus two machine cycles.

On power-up, V_{CC} should rise within approximately 10 ms. The oscillator start-up time depends on the oscillator frequency. For a 10 MHz crystal, the start-up time is typically 1 ms. For a 1 MHz crystal, the start-up time is typically 10 ms.

With the given circuit, reducing V_{CC} quickly to 0 causes the RST pin voltage to momentarily fall below 0 V. However, this voltage is internally limited and will not harm the device.

Note: The port pins will be in a random state until the oscillator has started and the internal reset algorithm has written 1s to them.
Powering up the device without a valid reset could cause the CPU to start executing instructions from an indeterminate location. This is because the SFRs, specifically the Program Counter, may not get properly initialized.

Figure 27. Power-On Reset Circuit



Power-Saving Modes of Operation

The Atmel Microcontrollers have two power-reducing modes, Idle and Power Down. The input through which backup power is supplied during these operations is V_{CC}. Figure 28 shows the internal circuitry which implements these features. In the Idle mode (IDL = 1), the oscillator continues to run and the Interrupt, Serial Port, and Timer blocks continue to be clocked, but the clock signal is gated off to the CPU. In Power Down (PD = 1), the oscillator is frozen. The Idle and Power Down modes are activated by setting bits in Special Function Register PCON. The address of this register is 87H. Figure 29 details its contents.

Table 3. Reset Values of the SFRs

SFR Name	Reset Value
PC	0000H
ACC	00H
B	00H
PSW	00H
SP	07H
DPTR	0000H
P0-P3	FFH
IP (AT89C51)	XXX00000B
IP (AT89C52)	XX000000B
IE (AT89C51)	0XX00000B
IE (AT89C52)	0X000000B
TMOD	00H
T2MOD (AT89C52)	XXXXXX00B
TCON	00H
T2CON (AT89C52)	00H
TH0	00H
TL0	00H
TH1	00H
TL1	00H
TH2 (AT89C52)	00H
TL2 (AT89C52)	00H
RCAP2H (AT89C52)	00H
RCAP2L (AT89C52)	00H
SCON	00H
SBUF	Indeterminate
PCON (CHMOS)	0XXX0000B

Idle Mode

An instruction that sets PCON.0 is the last instruction executed before the Idle mode begins. In the Idle mode, the internal clock signal is gated off to the CPU, but not to the Interrupt, Timer, and Serial Port functions. The CPU status is preserved in its entirety: the Stack Pointer, Program Counter, Program Status Word, Accumulator, and all other registers maintain their data during Idle. The port pins hold the logical states they had at the time Idle was activated. ALE and $\overline{PSEN}$ hold at logic high levels.

There are two ways to terminate the Idle. Activation of any enabled interrupt will cause PCON.0 to be cleared by hardware, terminating the Idle mode. The interrupt will be serviced, and following RETI the next instruction to be executed will be the one following the instruction that put the device into Idle.

The flag bits GF0 and GF1 can be used to indicate whether an interrupt occurred during normal operation or during an Idle. For example, an instruction that activates Idle can also set one or both flag bits. When Idle is terminated by an interrupt, the interrupt service routine can examine the flag bits.

The other way of terminating the Idle mode is with a hardware reset. Since the clock oscillator is still running, the hardware reset must be held active for only two machine cycles (24 oscillator periods) to complete the reset.

The signal at the RST pin clears the IDL bit directly and asynchronously. At this time, the CPU resumes program execution from where it left off; that is, at the instruction following the one that invoked the Idle Mode. As shown in Figure 26, two or three machine cycles of program execution may take place before the internal reset algorithm takes control. On-chip hardware inhibits access to the internal RAM during this time, but access to the port pins is not inhibited. To eliminate the possibility of unexpected outputs at the port pins, the instruction following the one that invokes Idle should not write to a port pin or to external data RAM.

Figure 28. Idle and Power-Down Hardware

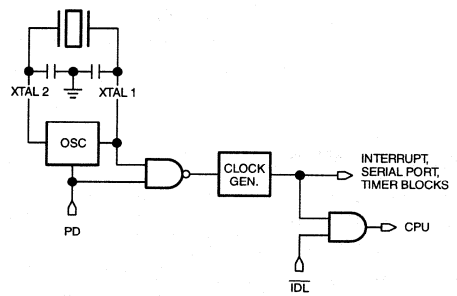


Figure 29. PCON Power Control Register

(MSB)							(LSB)
SMOD	—	—	—	GF1	GF0	PD	IDL

Symbol	Position	Function
SMOD	PCON.7	Double Baud rate bit. When set to a 1 and Timer 1 is used to generate baud rate, and the Serial Port is used in modes 1, 2, or 3.
—	PCON.6	(Reserved)
—	PCON.5	(Reserved)
—	PCON.4	(Reserved)
GF1	PCON.3	General-purpose flag bit.
GF0	PCON.2	General-purpose flag bit.
PD	PCON.1	Power Down bit. Setting this bit activates power down operation.
IDL	PCON.0	Idle mode bit. Setting this bit activates idle mode operation.

If 1s are written to PD and IDL at the same time, PD takes precedence. The reset value of PCON is (0XXX000).
User software should never write 1s to unimplemented bits, since they may be used in future products.

Table 4. Flash AT89C51 and AT89C52

Device Name	Flash Bytes	Ckt Type	VPP	Time Required to Program Entire Array
AT89C51	4 K	CMOS	12 V	6 seconds
AT89C52	8 K	CMOS	12 V	12 seconds

Power Down Mode

An instruction that sets PCON 1 is the last instruction executed before Power Down mode begins. In the Power Down mode, the on-chip oscillator stops. With the clock frozen, all functions are stopped, but the on-chip RAM and Special Function Registers are held. The port pins output the values held by their respective SFRs. ALE and $\overline{PSEN}$ output lows.

The only exit from Power Down for the AT89CC51 is a hardware reset. Reset redefines all the SFRs but does not change the on-chip RAM.

In the Power Down mode of operation, V_{CC} can be reduced to as low as 2V. However, V_{CC} must not be reduced before the Power Down mode is invoked, and V_{CC} must be restored to its normal operating level before the Power Down mode is terminated. The reset that terminates Power Down also frees the oscillator. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize (normally less than 10 msec).

Programming

The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. A list of programming companies that support Atmel's products can be found on the Atmel Bulletin Board and in the Microcontroller Programmer Support section of this Data Book. To access the bulletin board, dial 408-436-4309.

The AT89C51/52 programs at VPP = 12 V using one 100-msec PROG pulse per byte programmed. This results in a programming time of approximately 1.5 msec per byte, for a total programming time of 6 sec for the 4 Kbyte device and 12 sec for the 8Kbyte device.

Detailed procedures for programming and verifying each device are given in the data sheets.

Program Memory Locks

In some microcontroller applications, the program memory must be secure from software piracy. Atmel has responded to this need by implementing a program memory locking scheme in all of its devices. While it is impossible for anyone to guarantee absolute security against all levels of technological sophistication, the program memory locks present a substantial barrier against illegal readout of protected software.

The procedure for programming the lock bits is detailed in the data sheets.

Table 5 lists the Lock Bits and their corresponding effects on the microcontroller.

Erasing the Flash also erases the Lock Bits, returning the microcontroller to full functionality.

Table 5. Program Lock Bits and Their Features

Program Lock Bits				Protection Type
Mode	LB1	LB2	LB3	
1	U	U	U	No program lock features enabled.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, EA is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as 2, also verify is disabled.
4	P	P	P	Same as 3, also external execution is disabled.

P = Programmed U = Unprogrammed

Any other combination of the Lock Bits is not defined.

Table 6. Program Protection

Device	Lock Bits
AT89C51	LB1, LB2, LB3
AT89C52	LB1, LB2, LB3
AT89C2051	LB1, LB2
AT89C1051	LB1, LB2

When Lock Bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. The latched value of $\overline{EA}$ must agree with the current logic level at that pin in order for the device to function properly.

ONCE™ Mode

The ONCE ("on-circuit emulation") mode facilitates testing and debugging of systems using the device without requiring the device to be removed from the circuit. The ONCE mode is invoked by taking the following steps.

1. Pull ALE low while the device is in reset and $\overline{PSEN}$ is high;
2. Hold ALE low as RST is deactivated.

While the device is in ONCE mode, the Port 0 pins go into a float state, and the other port pins and ALE and $\overline{PSEN}$ are weakly pulled high. The oscillator circuit remains active. While the device is in this mode, an emulator or test CPU can be used to drive the circuit. A reset restores normal operation.

On-Chip Oscillators

The crystal specifications and capacitance values (C1 and C2 in Figure 30) are not critical. 30 pF can be used in these positions at any frequency with good quality crystals. A ceramic resonator can be used in place of the crystal in cost-sensitive applications. When a ceramic resonator is used, C1 and C2 are normally selected to be of somewhat higher values, typically, 47 pF. The manufacturer of the ceramic resonator should be consulted for recommendations on the values of these capacitors.

In general, crystals used with these devices typically have the following specifications.

ESR (Equivalent Series Resistance)	see Figure 31
C_O (Shunt Capacitance)	7.0 pF max.
C_L (Load Capacitance)	30 pF + 3 pF
Drive Level	1 mW

Frequency, tolerance and temperature range are determined by the system requirements.

Figure 30. Using the On-Chip Oscillator

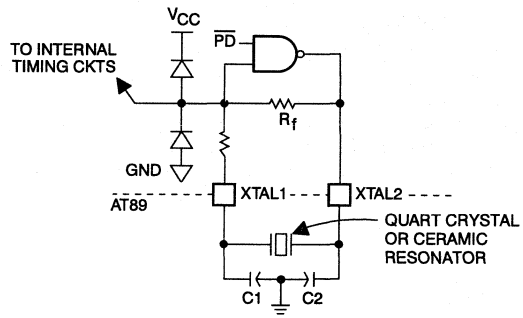
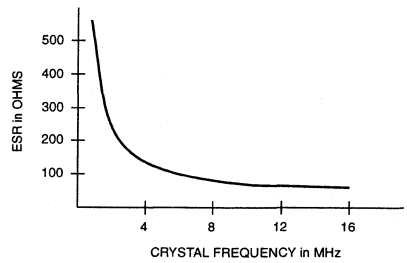


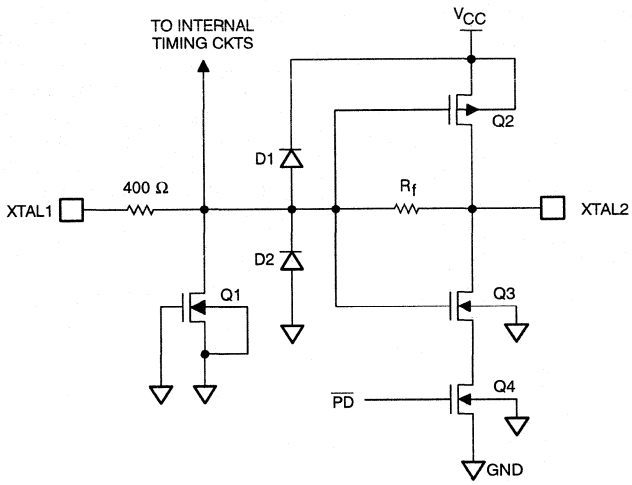
Figure 31. ESR versus Frequency



The on-chip oscillator circuitry shown in Figure 32, consists of a single stage linear inverter intended for use as a crystal-controlled, positive reactance oscillator.

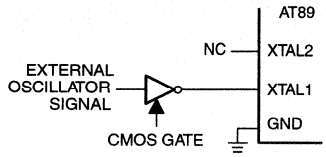
To drive the parts with an external clock source, apply the external clock signal to XTAL1, and leave XTAL2 floating, as shown in Figure 33.

Figure 32. On-Chip Oscillator Circuitry for the AT89C51



Note: In Atmel's CMOS microcontrollers the Oscillator Specification differs from that in NMOS versions.

Figure 33. Using an External Clock Source



Internal Timing

Figures 34 through 37 show the various strobe and port signals being clocked internally. The figures do not show rise and fall times of the signals, nor do they show propagation delays between the XTAL signal and events at other pins.

Rise and fall times are dependent on the external loading that each pin must drive. They are often taken to be about 10 ns, measured between 0.8 V and 2.0 V.

Propagation delays are different for different pins. For a given pin the delays vary with pin loading, temperature, V_{CC}, and manufacturing lot. If the XTAL waveform is taken as the timing reference, propagation delays may vary from 25 to 125 ns.

The AC Timings section of the data sheets do not reference any timing to the XTAL waveform. Rather, they relate the critical edges of control and input signals to each other. The timings published in the data sheet include the effects of propagation delays under the specified test conditions.

Figure 34. External Program Memory Fetches

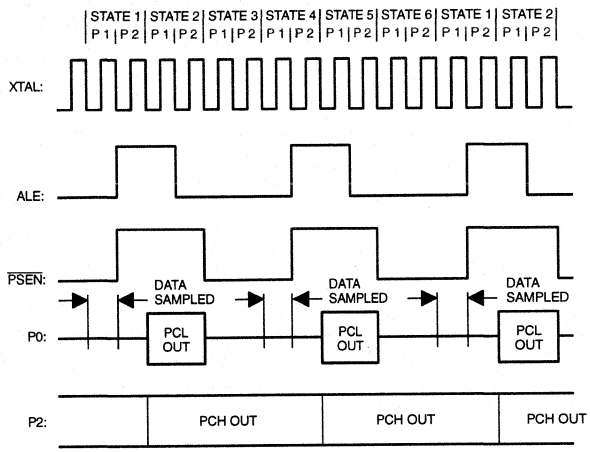


Figure 35. External Data Memory Read Cycle

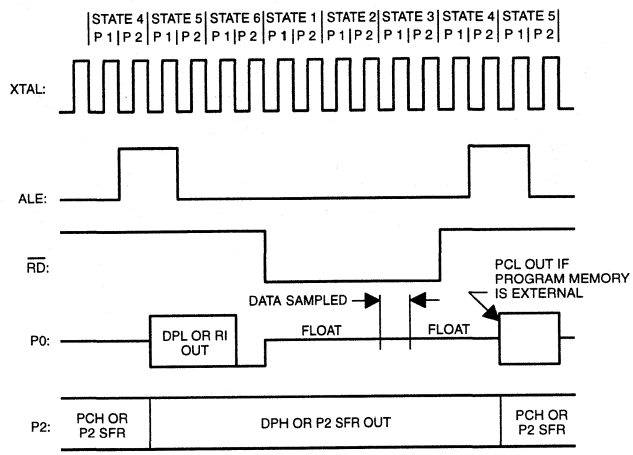


Figure 36. External Data Memory Write Cycle

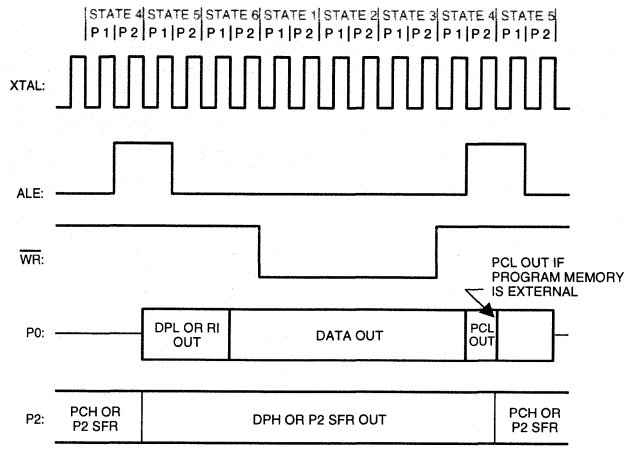
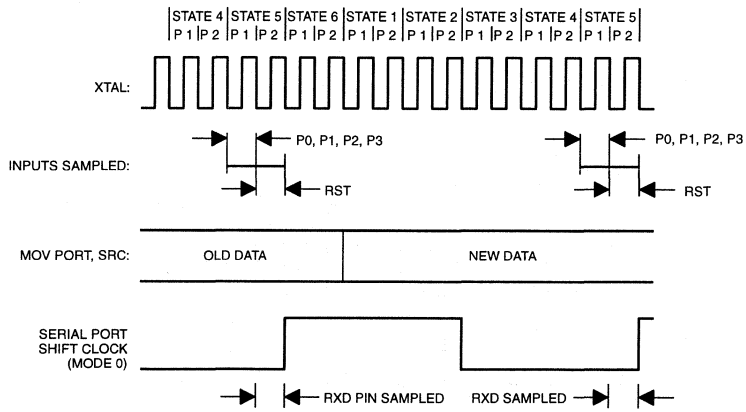


Figure 37. Port Operation



Microcontroller Instruction Set

For interrupt response time information, refer to the hardware description chapter.

Instructions that Affect Flag Settings[1]

Instruction	Flag			Instruction	Flag		
	C	OV	AC		C	OV	AC
ADD	X	X	X	CLR C	O		
ADDC	X	X	X	CPL C	X		
SUBB	X	X	X	ANL C,bit	X		
MUL	O	X		ANL C,/bit	X		
DIV	O	X		ORL C,bit	X		
DA	X			ORL C, bit	X		
RRC	X			MOV C,bit	X		
RLC	X			CJNE	X		
SETB C	1						

Note 1. Operations on SFR byte address 208 or bit addresses 209-215 (that is, the PSW or bits in the PSW) also affect flag settings.

The Instruction Set and Addressing Modes

Rn	Register R7-R0 of the currently selected Register Bank.
direct	8-bit internal data location's address. This could be an Internal Data RAM location (0-127) or a SFR [i.e., I/O port, control register, status register, etc. (128-255)].
@RI	8-bit internal data RAM location (0-255) addressed indirectly through register R1or R0.
#data	8-bit constant included in instruction.
#data 16	16-bit constant included in instruction.
addr 16	16-bit destination address. Used by LCALL and LJMP. A branch can be anywhere within the 64 Kbyte Program Memory address space.
addr 11	11-bit destination address. Used by ACALL and AJMP. The branch will be within the same 2 Kbyte page of program memory as the first byte of the following instruction.
rel	Signed (two's complement) 8-bit offset byte. Used by SJMP and all conditional jumps. Range is -128 to +127 bytes relative to first byte of the following instruction.
bit	Direct Addressed bit in Internal Data RAM or Special Function Register.

Instruction Set Summary

	0	1	2	3	4	5	6	7
0	NOP	JBC bit,rel [3B, 2C]	JB bit, rel [3B, 2C]	JNB bit, rel [3B, 2C]	JC rel [2B, 2C]	JNC rel [2B, 2C]	JZ rel [2B, 2C]	JNZ rel [2B, 2C]
1	AJMP (P0) [2B, 2C]	ACALL (P0) [2B, 2C]	AJMP (P1) [2B, 2C]	ACALL (P1) [2B, 2C]	AJMP (P2) [2B, 2C]	ACALL (P2) [2B, 2C]	AJMP (P3) [2B, 2C]	ACALL (P3) [2B, 2C]
2	LJMP addr16 [3B, 2C]	LCALL addr16 [3B, 2C]	RET [2C]	RETI [2C]	ORL dir, A [2B]	ANL dir, A [2B]	XRL dir, a [2B]	ORL C, bit [2B, 2C]
3	RR A	RRC A	RL A	RLC A	ORL dir, #data [3B, 2C]	ANL dir, #data [3B, 2C]	XRL dir, #data [3B, 2C]	JMP @A + DPTR [2C]
4	INC A	DEC A	ADD A, #data [2B]	ADDC A, #data [2B]	ORL A, #data [2B]	ANL A, #data [2B]	XRL A, #data [2B]	MOV A, #data [2B]
5	INC dir [2B]	DEC dir [2B]	ADD A, dir [2B]	ADDC A, dir [2B]	ORL A, dir [2B]	ANL A, dir [2B]	XRL A, dir [2B]	MOV dir, #data [3B, 2C]
6	INC @R0	DEC @R0	ADD A, @R0	ADDC A, @R0	ORL A, @R0	ANL A, @R0	XRL A, @R0	MOV @R0, @data [2B]
7	INC @R1	DEC @R1	ADD A, @R1	ADDC A, @R1	ORL A, @R1	ANL A, @R1	XRL A, @R1	MOV @R1, #data [2B]
8	INC R0	DEC R0	ADD A, R0	ADDC A, R0	ORL A, R0	ANL A, R0	XRL A, R0	MOV R0, #data [2B]
9	INC R1	DEC R1	ADD A, R1	ADDC A, R1	ORL A, R1	ANL A, R1	XRL A, R1	MOV R1, #data [2B]
A	INC R2	DEC R2	ADD A, R2	ADDC A, R2	ORL A, R2	ANL A, R2	XRL A, R2	MOV R2, #data [2B]
B	INC R3	DEC R3	ADD A, R3	ADDC A, R3	ORL A, R3	ANL A, R3	XRL A, R3	MOV R3, #data [2B]
C	INC R4	DEC R4	ADD A, R4	ADDC A, R4	ORL A, R4	ANL A, R4	XRL A, R4	MOV R4, #data [2B]
D	INC R5	DEC R5	ADD A, R5	ADDC A, R5	ORL A, R5	ANL A, R5	XRL A, R5	MOV R5, #data [2B]
E	INC R6	DEC R6	ADD A, R6	ADDC A, R6	ORL A, R6	ANL A, R6	XRL A, R6	MOV R6, #data [2B]
F	INC R7	DEC R7	ADD A, R7	ADDC A, R7	ORL A, R7	ANL A, R7	XRL A, R7	MOV R7, #data [2B]

Key:
[2B] = 2 Byte, [3B] = 3 Byte, [2C] = 2 Cycle, [4C] = 4 Cycle, Blank = 1 byte/1 cycle

Instruction Set Summary (Continued)

	8	9	A	B	C	D	E	F
0	SJMP REL [2B, 2C]	MOV DPTR,# data 16 [3B, 2C]	ORL C, /bit [2B, 2C]	ANL C, /bit [2B, 2C]	PUSH dir [2B, 2C]	POP dir [2B, 2C]	MOVX A, @DPTR [2C]	MOVX @DPTR, A [2C]
1	AJMP (P4) [2B, 2C]	ACALL (P4) [2B, 2C]	AJMP (P5) [2B, 2C]	ACALL (P5) [2B, 2C]	AJMP (P6) [2B, 2C]	ACALL (P6) [2B, 2C]	AJMP (P7) [2B, 2C]	ACALL (P7) [2B, 2C]
2	ANL C, bit [2B, 2C]	MOV bit, C [2B, 2C]	MOV C, bit [2B]	CPL bit [2B]	CLR bit [2B]	SETB bit [2B]	MOVX A, @R0 [2C]	MOVX wR0, A [2C]
3	MOVC A, @A + PC [2C]	MOVC A, @A + DPTR [2C]	INC DPTR [2C]	CPL C	CLR C	SETB C	MOVX A, @RI [2C]	MOVX @RI, A [2C]
4	DIV AB [2B, 4C]	SUBB A, #data [2B]	MUL AB [4C]	CJNE A, #data, rel [3B, 2C]	SWAP A	DA A	CLR A	CPL A
5	MOV dir, dir [3B, 2C]	SUBB A, dir [2B]		CJNE A, dir, rel [3B, 2C]	XCH A, dir [2B]	DJNZ dir, rel [3B, 2C]	MOV A, dir [2B]	MOV dir, A [2B]
6	MOV dir, @R0 [2B, 2C]	SUBB A, @R0	MOV @R0, dir [2B, 2C]	CJNE @R0, #data, rel [3B, 2C]	XCH A, @R0	XCHD A, @R0	MOV A, @R0	MOV @R0, A
7	MOV dir, @R1 [2B, 2C]	SUBB A, @R1	MOV @R1, dir [2B, 2C]	CJNE @R1, #data, rel [3B, 2C]	XCH A, @R1	XCHD A, @R1	MOV A, @R1	MOV @R1, A
8	MOV dir, R0 [2B, 2C]	SUBB A, R0	MOV R0, dir [2B, 2C]	CJNE R0, #data, rel [3B, 2C]	XCH A, R0	DJNZ R0, rel [2B, 2C]	MOV A, R0	MOV R0, A
9	MOV dir, R1 [2B, 2C]	SUBB A, R1	MOV R1, dir [2B, 2C]	CJNE R1, #data, rel [3B, 2C]	XCH A, R1	DJNZ R1, rel [2B, 2C]	MOV A, R1	MOV R1, A
A	MOV dir, R2 [2B, 2C]	SUBB A, R2	MOV R2, dir [2B, 2C]	CJNE R2, #data, rel [3B, 2C]	XCH A, R2	DJNZ R2, rel [2B, 2C]	MOV A, R2	MOV R2, A
B	MOV dir, R3 [2B, 2C]	SUBB A, R3	MOV R3, dir [2B, 2C]	CJNE R3, #data, rel [3B, 2C]	XCH A, R3	DJNZ R3, rel [2B, 2C]	MOV A, R3	MOV R3, A
C	MOV dir, R4 [2B, 2C]	SUBB A, R4	MOV R4, dir [2B, 2C]	CJNE R4, #data, rel [3B, 2C]	XCH A, R4	DJNZ R4, rel [2B, 2C]	MOV A, R4	MOV R4, A
D	MOV dir, R5 [2B, 2C]	SUBB A, R5	MOV R5, dir [2B, 2C]	CJNE R5, #data, rel [3B, 2C]	XCH A, R5	DJNZ R5, rel [2B, 2C]	MOV A, R5	MOV R5, A
E	MOV dir, R6 [2B, 2C]	SUBB A, R6	MOV R6, dir [2B, 2C]	CJNE R6, #data, rel [3B, 2C]	XCH A, R6	DJNZ R6, rel [2B, 2C]	MOV A, R6	MOV R6. A
F	MOV dir, R7 [2B, 2C]	SUBB A, R7	MOV R7, dir [2B, 2C]	CJNE R7, #data, rel [3B, 2C]	XCH A, R7	DJNZ R7, rel [2B, 2C]	MOV A, R7	MOV R7, A

Key:
[2B] = 2 Byte, [3B] = 3 Byte, [2C] = 2 Cycle, [4C] = 4 Cycle, Blank = 1 byte/1 cycle

Table 1. AT89 Instruction Set Summary[1]

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS				
ADD	A,Rn	Add register to Accumulator	1	12
ADD	A,direct	Add direct byte to Accumulator	2	12
ADD	A,@Ri	Add indirect RAM to Accumulator	1	12
ADD	A,#data	Add immediate data to Accumulator	2	12
ADDC	A,Rn	Add register to Accumulator with Carry	1	12
ADDC	A,direct	Add direct byte to Accumulator with Carry	2	12
ADDC	A,@Ri	Add indirect RAM to Accumulator with Carry	1	12
ADDC	A,#data	Add immediate data to Acc with Carry	2	12
SUBB	A,Rn	Subtract Register from Acc with borrow	1	12
SUBB	A,direct	Subtract direct byte from Acc with borrow	2	12
SUBB	A,@Ri	Subtract indirect RAM from ACC with borrow	1	12
SUBB	A,#data	Subtract immediate data from Acc with borrow	2	12
INC	A	Increment Accumulator	1	12
INC	Rn	Increment register	1	12
INC	direct	Increment direct byte	2	12
INC	@Ri	Increment direct RAM	1	12
DEC	A	Decrement Accumulator	1	12
DEC	Rn	Decrement Register	1	12
DEC	direct	Decrement direct byte	2	12
DEC	@Ri	Decrement indirect RAM	1	12

Mnemonic		Description	Byte	Oscillator Period
ARITHMETIC OPERATIONS (continued)				
INC	DPTR	Increment Data Pointer	1	24
MUL	AB	Multiply A & B	1	48
DIV	AB	Divide A by B	1	48
DA	A	Decimal Adjust Accumulator	1	12
LOGICAL OPERATIONS				
ANL	A,Rn	AND Register to Accumulator	1	12
ANL	A,direct	AND direct byte to Accumulator	2	12
ANL	A,@Ri	AND indirect RAM to Accumulator	1	12
ANL	A,#data	AND immediate data to Accumulator	2	12
ANL	direct,A	AND Accumulator to direct byte	2	12
ANL	direct,#data	AND immediate data to direct byte	3	24
ORL	A,Rn	OR register to Accumulator	1	12
ORL	A,direct	OR direct byte to Accumulator	2	12
ORL	A,@Ri	OR indirect RAM to Accumulator	1	12
ORL	A,#data	OR immediate data to Accumulator	2	12
ORL	direct,A	OR Accumulator to direct byte	2	12
ORL	direct,#data	OR immediate data to direct byte	3	24
XRL	A,Rn	Exclusive-OR register to Accumulator	1	12
XRL	A,direct	Exclusive-OR direct byte to Accumulator	2	12
XRL	A,@Ri	Exclusive-OR indirect RAM to Accumulator	1	12
XRL	A,#data	Exclusive-OR immediate data to Accumulator	2	12

Note: 1. All mnemonics copyrighted © Intel Corp., 1980.

Table 1. AT89 Instruction Set Summary (continued)

Mnemonic		Description	Byte	Oscillator Period
LOGICAL OPERATIONS (continued)				
XRL	direct,A	Exclusive-OR Accumulator to direct byte	2	12
XRL	direct,#data	Exclusive-OR immediate data to direct byte	3	24
CLR	A	Clear Accumulator	1	12
CPL	A	Complement Accumulator	1	12
RL	A	Rotate Accumulator Left	1	12
RLC	A	Rotate Accumulator Left through the Carry	1	12
RR	A	Rotate Accumulator Right	1	12
RRC	A	Rotate Accumulator Right through the Carry	1	12
SWAP	A	Swap nibbles within the Accumulator	1	12
DATA TRANSFER				
MOV	A,Rn	Move register to Accumulator	1	12
MOV	A,direct	Move direct byte to Accumulator	2	12
MOV	A,@Ri	Move indirect RAM to Accumulator	1	12
MOV	A,#data	Move immediate data to Accumulator	2	12
MOV	Rn,A	Move Accumulator to register	1	12
MOV	Rn,direct	Move direct byte to register	2	24
MOV	Rn,#data	Move immediate data to register	2	12
MOV	direct,A	Move Accumulator to direct byte	2	12
MOV	direct,Rn	Move register to direct byte	2	24
MOV	direct,direct	Move direct byte to direct	3	24

Mnemonic		Description	Byte	Oscillator Period
DATA TRANSFER (continued)				
MOV	direct,@Ri	Move indirect RAM to direct byte	2	24
MOV	direct,#data	Move immediate data to direct byte	3	24
MOV	@Ri,A	Move Accumulator to indirect RAM	1	12
MOV	@Ri,direct	Move direct byte to indirect RAM	2	24
MOV	@Ri,#data	Move immediate data to indirect RAM	2	12
MOV	DPTR,#data16	Load Data Pointer with a 16-bit constant	3	24
MOVC	A,@A+DPTR	Move Code byte relative to DPTR to Acc	1	24
MOVC	A,@A+PC	Move Code byte relative to PC to Acc	1	24
MOVX	A,@Ri	Move External RAM (8-bit addr) to Acc	1	24
MOVX	A,@DPTR	Move Exernal RAM (16-bit addr) to Acc	1	24
MOVX	@Ri,A	Move Acc to External RAM (8-bit addr)	1	24
MOVX	@DPTR,A	Move Acc to External RAM (16-bit addr)	1	24
PUSH	direct	Push direct byte onto stack	2	24
POP	direct	Pop direct byte from stack	2	24
XCH	A,Rn	Exchange register with Accumulator	1	12
XCH	A,direct	Exchange direct byte with Accumulator	2	12
XCH	A,@Ri	Exchange indirect RAM with Accumulator	1	12
XCHD	A,@Ri	Exchange low-order Digit indirect RAM with Acc	1	12

Table 1. AT89 Instruction Set Summary (continued)

Mnemonic		Description	Byte	Oscillator Period
BOOLEAN VARIABLE MANIPULATION				
CLR	C	Clear Carry	1	12
CLR	bit	Clear direct bit	2	12
SETB	C	Set Carry	1	12
SETB	bit	Set direct bit	2	12
CPL	C	Complement Carry	1	12
CPL	bit	Complement direct bit	2	12
ANL	C,bit	AND direct bit to CARRY	2	24
ANL	C,/bit	AND complement of direct bit to Carry	2	24
ORL	C,bit	OR direct bit to Carry	2	24
ORL	C,/bit	OR complement of direct bit to Carry	2	24
MOV	C,bit	Move direct bit to Carry	2	12
MOV	bit,C	Move Carry to direct bit	2	24
JC	rel	Jump if Carry is set	2	24
JNC	rel	Jump if Carry not set	2	24
JB	bit,rel	Jump if direct Bit is set	3	24
JNB	bit,rel	Jump if direct Bit is Not set	3	24
JBC	bit,rel	Jump if direct Bit is set & clear bit	3	24
PROGRAM BRANCHING				
ACALL	addr11	Absolute Subroutine Call	2	24
LCALL	addr16	Long Subroutine Call	3	24
RET		Return from Subroutine	1	24
RETI		Return from interrupt	1	24
AJMP	addr11	Absolute Jump	2	24
LJMP	addr16	Long Jump	3	24
SJMP	rel	Short Jump (relative addr)	2	24

Mnemonic		Description	Byte	Oscillator Period
PROGRAM BRANCHING (continued)				
JMP	@A+DPTR	Jump indirect relative to the DPTR	1	24
JZ	rel	Jump if Accumulator is Zero	2	24
JNZ	rel	Jump if Accumulator is Not Zero	2	24
CJNE	A,direct,rel	Compare direct byte to Acc and Jump if Not Equal	3	24
CJNE	A,#data,rel	Compare immediate to Acc and Jump if Not Equal	3	24
CJNE	Rn,#data,rel	Compare immediate to register and Jump if Not Equal	3	24
CJNE	@Ri,#data,rel	Compare immediate to indirect and Jump if Not Equal	3	24
DJNZ	Rn,rel	Decrement register and Jump if Not Zero	2	24
DJNZ	direct,rel	Decrement direct byte and Jump if Not Zero	3	24
NOP		No Operation	1	12

Table 2. Instruction Opcodes in Hexadecimal Order

Hex Code	Number of Bytes	Mnemonic	Operands
00	1	NOP	
01	2	AJMP	code addr
02	3	LJMP	code addr
03	1	RR	A
04	1	INC	A
05	2	INC	data addr
06	1	INC	@R0
07	1	INC	@R1
08	1	INC	R0
09	1	INC	R1
0A	1	INC	R2
0B	1	INC	R3
0C	1	INC	R4
0D	1	INC	R5
0E	1	INC	R6
0F	1	INC	R7
10	3	JBC	bit addr, code addr
11	2	ACALL	code addr
12	3	LCALL	code addr
13	1	RRC	A
14	1	DEC	A
15	2	DEC	data addr
16	1	DEC	@R0
17	1	DEC	@R1
18	1	DEC	R0
19	1	DEC	R1
1A	1	DEC	R2
1B	1	DEC	R3
1C	1	DEC	R4
1D	1	DEC	R5
1E	1	DEC	R6
1F	1	DEC	R7
20	3	JB	bit addr,code addr
21	2	AJMP	code addr
22	1	RET	
23	1	RL	A
24	2	ADD	A,#data
25	2	ADD	A,data addr
26	1	ADD	A,@R0
27	1	ADD	A,@R1
28	1	ADD	A,R0
29	1	ADD	A,R1
2A	1	ADD	A,R2
2B	1	ADD	A,R3
2C	1	ADD	A,R4
2D	1	ADD	A,R5
2E	1	ADD	A,R6
2F	1	ADD	A,R7
30	3	JNB	bit addr,code addr
31	2	ACALL	code addr
32	1	RETI	
33	1	RLC	A
34	2	ADDC	A,#data
35	2	ADDC	A,data addr
36	1	ADDC	A,@R0
37	1	ADDC	A,@R1
38	1	ADDC	A,R0
39	1	ADDC	A,R1
3A	1	ADDC	A,R2
3B	1	ADDC	A,R3
3C	1	ADDC	A,R4
3D	1	ADDC	A,R5
3E	1	ADDC	A,R6
3F	1	ADDC	A,R7
40	2	JC	code addr

Table 2. Instruction Opcodes in Hexadecimal Order (continued)

Hex Code	Number of Bytes	Mnemonic	Operands
41	2	AJMP	code addr
42	2	ORL	data addr,A
43	3	ORL	data addr,#data
44	2	ORL	A,#data
45	2	ORL	A,data addr
46	1	ORL	A,@R0
47	1	ORL	A,@R1
48	1	ORL	A,R0
49	1	ORL	A,R1
4A	1	ORL	A,R2
4B	1	ORL	A,R3
4C	1	ORL	A,R4
4D	1	ORL	A,R5
4E	1	ORL	A,R6
4F	1	ORL	A,R7
50	2	JNC	code addr
51	2	ACALL	code addr
52	2	ANL	data addr,A
53	3	ANL	data addr,#data
54	2	ANL	A,#data
55	2	ANL	A, data addr
56	1	ANL	A,@R0
57	1	ANL	A,@R1
58	1	ANL	A,R0
59	1	ANL	A,R1
5A	1	ANL	A,R2
5B	1	ANL	A,R3
5C	1	ANL	A,R4
5D	1	ANL	A,R5
5E	1	ANL	A,R6
5F	1	ANL	A,R7

Hex Code	Number of Bytes	Mnemonic	Operands
60	2	JZ	code addr
61	2	AJMP	code addr
62	2	XRL	data addr,A
63	3	XRL	data addr,#data
64	2	XRL	A,#data
65	2	XRL	A,data addr
66	1	XRL	A,@R0
67	1	XRL	A,@R1
68	1	XRL	A,R0
69	1	XRL	A,R1
6A	1	XRL	A,R2
6B	1	XRL	A,R3
6C	1	XRL	A,R4
6D	1	XRL	A,R5
6E	1	XRL	A,R6
6F	1	XRL	A,R7
70	2	JNZ	code addr
71	2	ACALL	code addr
72	2	ORL	C,bit addr
73	1	JMP	@A+DPTR
74	2	MOV	A,#data
75	3	MOV	data addr,#data
76	2	MOV	@R0,#data
77	2	MOV	@R1,#data
78	2	MOV	R0,#data
79	2	MOV	R1,#data
7A	2	MOV	R2,#data
7B	2	MOV	R3,#data
7C	2	MOV	R4,#data
7D	2	MOV	R5,#data
7E	2	MOV	R6,#data

Table 2. Instruction Opcodes in Hexadecimal Order (continued)

Hex Code	Number of Bytes	Mnemonic	Operands
7F	2	MOV	R7,#data
80	2	SJMP	code addr
81	2	AJMP	code addr
82	2	ANL	C,bit addr
83	1	MOVC	A,@A+PC
84	1	DIV	AB
85	3	MOV	data addr,data addr
86	2	MOV	data addr,@R0
87	2	MOV	data addr,@R1
88	2	MOV	data addr,R0
89	2	MOV	data addr,R1
8A	2	MOV	data addr,R2
8B	2	MOV	data addr,R3
8C	2	MOV	data addr,R4
8D	2	MOV	data addr,R5
8E	2	MOV	data addr,R6
8F	2	MOV	data addr,R7
90	3	MOV	DPTR,#data
91	2	ACALL	code addr
92	2	MOV	bit addr,C
93	1	MOVC	A,@A+DPTR
94	2	SUBB	A,#data
95	2	SUBB	A,data addr
96	1	SUBB	A,@R0
97	1	SUBB	A,@R1
98	1	SUBB	A,R0
99	1	SUBB	A,R1
9A	1	SUBB	A,R2
9B	1	SUBB	A,R3
9C	1	SUBB	A,R4
9D	1	SUBB	A,R5
9E	1	SUBB	A,R6
9F	1	SUBB	A,R7

Hex Code	Number of Bytes	Mnemonic	Operands
A0	2	ORL	C,/bit addr
A1	2	AJMP	code addr
A2	2	MOV	C,bit addr
A3	1	INC	DPTR
A4	1	MUL	AB
A5		reserved	
A6	2	MOV	@R0,data addr
A7	2	MOV	@R1,data addr
A8	2	MOV	R0,data addr
A9	2	MOV	R1,data addr
AA	2	MOV	R2,data addr
AB	2	MOV	R3,data addr
AC	2	MOV	R4,data addr
AD	2	MOV	R5,data addr
AE	2	MOV	R6,data addr
AF	2	MOV	R7,data addr
B0	2	ANL	C,/bit addr
B1	2	ACALL	code addr
B2	2	CPL	bit addr
B3	1	CPL	C
B4	3	CJNE	A,#data,code addr
B5	3	CJNE	A,data addr,code addr
B6	3	CJNE	@R0,#data,code addr
B7	3	CJNE	@R1,#data,code addr
B8	3	CJNE	R0,#data,code addr
B9	3	CJNE	R1,#data,code addr
BA	3	CJNE	R2,#data,code addr
BB	3	CJNE	R3,#data,code addr
BC	3	CJNE	R4,#data,code addr
BD	3	CJNE	R5,#data,code addr
BE	3	CJNE	R6,#data,code addr
BF	3	CJNE	R7,#data,code addr

Table 2. Instruction Opcodes in Hexadecimal Order (continued)

Hex Code	Number of Bytes	Mnemonic	Operands
C0	2	PUSH	data addr
C1	2	AJMP	code addr
C2	2	CLR	bit addr
C3	1	CLR	C
C4	1	SWAP	A
C5	2	XCH	A,data addr
C6	1	XCH	A,@R0
C7	1	XCH	A,@R1
C8	1	XCH	A,R0
C9	1	XCH	A,R1
CA	1	XCH	A,R2
CB	1	XCH	A,R3
CC	1	XCH	A,R4
CD	1	XCH	A,R5
CE	1	XCH	A,R6
CF	1	XCH	A,R7
D0	2	POP	data addr
D1	2	ACALL	code addr
D2	2	SETB	bit addr
D3	1	SETB	C
D4	1	DA	A
D5	3	DJNZ	data addr,code addr
D6	1	XCHD	A,@R0
D7	1	XCHD	A,@R1
D8	2	DJNZ	R0,code addr
D9	2	DJNZ	R1,code addr
DA	2	DJNZ	R2,code addr
DB	2	DJNZ	R3,code addr
DC	2	DJNZ	R4,code addr
DD	2	DJNZ	R5,code addr
DE	2	DJNZ	R6,code addr
DF	2	DJNZ	R7,code addr

Hex Code	Number of Bytes	Mnemonic	Operands
E0	1	MOVX	A,@DPTR
E1	2	AJMP	code addr
E2	1	MOVX	A,@R0
E3	1	MOVX	A,@R1
E4	1	CLR	A
E5	2	MOV	A,data addr
E6	1	MOV	A,@R0
E7	1	MOV	A,@R1
E8	1	MOV	A,R0
E9	1	MOV	A,R1
EA	1	MOV	A,R2
EB	1	MOV	A,R3
EC	1	MOV	A,R4
ED	1	MOV	A,R5
EE	1	MOV	A,R6
EF	1	MOV	A,R7
F0	1	MOVX	@DPTR,A
F1	2	ACALL	code addr
F2	1	MOVX	@R0,A
F3	1	MOVX	@R1,A
F4	1	CPL	A
F5	2	MOV	data addr,A
F6	1	MOV	@R0,A
F7	1	MOV	@R1,A
F8	1	MOV	R0,A
F9	1	MOV	R1,A
FA	1	MOV	R2,A
FB	1	MOV	R3,A
FC	1	MOV	R4,A
FD	1	MOV	R5,A
FE	1	MOV	R6,A
FF	1	MOV	R7,A

Instruction Definitions

ACALL addr11

Function: Absolute Call

Description: ACALL unconditionally calls a subroutine located at the indicated address. The instruction increments the PC twice to obtain the address of the following instruction, then pushes the 16-bit result onto the stack (low-order byte first) and increments the Stack Pointer twice. The destination address is obtained by successively concatenating the five high-order bits of the incremented PC, opcode bits 7 through 5, and the second byte of the instruction. The subroutine called must therefore start within the same 2 K block of the program memory as the first byte of the instruction following ACALL. No flags are affected.

Example: Initially SP equals 07H. The label SUBRTN is at program memory location 0345 H. After executing the following instruction,

ACALL SUBRTN

at location 0123H, SP contains 09H, internal RAM locations 08H and 09H will contain 25H and 01H, respectively, and the PC contains 0345H.

Bytes: 2

Cycles: 2

Encoding:

a10	a9	a8	1	0	0	0	1

a7	a6	a5	a4	a3	a2	a1	a0

Operation: ACALL
(PC) ← (PC) + 2
(SP) ← (SP) + 1
((SP)) ← (PC_{7-0})
(SP) ← (SP) + 1
((SP)) ← (PC_{15-8})
(PC_{10-0}) ← page address

ADD A,<src-byte>

Function: Add

Description: ADD adds the byte variable indicated to the Accumulator, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set, respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not bit 6; otherwise, OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands, or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C3H (11000011B), and register 0 holds 0AAH (10101010B). The following instruction,

ADD A,R0

leaves 6DH (01101101B) in the Accumulator with the AC flag cleared and both the carry flag and OV set to 1.

ADD A,Rn

Bytes: 1

Cycles: 1

Encoding:

0	0	1	0	1	r	r	r

Operation: ADD
(A) ← (A) + (Rn)

ADD A,direct

Bytes: 2

Cycles: 1

Encoding:

0	0	1	0	0	1	0	1

direct address

Operation: ADD
(A) ← (A) + (direct)

ADD A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0	0	1	0	0	1	1	i

Operation: ADD
(A) ← (A) + ((Ri))

ADD A,#data

Bytes: 2

Cycles: 1

Encoding:

0	0	1	0	0	1	0	0

immediate data

Operation: ADD
(A) ← (A) + #data

ADDC A, <src-byte>

Function: Add with Carry

Description: ADDC simultaneously adds the byte variable indicated, the carry flag and the Accumulator contents, leaving the result in the Accumulator. The carry and auxiliary-carry flags are set respectively, if there is a carry-out from bit 7 or bit 3, and cleared otherwise. When adding unsigned integers, the carry flag indicates an overflow occurred.

OV is set if there is a carry-out of bit 6 but not out of bit 7, or a carry-out of bit 7 but not out of bit 6; otherwise OV is cleared. When adding signed integers, OV indicates a negative number produced as the sum of two positive operands or a positive sum from two negative operands.

Four source operand addressing modes are allowed: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) with the carry flag set. The following instruction,

ADDC A,R0

leaves 6EH (01101110B) in the Accumulator with AC cleared and both the Carry flag and OV set to 1.

ADDC A,Rn

Bytes: 1

Cycles: 1

Encoding:

0	0	1	1	1	r	r	r

Operation: ADDC
(A) ← (A) + (C) + (R_n)

ADDC A,direct

Bytes: 2

Cycles: 1

Encoding:

0	0	1	1	0	1	0	1	direct address

Operation: ADDC
(A) ← (A) + (C) + (direct)

ADDC A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0	0	1	1	0	1	1	i

Operation: ADDC
(A) ← (A) + (C) + ((R_i))

ADDC A,#data

Bytes: 2

Cycles: 1

Encoding:

0	0	1	1	0	1	0	0	immediate data

Operation: ADDC
(A) ← (A) + (C) + #data

AJMP addr11

Function: Absolute Jump

Description: AJMP transfers program execution to the indicated address, which is formed at run-time by concatenating the high-order five bits of the PC (after incrementing the PC twice), opcode bits 7 through 5, and the second byte of the instruction. The destination must therfore be within the same 2 K block of program memory as the first byte of the instruction following AJMP.

Example: The label JMPADR is at program memory location 0123H. The following instruction,

```
AJMP        JMPADR
```

is at location 0345H and loads the PC with 0123H.

Bytes: 2

Cycles: 2

Encoding:

a10	a9	a8	0	0	0	0	1	a7	a6	a5	a4	a3	a2	a1	a0

Operation: AJMP
(PC) ← (PC) + 2
(PC_{10-0}) ← page address

ANL <dest-byte>,<src-byte>

Function: Logical-AND for byte variables

Description: ANL performs the bitwise logical-AND operation between the variables indicated and stores the results in the destination variable. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: If the Accumulator holds 0C3H (11000011B), and register 0 holds 55H (01010101B), then the following instruction,

```
ANL  A,R0
```

leaves 41H (01000001B) in the Accumulator.

When the destination is a directly addressed byte, this instruction clears combinations of bits in any RAM location or hardware register. The mask byte determining the pattern of bits to be cleared would either be a constant contained in the instruction or a value computed in the Accumulator at run-time. The following instruction,

```
ANL  P1,#01110011B
```

clears bits 7, 3, and 2 of output port 1.

ANL A,Rn

Bytes: 1

Cycles: 1

Encoding:

0	1	0	1	1	r	r	r

Operation: ANL
(A) ← (A) ∧ (Rn)

ANL A,direct

Bytes: 2

Cycles: 1

Encoding:

0	1	0	1	0	1	0	1	direct address

Operation: ANL
(A) ← (A) ∧ (direct)

ANL A,@RI

Bytes: 1

Cycles: 1

Encoding:

0	1	0	1	0	1	1	i

Operation: ANL
(A) ← (A) ∧ ((Ri))

ANL A,#data

Bytes: 2

Cycles: 1

Encoding:

0	1	0	1	0	1	0	0	immediate data

Operation: ANL
(A) ← (A) ∧ #data

ANL direct,A

Bytes: 2

Cycles: 1

Encoding:

0	1	0	1	0	0	1	0	direct address

Operation: ANL
(direct) ← (direct) ∧ (A)

ANL direct,#data

Bytes: 3

Cycles: 2

Encoding:

0	1	0	1	0	0	1	1	direct address	immediate data

Operation: ANL
(direct) ← (direct) ∧ #data

ANL C,<src-bit>

Function: Logical-AND for bit variables

Description: If the Boolean value of the source bit is a logical 0, then ANL C clears the carry flag; otherwise, this instruction leaves the carry flag in its current state. A slash (/) preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, *but the source bit itself is not affected.* No other flags are affected.

Only direct addressing is allowed for the source operand.

Example: Set the carry flag if, and only if, P1.0 = 1, ACC.7 = 1, and OV = 0:

```
MOV     C,P1.0      ;LOAD CARRY WITH INPUT PIN STATE
ANL     C,ACC.7     ;AND CARRY WITH ACCUM. BIT 7
ANL     C,/OV       ;AND WITH INVERSE OF OVERFLOW FLAG
```

ANL C,bit

Bytes: 2

Cycles: 2

Encoding:

1	0	0	0	0	0	1	0	bit address

Operation: ANL
(C) ← (C) ∧ (bit)

ANL C,/bit

Bytes: 2

Cycles: 2

Encoding:

1	0	1	1	0	0	0	0	bit address

Operation: ANL
(C) ← (C) ∧ ¬ (bit)

CJNE <dest-byte>,<src-byte>, rel

Function: Compare and Jump if Not Equal.

Description: CJNE compares the magnitudes of the first two operands and branches if their values are not equal. The branch destination is computed by adding the signed relative-displacement in the last instruction byte to the PC, after incrementing the PC to the start of the next instruction. The carry flag is set if the unsigned integer value of <dest-byte> is less than the unsigned integer value of <src-byte>; otherwise, the carry is cleared. Neither operand is affected.

The first two operands allow four addressing mode combinations: the Accumulator may be compared with any directly addressed byte or immediate data, and any indirect RAM location or working register can be compared with an immediate constant.

Example: The Accumulator contains 34H. Register 7 contains 56H. The first instruction in the sequence,

```
                CJNE    R7, # 60H, NOT_EQ
;               ...     .....           ;  R7 =  60H.
NOT_EQ:         JC      REQ_LOW         ;  IF R7 < 60H.
;               ...     .....           ;  R7 > 60H.
```

sets the carry flag and branches to the instruction at label NOT_EQ. By testing the carry flag, this instruction determines whether R7 is greater or less than 60H.

If the data being presented to Port 1 is also 34H, then the following instruction,

```
WAIT:   CJNE    A, P1,WAIT
```

clears the carry flag and continues with the next instruction in sequence, since the Accumulator does equal the data read from P1. (If some other value was being input on P1, the program loops at this point until the P1 data changes to 34H.)

CJNE A,direct,rel

Bytes: 3

Cycles: 2

Encoding:

1	0	1	1	0	1	0	1	direct address	rel. address

Operation:

```
(PC) ← (PC) + 3
IF (A) < > (direct)
THEN
    (PC) ← (PC) + relative offset
IF (A) < (direct)
THEN
    (C) ← 1
ELSE
    (C) ← 0
```

CJNE A,#data,rel

Bytes: 3

Cycles: 2

Encoding:

1	0	1	1	0	1	0	0	immediate data	rel. address

Operation:
(PC) ← (PC) + 3
IF (A) < > *data*
THEN
    (PC) ← (PC) + *relative offset*
IF (A) < *data*
THEN
    (C) ← 1
ELSE
    (C) ← 0

CJNE Rn,#data,rel

Bytes: 3

Cycles: 2

Encoding:

1	0	1	1	1	r	r	r	immediate data	rel. address

Operation:
(PC) ← (PC) + 3
IF (Rn) < > *data*
THEN
    (PC) ← (PC) + *relative offset*
IF (Rn) < *data*
THEN
    (C) ← 1
ELSE
    (C) ← 0

CJNE @Ri,data,rel

Bytes: 3

Cycles: 2

Encoding:

1	0	1	1	0	1	1	i	immediate data	rel. address

Operation:
(PC) ← (PC) + 3
IF ((Ri)) < > *data*
THEN
    (PC) ← (PC) + *relative offset*
IF ((Ri)) < *data*
THEN
    (C) ← 1
ELSE
    (C) ← 0

CLR A

Function: Clear Accumulator

Description: CLR A clears the Accumulator (all bits set to 0). No flags are affected

Example: The Accumulator contains 5CH (01011100B). The following instruction,

CLR A

leaves the Accumulator set to 00H (00000000B).

Bytes: 1

Cycles: 1

Encoding:

1	1	1	0	0	1	0	0

Operation: CLR
(A) ← 0

CLR bit

Function: Clear bit

Description: CLR bit clears the indicated bit (reset to 0). No other flags are affected. CLR can operate on the carry flag or any directly addressable bit.

Example: Port 1 has previously been written with 5DH (01011101B). The following instruction,

CLR P1.2

leaves the port set to 59H (01011001B).

CLR C

Bytes: 1

Cycles: 1

Encoding:

1	1	0	0	0	0	1	1

Operation: CLR
(C) ← 0

CLR bit

Bytes: 2

Cycles: 1

Encoding:

1	1	0	0	0	0	1	0

bit address

Operation: CLR
(bit) ← 0

CPL A

Function: Complement Accumulator

Description: CPLA logically complements each bit of the Accumulator (one's complement). Bits which previously contained a 1 are changed to a 0 and vice-versa. No flags are affected.

Example: The Accumulator contains 5CH (01011100B). The following instruction,

```
CPL   A
```

leaves the Accumulator set to 0A3H (10100011B).

Bytes: 1

Cycles: 1

Encoding:

1	1	1	1	0	1	0	0

Operation: CPL
(A) ← ¬ (A)

CPL bit

Function: Complement bit

Description: CPL bit complements the bit variable specified. A bit that had been a 1 is changed to 0 and vice-versa. No other flags are affected. CLR can operate on the carry or any directly addressable bit.

Note: When this instruction is used to modify an output pin, the value used as the original data is read from the output data latch, *not* the input pin.

Example: Port 1 has previously been written with 5BH (01011101B). The following instruction sequence,

```
CPL P1.1

CPL P1.2
```

leaves the port set to 5BH (01011011B).

CPL C

Bytes: 1

Cycles: 1

Encoding:

1	0	1	1	0	0	1	1

Operation: CPL
(C) ← ¬ (C)

CPL bit

Bytes: 2

Cycles: 1

Encoding:

1	0	1	1	0	0	1	0	bit address

Operation: CPL
(bit) ← ¬ (bit)

DA A

Function: Decimal-adjust Accumulator for Addition

Description: DA A adjusts the eight-bit value in the Accumulator resulting from the earlier addition of two variables (each in packed-BCD format), producing two four-bit digits. Any ADD or ADDC instruction may have been used to perform the addition.

If Accumulator bits 3 through 0 are greater than nine (xxxx1010-xxxx1111), or if the AC flag is one, six is added to the Accumulator producing the proper BCD digit in the low-order nibble. This internal addition sets the carry flag if a carry-out of the low-order four-bit field propagates through all high-order bits, but it does not clear the carry flag otherwise.

If the carry flag is now set, or if the four high-order bits now exceed nine (1010xxxx-1111xxxx), these high-order bits are incremented by six, producing the proper BCD digit in the high-order nibble. Again, this sets the carry flag if there is a carry-out of the high-order bits, but does not clear the carry. The carry flag thus indicates if the sum of the original two BCD variables is greater than 100, allowing multiple precision decimal addition. OV is not affected.

All of this occurs during the one instruction cycle. Essentially, this instruction performs the decimal conversion by adding 00H, 06H, 60H, or 66H to the Accumulator, depending on initial Accumulator and PSW conditions.

Note: DA A *cannot* simply convert a hexadecimal number in the Accumulator to BCD notation, nor does DA A apply to decimal subtraction.

Example: The Accumulator holds the value 56H (01010110B), representing the packed BCD digits of the decimal number 56. Register 3 contains the value 67H (01100111B), representing the packed BCD digits of the decimal number 67. The carry flag is set. The following instruction sequence

```
ADDC    A,R3
DA      A
```

first performs a standard two's-complement binary addition, resulting in the value 0BEH (10111110) in the Accumulator. The carry and auxiliary carry flags are cleared.

The Decimal Adjust instruction then alters the Accumulator to the value 24H (00100100B), indicating the packed BCD digits of the decimal number 24, the low-order two digits of the decimal sum of 56, 67, and the carry-in. The carry flag is set by the Decimal Adjust instruction, indicating that a decimal overflow occurred. The true sum of 56, 67, and 1 is 124.

BCD variables can be incremented or decremented by adding 01H or 99H. If the Accumulator initially holds 30H (representing the digits of 30 decimal), then the following instruction sequence,

```
ADD     A, # 99H
DA      A
```

leaves the carry set and 29H in the Accumulator, since 30 + 99 = 129. The low-order byte of the sum can be interpreted to mean 30 - 1 = 29.

Bytes: 1

Cycles: 1

Encoding:

1	1	0	1	0	1	0	0

Operation: DA

-contents of Accumulator are BCD

IF [[(A_{3-0}) > 9] ∨ [(AC) = 1]]
THEN $(A_{3-0}) \leftarrow (A_{3-0}) + 6$
AND
IF [[(A_{7-4}) > 9] ∨ [(C) = 1]]
THEN $(A_{7-4}) \leftarrow (A_{7-4}) + 6$

DEC byte

Function: Decrement

Description: DEC byte decrements the variable indicated by 1. An original value of 00H underflows to 0FFH. No flags are affected. Four operand addressing modes are allowed: accumulator, register, direct, or register-indirect.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: Register 0 contains 7FH (01111111B). Internal RAM locations 7EH and 7FH contain 00H and 40H, respectively. The following instruction sequence,

```
DEC     @R0
DEC     R0
DEC     @R0
```

leaves register 0 set to 7EH and internal RAM locations 7EH and 7FH set to 0FFH and 3FH.

DEC A

Bytes: 1

Cycles: 1

Encoding:

0	0	0	1	0	1	0	0

Operation: DEC
(A) ← (A) - 1

DEC Rn

Bytes: 1

Cycles: 1

Encoding:

0	0	0	1	1	r	r	r

Operation: DEC
(Rn) ← (Rn) - 1

DEC direct

Bytes: 2

Cycles: 1

Encoding:

0	0	0	1	0	1	0	1	direct address

Operation: DEC
(direct) ← (direct) - 1

DEC @Ri

Bytes: 1

Cycles: 1

Encoding:

0	0	0	1	0	1	1	i

Operation: DEC
((Ri)) ← ((Ri)) - 1

DIV AB

Function: Divide

Description: DIV AB divides the unsigned eight-bit integer in the Accumulator by the unsigned eight-bit integer in register B. The Accumulator receives the integer part of the quotient; register B receives the integer remainder. The carry and OV flags are cleared.

Exception: if B had originally contained 00H, the values returned in the Accumulator and B-register are undefined and the overflow flag are set. The carry flag is cleared in any case.

Example: The Accumulator contains 251 (0FBH or 11111011B) and B contains 18 (12H or 00010010B). The following instruction,

```
DIV    AB
```

leaves 13 in the Accumulator (0DH or 00001101B) and the value 17 (11H or 00010001B) in B, since 251 = (13 X 18) + 17. Carry and OV are both cleared.

Bytes: 1

Cycles: 4

Encoding:

1	0	0	0	0	1	0	0

Operation: DIV
$(A)_{15-8} \leftarrow (A)/(B)$
$(B)_{7-0}$

DJNZ <byte>,<rel-addr>

Function: Decrement and Jump if Not Zero

Description: DJNZ decrements the location indicated by 1, and branches to the address indicated by the second operand if the resulting value is not zero. An original value of 00H underflows to 0FFH. No flags are affected. The branch destination is computed by adding the signed relative-displacement value in the last instruction byte to the PC, after incrementing the PC to the first byte of the following instruction.

The location decremented may be a register or directly addressed byte.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins

Example: Internal RAM locations 40H, 50H, and 60H contain the values 01H, 70H, and 15H, respectively. The following instruction sequence,

```
DJNZ    40H,LABEL_1
DJNZ    50H,LABEL_2
DJNZ    60H,LABEL_3
```

causes a jump to the instruction at label LABEL_2 with the values 00H, 6FH, and 15H in the three RAM locations. The first jump was *not* taken because the result was zero.

This instruction provides a simple way to execute a program loop a given number of times or for adding a moderate time delay (from 2 to 512 machine cycles) with a single instruction. The following instruction sequence,

```
        MOV   R2, # 8
TOGGLE: CPL   P1.7
        DJNZ  R2,TOGGLE
```

toggles P1.7 eight times, causing four output pulses to appear at bit 7 of output Port 1. Each pulse lasts three machine cycles; two for DJNZ and one to alter the pin.

DJNZ Rn,rel

Bytes: 2

Cycles: 2

Encoding:

1	1	0	1	1	r	r	r	rel. address

Operation: DJNZ

```
(PC) ← (PC) + 2
(Rn) ← (Rn) - 1
IF  (Rn) > 0 or (Rn) < 0
    THEN
        (PC) ← (PC) + rel
```

DJNZ direct,rel

Bytes: 3

Cycles: 2

Encoding:

1	1	0	1	0	1	0	1	direct address	rel. address

Operation: DJNZ
(PC) ← (PC) + 2
(direct) ← (direct) - 1
IF (direct) > 0 or (direct) < 0
THEN
(PC) ← (PC) + rel

INC <byte>

Function: Increment

Description: INC increments the indicated variable by 1. An original value of 0FFH overflows to 00H. No flags are affected. Three addressing modes are allowed: register, direct, or register-indirect.

Note: When this instruction is used to modify an output port, the value used as the original port data will be read from the output data latch, *not* the input pins.

Example: Register 0 contains 7EH (011111110B). Internal RAM locations 7EH and 7FH contain 0FFH and 40H, respectively. The following instruction sequence,

```
INC     @R0
INC     R0
INC     @R0
```

leaves register 0 set to 7FH and internal RAM locations 7EH and 7FH holding 00H and 41H, respectively.

INC A

Bytes: 1

Cycles: 1

Encoding:

0	0	0	0	0	1	0	0

Operation: INC
(A) ← (A) + 1

INC Rn

Bytes: 1

Cycles: 1

Encoding:

0	0	0	0	1	r	r	r

Operation: INC
(Rn) ← (Rn) + 1

INC direct

Bytes: 2

Cycles: 1

Encoding:

0	0	0	0	0	1	0	1	direct address

Operation: INC
(direct) ← (direct) + 1

INC @Ri

Bytes: 1

Cycles: 1

Encoding:

0	0	0	0	0	1	1	i

Operation: INC
((Ri)) ← ((Ri)) + 1

INC DPTR

Function: Increment Data Pointer

Description: INC DPTR increments the 16-bit data pointer by 1. A 16-bit increment (modulo 2^{16}) is performed, and an overflow of the low-order byte of the data pointer (DPL) from 0FFH to 00H increments the high-order byte (DPH). No flags are affected.

This is the only 16-bit register which can be incremented.

Example: Registers DPH and DPL contain 12H and 0FEH, respectively. The following instruction sequence,

```
INC   DPTR
INC   DPTR
INC   DPTR
```

changes DPH and DPL to 13H and 01H.

Bytes: 1

Cycles: 2

Encoding:

1	0	1	0	0	0	1	1

Operation: INC
(DPTR) ← (DPTR) + 1

JB bit,rel

Function: Jump if Bit set

Description: If the indicated bit is a one, JB jump to the address indicated; otherwise, it proceeds with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example: The data present at input port 1 is 11001010B. The Accumulator holds 56 (01010110B). The following instruction sequence,

```
JB   P1.2,LABEL1
JB   ACC. 2,LABEL2
```

causes program execution to branch to the instruction at label LABEL2.

Bytes: 3

Cycles: 2

Encoding:

0	0	1	0	0	0	0	0	bit address	rel. address

Operation:

```
JB
(PC) ← (PC) + 3
IF (bit) = 1
    THEN
        (PC) ← (PC) + rel
```

JBC bit,rel

Function: Jump if Bit is set and Clear bit

Description: If the indicated bit is one, JBC branches to the address indicated; otherwise, it proceeds with the next instruction. *The bit will not be cleared if it is already a zero.* The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. No flags are affected.

Note: When this instruction is used to test an output pin, the value used as the original data will be read from the output data latch, *not* the input pin.

Example: The Accumulator holds 56H (01010110B). The following instruction sequence,

```
JBC   ACC.3,LABEL1
JBC   ACC.2,LABEL2
```

causes program execution to continue at the instruction identified by the label LABEL2, with the Accumulator modified to 52H (01010010B).

Bytes: 3

Cycles: 2

Encoding:

0	0	0	1	0	0	0	0	bit address	rel. address

Operation: JBC

```
(PC) ← (PC) + 3
IF  (bit) = 1
      THEN
            (bit) ← 0
            (PC) ← (PC) +rel
```

JC rel

Function: Jump if Carry is set

Description: If the carry flag is set, JC branches to the address indicated; otherwise, it proceeds with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. No flags are affected.

Example: The carry flag is cleared. The following instruction sequence,

```
JC      LABEL1
CPL     C
JC      LABEL 2
```

sets the carry and causes program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0	1	0	0	0	0	0	0	rel. address

Operation: JC

```
(PC) ← (PC) + 2
IF (C) = 1
      THEN
            (PC) ← (PC) + rel
```

JMP @A+DPTR

Function: Jump indirect

Description: JMP @A+DPTR adds the eight-bit unsigned contents of the Accumulator with the 16-bit data pointer and loads the resulting sum to the program counter. This is the address for subsequent instruction fetches. Sixteen-bit addition is performed (modulo 2^{16}): a carry-out from the low-order eight bits propagates through the higher-order bits. Neither the Accumulator nor the Data Pointer is altered. No flags are affected.

Example: An even number from 0 to 6 is in the Accumulator. The following sequence of instructions branches to one of four AJMP instructions in a jump table starting at JMP_TBL.

```
            MOV   DPTR, # JMP_TBL
            JMP   @A + DPTR
JMP_TBL:    AJMP  LABEL0
            AJMP  LABEL1
            AJMP  LABEL2
            AJMP  LABEL3
```

If the Accumulator equals 04H when starting this sequence, execution jumps to label LABEL2. Because AJMP is a 2-byte instruction, the jump instructions start at every other address.

Bytes: 1

Cycles: 2

Encoding:

0	1	1	1	0	0	1	1

Operation: JMP
(PC) ← (A) + (DPTR)

JNB bit,rel

Function: Jump if Bit Not set

Description: If the indicated bit is a 0, JNB branches to the indicated address; otherwise, it proceeds with the next instruction. The branch destination is computed by adding the signed relative-displacement in the third instruction byte to the PC, after incrementing the PC to the first byte of the next instruction. *The bit tested is not modified.* No flags are affected.

Example: The data present at input port 1 is 11001010B. The Accumulator holds 56H (01010110B). The following instruction sequence,

```
JNB   P1.3,LABEL1
JNB   ACC.3,LABEL2
```

causes program execution to continue at the instruction at label LABEL2.

Bytes: 3

Cycles: 2

Encoding:

0 0 1 1 0 0 0 0	bit address	rel. address

Operation: JNB
(PC) ← (PC) + 3
IF (bit) = 0
THEN (PC) ← (PC) +rel

JNC rel

Function: Jump if Carry not set

Description: If the carry flag is a 0, JNC branches to the address indicated; otherwise, it proceeds with the next instruction. The branch destination is computed by adding the signal relative-displacement in the second instruction byte to the PC, after incrementing the PC twice to point to the next instruction. The carry flag is not modified.

Example: The carry flag is set. The following instruction sequence,

```
JNC   LABEL1

CPL   C

JNC   LABEL2
```

clears the carry and causes program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0 1 0 1 0 0 0 0	rel. address

Operation: JNC
(PC) ← (PC) + 2
IF (C) = 0
THEN (PC) ← (PC) +rel

JNZ rel

Function: Jump if Accumulator Not Zero

Description: If any bit of the Accumulator is a one, JNZ branches to the indicated address; otherwise, it proceeds with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example: The Accumulator originally holds 00H. The following instruction sequence,

```
JNZ     LABEL1
INC     A
JNZ     LABEL2
```

sets the Accumulator to 01H and continues at label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0	1	1	1	0	0	0	0	rel. address

Operation:

```
JNZ
(PC) ← (PC) + 2
IF   (A) ≠ 0
     THEN (PC) ← (PC) + rel
```

JZ rel

Function: Jump if Accumulator Zero

Description If all bits of the Accumulator are 0, JZ branches to the address indicated; otherwise, it proceeds with the next instruction. The branch destination is computed by adding the signed relative-displacement in the second instruction byte to the PC, after incrementing the PC twice. The Accumulator is not modified. No flags are affected.

Example: The Accumulator originally contains 01H. The following instruction sequence,

```
JZ      LABEL1

DEC     A

JZ      LABEL2
```

changes the Accumulator to 00H and causes program execution to continue at the instruction identified by the label LABEL2.

Bytes: 2

Cycles: 2

Encoding:

0	1	1	0	0	0	0	0	rel. address

Operation:

```
JZ
(PC) ← (PC) + 2
IF   (A) = 0
     THEN (PC) ← (PC) + rel
```

LCALL addr16

Function: Long call

Description: LCALL calls a subroutine located at the indicated address. The instruction adds three to the program counter to generate the address of the next instruction and then pushes the 16-bit result onto the stack (low byte first), incrementing the Stack Pointer by two. The high-order and low-order bytes of the PC are then loaded, respectively, with the second and third bytes of the LCALL instruction. Program execution continues with the instruction at this address. The subroutine may therefore begin anywhere in the full 64K-byte program memory address space. No flags are affected.

Example: Initially the Stack Pointer equals 07H. The label SUBRTN is assigned to program memory location 1234H. After executing the instruction,

```
LCALL   SUBRTN
```

at location 0123H, the Stack Pointer will contain 09H, internal RAM locations 08H and 09H will contain 26H and 01H, and the PC will contain 1234H.

Bytes: 3

Cycles: 2

Encoding:

0 0 0 1	0 0 1 0	addr15-addr8	addr7-addr0

Operation: LCALL
(PC) ← (PC) + 3
(SP) ← (SP) + 1
((SP)) ← (PC_{7-0})
(SP) ← (SP) + 1
((SP)) ← (PC_{15-8})
(PC) ← $addr_{15-0}$

LJMP addr16

Function: Long Jump

Description: LJMP causes an unconditional branch to the indicated address, by loading the high-order and low-order bytes of the PC (respectively) with the second and third instruction bytes. The destination may therefore be anywhere in the full 64K program memory address space. No flags are affected.

Example: The label JMPADR is assigned to the instruction at program memory location 1234H. The instruction,

```
LJMP   JMPADR
```

at location 0123H will load the program counter with 1234H.

Bytes: 3

Cycles: 2

Encoding:

0 0 0 0	0 0 1 0	addr15-addr8	addr7-addr0

Operation: LJMP
(PC) ← $addr_{15-0}$

MOV <dest-byte>,<src-byte>

Function: Move byte variable

Description: The byte variable indicated by the second operand is copied into the location specified by the first operand. The source byte is not affected. No other register or flag is affected.

This is by far the most flexible operation. Fifteen combinations of source and destination addressing modes are allowed.

Example: Internal RAM location 30H holds 40H. The value of RAM location 40H is 10H. The data present at input port 1 is 11001010B (0CAH).

```
MOV   R0,#30H     ;R0 <= 30H
MOV   A,@R0       ;A <= 40H
MOV   R1,A        ;R1 <= 40H
MOV   B,@R1       ;B <= 10H
MOV   @R1,P1      ;RAM (40H) <= 0CAH
MOV   P2,P1       ;P2 #0CAH
```

leaves the value 30H in register 0, 40H in both the Accumulator and register 1, 10H in register B, and 0CAH (11001010B) both in RAM location 40H and output on port 2.

MOV A,Rn

Bytes: 1

Cycles: 1

Encoding:

1	1	1	0	1	r	r	r

Operation: MOV
(A) ← (Rn)

***MOV A,direct**

Bytes: 2

Cycles: 1

Encoding:

1	1	1	0	0	1	0	1

direct address

Operation: MOV
(A) ← (direct)

*** MOV A,ACC is not a valid Instruction.**

MOV A,@Ri

Bytes: 1

Cycles: 1

Encoding:

1	1	1	0	0	1	1	i

Operation: MOV
(A) ← ((Ri))

MOV A,#data

Bytes: 2

Cycles: 1

Encoding:

0	1	1	1	0	1	0	0	immediate data

Operation: MOV
(A) ← #data

MOV Rn,A

Bytes: 1

Cycles: 1

Encoding:

1	1	1	1	1	r	r	r

Operation: MOV
(Rn) ← (A)

MOV Rn,direct

Bytes: 2

Cycles: 2

Encoding:

1	0	1	0	1	r	r	r	direct addr.

Operation: MOV
(Rn) ← (direct)

MOV Rn,#data

Bytes: 2

Cycles: 1

Encoding:

0	1	1	1	1	r	r	r	immediate data

Operation: MOV
(Rn) ← #data

MOV direct,A

Bytes: 2

Cycles: 1

Encoding:

1	1	1	1	0	1	0	1	direct address

Operation: MOV
(direct) ← (A)

MOV direct,Rn

Bytes: 2

Cycles: 2

Encoding:

1	0	0	0	1	r	r	r	direct address

Operation: MOV
(direct) ← (Rn)

MOV direct,direct

Bytes: 3

Cycles: 2

Encoding:

1	0	0	0	0	1	0	1	dir. addr. (scr)	dir. addr. (dest)

Operation: MOV
(direct) ← (direct)

MOV direct,@Ri

Bytes: 2

Cycles: 2

Encoding:

1	0	0	0	0	1	1	i	direct addr.

Operation: MOV
(direct) ← ((Ri))

MOV direct,#data

Bytes: 3

Cycles: 2

Encoding:

0	1	1	1	0	1	0	1	direct address	immediate data

Operation: MOV
(direct) ← #data

MOV @Ri,A

Bytes: 1

Cycles: 1

Encoding:

1	1	1	1	0	1	1	i

Operation: MOV
((Ri)) ← (A)

MOV @Ri,direct

Bytes: 2

Cycles: 2

Encoding:

1	0	1	0	0	1	1	i

direct addr.

Operation: MOV
((Ri)) ← (direct)

MOV @Ri,#data

Bytes: 2

Cycles: 1

Encoding:

0	1	1	1	0	1	1	i

immediate data

Operation: MOV
((Ri)) ← #data

MOV <dest-bit>,<src-bit>

Function: Move bit data

Description: MOV <dest-bit>,<src-bit> copies the Boolean variable indicated by the second operand into the location specified by the first operand. One of the operands must be the carry flag; the other may be any directly addressable bit. No other register or flag is affected.

Example: The carry flag is originally set. The data present at input Port 3 is 11000101B. The data previously written to output Port 1 is 35H (00110101B).

```
MOV     P1.3,C
MOV     C,P3.3
MOV     P1.2,C
```

leaves the carry cleared and changes Port 1 to 39H (00111001B).

MOV C,bit

Bytes: 2

Cycles: 1

Encoding:

1	0	1	0	0	0	1	0	bit address

Operation: MOV
(C) ← (bit)

MOV bit,C

Bytes: 2

Cycles: 2

Encoding:

1	0	0	1	0	0	1	0	bit address

Operation: MOV
(bit) ← (C)

MOV DPTR,#data16

Function: Load Data Pointer with a 16-bit constant

Description: MOV DPTR,#data16 loads the Data Pointer with the 16-bit constant indicated. The 16-bit constant is loaded into the second and third bytes of the instruction. The second byte (DPH) is the high-order byte, while the third byte (DPL) holds the lower-order byte. No flags are affected.

This is the only instruction which moves 16 bits of data at once.

Example: The instruction,

MOV DPTR, # 1234H

loads the value 1234H into the Data Pointer: DPH holds 12H, and DPL holds 34H.

Bytes: 3

Cycles: 2

Encoding:

1	0	0	1	0	0	0	0	immed. data15-8	immed. data7-0

Operation: MOV
(DPTR) ← $\#data_{15-0}$
DPH ☐ DPL ← $\#data_{15-8}$ ☐ $\#data_{7-0}$

MOVC A,@A+ <base-reg>

Function: Move Code byte

Description: The MOVC instructions load the Accumulator with a code byte or constant from program memory. The address of the byte fetched is the sum of the original unsigned 8-bit Accumulator contents and the contents of a 16-bit base register, which may be either the Data Pointer or the PC. In the latter case, the PC is incremented to the address of the following instruction before being added with the Accumulator; otherwise the base register is not altered. Sixteen-bit addition is performed so a carry-out from the low-order eight bits may propagate through higher-order bits. No flags are affected.

Example: A value between 0 and 3 is in the Accumulator. The following instructions will translate the value in the Accumulator to one of four values defined by the DB (define byte) directive.

```
REL_PC:  INC   A
         MOVC  A,@A+PC
         RET
         DB    66H
         DB    77H
         DB    88H
         DB    99H
```

If the subroutine is called with the Accumulator equal to 01H, it returns with 77H in the Accumulator. The INC A before the MOVC instruction is needed to "get around" the RET instruction above the table. If several bytes of code separate the MOVC from the table, the corresponding number is added to the Accumulator instead.

MOVC A,@A+DPTR

Bytes: 1

Cycles: 2

Encoding:

1	0	0	1	0	0	1	1

Operation: MOVC
(A) ← ((A) + (DPTR))

MOVC A,@A+PC

Bytes: 1

Cycles: 2

Encoding:

1	0	0	0	0	0	1	1

Operation: MOVC
(PC) ← (PC) + 1
(A) ← ((A) + (PC))

MOVX <dest-byte>,<src-byte>

Function: Move External

Description: The MOVX instructions transfer data between the Accumulator and a byte of external data memory, which is why "X" is appended to MOV. There are two types of instructions, differing in whether they provide an 8-bit or 16-bit indirect address to the external data RAM.

In the first type, the contents of R0 or R1 in the current register bank provide an 8-bit address multiplexed with data on P0. Eight bits are sufficient for external I/O expansion decoding or for a relatively small RAM array. For somewhat larger arrays, any output port pins can be used to output higher-order address bits. These pins are controlled by an output instruction preceding the MOVX.

In the second type of MOVX instruction, the Data Pointer generates a 16-bit address. P2 outputs the high-order eight address bits (the contents of DPH), while P0 multiplexes the low-order eight bits (DPL) with data. The P2 Special Function Register retains its previous contents, while the P2 output buffers emit the contents of DPH. This form of MOVX is faster and more efficient when accessing very large data arrays (up to 64K bytes), since no additional instructions are needed to set up the output ports.

It is possible to use both MOVX types in some situations. A large RAM array with its high-order address lines driven by P2 can be addressed via the Data Pointer, or with code to output high-order address bits to P2, followed by a MOVX instruction using R0 or R1.

Example: An external 256 byte RAM using multiplexed address/data lines is connected to the 8051 Port 0. Port 3 provides control lines for the external RAM. Ports 1 and 2 are used for normal I/O. Registers 0 and 1 contain 12H and 34H. Location 34H of the external RAM holds the value 56H. The instruction sequence,

```
MOVX    A,@R1

MOVX    @R0,A
```

copies the value 56H into both the Accumulator and external RAM location 12H.

MOVX A,@Ri

Bytes: 1

Cycles: 2

Encoding:

1	1	1	0	0	0	1	i

Operation: MOVX
(A) ← ((Ri))

MOVX A,@DPTR

Bytes: 1

Cycles: 2

Encoding:

1	1	1	0	0	0	0	0

Operation: MOVX
(A) ← ((DPTR))

MOVX @Ri,A

Bytes: 1

Cycles: 2

Encoding:

1	1	1	1	0	0	1	i

Operation: MOVX
((Ri)) ← (A)

MOVX @DPTR,A

Bytes: 1

Cycles: 2

Encoding:

1	1	1	1	0	0	0	0

Operation: MOVX
(DPTR) ← (A)

MUL AB

Function: Multiply

Description: MUL AB multiplies the unsigned 8-bit integers in the Accumulator and register B. The low-order byte of the 16-bit product is left in the Accumulator, and the high-order byte in B. If the product is greater than 255 (0FFH), the overflow flag is set; otherwise it is cleared. The carry flag is always cleared.

Example: Originally the Accumulator holds the value 80 (50H). Register B holds the value 160 (0A0H). The instruction,

```
MUL     AB
```

will give the product 12,800 (3200H), so B is changed to 32H (00110010B) and the Accumulator is cleared. The overflow flag is set, carry is cleared.

Bytes: 1

Cycles: 4

Encoding:

1	0	1	0	0	1	0	0

Operation: MUL

$(A)_{7-0} \leftarrow (A) \times (B)$

$(B)_{15-8}$

NOP

Function: No Operation

Description: Execution continues at the following instruction. Other than the PC, no registers or flags are affected.

Example: A low-going output pulse on bit 7 of Port 2 must last exactly 5 cycles. A simple SETB/CLR sequence generates a one-cycle pulse, so four additional cycles must be inserted. This may be done (assuming no interrupts are enabled) with the following instruction sequence,

```
CLR     P2.7
NOP
NOP
NOP
NOP
SETB    P2.7
```

Bytes: 1

Cycles: 1

Encoding:

0	0	0	0	0	0	0	0

Operation: NOP

$(PC) \leftarrow (PC) + 1$

ORL <dest-byte> <src-byte>

Function: Logical-OR for byte variables

Description: ORL performs the bitwise logical-OR operation between the indicated variables, storing the results in the destination byte. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

Note: When this instruction is used to modify an output port, the value used as the original port data is read from the output data latch, *not* the input pins.

Example: If the Accumulator holds 0C3H (11000011B) and R0 holds 55H (01010101B) then the following instruction,

```
ORL     A,R0
```

leaves the Accumulator holding the value 0D7H (11010111B).

When the destination is a directly addressed byte, the instruction can set combinations of bits in any RAM location or hardware register. The pattern of bits to be set is determined by a mask byte, which may be either a constant data value in the instruction or a variable computed in the Accumulator at run-time. The instruction,

```
ORL     P1,#00110010B
```

sets bits 5, 4, and 1 of output Port 1.

ORL A,Rn

Bytes: 1

Cycles: 1

Encoding:

0	1	0	0	1	r	r	r

Operation: ORL
(A) ← (A) ∨ (Rn)

ORL A,direct

Bytes: 2

Cycles: 1

Encoding:

0	1	0	0	0	1	0	1	direct address

Operation: ORL
(A) ← (A) ∨ (direct)

ORL A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0	1	0	0	0	1	1	i

Operation: ORL
(A) ← (A) ∨((Ri))

ORL A,#data

Bytes: 2

Cycles: 1

Encoding:

0	1	0	0	0	1	0	0	immediate data

Operation: ORL
(A) ← (A) ∨ #data

ORL direct,A

Bytes: 2

Cycles: 1

Encoding:

0	1	0	0	0	0	1	0	direct address

Operation: ORL
(direct) ← (direct) ∨ (A)

ORL direct,#data

Bytes: 3

Cycles: 2

Encoding:

0	1	0	0	0	0	1	1	direct addr.	immediate data

Operation: ORL
(direct) ← (direct) ∨ #data

ORL C,<src-bit>

Function: Logical-OR for bit variables

Description: Set the carry flag if the Boolean value is a logical 1; leave the carry in its current state otherwise. A slash (/) preceding the operand in the assembly language indicates that the logical complement of the addressed bit is used as the source value, but the source bit itself is not affected. No other flags are affected.

Example: Set the carry flag if and only if P1.0 = 1, ACC. 7 = 1, or OV = 0:

```
MOV     C,P1.0      ;LOAD CARRY WITH INPUT PIN P10
ORL     C,ACC.7     ;OR CARRY WITH THE ACC. BIT 7
ORL     C,/OV       ;OR CARRY WITH THE INVERSE OF OV.
```

ORL C,bit

Bytes: 2

Cycles: 2

Encoding:

0	1	1	1	0	0	1	0	bit address

Operation: ORL
(C) ← (C) ∨ (bit)

ORL C,/bit

Bytes: 2

Cycles: 2

Encoding:

1	0	1	0	0	0	0	0	bit address

Operation: ORL
(C) ← (C) ∨ ($\overline{\text{bit}}$)

POP direct

Function: Pop from stack.

Description: The contents of the internal RAM location addressed by the Stack Pointer is read, and the Stack Pointer is decremented by one. The value read is then transferred to the directly addressed byte indicated. No flags are affected.

Example: The Stack Pointer originally contains the value 32H, and internal RAM locations 30H through 32H contain the values 20H, 23H, and 01H, respectively. The following instruction sequence,

```
POP     DPH

POP     DPL
```

leaves the Stack Pointer equal to the value 30H and sets the Data Pointer to 0123H. At this point, the following instruction,

```
POP     SP
```

leaves the Stack Pointer set to 20H. In this special case, the Stack Pointer was decremented to 2FH before being loaded with the value popped (20H).

Bytes: 2

Cycles: 2

Encoding:

1	1	0	1	0	0	0	0	direct address

Operation: POP
(direct) ← ((SP))
(SP) ← (SP) - 1

PUSH direct

Function: Push onto stack

Description: The Stack Pointer is incremented by one. The contents of the indicated variable is then copied into the internal RAM location addressed by the Stack Pointer. Otherwise no flags are affected.

Example: On entering an interrupt routine, the Stack Pointer contains 09H. The Data Pointer holds the value 0123H. The following instruction sequence,

```
PUSH    DPL

PUSH    DPH
```

leaves the Stack Pointer set to 0BH and stores 23H and 01H in internal RAM locations 0AH and 0BH, respectively.

Bytes: 2

Cycles: 2

Encoding:

1	1	0	0	0	0	0	0	direct address

Operation: PUSH
(SP) ← (SP) + 1
((SP)) ← (direct)

RET

Function: Return from subroutine

Description: RET pops the high- and low-order bytes of the PC successively from the stack, decrementing the Stack Pointer by two. Program execution continues at the resulting address, generally the instruction immediately following an ACALL or LCALL. No flags are affected.

Example: The Stack Pointer originally contains the value 0BH. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The following instruction,

RET

leaves the Stack Pointer equal to the value 09H. Program execution continues at location 0123H.

Bytes: 1

Cycles: 2

Encoding:

0	0	1	0	0	0	1	0

Operation: RET
$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

RETI

Function: Return from interrupt

Description: RETI pops the high- and low-order bytes of the PC successively from the stack and restores the interrupt logic to accept additional interrupts at the same priority level as the one just processed. The Stack Pointer is left decremented by two. No other registers are affected; the PSW is *not* automatically restored to its pre-interrupt status. Program execution continues at the resulting address, which is generally the instruction immediately after the point at which the interrupt request was detected. If a lower- or same-level interrupt was pending when the RETI instruction is executed, that one instruction is executed before the pending interrupt is processed.

Example: The Stack Pointer originally contains the value 0BH. An interrupt was detected during the instruction ending at location 0122H. Internal RAM locations 0AH and 0BH contain the values 23H and 01H, respectively. The following instruction,

RETI

leaves the Stack Pointer equal to 09H and returns program execution to location 0123H.

Bytes: 1

Cycles: 2

Encoding:

0	0	1	1	0	0	1	0

Operation: RETI
$(PC_{15-8}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$
$(PC_{7-0}) \leftarrow ((SP))$
$(SP) \leftarrow (SP) - 1$

RL A

Function: Rotate Accumulator Left

Description: The eight bits in the Accumulator are rotated one bit to the left. Bit 7 is rotated into the bit 0 position. No flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B). The following instruction,

```
RL    A
```

leaves the Accumulator holding the value 8BH (10001011B) with the carry unaffected.

Bytes: 1

Cycles: 1

Encoding:

0	0	1	0	0	0	1	1

Operation: RL
$(A_n + 1) \leftarrow (An)$ n = 0 - 6
$(A0) \leftarrow (A7)$

RLC A

Function: Rotate Accumulator Left through the Carry flag

Description: The eight bits in the Accumulator and the carry flag are together rotated one bit to the left. Bit 7 moves into the carry flag; the original state of the carry flag moves into the bit 0 position. No other flags are affected.

Example: The Accumulator holds the value 0C5H(11000101B), and the carry is zero. The following instruction,

```
RLC    A
```

leaves the Accumulator holding the value 8BH (10001010B) with the carry set.

Bytes: 1

Cycles: 1

Encoding:

0	0	1	1	0	0	1	1

Operation: RLC
$(An + 1) \leftarrow (An)$ n = 0 - 6
$(A0) \leftarrow (C)$
$(C) \leftarrow (A7)$

RR A

Function: Rotate Accumulator Right

Description: The eight bits in the Accumulator are rotated one bit to the right. Bit 0 is rotated into the bit 7 position. No flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B). The following instruction,

```
RR    A
```

leaves the Accumulator holding the value 0E2H (11100010B) with the carry unaffected.

Bytes: 1

Cycles: 1

Encoding:

0	0	0	0	0	0	1	1

Operation: RR
(An) ← (A_n + 1) n = 0 - 6
(A7) ← (A0)

RRC A

Function: Rotate Accumulator Right through Carry flag

Description: The eight bits in the Accumulator and the carry flag are together rotated one bit to the right. Bit 0 moves into the carry flag; the original value of the carry flag moves into the bit 7 position. No other flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B), the carry is zero. The following instruction,

```
RRC    A
```

leaves the Accumulator holding the value 62 (01100010B) with the carry set.

Bytes: 1

Cycles: 1

Encoding:

0	0	0	1	0	0	1	1

Operation: RRC
(An) ← (An + 1) n = 0 - 6
(A7) ← (C)
(C) ← (A0)

SETB <bit>

Function: Set Bit

Description: SETB sets the indicated bit to one. SETB can operate on the carry flag or any directly addressable bit. No other flags are affected.

Example: The carry flag is cleared. Output Port 1 has been written with the value 34H (00110100B). The following instructions,

```
SETB    C

SETB    P1.0
```

sets the carry flag to 1 and changes the data output on Port 1 to 35H (00110101B).

SETB C

Bytes: 1

Cycles: 1

Encoding:

1	1	0	1	0	0	1	1

Operation: SETB
(C) ← 1

SETB bit

Bytes: 2

Cycles: 1

Encoding:

1	1	0	1	0	0	1	0	bit address

Operation: SETB
(bit) ← 1

SJMP rel

Function: Short Jump

Description: Program control branches unconditionally to the address indicated. The branch destination is computed by adding the signed displacement in the second instruction byte to the PC, after incrementing the PC twice. Therefore, the range of destinations allowed is from 128 bytes preceding this instruction 127 bytes following it.

Example: The label RELADR is assigned to an instruction at program memory location 0123H. The following instruction,

```
SJMP   RELADR
```

assembles into location 0100H. After the instruction is executed, the PC contains the value 0123H.

(*Note:* Under the above conditions the instruction following SJMP is at 102H. Therefore, the displacement byte of the instruction is the relative offset (0123H-0102H) = 21H. Put another way, an SJMP with a displacement of 0FEH is a one-instruction infinite loop.)

Bytes: 2

Cycles: 2

Encoding:

1	0	0	0	0	0	0	0	rel. address

Operation: SJMP
(PC) ← (PC) + 2
(PC) ← (PC) + rel

SUBB A,<src-byte>

Function: Subtract with borrow

Description: SUBB subtracts the indicated variable and the carry flag together from the Accumulator, leaving the result in the Accumulator. SUBB sets the carry (borrow) flag if a borrow is needed for bit 7 and clears C otherwise. (If C was set *before* executing a SUBB instruction, this indicates that a borrow was needed for the previous step in a multiple-precision subtraction, so the carry is subtracted from the Accumulator along with the source operand.) AC is set if a borrow is needed for bit 3 and cleared otherwise. OV is set if a borrow is needed into bit 6, but not into bit 7, or into bit 7, but not bit 6.

When subtracting signed integers, OV indicates a negative number produced when a negative value is subtracted from a positive value, or a positive result when a positive number is subtracted from a negative number.

The source operand allows four addressing modes: register, direct, register-indirect, or immediate.

Example: The Accumulator holds 0C9H (11001001B), register 2 holds 54H (01010100B), and the carry flag is set. The instruction,

SUBB A,R2

will leave the value 74H (01110100B) in the accumulator, with the carry flag and AC cleared but OV set.

Notice that 0C9H minus 54H is 75H. The difference between this and the above result is due to the carry (borrow) flag being set before the operation. If the state of the carry is not known before starting a single or multiple-precision subtraction, it should be explicitly cleared by CLR C instruction.

SUBB A,Rn

Bytes: 1

Cycles: 1

Encoding:

1	0	0	1	1	r	r	r

Operation: SUBB
(A) ← (A) - (C) - (Rn)

SUBB A,direct

Bytes: 2

Cycles: 1

Encoding:

1	0	0	1	0	1	0	1	direct address

Operation: SUBB
(A) ← (A) - (C) - (direct)

SUBB A,@Ri

Bytes: 1

Cycles: 1

Encoding:

1	0	0	1	0	1	1	i

Operation: SUBB
(A) ← (A) - (C) - ((Ri))

SUBB A,#data

Bytes: 2

Cycles: 1

Encoding:

1	0	0	1	0	1	0	0	immediate data

Operation: SUBB
(A) ← (A) - (C) - #data

SWAP A

Function: Swap nibbles within the Accumulator

Description: SWAP A interchanges the low- and high-order nibbles (four-bit fields) of the Accumulator (bits 3 through 0 and bits 7 through 4). The operation can also be thought of as a 4-bit rotate instruction. No flags are affected.

Example: The Accumulator holds the value 0C5H (11000101B). The instruction,

SWAP A

leaves the Accumulator holding the value 5CH (01011100B).

Bytes: 1

Cycles: 1

Encoding:

1	1	0	0	0	1	0	0

Operation: SWAP
$(A_{3-0}) \rightleftarrows (A_{7-4})$

XCH A,<byte>

Function: Exchange Accumulator with byte variable

Description: XCH loads the Accumulator with the contents of the indicated variable, at the same time writing the original Accumulator contents to the indicated variable. The source/destination operand can use register, direct, or register-indirect addressing.

Example: R0 contains the address 20H. The Accumulator holds the value 3FH (0011111lB). Internal RAM location 20H holds the value 75H (01110101B). The following instruction,

```
XCH    A,@R0
```

leaves RAM location 20H holding the values 3FH (00111111B) and 75H (01110101B) in the accumulator.

XCH A,Rn

Bytes: 1

Cycles: 1

Encoding:

1	1	0	0	1	r	r	r

Operation: XCH
(A) $\rightleftarrows$ ((Rn)

XCH A,direct

Bytes: 2

Cycles: 1

Encoding:

1	1	0	0	0	1	0	1

direct address

Operation: XCH
(A) $\rightleftarrows$ (direct)

XCH A,@Ri

Bytes: 1

Cycles: 1

Encoding:

1	1	0	0	0	1	1	i

Operation: XCH
(A) $\rightleftarrows$ ((Ri))

XCHD A,@RI

Function: Exchange Digit

Description: XCHD exchanges the low-order nibble of the Accumulator (bits 3 through 0), generally representing a hexadecimal or BCD digit, with that of the internal RAM location indirectly addressed by the specified register. The high-order nibbles (bits 7-4) of each register are not affected. No flags are affected.

Example: R0 contains the address 20H. The Accumulator holds the value 36H (00110110B). Internal RAM location 20H holds the value 75H (01110101B). The following instruction,

```
XCHD    A,@R0
```

leaves RAM location 20H holding the value 76H (01110110B) and 35H (00110101B) in the Accumulator.

Bytes: 1

Cycles: 1

Encoding:

1	1	0	1	0	1	1	i

Operation: XCHD
$(A_{3-0}) \rightleftarrows ((Ri_{3-0}))$

XRL <dest-byte>,<src-byte>

Function: Logical Exclusive-OR for byte variables

Description: XRL performs the bitwise logical Exclusive-OR operation between the indicated variables, storing the results in the destination. No flags are affected.

The two operands allow six addressing mode combinations. When the destination is the Accumulator, the source can use register, direct, register-indirect, or immediate addressing; when the destination is a direct address, the source can be the Accumulator or immediate data.

(Note: When this instruction is used to modify an output port, the value used as the original port data is read from the output data latch, *not* the input pins.)

Example: If the Accumulator holds 0C3H (11000011B) and register 0 holds 0AAH (10101010B) then the instruction,

```
XRL     A,R0
```

leaves the Accumulator holding the value 69H (01101001B).

When the destination is a directly addressed byte, this instruction can complement combinations of bits in any RAM location or hardware register. The pattern of bits to be complemented is then determined by a mask byte, either a constant contained in the instruction or a variable computed in the Accumulator at run-time. The following instruction,

```
XRL     P1,#00110001B
```

complements bits 5, 4, and 0 of output Port 1.

XRL A,Rn

Bytes: 1

Cycles: 1

Encoding:

0	1	1	0	1	r	r	r

Operation: XRL
(A) ← (A) ∀ (Rn)

XRL A,direct

Bytes: 2

Cycles: 1

Encoding:

0	1	1	0	0	1	0	1

direct address

Operation: XRL
(A) ← (A) ∀ (direct)

XRL A,@Ri

Bytes: 1

Cycles: 1

Encoding:

0	1	1	0	0	1	1	i

Operation: XRL
(A) ← (A) ∀ ((Ri))

XRL A,#data

Bytes: 2

Cycles: 1

Encoding:

0	1	1	0	0	1	0	0

immediate data

Operation: XRL
(A) ← (A) ∀ #data

XRL direct,A

Bytes: 2

Cycles: 1

Encoding:

0	1	1	0	0	0	1	0

direct address

Operation: XRL
(direct) ← (direct) ∀ (A)

XRL direct,#data

Bytes: 3

Cycles: 2

Encoding:

0	1	1	0	0	0	1	1	direct address	immediate data

Operation: XRL
(direct) ← (direct) ∀ #data

Contents

Section 3 Microcontroller Data Sheets

Features

- Compatible with MCS-51™ Products
- 1 Kbyte of Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 2.7 V to 6 V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-Level Program Memory Lock
- 64 bytes SRAM
- 15 Programmable I/O Lines
- One 16-Bit Timer/Counter
- Three Interrupt Sources
- Direct LED Drive Outputs
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes

8-Bit Microcontroller with 1 Kbyte Flash

3

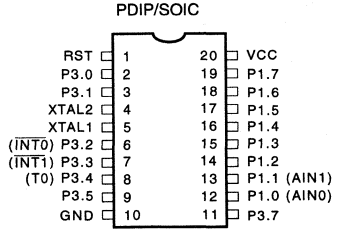
Description

The AT89C1051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 1 Kbyte of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C1051 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C1051 provides the following standard features: 1 Kbyte of Flash, 64 bytes of RAM, 15 I/O lines, one 16-bit timer/counter, a three vector two-level interrupt architecture, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C1051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

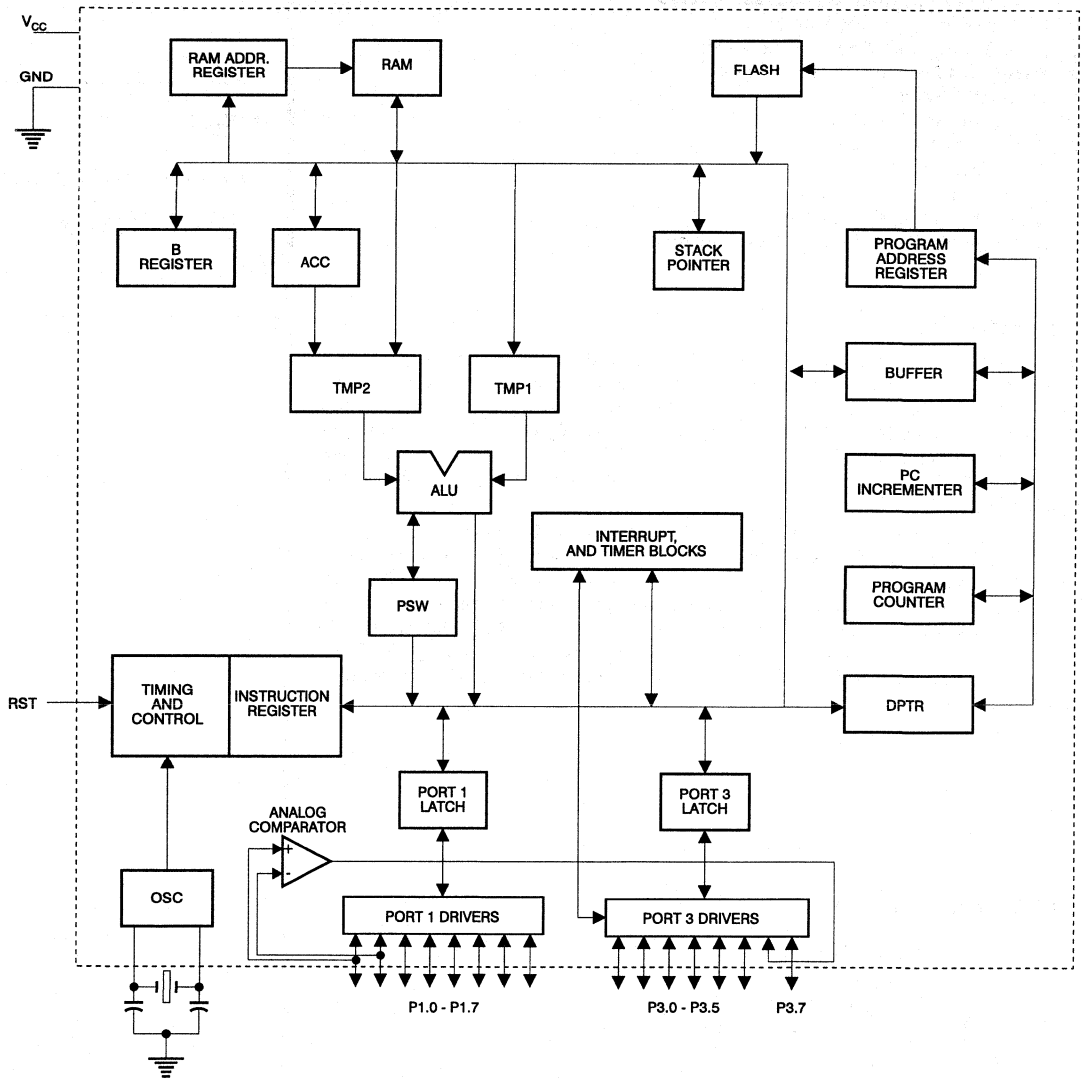
Pin Configuration

PDIP/SOIC



0366C

Block Diagram



Pin Description

Vcc

Supply voltage.

GND

Ground.

Port 1

Port 1 is an 8-bit bidirectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (IIL) because of the internal pullups.

Port 1 also receives code data during Flash programming and program verification.

Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bidirectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C1051 as listed below:

Port Pin	Alternate Functions
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)

Port 3 also receives some control signals for Flash programming and programming verification.

RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

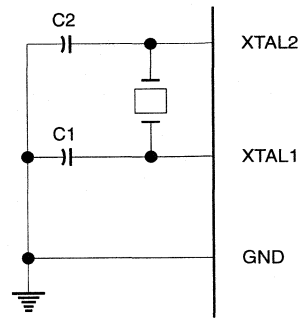
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

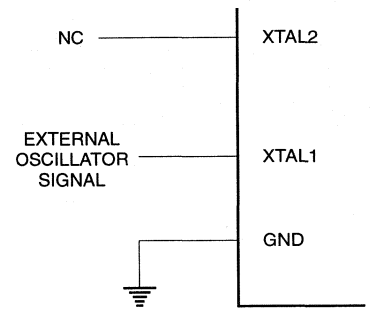
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in the table below.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Table 1. AT89C1051 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XXX00000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0XX00000								0AFH
0A0H									0A7H
98H									9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000		TH0 00000000				8FH
80H		SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000	87H

Restrictions on Certain Instructions

The AT89C1051 is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 1 Kbyte of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 1K for the AT89C1051. This should be the responsibility of the software programmer. For example, LJMP 3FEH would be a valid instruction for the AT89C1051 (with 1K of memory), whereas LJMP 410H would not.

1. Branching instructions:

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 3FFH for the 89C1051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related instructions, Data Memory:

The AT89C1051 contains 64 bytes of internal data memory. Thus, in the AT89C1051 the stack depth is limited to 64 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

Lock Bit Protection Modes[1]

Program Lock Bits			
	LB1	LB2	Protection Type
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

Programming The Flash

The AT89C1051 is shipped with the 1 Kbyte of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the entire memory array needs to be erased electrically.*

Internal Address Counter: The AT89C1051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

Programming Algorithm: To program the AT89C1051, the following sequence is recommended.

1. Power-up sequence:
   Apply power between V_{CC} and GND pins
   Set RST and XTAL1 to GND
   With all other pins floating, wait for greater than 10 milliseconds
2. Set pin RST to 'H'
   Set pin P3.2 to 'H'
3. Apply the appropriate combination of 'H' or 'L' logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.

To Program and Verify the Array:

4. Apply data for Code byte at location 000H to P1.0 to P1.7.
5. Raise RST to 12V to enable programming.
6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
7. To verify the programmed data, lower RST from 12V to logic 'H' level and set pins P3.3 to P3.7 to the appropiate levels. Output data can be read at the port P1 pins.
8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 1 Kbyte array or until the end of the object file is reached.
10. Power-off sequence:
    set XTAL1 to 'L'
    set RST to 'L'
    Float all other I/O pins
    Turn Vcc power off

Data Polling: The AT89C1051 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The Progress of byte programming can also be monitored by the RDY/BSY output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from 'L' to 'H'.
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire PEROM array (1 Kbyte) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(001H) = 11H indicates 89C1051

Programming Interface

3

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	P3.2/PROG	P3.3	P3.4	P3.5	P3.7
Write Code Data[1,3]		12V	⎍	L	H	H	H
Read Code Data[1]		H	H	L	L	H	H
Write Lock	Bit - 1	12V	⎍	H	H	H	H
	Bit - 2	12V	⎍	H	H	L	L
Chip Erase		12V	⎍[2]	H	L	L	L
Read Signature Byte		H	H	L	L	L	L

Notes: 1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL1 pin.
2. Chip Erase requires a 10 ms PROG pulse.
3. P3.1 is pulled Low during programming to indicate RDY/BSY.

Figure 3. Programming the Flash Memory

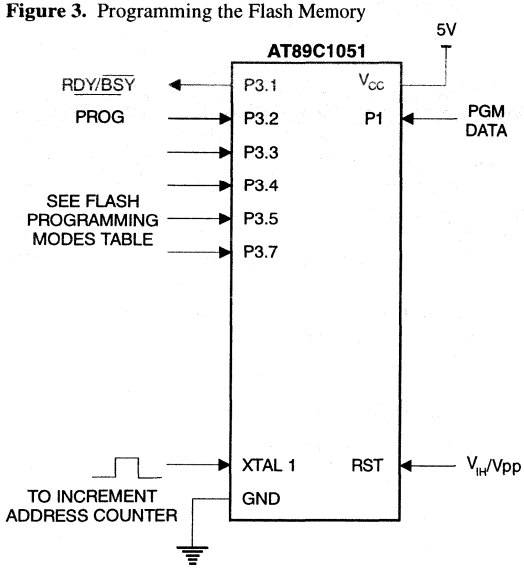
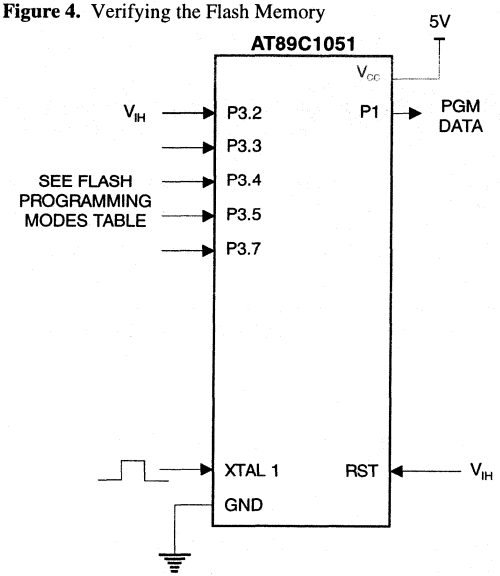


Figure 4. Verifying the Flash Memory

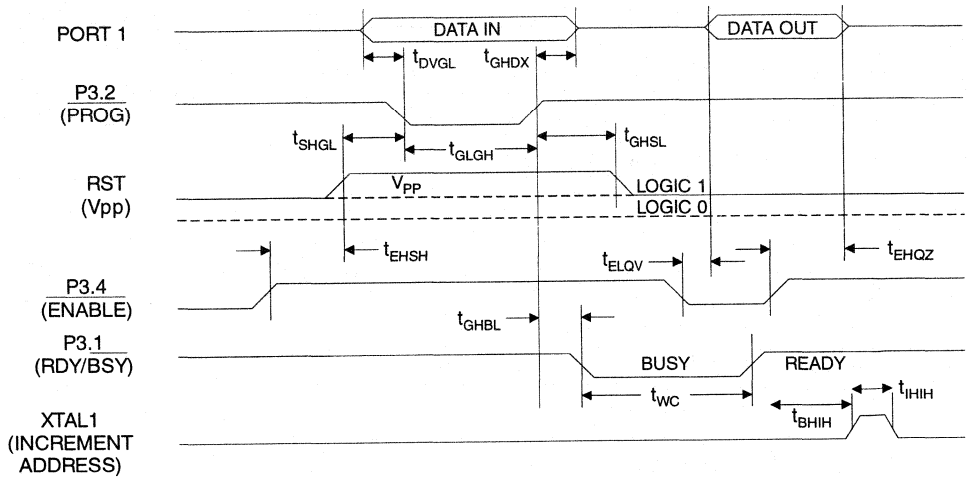


Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	µA
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	1.0		µs
t_{GHDX}	Data Hold After $\overline{PROG}$	1.0		µs
t_{EHSH}	P3.4 ($\overline{ENABLE}$) High to V_{PP}	1.0		µs
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		1.0	µs
t_{EHQZ}	Data Float After $\overline{ENABLE}$	0	1.0	µs
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHIH}	RDY/$\overline{BSY}$ to Increment Clock Delay	1.0		µs
t_{IHIL}	Increment Clock High	200		ns

Flash Programming and Verification Waveforms



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current	25.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

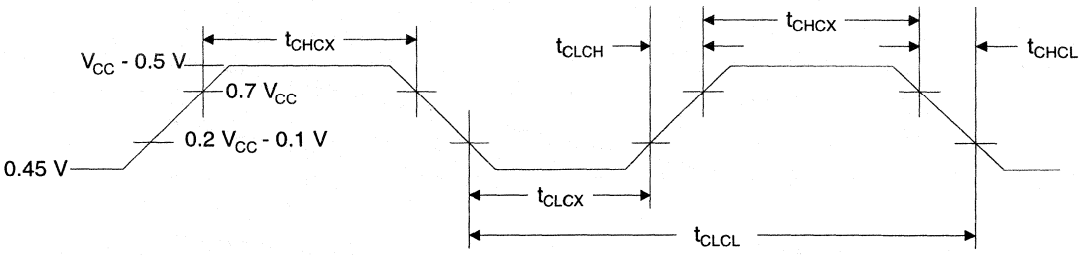
D.C. Characteristics

T_A = -40°C to 85°C, V_{CC} = 2.7 V to 6.0 V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage		-0.5	0.2 V_{CC}-0.1	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1, 3)	I_{OL} = 20 mA, V_{CC} = 5 V I_{OL} = 10 mA, V_{CC} = 2.7 V		0.50	V
V_{OH}	Output High Voltage (Ports 1, 3)	I_{OH} = -80 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -30 µA	0.75 V_{CC}		V
		I_{OH} = -12 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)	V_{IN} = 0.45 V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)	V_{IN} = 2 V		-750	µA
I_{LI}	Input Leakage Current (Port P1.0, P1.1)	0 < V_{IN} < V_{CC}		±10	µA
V_{OS}	Comparator Input Offset Voltage	V_{CC} = 5 V		20	mV
V_{CM}	Comparator Input Common Mode Voltage		0	V_{CC}	V
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz, V_{CC} = 6 V/3 V		15/5.5	mA
		Idle Mode, 12 MHz, V_{CC} = 6 V/3 V P1.0 & P1.1 = 0V or V_{CC}		5/1	mA
	Power Down Mode[2]	V_{CC} = 6 V P1.0 & P1.1 = 0V or V_{CC}		100	µA
		V_{CC} = 3 V P1.0 & P1.1 = 0V or V_{CC}		20	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin:20 mA
Maximum total IOL for all output pins:80 mA
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power Down is 2 V.

External Clock Drive Waveforms

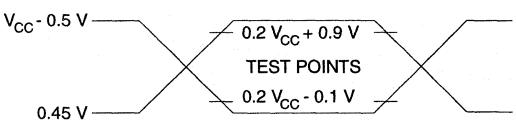


3

External Clock Drive

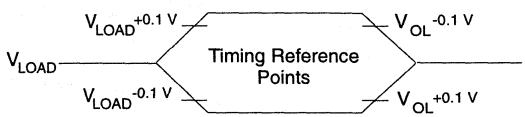
Symbol	Parameter	V_{CC} = 2.7 V to 6.0 V		V_{CC} = 4.0 V to 6.0 V		Units
		Min	Max	Min	Max	
1/t_{CLCL}	Oscillator Frequency	0	12	0	24	MHz
t_{CLCL}	Clock Period	83.3		41.6		ns
t_{CHCX}	High Time	30		15		ns
t_{CLCX}	Low Time	30		15		ns
t_{CLCH}	Rise Time		20		20	ns
t_{CHCL}	Fall Time		20		20	ns

AC Testing Input/Output Waveforms [1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic 1 and 0.45 V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

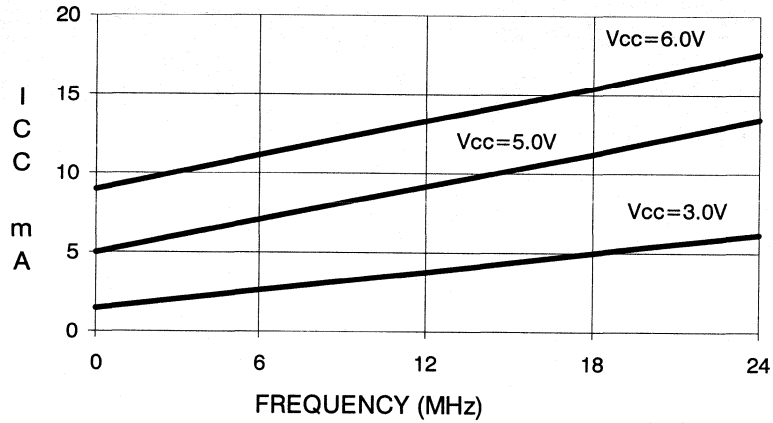
Float Waveforms [1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

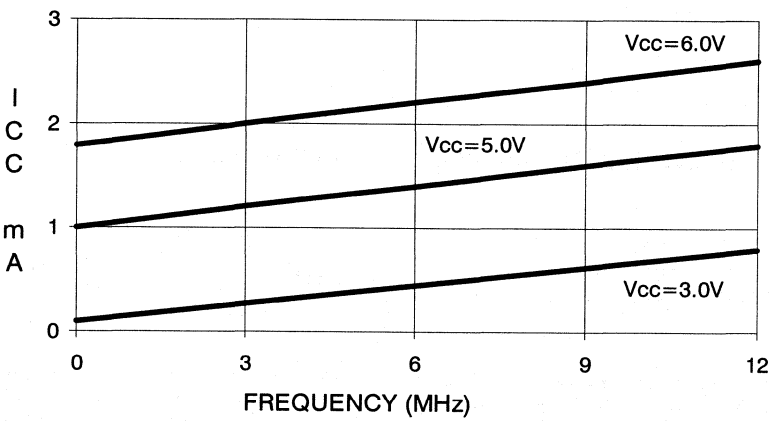
AT89C1051

TYPICAL ICC - ACTIVE (85°C)



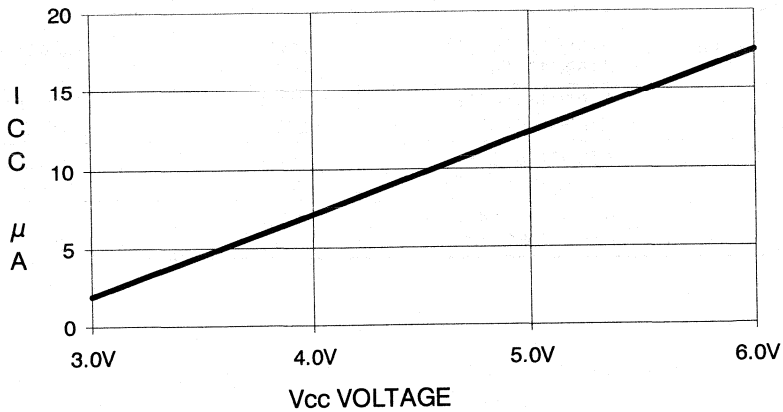
AT89C1051

TYPICAL ICC - IDLE (85°C)



AT89C1051

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



Note: 1. XTAL1 tied to GND for I_{CC} (power down).
2. P.1.0 and P1.1 = V_{CC} or GND.
3. Lock bits programmed.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7 V to 6.0 V	AT89C1051-12PC AT89C1051-12SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C1051-12PI AT89C1051-12SI	20P3 20S	Industrial (-40°C to 85°C)
24	4.0 V to 6.0 V	AT89C1051-24PC AT89C1051-24SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C1051-24PI AT89C1051-24SI	20P3 20S	Industrial (-40°C to 85°C)

Package Type	
20P3	20 Lead, 0.300" Wide, Plastic Dual Inline Package (PDIP)
20S	20 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

Features

- Compatible with MCS-51™ Products
- 2 Kbytes of Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 2.7 V to 6 V Operating Range
- Fully Static Operation: 0 Hz to 24 MHz
- Two-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 15 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial UART Channel
- Direct LED Drive Outputs
- On-Chip Analog Comparator
- Low Power Idle and Power Down Modes

8-Bit Microcontroller with 2 Kbytes Flash

3

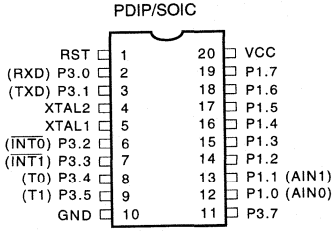
Description

The AT89C2051 is a low-voltage, high-performance CMOS 8-bit microcomputer with 2 Kbytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C2051 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C2051 provides the following standard features: 2 Kbytes of Flash, 128 bytes of RAM, 15 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, a precision analog comparator, on-chip oscillator and clock circuitry. In addition, the AT89C2051 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

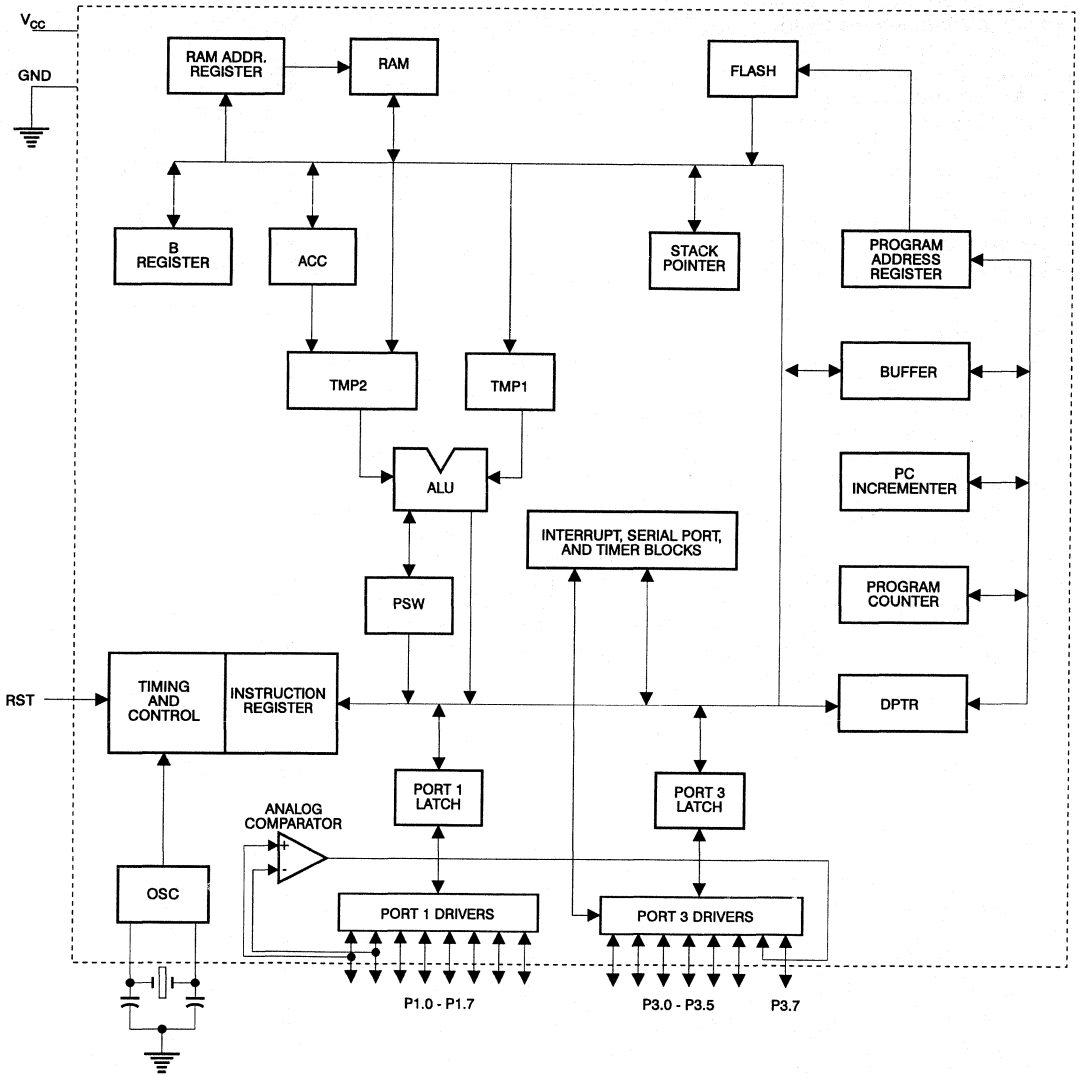
Pin Configuration

PDIP/SOIC



0368C

Block Diagram



Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 1

Port 1 is an 8-bit bidirectional I/O port. Port pins P1.2 to P1.7 provide internal pullups. P1.0 and P1.1 require external pullups. P1.0 and P1.1 also serve as the positive input (AIN0) and the negative input (AIN1), respectively, of the on-chip precision analog comparator. The Port 1 output buffers can sink 20 mA and can drive LED displays directly. When 1s are written to Port 1 pins, they can be used as inputs. When pins P1.2 to P1.7 are used as inputs and are externally pulled low, they will source current (IIL) because of the internal pullups.

Port 1 also receives code data during Flash programming and program verification.

Port 3

Port 3 pins P3.0 to P3.5, P3.7 are seven bidirectional I/O pins with internal pullups. P3.6 is hard-wired as an input to the output of the on-chip comparator and is not accessible as a general purpose I/O pin. The Port 3 output buffers can sink 20 mA. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C2051 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{\text{INT0}}$ (external interrupt 0)
P3.3	$\overline{\text{INT1}}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)

Port 3 also receives some control signals for Flash programming and programming verification.

RST

Reset input. All I/O pins are reset to 1s as soon as RST goes high. Holding the RST pin high for two machine cycles while the oscillator is running resets the device.

Each machine cycle takes 12 oscillator or clock cycles.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

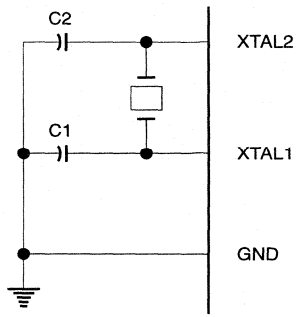
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

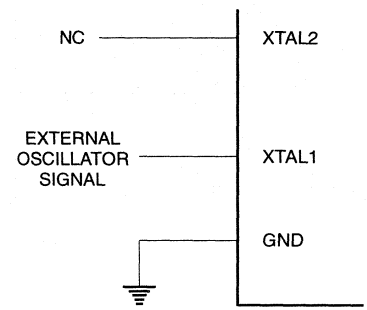
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in the table below.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Table 1. AT89C2051 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H									0CFH
0C0H									0C7H
0B8H	IP XXX00000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0XX00000								0AFH
0A0H									0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H		SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000	87H

Restrictions on Certain Instructions

The AT89C2051 and is an economical and cost-effective member of Atmel's growing family of microcontrollers. It contains 2 Kbytes of flash program memory. It is fully compatible with the MCS-51 architecture, and can be programmed using the MCS-51 instruction set. However, there are a few considerations one must keep in mind when utilizing certain instructions to program this device.

All the instructions related to jumping or branching should be restricted such that the destination address falls within the physical program memory space of the device, which is 2K for the AT89C2051. This should be the responsibility of the software programmer. For example, LJMP 7E0H would be a valid instruction for the AT89C2051 (with 2K of memory), whereas LJMP 900H would not.

1. Branching instructions:

LCALL, LJMP, ACALL, AJMP, SJMP, JMP @A+DPTR

These unconditional branching instructions will execute correctly as long as the programmer keeps in mind that the destination branching address must fall within the physical boundaries of the program memory size (locations 00H to 7FFH for the 89C2051). Violating the physical space limits may cause unknown program behavior.

CJNE [...], DJNZ [...], JB, JNB, JC, JNC, JBC, JZ, JNZ With these conditional branching instructions the same rule above applies. Again, violating the memory boundaries may cause erratic execution.

For applications involving interrupts the normal interrupt service routine address locations of the 80C51 family architecture have been preserved.

2. MOVX-related instructions, Data Memory:

The AT89C2051 contains 128 bytes of internal data memory. Thus, in the AT89C2051 the stack depth is limited to 128 bytes, the amount of available RAM. External DATA memory access is not supported in this device, nor is external PROGRAM memory execution. Therefore, no MOVX [...] instructions should be included in the program.

A typical 80C51 assembler will still assemble instructions, even if they are written in violation of the restrictions mentioned above. It is the responsibility of the controller user to know the physical features and limitations of the device being used and adjust the instructions used correspondingly.

Program Memory Lock Bits

On the chip are two lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

Lock Bit Protection Modes[1]

Program Lock Bits			
	LB1	LB2	Protection Type
1	U	U	No program lock features.
2	P	U	Further programming of the Flash is disabled.
3	P	P	Same as mode 2, also verify is disabled.

Note: 1. The Lock Bits can only be erased with the Chip Erase operation

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

P1.0 and P1.1 should be set to '0' if no external pullups are used, or set to '1' if external pullups are used.

Programming The Flash

The AT89C2051 is shipped with the 2 Kbytes of on-chip PEROM code memory array in the erased state (i.e., contents = FFH) and ready to be programmed. The code memory array is programmed one byte at a time. *Once the array is programmed, to re-program any non-blank byte, the entire memory array needs to be erased electrically.*

Internal Address Counter: The AT89C2051 contains an internal PEROM address counter which is always reset to 000H on the rising edge of RST and is advanced by applying a positive going pulse to pin XTAL1.

Programming Algorithm: To program the AT89C2051, the following sequence is recommended.

1. Power-up sequence:
   Apply power between V_{CC} and GND pins
   Set RST and XTAL1 to GND
   With all other pins floating, wait for greater than 10 milliseconds
2. Set pin RST to 'H'
   Set pin P3.2 to 'H'
3. Apply the appropriate combination of 'H' or 'L' logic levels to pins P3.3, P3.4, P3.5, P3.7 to select one of the programming operations shown in the PEROM Programming Modes table.

To Program and Verify the Array:

4. Apply data for Code byte at location 000H to P1.0 to P1.7.
5. Raise RST to 12V to enable programming.
6. Pulse P3.2 once to program a byte in the PEROM array or the lock bits. The byte-write cycle is self-timed and typically takes 1.2 ms.
7. To verify the programmed data, lower RST from 12V to logic 'H' level and set pins P3.3 to P3.7 to the appropiate levels. Output data can be read at the port P1 pins.
8. To program a byte at the next address location, pulse XTAL1 pin once to advance the internal address counter. Apply new data to the port P1 pins.
9. Repeat steps 5 through 8, changing data and advancing the address counter for the entire 2 Kbytes array or until the end of the object file is reached.
10. Power-off sequence:
    set XTAL1 to 'L'
    set RST to 'L'
    Float all other I/O pins
    Turn Vcc power off

Data Polling: The AT89C2051 features Data Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on P1.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. Data Polling may begin any time after a write cycle has been initiated.

Ready/Busy: The Progress of byte programming can also be monitored by the RDY/BSY output signal. Pin P3.1 is pulled low after P3.2 goes High during programming to indicate BUSY. P3.1 is pulled High again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed code data can be read back via the data lines for verification:

1. Reset the internal address counter to 000H by bringing RST from 'L' to 'H'.
2. Apply the appropriate control signals for Read Code data and read the output data at the port P1 pins.
3. Pulse pin XTAL1 once to advance the internal address counter.
4. Read the next code data byte at the port P1 pins.
5. Repeat steps 3 and 4 until the entire array is read.

The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire PEROM array (2 Kbytes) and the two Lock Bits are erased electrically by using the proper combination of control signals and by holding P3.2 low for 10 ms. The code array is written with all "1"s in the Chip Erase operation and must be executed before any non-blank memory byte can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 000H, 001H, and 002H, except that P3.5 and P3.7 must be pulled to a logic low. The values returned are as follows.

(000H) = 1EH indicates manufactured by Atmel
(001H) = 21H indicates 89C2051

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

3

Flash Programming Modes

Mode		RST	P3.2/ PROG	P3.3	P3.4	P3.5	P3.7
Write Code Data[1,3]		12V	⎍	L	H	H	H
Read Code Data[1]		H	H	L	L	H	H
Write Lock	Bit - 1	12V	⎍	H	H	H	H
	Bit - 2	12V	⎍	H	H	L	L
Chip Erase		12V	⎍[2]	H	L	L	L
Read Signature Byte		H	H	L	L	L	L

Notes: 1. The internal PEROM address counter is reset to 000H on the rising edge of RST and is advanced by a positive pulse at XTAL1 pin.
2. Chip Erase requires a 10 ms PROG pulse.
3. P3.1 is pulled Low during programming to indicate RDY/BSY\.

Figure 3. Programming the Flash Memory

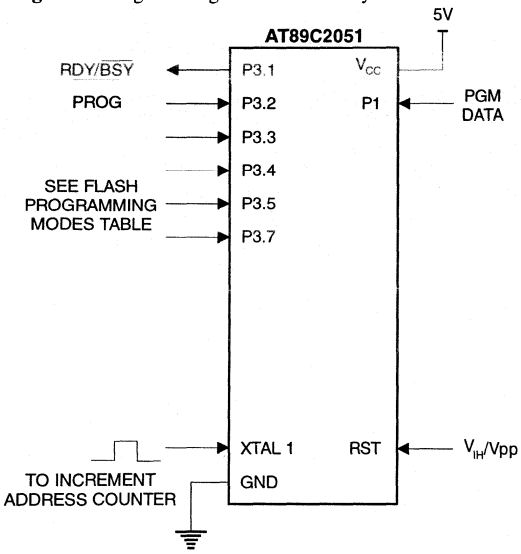
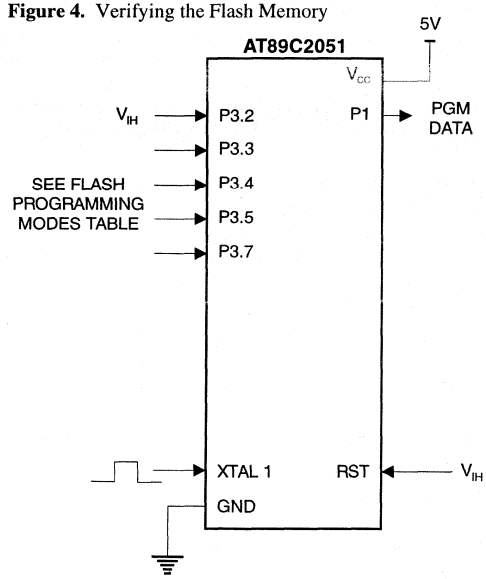


Figure 4. Verifying the Flash Memory

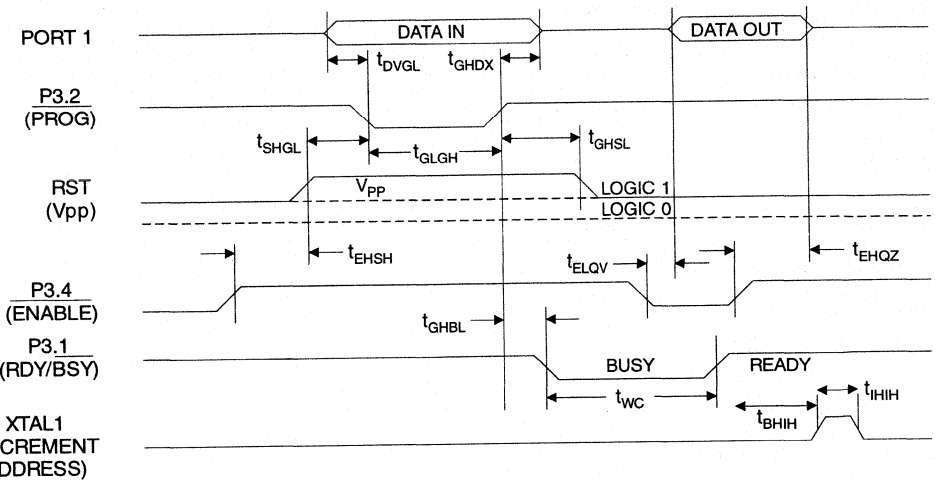


Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}	Programming Enable Voltage	11.5	12.5	V
I_{PP}	Programming Enable Current		250	µA
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	1.0		µs
t_{GHDX}	Data Hold After $\overline{PROG}$	1.0		µs
t_{EHSH}	P3.4 ($\overline{ENABLE}$) High to V_{PP}	1.0		µs
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		1.0	µs
t_{EHQZ}	Data Float After $\overline{ENABLE}$	0	1.0	µs
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		50	ns
t_{WC}	Byte Write Cycle Time		2.0	ms
t_{BHIH}	RDY/$\overline{BSY}$\ to Increment Clock Delay	1.0		µs
t_{IHIL}	Increment Clock High	200		ns

Flash Programming and Verification Waveforms



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current	25.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

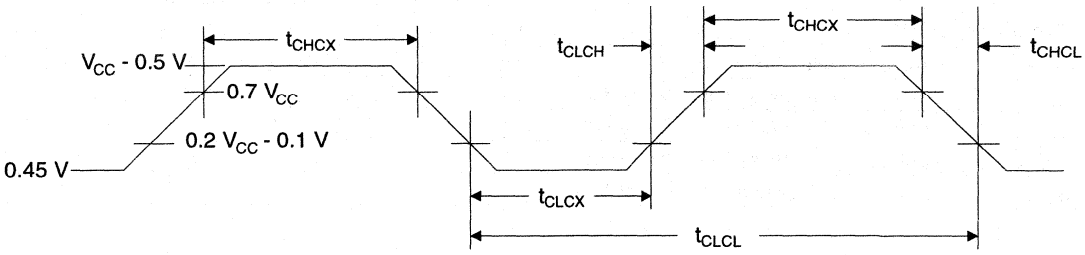
D.C. Characteristics

T_A = -40°C to 85°C, V_{CC} = 2.7 V to 6.0 V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage		-0.5	0.2 V_{CC}-0.1	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1, 3)	I_{OL} = 20 mA, V_{CC} = 5 V I_{OL} = 10 mA, V_{CC} = 2.7 V		0.5	V
V_{OH}	Output High Voltage (Ports 1, 3)	I_{OH} = -80 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -30 µA	0.75 V_{CC}		V
		I_{OH} = -12 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1, 2, 3)	V_{IN} = 0.45 V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1, 2, 3)	V_{IN} = 2 V		-750	µA
I_{LI}	Input Leakage Current (Port P1.0, P1.1)	0 < V_{IN} < V_{CC}		±10	µA
V_{OS}	Comparator Input Offset Voltage	V_{CC} = 5 V		20	mV
V_{CM}	Comparator Input Common Mode Voltage		0	V_{CC}	V
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz, V_{CC} = 6 V/3 V		15/5.5	mA
		Idle Mode, 12 MHz, V_{CC} = 6 V/3 V P1.0 & P1.1 = 0V or V_{CC}		5/1	mA
	Power Down Mode[2]	V_{CC} = 6 V P1.0 & P1.1 = 0V or V_{CC}		100	µA
		V_{CC} = 3 V P1.0 & P1.1 = 0V or V_{CC}		20	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin:20 mA
Maximum total IOL for all output pins:80 mA
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power Down is 2 V.

External Clock Drive Waveforms



3

External Clock Drive

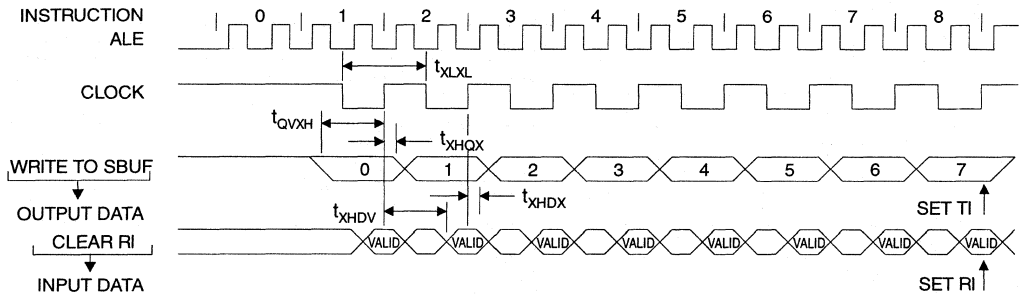
Symbol	Parameter	V_{CC} = 2.7 V to 6.0 V		V_{CC} = 4.0 V to 6.0 V		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency	0	12	0	24	MHz
t_{CLCL}	Clock Period	83.3		41.6		ns
t_{CHCX}	High Time	30		15		ns
t_{CLCX}	Low Time	30		15		ns
t_{CLCH}	Rise Time		20		20	ns
t_{CHCL}	Fall Time		20		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

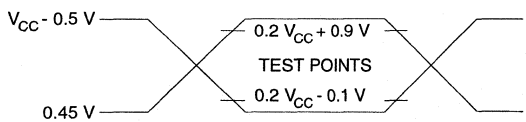
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-33		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		10tCLCL-133	ns

Shift Register Mode Timing Waveforms

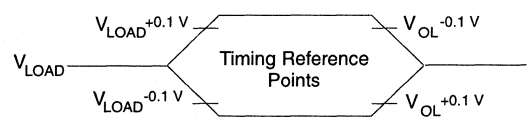


AC Testing Input/Output Waveforms [1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic 1 and 0.45 V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

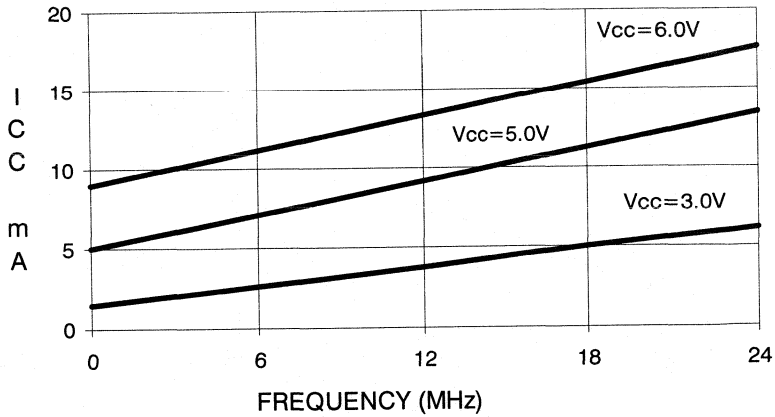
Float Waveforms [1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

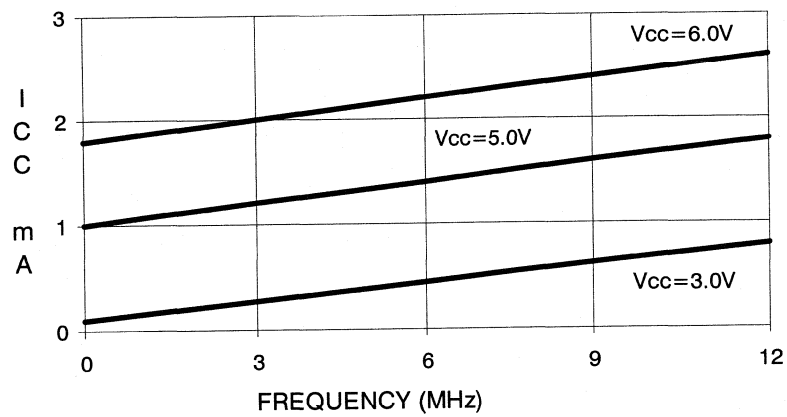
AT89C2051

TYPICAL ICC - ACTIVE (85°C)



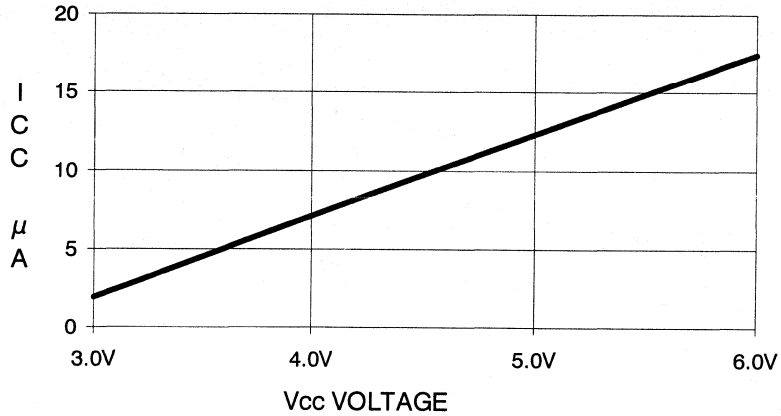
AT89C2051

TYPICAL ICC - IDLE (85°C)



AT89C2051

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



Note: 1. XTAL1 tied to GND for I_{CC} (power down).
2. P.1.0 and P1.1 = V_{CC} or GND.
3. Lock bits programmed.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7 V to 6.0 V	AT89C2051-12PC AT89C2051-12SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-12PI AT89C2051-12SI	20P3 20S	Industrial (-40°C to 85°C)
24	4.0 V to 6.0 V	AT89C2051-24PC AT89C2051-24SC	20P3 20S	Commercial (0°C to 70°C)
		AT89C2051-24PI AT89C2051-24SI	20P3 20S	Industrial (-40°C to 85°C)

Package Type	
20P3	20 Lead, 0.300" Wide, Plastic Dual In-line Package (PDIP)
20S	20 Lead, 0.300" Wide, Plastic Gull Wing Small Outline (SOIC)

Features

- Compatible with MCS-51™ Products
- 4 Kbytes of In-System Reprogrammable Flash Memory
  Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

8-Bit Microcontroller with 4 Kbytes Flash

3

Description

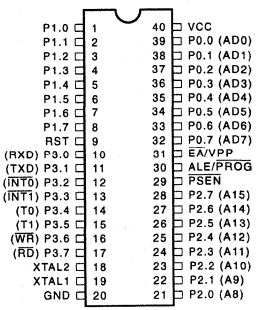
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4 Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C51 provides the following standard features: 4 Kbytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is

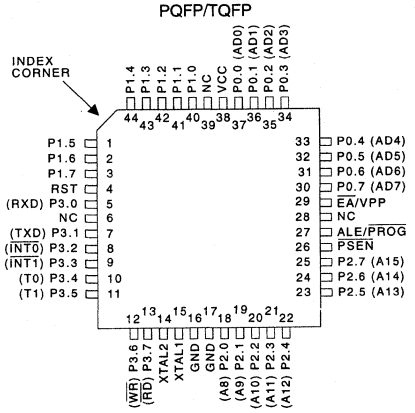
(continued)

Pin Configurations

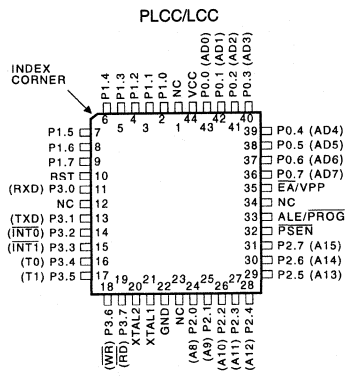
PDIP/Cerdip



PQFP/TQFP

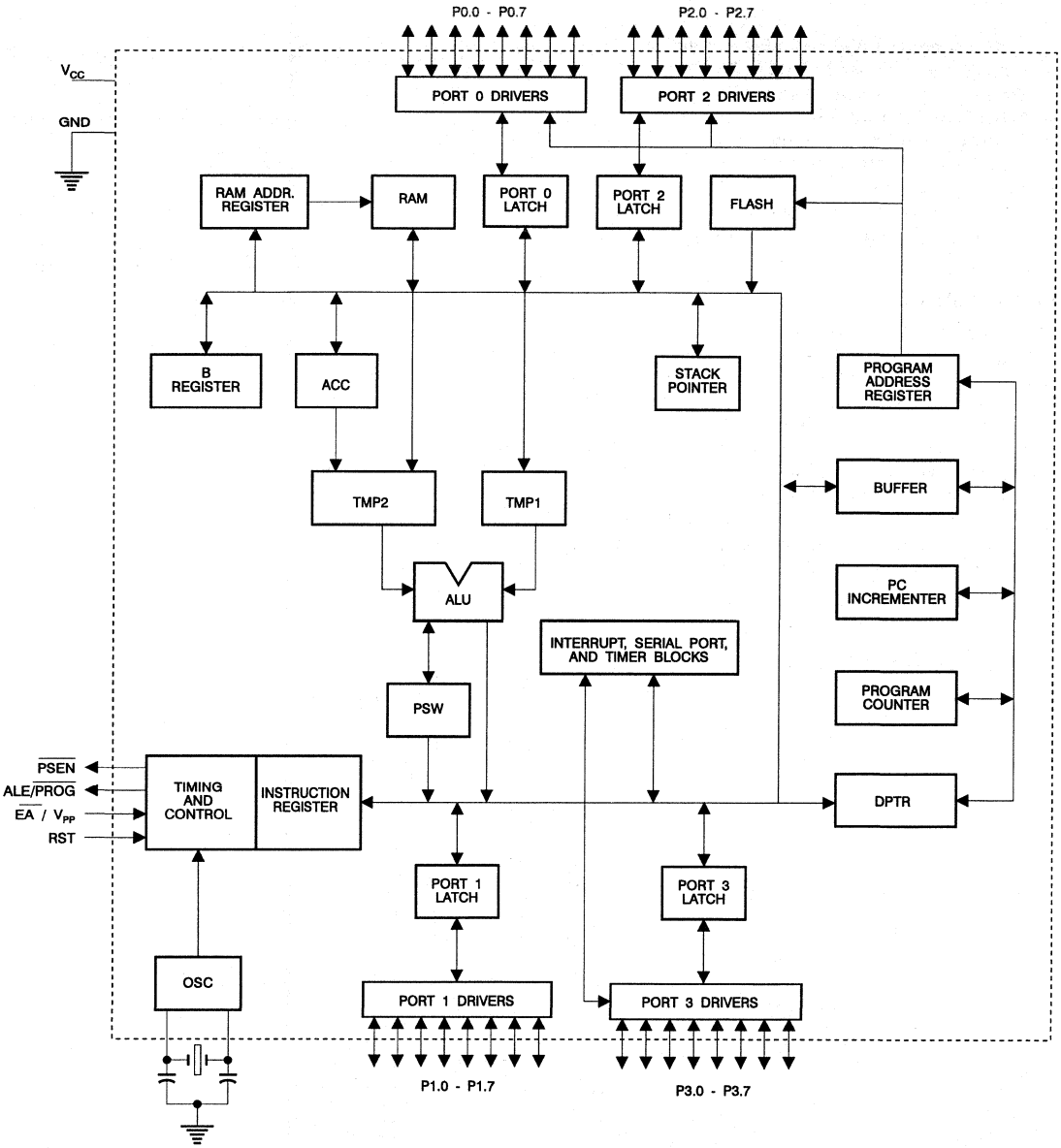


PLCC/LCC



0265E

Block Diagram



Description (Continued)

designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (extenal interrupt 0)
P3.3	$\overline{INT1}$ (extenal interrupt 1)
P3.4	T0 (timer 0 extenal input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (extenal data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and programming verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcrontroller is in external execution mode.

$\overline{PSEN}$

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

$\overline{EA}$/V_{PP}

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require 12-volt V_{PP}.

(continued)

Pin Description (Continued)

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

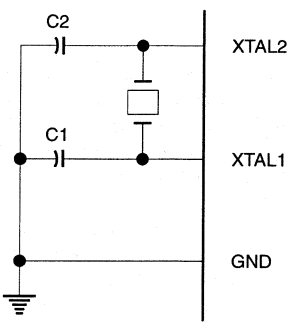
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Power Down Mode

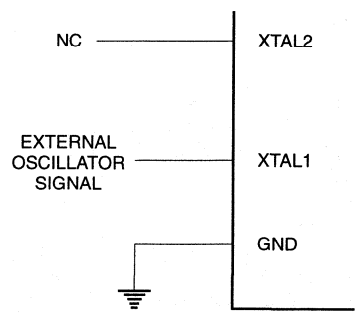
In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Figure 1. Oscillator Connections



Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 2. External Clock Drive Configuration



Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

3

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12 V	V_{PP} = 5 V
Top-Side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=51H (032H)=FFH	(030H)=1EH (031H)=51H (032H)=05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}$/V_{PP} to 12 V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H,

031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12 V programming
(032H) = 05H indicates 5 V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L	⎍	H/12V(1)	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L	⎍	H/12V	H	H	H	H
	Bit - 2	H	L	⎍ (2)	H/12V	H	H	L	L
	Bit - 3	H	L	⎍	H/12V	H	L	H	L
Chip Erase		H	L	⎍	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Notes: 1. The signature byte at location 032H designates whether V_{PP} = 12 V or V_{PP} = 5 V should be used to enable programming.
2. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.

Figure 3. Programming the Flash

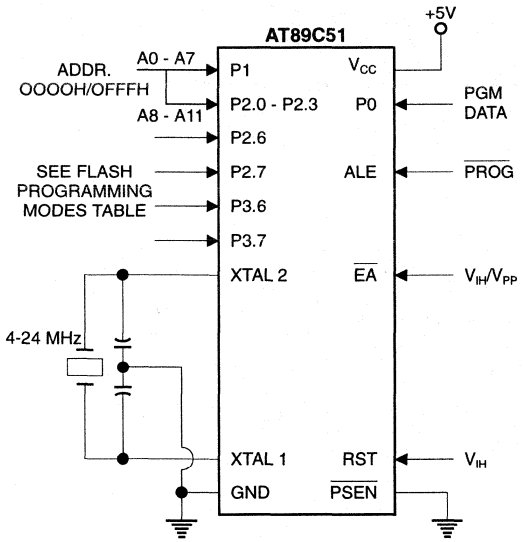
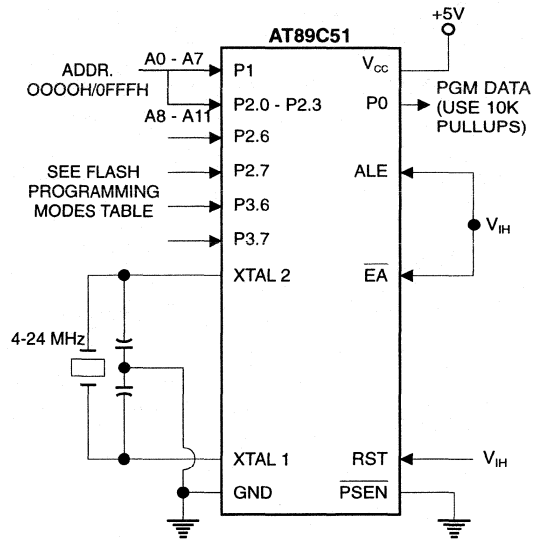


Figure 4. Verifying the Flash



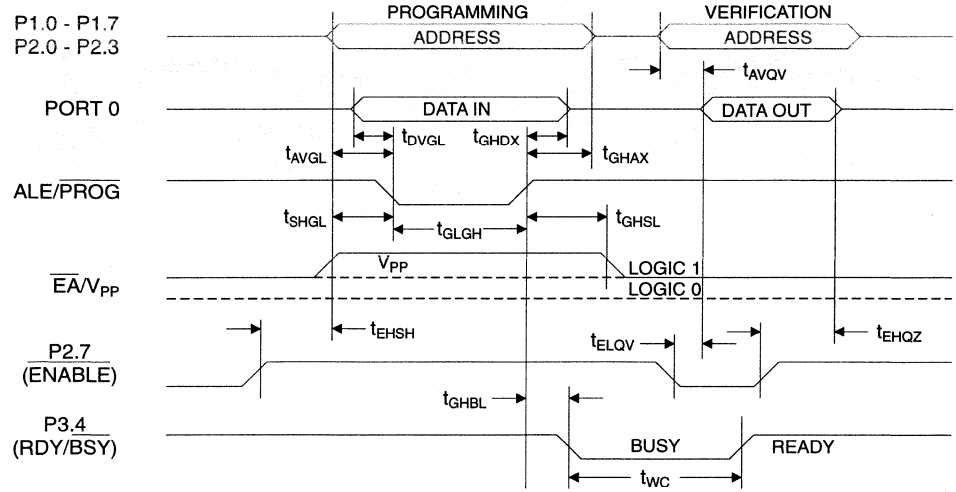
Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, V_{CC} = 5.0 ± 10%

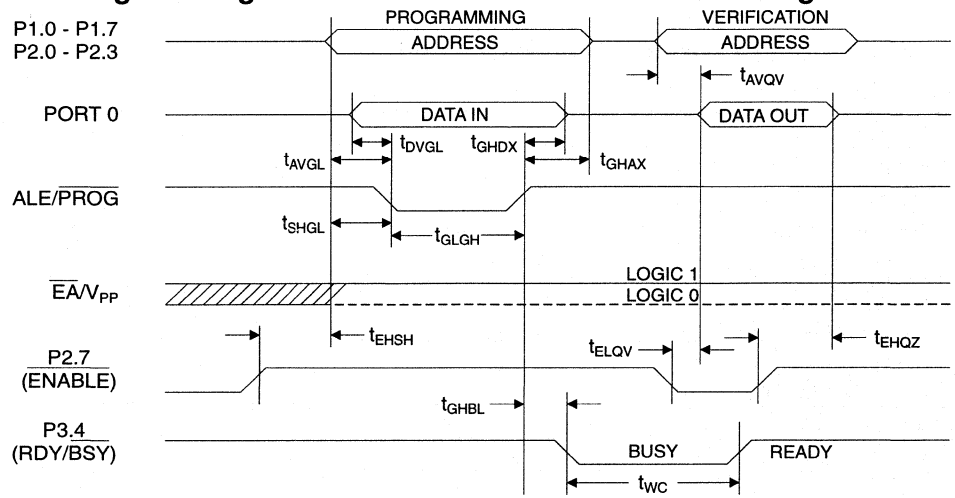
Symbol	Parameter	Min	Max	Units
V_{PP}(1)	Programming Enable Voltage	11.5	12.5	V
I_{PP}(1)	Programming Enable Current		1.0	mA
1/t_{CLCL}	Oscillator Frequency	4	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	48t_{CLCL}		
t_{GHAX}	Address Hold After $\overline{PROG}$	48t_{CLCL}		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	48t_{CLCL}		
t_{GHDX}	Data Hold After $\overline{PROG}$	48t_{CLCL}		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	48tCLCL		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}(1)	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		48tCLCL	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		48t_{CLCL}	
t_{EHQV}	Data Float After $\overline{ENABLE}$	0	48t_{CLCL}	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

Flash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0 V ± 20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage(1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage(1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45 V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2 V		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power Down Mode(2)	V_{CC} = 6 V		100	µA
		V_{CC} = 3 V		40	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin:10 mA
Maximum I_{OL} per 8-bit port:
Port 0:26 mA
Ports 1,2, 3:15 mA
Maximum total IOL for all output pins:71 mA
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum VCC for Power Down is 2 V.

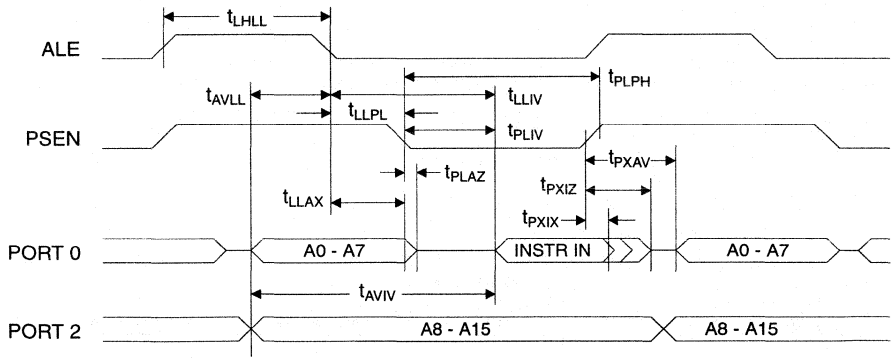
A.C. Characteristics

(Under Operating Conditions; Load Capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; Load Capacitance for all other outputs = 80 pF)

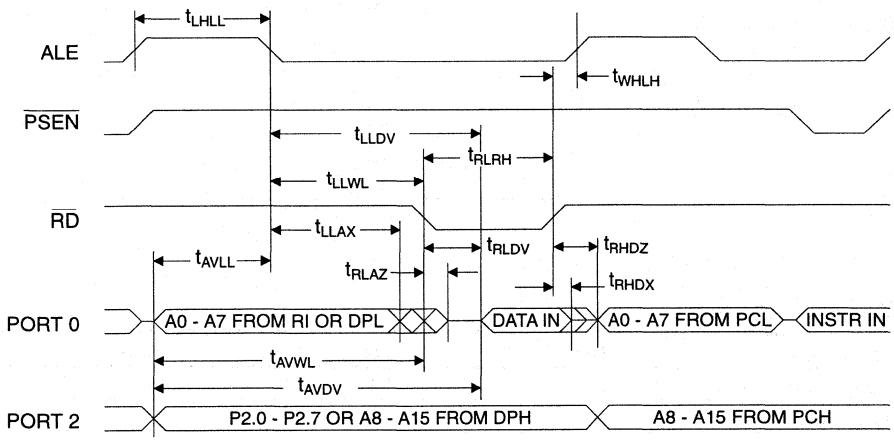
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	28		tCLCL-13		ns
t_{LLAX}	Address Hold After ALE Low	48		tCLCL-20		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		4tCLCL-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-13		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-20		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-45	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-10	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-55	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		8tCLCL-150	ns
t_{AVDV}	Address to Valid Data In		585		9tCLCL-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-20		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-120		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-20		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-20	t_{CLCL}+25	ns

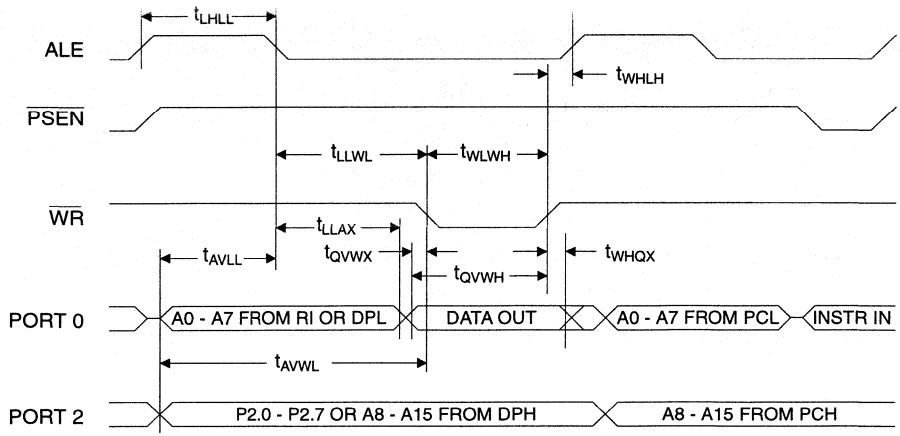
External Program Memory Read Cycle



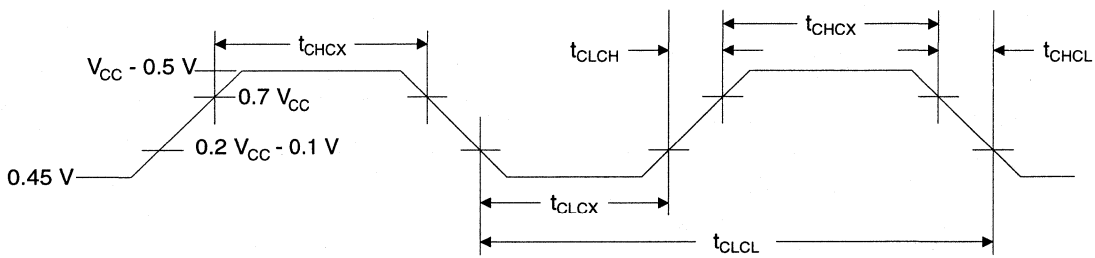
External Data Memory Read Cycle



External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

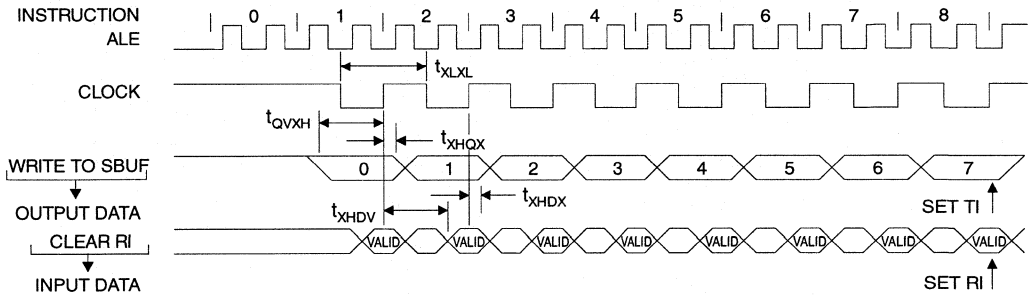
Symbol	Parameter	Min	Max	Units
1/t_{CLCL}	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

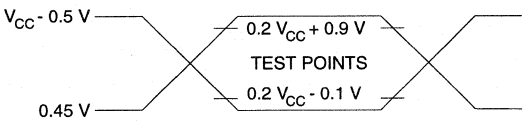
(V_{CC} = 5.0 V ± 20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-33		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		10tCLCL-133	ns

Shift Register Mode Timing Waveforms

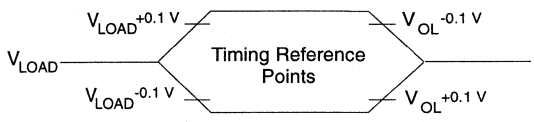


AC Testing Input/Output Waveforms [1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic 1 and 0.45 V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms [1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5 V ± 20%	AT89C51-12AC AT89C51-12JC AT89C51-12PC AT89C51-12QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C51-12AI AT89C51-12JI AT89C51-12PI AT89C51-12QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
		AT89C51-12AA AT89C51-12JA AT89C51-12PA AT89C51-12QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
	5 V ± 10%	AT89C51-12DM AT89C51-12LM	40D6 44L	Military (-55°C to 125°C)
		AT89C51-12DM/883 AT89C51-12LM/883	40D6 44L	Military/883C Class B, Fully Compliant (-55°C to 125°C)
16	5 V ± 20%	AT89C51-16AC AT89C51-16JC AT89C51-16PC AT89C51-16QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C51-16AI AT89C51-16JI AT89C51-16PI AT89C51-16QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
		AT89C51-16AA AT89C51-16JA AT89C51-16PA AT89C51-16QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
20	5 V ± 20%	AT89C51-20AC AT89C51-20JC AT89C51-20PC AT89C51-20QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C51-20AI AT89C51-20JI AT89C51-20PI AT89C51-20QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
24	5 V ± 20%	AT89C51-24AC AT89C51-24JC AT89C51-24PC AT89C51-24QC	44A 44J 44P6 44Q	Commercial (0°C to 70°C)
		AT89C51-24AI AT89C51-24JI AT89C51-24PI AT89C51-24QI	44A 44J 44P6 44Q	Industrial (-40°C to 85°C)

Ordering Information

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
40D6	40 Lead, 0.600" Wide, Non-Windowed, Ceramic Dual Inline Package (Cerdip)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
44L	44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

Features

- Compatible with MCS-51™ Products
- 4 Kbytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- 2.7 V to 6 V Operating Range
- Fully Static Operation: 0 Hz to 12 MHz
- Three-Level Program Memory Lock
- 128 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-Bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

8-Bit Microcontroller with 4 Kbytes Flash

3

Description

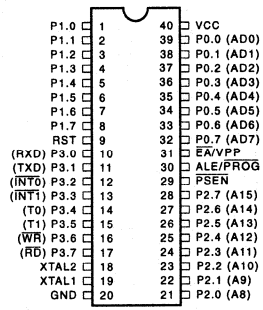
The AT89LV51 is a low-voltage, high-performance CMOS 8-bit microcomputer with 4 Kbytes of Flash Programmable and Erasable Read Only Memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard MCS-51™ instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89LV51 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89LV51 provides the following standard features: 4 Kbytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89LV51 is

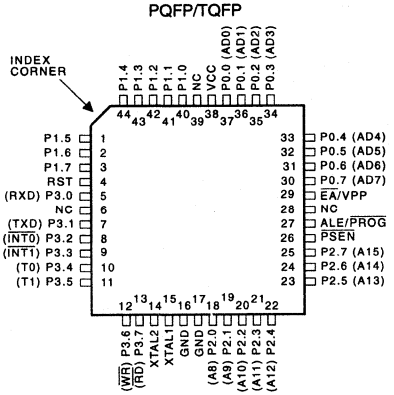
(continued)

Pin Configurations

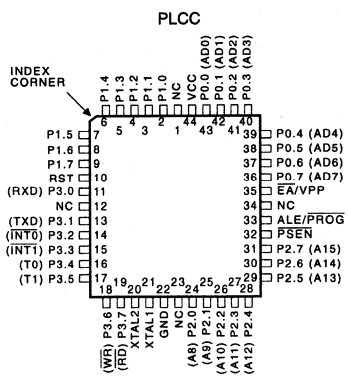
PDIP



PQFP/TQFP

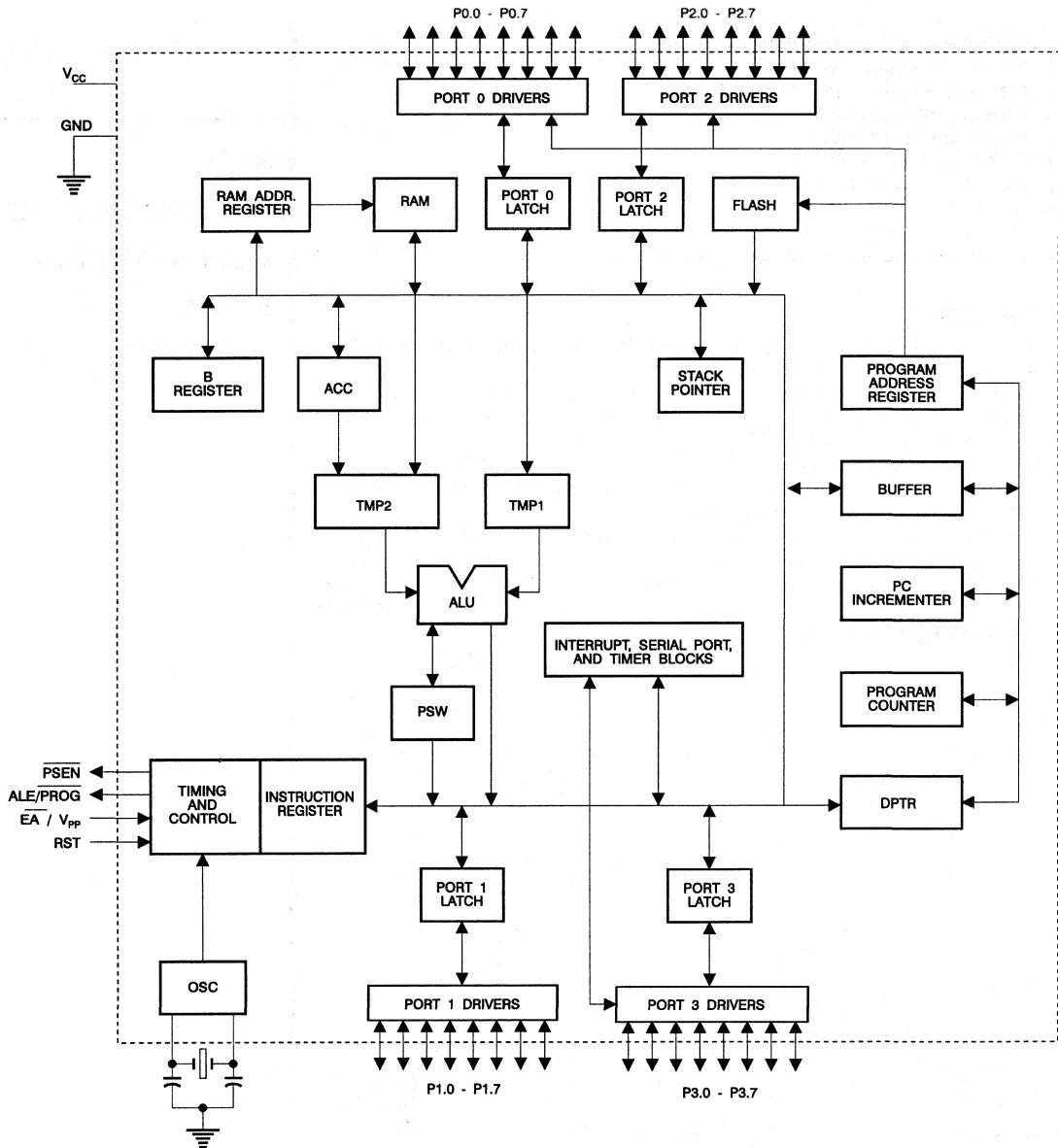


PLCC



0303C

Block Diagram



Description (Continued)

designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89LV51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (extenal interrupt 0)
P3.3	$\overline{INT1}$ (extenal interrupt 1)
P3.4	T0 (timer 0 extenal input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (extenal data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and programming verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external Data Memory.

$\overline{PSEN}$

Program Store Enable is the read strobe to external program memory.

When the AT89LV51 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

$\overline{EA}$/V_{PP}

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, when 12-volt programming is selected.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

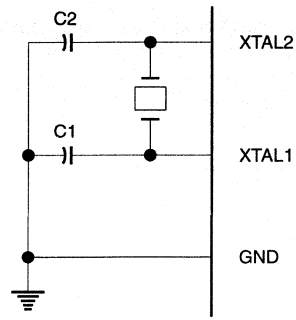
XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 1. Oscillator Connections



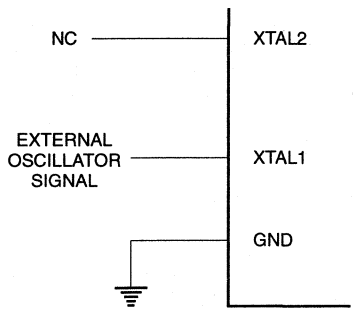
Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hardware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 2. External Clock Drive Configuration



Power Down Mode

In the power down mode the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below:

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes[1]

	Program Lock Bits			
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled.
3	P	P	U	Same as mode 2, also verify is disabled.
4	P	P	P	Same as mode 3, also external execution is disabled.

Note: 1. The lock bits can only be erased with the chip erase operation.

3

Programming the Flash

The AT89LV51 is normally shipped with the on-chip Flash memory array in the erased state (i.e. contents=FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (5-volt) program enable signal. The low voltage programming mode provides a convenient way to program the AT89LV51 inside the user's system while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89LV51 is shipped with either the High-Voltage or Low-Voltage programming mode enabled. The respective top-side marking and device signature codes are listed below:

	V_{PP} = 12 V	V_{PP} = 5 V
Top-Side Mark	AT89LV51 xxxx yyww	AT89LV51 xxxx-5 yyww
Signature	(030H)=1EH (031H)=61H (032H)=FFH	(030H)=1EH (031H)=61H (032H)=05H

The AT89LV51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip PEROM Code Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89LV51, the address, data and control signals should be set up according to the Flash programming mode table and Figures 3 and 4. To program the AT89LV51, the following sequence should be followed:

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}$/V_{PP} to 12-V if in the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5 changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89LV51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on PO.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array and the lock bits are erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H and 031H, except that P3.6 and P3.7 need to be pulled to a logic low. The values returned are:

(030H) = 1EH indicates manufactured by Atmel

(031H) = 61H indicates 89LV51

(032H) = FFH (High-Voltage) or 05H (Low-Voltage) programming mode

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L	⎍	H/12V(1)	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L	⎍	H/12V	H	H	H	H
	Bit - 2	H	L	⎍ (2)	H/12V	H	H	L	L
	Bit - 3	H	L	⎍	H/12V	H	L	H	L
Chip Erase		H	L	⎍	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Notes: 1. The signature byte at location 032H designates whether V_{PP} = 12 V or V_{PP} = 5 V should be used to enable programming.
2. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.

Figure 3. Programming the Flash

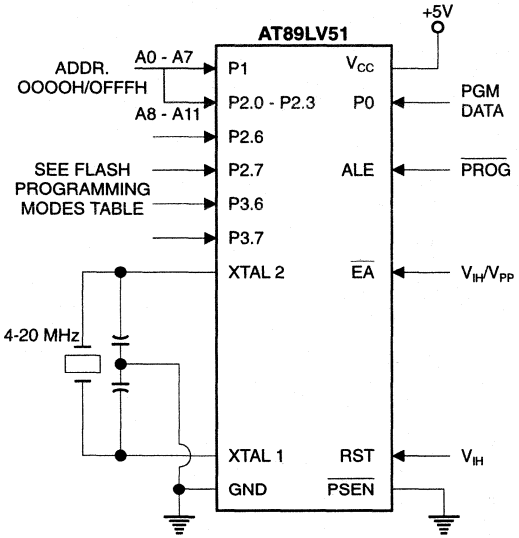
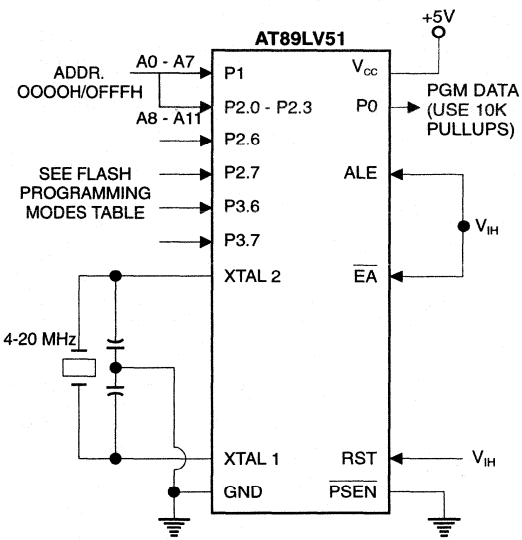


Figure 4. Verifying the Flash



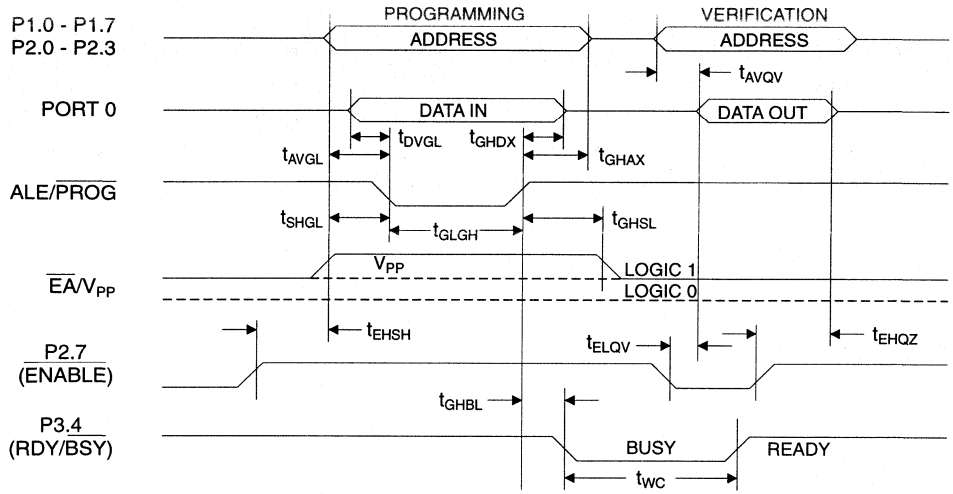
Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, V_{CC} = 5.0 ± 10%

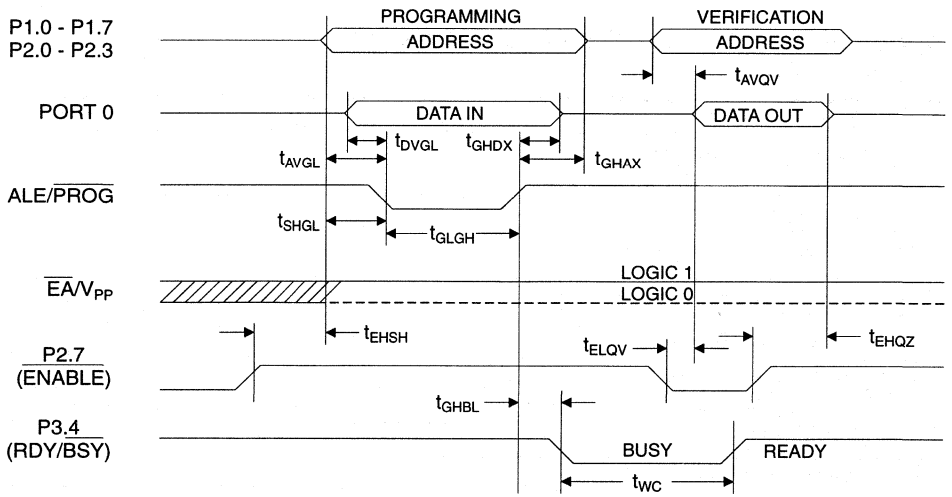
Symbol	Parameter	Min	Max	Units
V_{PP}(1)	Programming Enable Voltage	11.5	12.5	V
I_{PP}(1)	Programming Enable Current		25	µA
$1/t_{CLCL}$	Oscillator Frequency	4	12	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	48tCLCL		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}(1)	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		48tCLCL	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQV}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

Flash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

T_A = -40°C to 85°C, V_{CC} = 2.7 V to 6.0 V (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage(1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage(1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -20 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45 V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2 V		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz, V_{CC} = 6 V/3 V		20/5.5	mA
		Idle Mode, 12 MHz, V_{CC} = 6 V/3 V		5/1	mA
	Power Down Mode(2)	V_{CC} = 6 V		100	µA
		V_{CC} = 3 V		20	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin:10 mA
Maximum I_{OL} per 8-bit port:
Port 0:26 mA
Ports 1,2, 3:15 mA
Maximum total IOL for all output pins:71 mA
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum VCC for Power Down is 2 V.

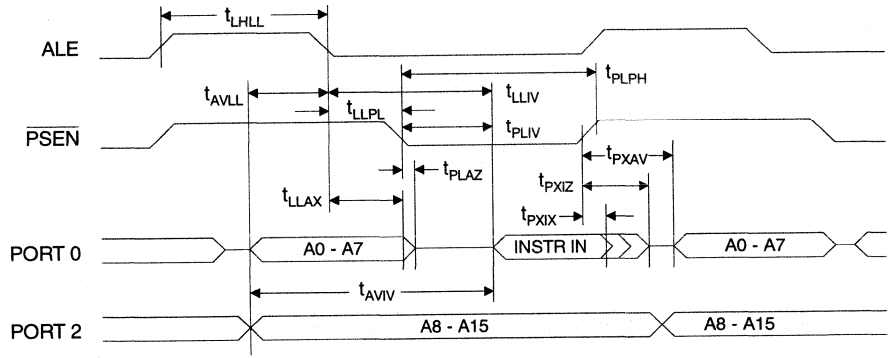
A.C. Characteristics

Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

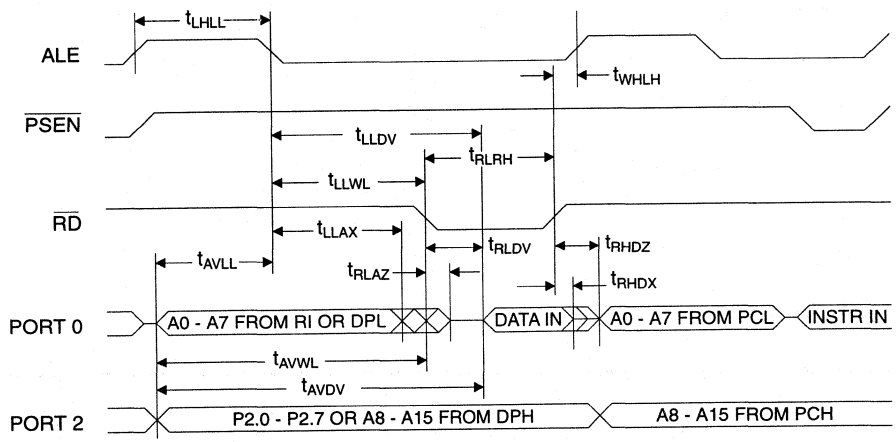
Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	12	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}-40$		ns
t_{AVLL}	Address Valid to ALE Low	28		$t_{CLCL}-25$		ns
t_{LLAX}	Address Hold After ALE Low	48		$t_{CLCL}-25$		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}-65$	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		$t_{CLCL}-25$		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}-45$		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}-60$	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		$t_{CLCL}-25$	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		$t_{CLCL}-8$		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}-80$	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}-100$		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}-100$		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}-90$	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}-28$	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}-150$	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}-165$	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}-50$	$3t_{CLCL}+50$	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}-75$		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		$t_{CLCL}-30$		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}-120$		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		$t_{CLCL}-25$		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	$t_{CLCL}-25$	$t_{CLCL}+25$	ns

External Program Memory Read Cycle

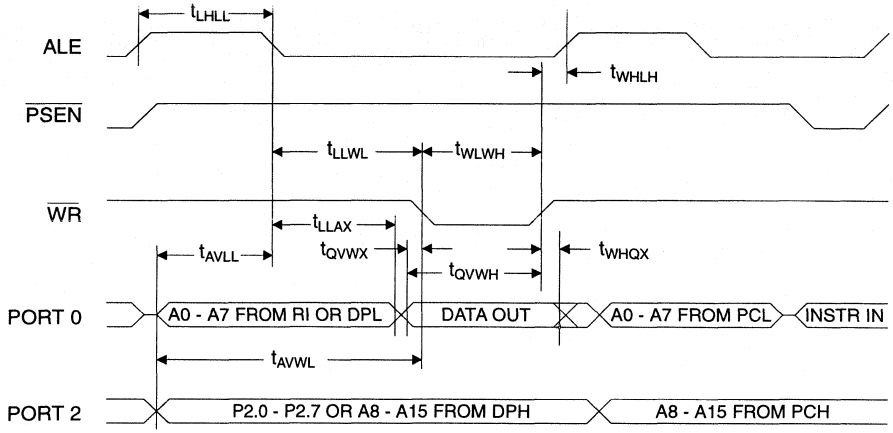


3

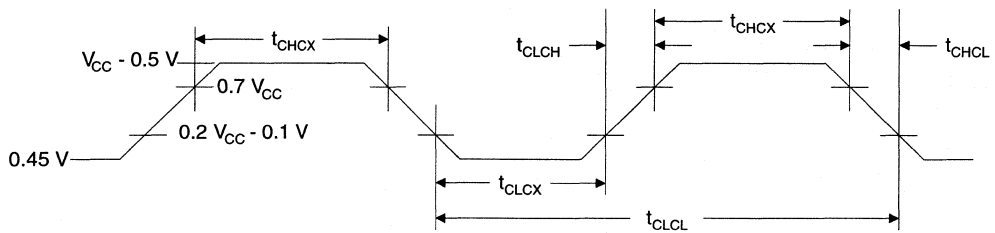
External Data Memory Read Cycle



External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

T_A = -40°C to 85°C

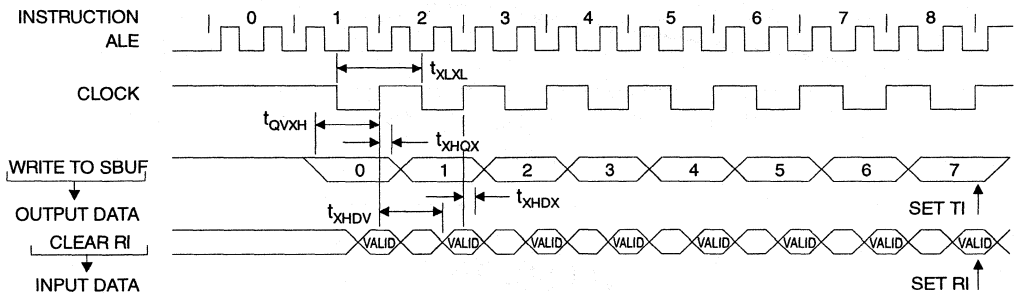
Symbol	Parameter	Min			Max			Units
		V_{CC} = 2.7 V	V_{CC} = 3.0 V	V_{CC} = 3.3 V	V_{CC} = 2.7 V	V_{CC} = 3.0 V	V_{CC} = 3.3 V	
1/t_{CLCL}	Oscillator Frequency	0	0	0	12	16	20	MHz
t_{CLCL}	Clock Period	83.3	62.5	50				ns
t_{CHCX}	High Time	20	15	10				ns
t_{CLCX}	Low Time	20	15	10				ns
t_{CLCH}	Rise Time				20	15	10	ns
t_{CHCL}	Fall Time				20	15	10	ns

Serial Port Timing: Shift Register Mode Test Conditions

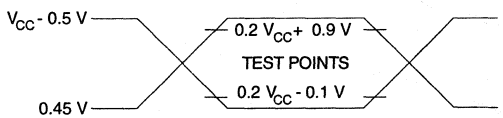
(V_{CC} = 2.7 V to 6 V; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-117		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		10tCLCL-133	ns

Shift Register Mode Timing Waveforms

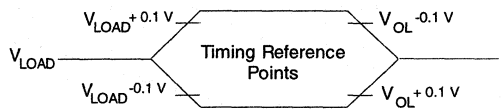


AC Testing Input/Output Waveforms [1]



Note: 1. AC Inputs during testing are driven at 2.4 V for a logic "1" and 0.45 V for a logic "0". Timing measurements are made at 2.0 V for a logic "1" and 0.8 V for a logic "0."

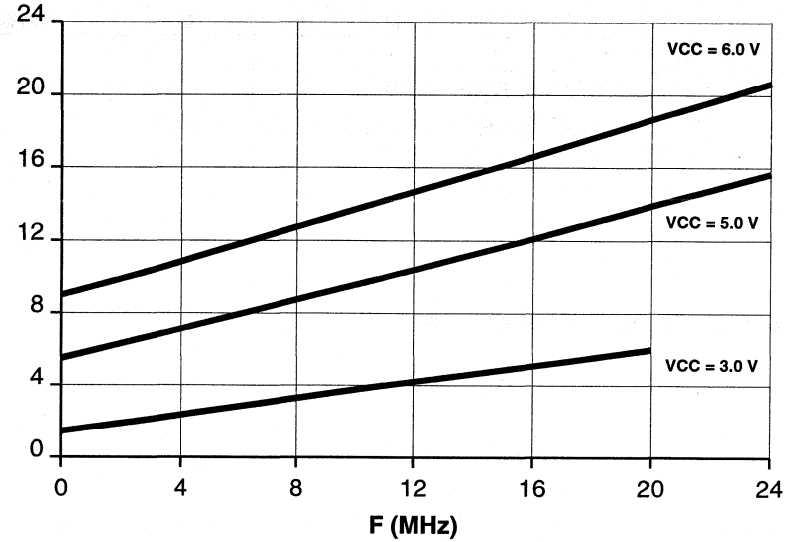
Float Waveforms [1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when a 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

ICC (mA)

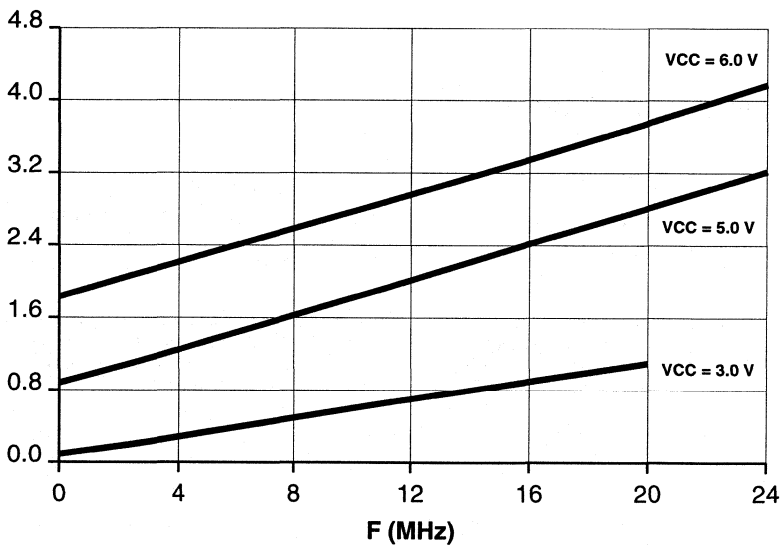
TYPICAL ICC (ACTIVE) at 25° C



AT89LV51

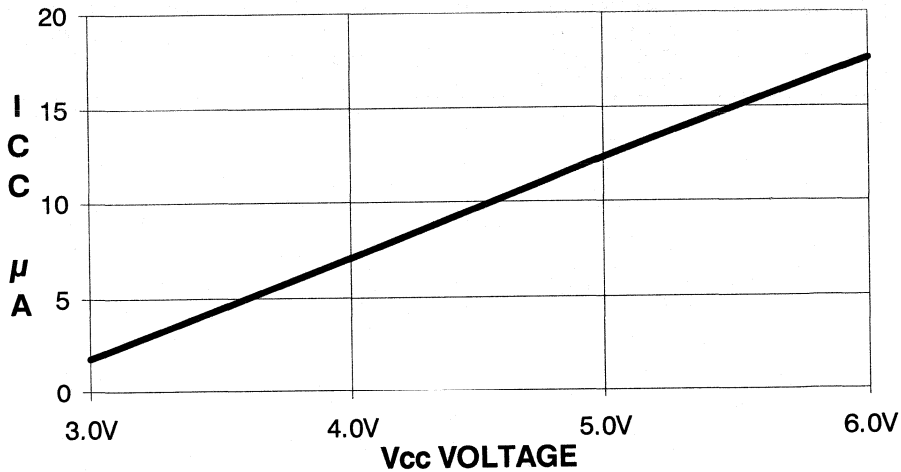
ICC (mA)

TYPICAL ICC (IDLE) at 25° C



AT89LV51

TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7 V to 6 V	AT89LV51-12AC AT89LV51-12JC AT89LV51-12PC AT89LV51-12QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)

Ordering Information

	Package Type
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

Features

- Compatible with MCS-51™ Products
- 8 Kbytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-Bit Timer/Counters
- Eight Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes

8-Bit Microcontroller with 8 Kbytes Flash

3

Description

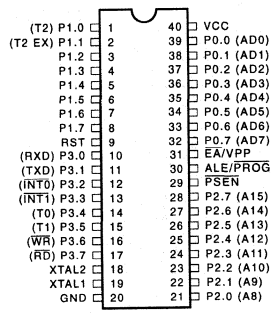
The AT89C52 is a low-power, high-performance CMOS 8-bit microcomputer with 8 Kbytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 and 80C52 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C52 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

The AT89C52 provides the following standard features: 8 Kbytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89C52 is

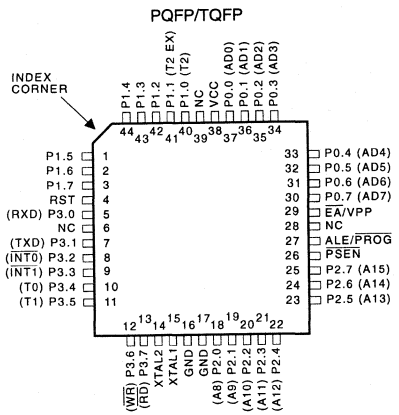
(continued)

Pin Configurations

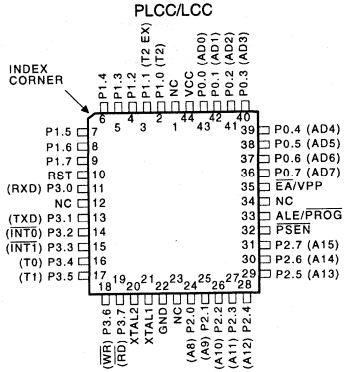
PDIP/Cerdip



PQFP/TQFP

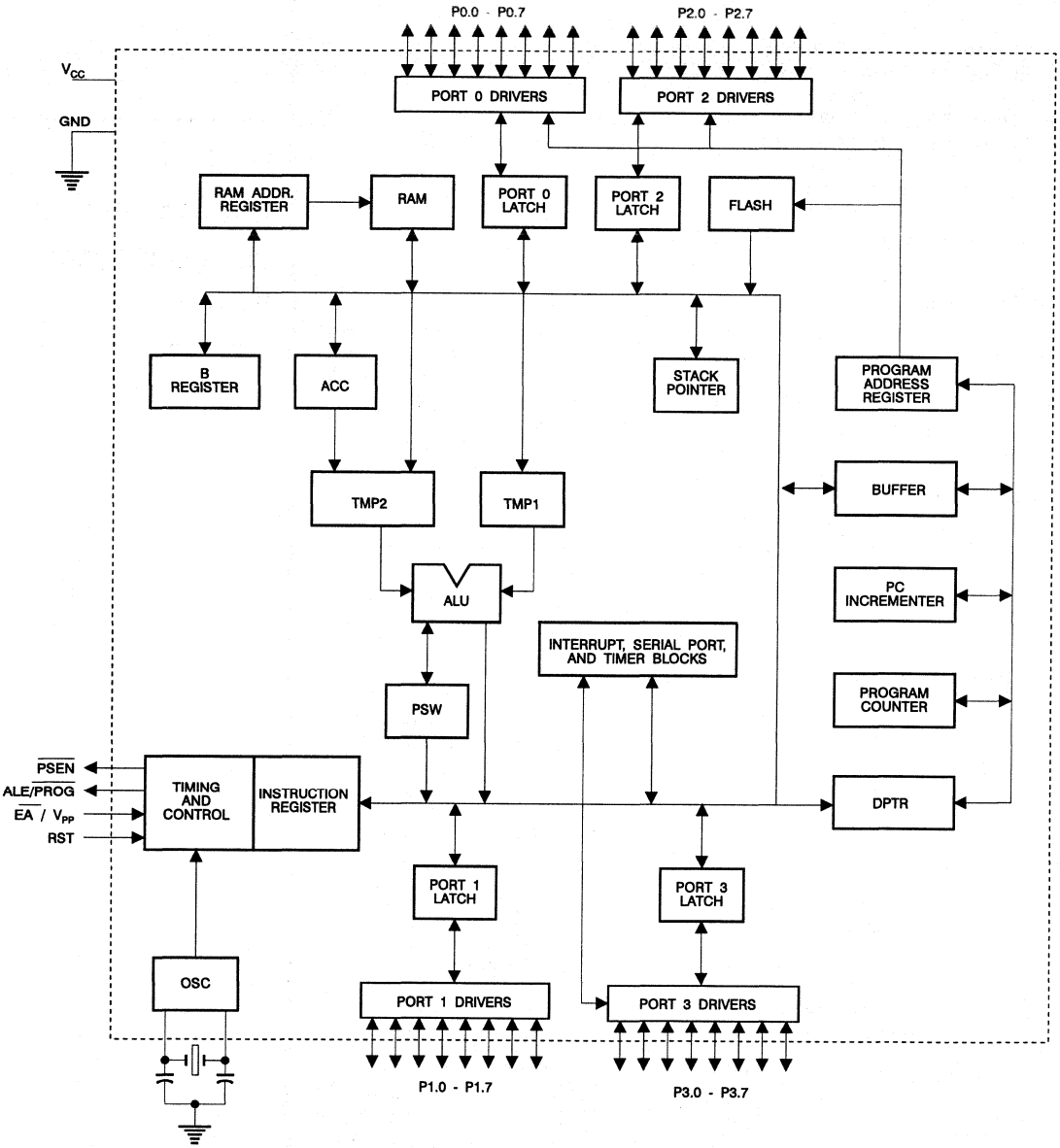


PLCC/LCC



0313E

Block Diagram



Description (Continued)

designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)

Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and programming verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcrontroller is in external execution mode.

$\overline{PSEN}$

Program Store Enable is the read strobe to external program memory.

When the AT89C52 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

(continued)

Pin Description (Continued)

$\overline{EA}$/V_{PP}

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode.

(continued)

Table 1. AT89C52 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111								0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000	87H

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

Bit	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/$\overline{T2}$	Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
CP/$\overline{RL2}$	Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Special Function Registers (Continued)

Interrupt Registers The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

Data Memory

The AT89C52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

MOV 0A0H, #data

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

MOV @R0, #data

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89C52 operate the same way as Timer 0 and Timer 1 in the AT89C51.

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a 1-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Table 3. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-Bit Auto-Reload
0	1	1	16-Bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a 1-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Auto-Reload (Up or Down Counter)

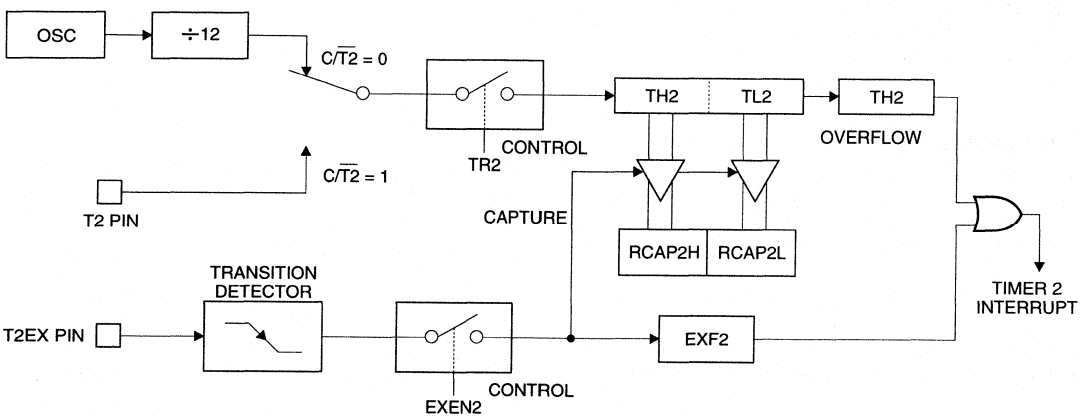
Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a 1-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

(continued)

Figure 1. Timer 2 in Capture Mode



Auto-Reload (Up or Down Counter) (Continued)

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 2. Timer 2 Auto Reload Mode (DCEN = 0)

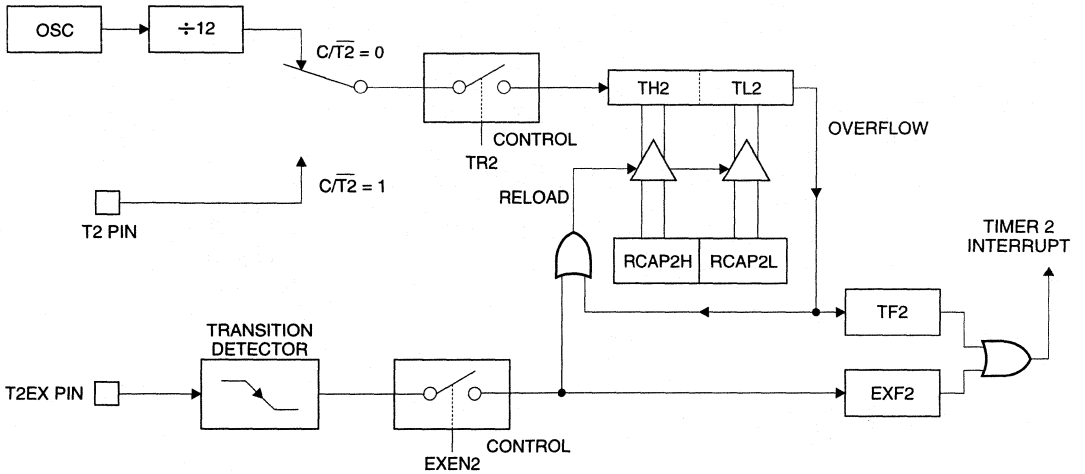


Table 4. T2MOD—Timer 2 Mode Control Register

T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

	—	—	—	—	—	—	T2OE	DCEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
—	Not implemented, reserved for future use.
T2OE	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

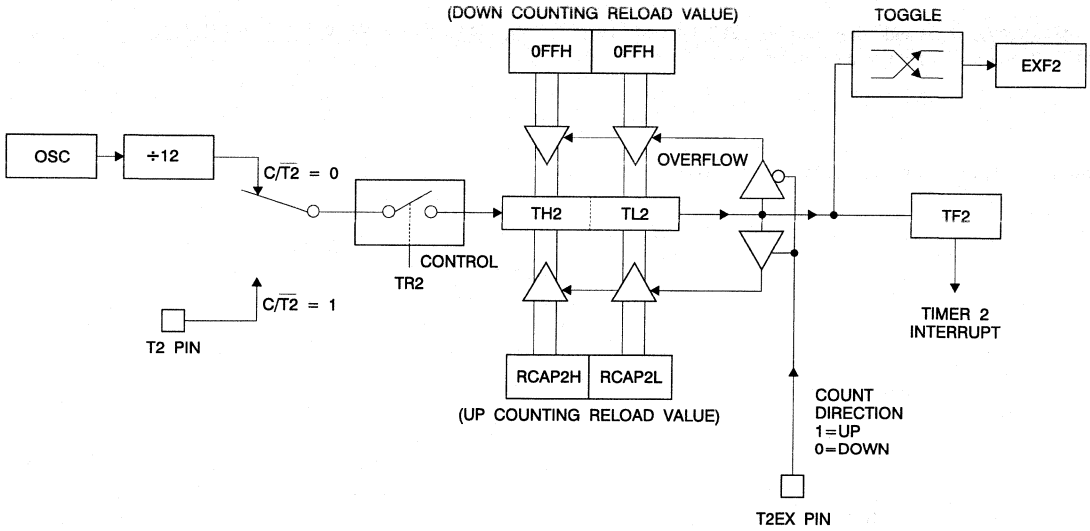
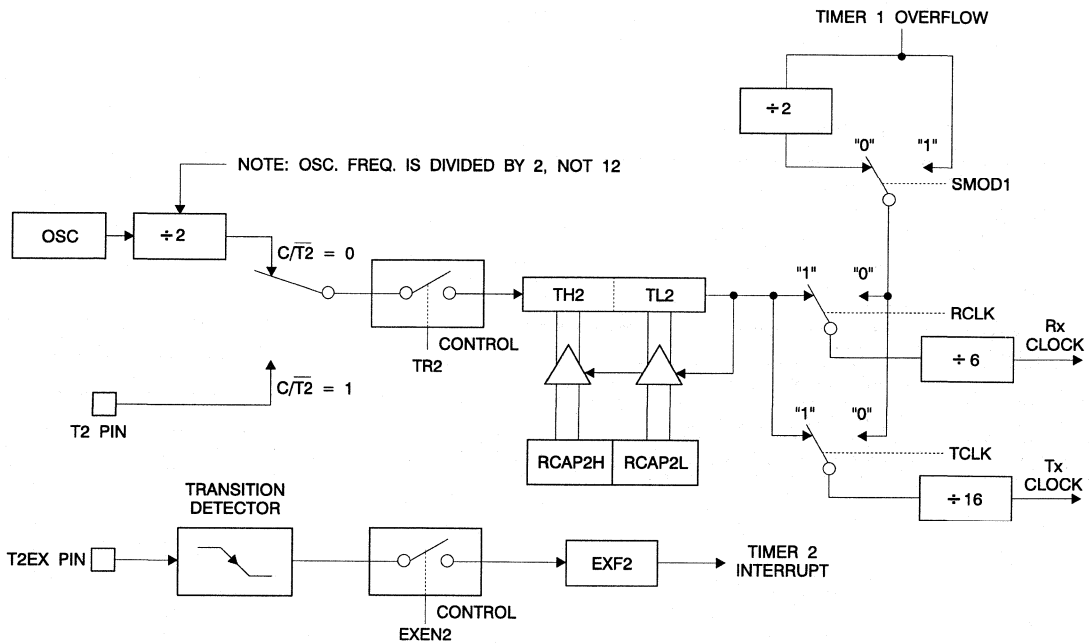


Figure 4. Timer 2 in Baud Rate Generator Mode



Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($C/\overline{T2} = 0$). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

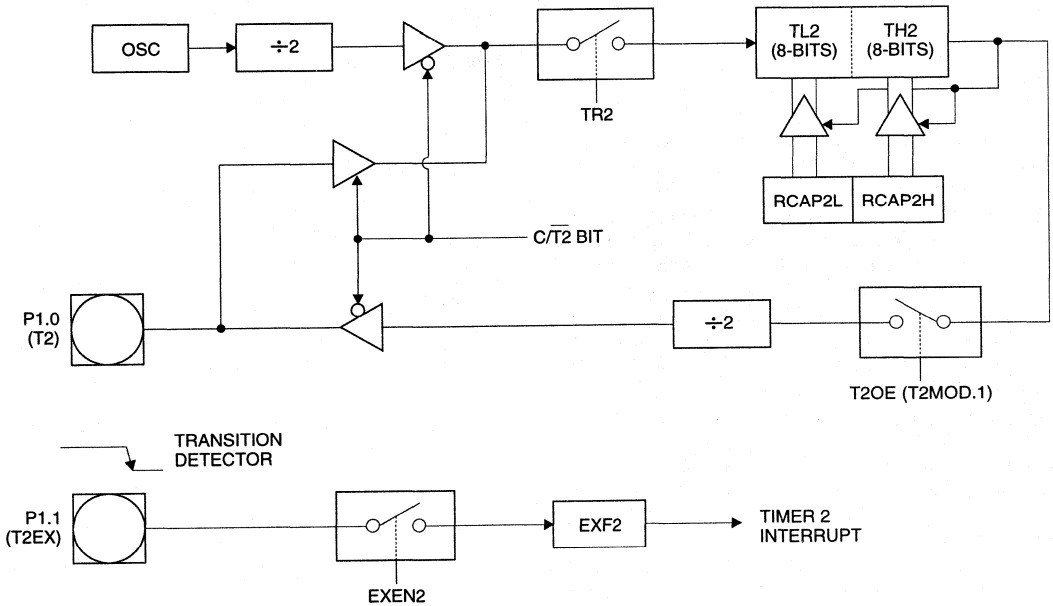
$$\frac{\text{Modes1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a l-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Figure 5. Timer 2 in Clock-Out Mode



Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 4 MHz at a 16 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/$\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock-Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

In the clock-out mode, Timer 2 roll-overs will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

UART

The UART in the AT89C52 operates the same way as the UART in the AT89C51.

Interrupts

The AT89C52 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 5 shows that bit position IE.6 is unimplemented. In the AT89C51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TFI, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Table 5. Interrupt Enable (IE) Register

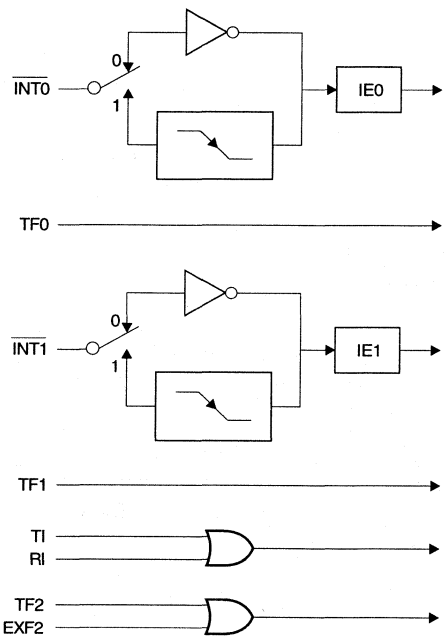
(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.
User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.		

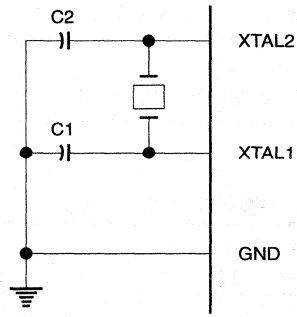
Figure 6. Interrupt Sources



Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 7. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 8. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Figure 7. Oscillator Connections



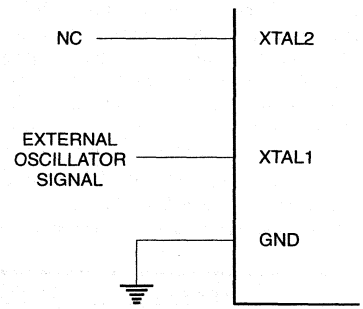
Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Figure 8. External Clock Drive Configuration



Power Down Mode

In the power down mode, the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

Program Memory Lock Bits

The AT89C52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of $\overline{EA}$ must agree with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash memory is disabled.
3	P	P	U	Same as mode 2, but verify is also disabled.
4	P	P	P	Same as mode 3, but external execution is also disabled.

Programming the Flash

The AT89C52 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89C52 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89C52 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12 V	V_{PP} = 5 V
Top-Side Mark	AT89C52 xxxx yyww	AT89C52 xxxx-5 yyww
Signature	(030H)=1EH (031H)=52H (032H)=FFH	(030H)=1EH (031H)=52H (032H)=05H

The AT89C52 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C52, the address, data and control signals should be set up according to the Flash programming mode table and Figures 9 and 10. To program the AT89C52, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}$/V_{PP} to 12 V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89C52 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on PO.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate $\overline{BUSY}$. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all 1s. The chip erase operation must be executed before the code memory can be reprogrammed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H,

(continued)

Programming the Flash (Continued)

031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 52H indicates 89C52
(032H) = FFH indicates 12 V programming
(032H) = 05H indicates 5 V programming

Programming Interface

Every code byte in the Flash array can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L	⎍	H/12V(1)	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L	⎍	H/12V	H	H	H	H
	Bit - 2	H	L	⎍ (2)	H/12V	H	H	L	L
	Bit - 3	H	L	⎍	H/12V	H	L	H	L
Chip Erase		H	L	⎍	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Notes: 1. The signature byte at location 032H designates whether V_{PP} = 12 V or V_{PP} = 5 V should be used to enable programming.
2. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.

Figure 9. Programming the Flash Memory

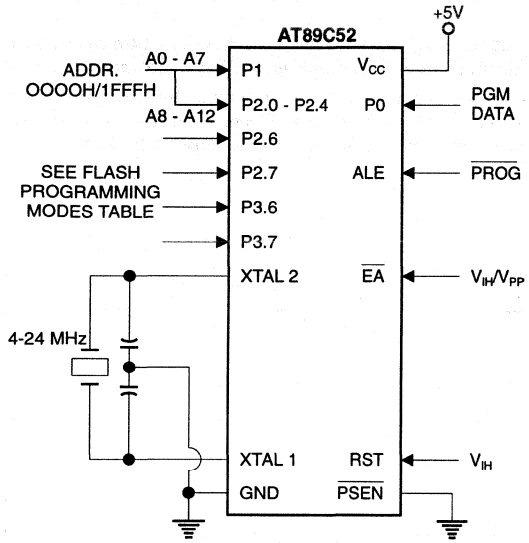
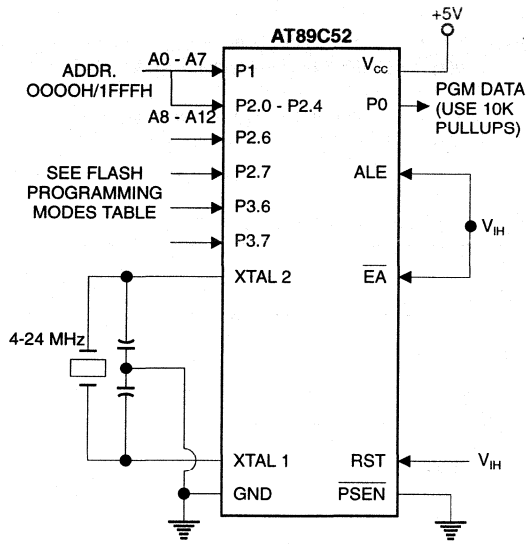


Figure 10. Verifying the Flash Memory



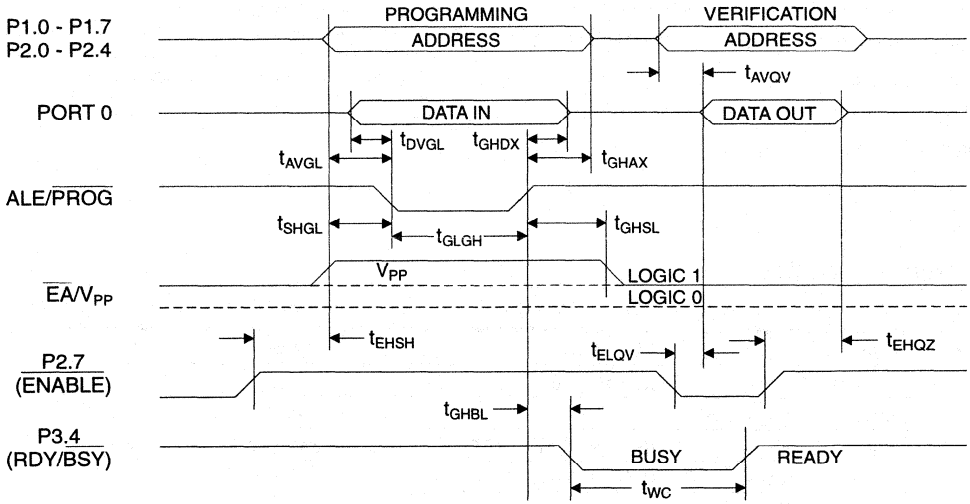
Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, V_{CC} = 5.0 ± 10%

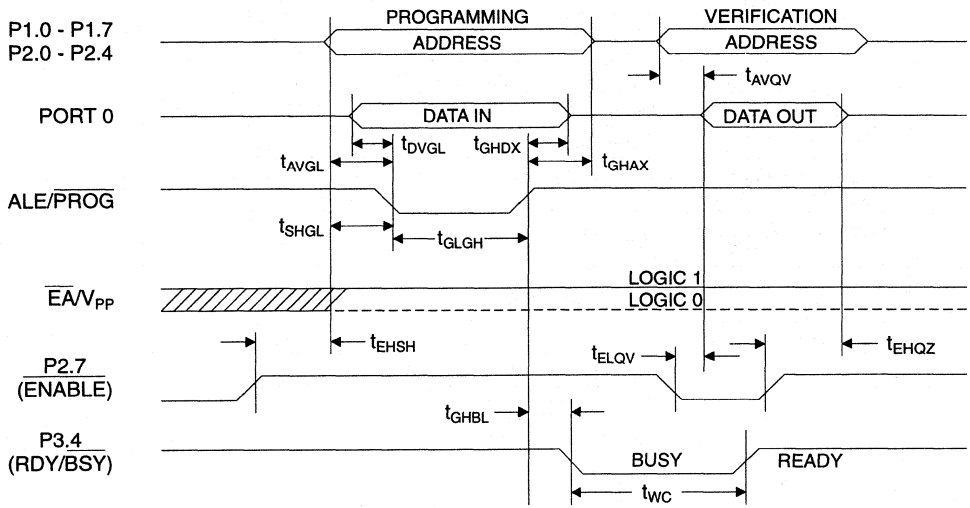
Symbol	Parameter	Min	Max	Units
V_{PP}(1)	Programming Enable Voltage	11.5	12.5	V
I_{PP}(1)	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	4	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold After $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}(1)	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQV}	Data Float After $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

Flash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 5.0 V ± 20%, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage[1] (Port 0, ALE, PSEN)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, PSEN)	I_{OH} = -60 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45 V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2 V		-650	µA
I_{LI}	Input Leakage Current (Port 0, EA)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power Down Mode[2]	V_{CC} = 6 V		100	µA
		V_{CC} = 3 V		40	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin:10 mA
Maximum I_{OL} per 8-bit port:
Port 0:26 mA
Ports 1,2, 3:15 mA
Maximum total IOL for all output pins:71 mA
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power Down is 2 V.

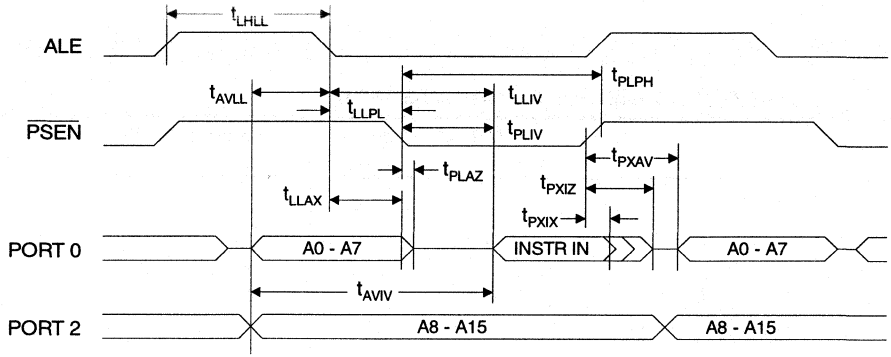
A.C. Characteristics

Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

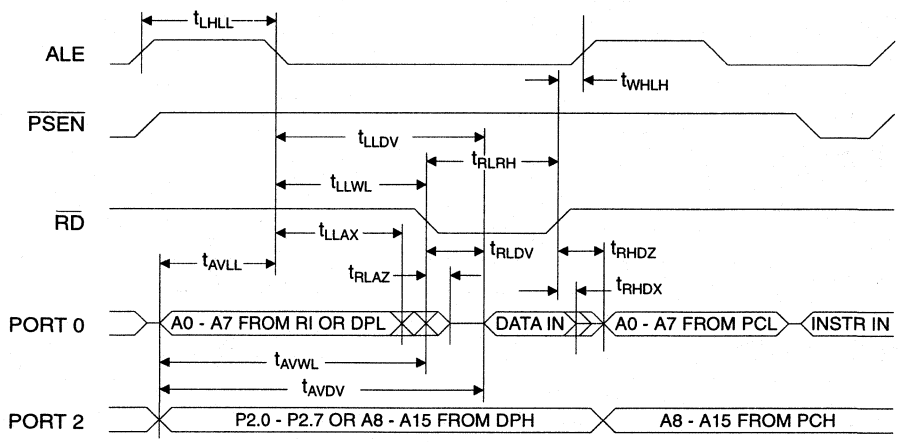
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	28		t_{CLCL}-13		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-13		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-20		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-45	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-10	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-55	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-20		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-120		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-20		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-20	t_{CLCL}+25	ns

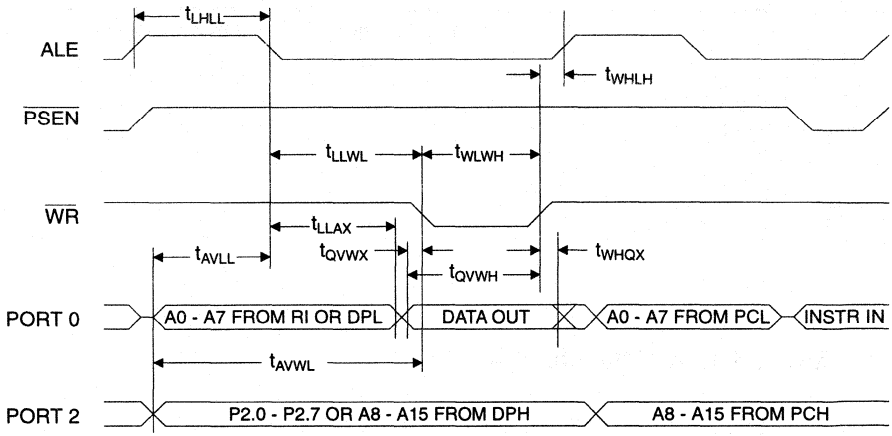
External Program Memory Read Cycle



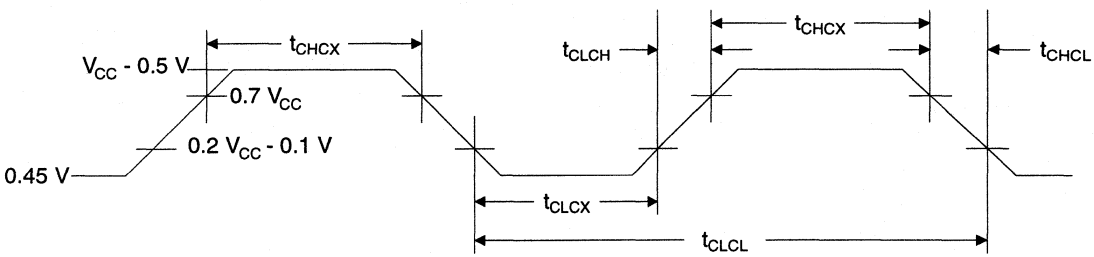
External Data Memory Read Cycle



External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

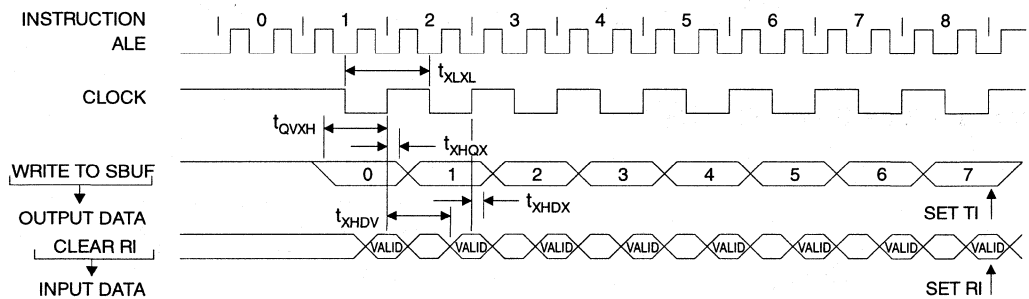
Symbol	Parameter	Min	Max	Units
1/t_{CLCL}	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

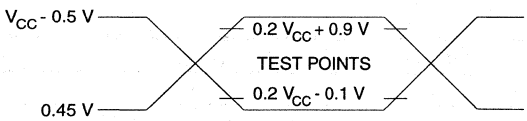
The values in this table are valid for V_{CC} = 5.0 V ± 20% and Load Capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-33		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

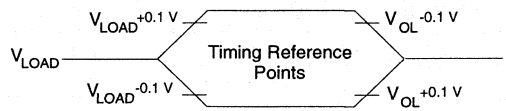


AC Testing Input/Output Waveforms [1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic 1 and 0.45 V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms [1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100-mV change from load voltage occurs. A port pin begins to float when a 100-mV change from the loaded V_{OH}/V_{OL} level occurs.

Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5 V ± 20%	AT89C52-12AC AT89C52-12JC AT89C52-12PC AT89C52-12QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C52-12AI AT89C52-12JI AT89C52-12PI AT89C52-12QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
		AT89C52-12AA AT89C52-12JA AT89C52-12PA AT89C52-12QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
	5 V ± 10%	AT89C52-12DM AT89C52-12LM	40D6 44L	Military (-55°C to 125°C)
		AT89C52-12DM/883 AT89C52-12LM/883	40D6 44L	Military/883C Class B, Fully Compliant (-55°C to 125°C)
16	5 V ± 20%	AT89C52-16AC AT89C52-16JC AT89C52-16PC AT89C52-16QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C52-16AI AT89C52-16JI AT89C52-16PI AT89C52-16QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
		AT89C52-16AA AT89C52-16JA AT89C52-16PA AT89C52-16QA	44A 44J 40P6 44Q	Automotive (-40°C to 125°C)
20	5 V ± 20%	AT89C52-20AC AT89C52-20JC AT89C52-20PC AT89C52-20QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89C52-20AI AT89C52-20JI AT89C52-20PI AT89C52-20QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
24	5 V ± 20%	AT89C52-24AC AT89C52-24JC AT89C52-24PC AT89C52-24QC	44A 44J 44P6 44Q	Commercial (0°C to 70°C)
		AT89C52-24AI AT89C52-24JI AT89C52-24PI AT89C52-24QI	44A 44J 44P6 44Q	Industrial (-40°C to 85°C)

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
40D6	40 Lead, 0.600" Wide, Non-Windowed, Ceramic Dual Inline Package (Cerdip)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
44L	44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

Features

- Compatible with MCS-51™ Products
- 8 Kbytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 12 MHz
- Three-Level Program Memory Lock
- 256 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-Bit Timer/Counters
- Eight Interrupt Sources
- Programmable Serial Channel
- Low Power Idle and Power Down Modes
- 2.7 V to 6 V Operating Range

8-Bit Microcontroller with 8 Kbytes Flash

3

Description

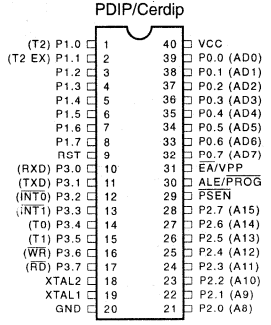
The AT89LV52 is a low-voltage, high-performance CMOS 8-bit microcomputer with 8 Kbytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 and 80C52 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89LV52 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications. The AT89LV52 operates at 2.7 volts up to 6.0 volts.

The AT89LV52 provides the following standard features: 8 Kbytes of Flash, 256 bytes of RAM, 32 I/O lines, three 16-bit timer/counters, a six-vector two-level interrupt architecture, a

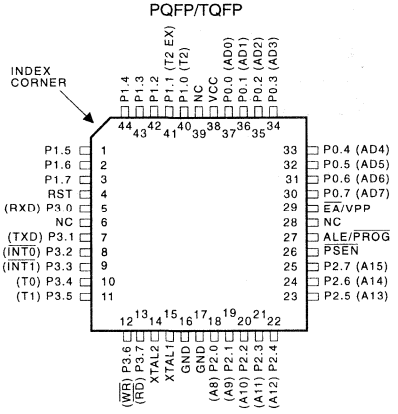
(continued)

Pin Configurations

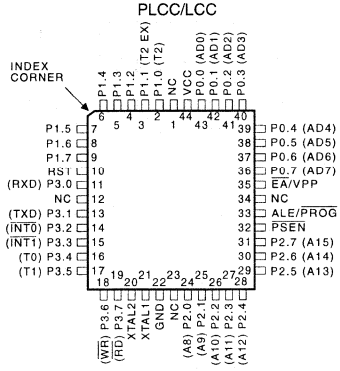
PDIP/Cerdip



PQFP/TQFP

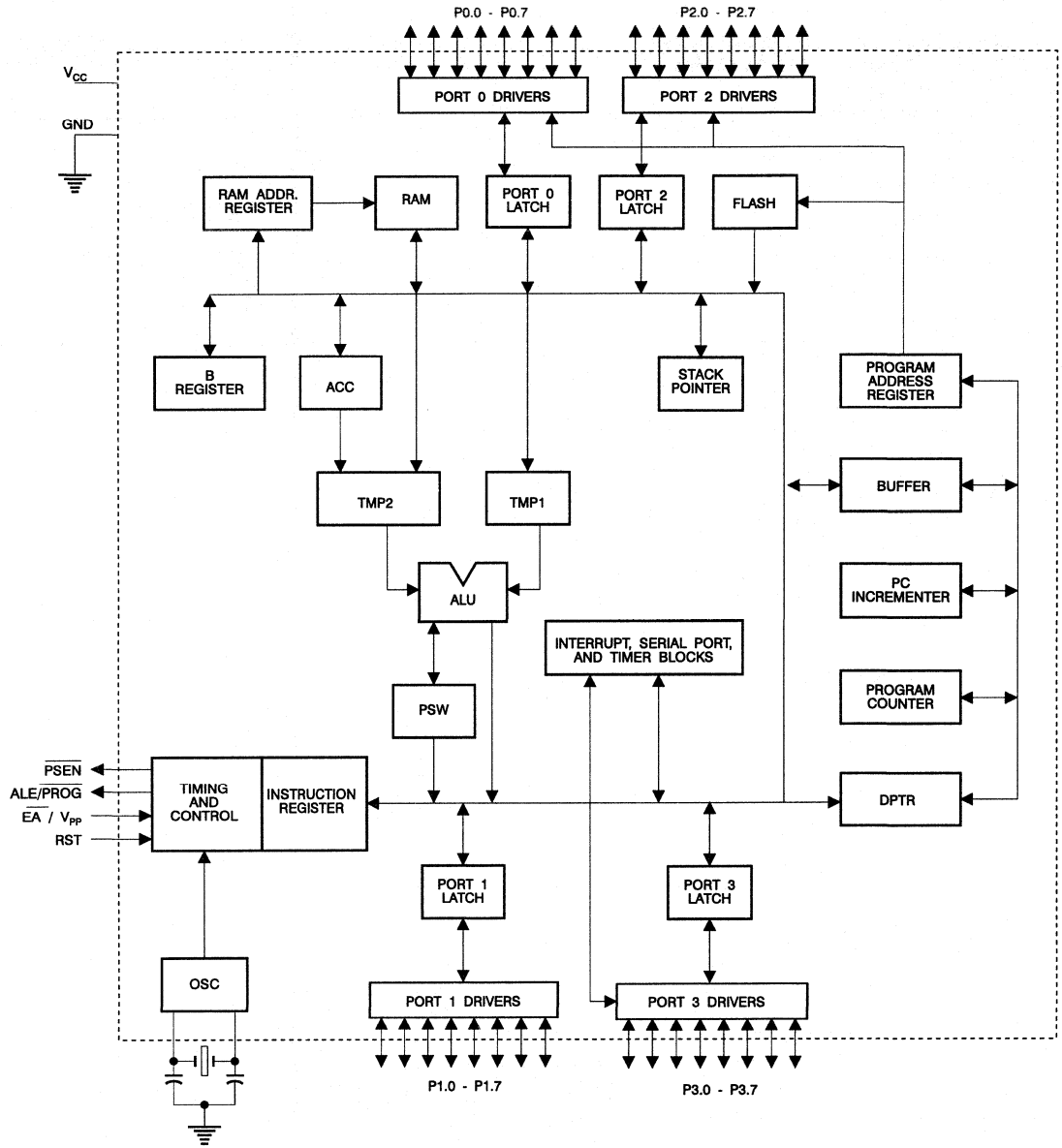


PLCC/LCC



0375B

Block Diagram



Description (Continued)

full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89LV52 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next hardware reset.

Pin Description

V_{CC}

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open drain bidirectional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 can also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode, P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bidirectional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

In addition, P1.0 and P1.1 can be configured to be the timer/counter 2 external count input (P1.0/T2) and the timer/counter 2 trigger input (P1.1/T2EX), respectively, as shown in the following table.

Port Pin	Alternate Functions
P1.0	T2 (external count input to Timer/Counter 2), clock-out
P1.1	T2EX (Timer/Counter 2 capture/reload trigger and direction control)

Port 1 also receives the low-order address bytes during Flash programming and program verification.

Port 2

Port 2 is an 8-bit bidirectional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, Port 2 uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bidirectional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins, they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89LV51, as shown in the following table.

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and programming verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable is an output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation, ALE is emitted at a constant rate of 1/6 the oscillator frequency and may be used for external timing or clocking purposes. Note, however, that one ALE pulse is skipped during each access to external data memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcrontroller is in external execution mode.

$\overline{PSEN}$

Program Store Enable is the read strobe to external program memory.

When the AT89LV52 is executing code from external program memory, $\overline{PSEN}$ is activated twice each machine cycle, except that two $\overline{PSEN}$ activations are skipped during each access to external data memory.

(continued)

Pin Description (Continued)

$\overline{EA}$/V_{PP}

External Access Enable. $\overline{EA}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{EA}$ will be internally latched on reset.

$\overline{EA}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming when 12-volt programming is selected.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Special Function Registers

A map of the on-chip memory area called the Special Function Register (SFR) space is shown in Table 1.

Note that not all of the addresses are occupied, and unoccupied addresses may not be implemented on the chip. Read accesses to these addresses will in general return random data, and write accesses will have an indeterminate effect.

User software should not write 1s to these unlisted locations, since they may be used in future products to invoke new features. In that case, the reset or inactive values of the new bits will always be 0.

Timer 2 Registers Control and status bits are contained in registers T2CON (shown in Table 2) and T2MOD (shown in Table 4) for Timer 2. The register pair (RCAP2H, RCAP2L) are the Capture/Reload registers for Timer 2 in 16-bit capture mode or 16-bit auto-reload mode. *(continued)*

Table 1. AT89LV52 SFR Map and Reset Values

0F8H									0FFH
0F0H	B 00000000								0F7H
0E8H									0EFH
0E0H	ACC 00000000								0E7H
0D8H									0DFH
0D0H	PSW 00000000								0D7H
0C8H	T2CON 00000000	T2MOD XXXXXX00	RCAP2L 00000000	RCAP2H 00000000	TL2 00000000	TH2 00000000			0CFH
0C0H									0C7H
0B8H	IP XX000000								0BFH
0B0H	P3 11111111								0B7H
0A8H	IE 0X000000								0AFH
0A0H	P2 11111111								0A7H
98H	SCON 00000000	SBUF XXXXXXXX							9FH
90H	P1 11111111								97H
88H	TCON 00000000	TMOD 00000000	TL0 00000000	TL1 00000000	TH0 00000000	TH1 00000000			8FH
80H	P0 11111111	SP 00000111	DPL 00000000	DPH 00000000				PCON 0XXX0000	87H

Table 2. T2CON—Timer/Counter 2 Control Register

T2CON Address = 0C8H — Reset Value = 0000 0000B

Bit Addressable

	TF2	EXF2	RCLK	TCLK	EXEN2	TR2	C/$\overline{T2}$	CP/$\overline{RL2}$
Bit	7	6	5	4	3	2	1	0

Symbol	Function
TF2	Timer 2 overflow flag set by a Timer 2 overflow and must be cleared by software. TF2 will not be set when either RCLK = 1 or TCLK = 1.
EXF2	Timer 2 external flag set when either a capture or reload is caused by a negative transition on T2EX and EXEN2 = 1. When Timer 2 interrupt is enabled, EXF2 = 1 will cause the CPU to vector to the Timer 2 interrupt routine. EXF2 must be cleared by software. EXF2 does not cause an interrupt in up/down counter mode (DCEN = 1).
RCLK	Receive clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its receive clock in serial port Modes 1 and 3. RCLK = 0 causes Timer 1 overflow to be used for the receive clock.
TCLK	Transmit clock enable. When set, causes the serial port to use Timer 2 overflow pulses for its transmit clock in serial port Modes 1 and 3. TCLK = 0 causes Timer 1 overflows to be used for the transmit clock.
EXEN2	Timer 2 external enable. When set, allows a capture or reload to occur as a result of a negative transition on T2EX if Timer 2 is not being used to clock the serial port. EXEN2 = 0 causes Timer 2 to ignore events at T2EX.
TR2	Start/Stop control for Timer 2. TR2 = 1 starts the timer.
C/$\overline{T2}$	Timer or counter select for Timer 2. C/$\overline{T2}$ = 0 for timer function. C/$\overline{T2}$ = 1 for external event counter (falling edge triggered).
CP/$\overline{RL2}$	Capture/Reload select. CP/$\overline{RL2}$ = 1 causes captures to occur on negative transitions at T2EX if EXEN2 = 1. CP/$\overline{RL2}$ = 0 causes automatic reloads to occur when Timer 2 overflows or negative transitions occur at T2EX when EXEN2 = 1. When either RCLK or TCLK = 1, this bit is ignored and the timer is forced to auto-reload on Timer 2 overflow.

Special Function Registers (Continued)

Interrupt Registers The individual interrupt enable bits are in the IE register. Two priorities can be set for each of the six interrupt sources in the IP register.

Data Memory

The AT89LV52 implements 256 bytes of on-chip RAM. The upper 128 bytes occupy a parallel address space to the Special Function Registers. That means the upper 128 bytes have the same addresses as the SFR space but are physically separate from SFR space.

When an instruction accesses an internal location above address 7FH, the address mode used in the instruction specifies whether the CPU accesses the upper 128 bytes of RAM or the SFR space. Instructions that use direct addressing access SFR space.

For example, the following direct addressing instruction accesses the SFR at location 0A0H (which is P2).

MOV 0A0H, #data

Instructions that use indirect addressing access the upper 128 bytes of RAM. For example, the following indirect addressing instruction, where R0 contains 0A0H, accesses the data byte at address 0A0H, rather than P2 (whose address is 0A0H).

MOV @R0, #data

Note that stack operations are examples of indirect addressing, so the upper 128 bytes of data RAM are available as stack space.

Timer 0 and 1

Timer 0 and Timer 1 in the AT89LV52 operate the same way as Timer 0 and Timer 1 in the AT89LV51.

Timer 2

Timer 2 is a 16-bit Timer/Counter that can operate as either a timer or an event counter. The type of operation is selected by bit C/$\overline{T2}$ in the SFR T2CON (shown in Table 2). Timer 2 has three operating modes: capture, auto-reload (up or down counting), and baud rate generator. The modes are selected by bits in T2CON, as shown in Table 3.

Timer 2 consists of two 8-bit registers, TH2 and TL2. In the Timer function, the TL2 register is incremented every machine cycle. Since a machine cycle consists of 12 oscillator periods, the count rate is 1/12 of the oscillator frequency.

In the Counter function, the register is incremented in response to a l-to-0 transition at its corresponding external input pin, T2. In this function, the external input is sampled during S5P2 of every machine cycle. When the samples show a high in one cycle and a low in the next cycle, the count is incremented. The new count value appears in the register during S3P1 of the cycle following the one in which the transition was detected. Since two machine cycles (24 oscillator periods) are required to recognize a 1-to-0 transition, the maximum count rate is 1/24 of the oscillator frequency. To ensure that a given level is sampled at least once before it changes, the level should be held for at least one full machine cycle.

Table 3. Timer 2 Operating Modes

RCLK + TCLK	CP/$\overline{RL2}$	TR2	MODE
0	0	1	16-Bit Auto-Reload
0	1	1	16-Bit Capture
1	X	1	Baud Rate Generator
X	X	0	(Off)

Capture Mode

In the capture mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 is a 16-bit timer or counter which upon overflow sets bit TF2 in T2CON. This bit can then be used to generate an interrupt. If EXEN2 = 1, Timer 2 performs the same operation, but a l-to-0 transition at external input T2EX also causes the current value in TH2 and TL2 to be captured into RCAP2H and RCAP2L, respectively. In addition, the transition at T2EX causes bit EXF2 in T2CON to be set. The EXF2 bit, like TF2, can generate an interrupt. The capture mode is illustrated in Figure 1.

Auto-Reload (Up or Down Counter)

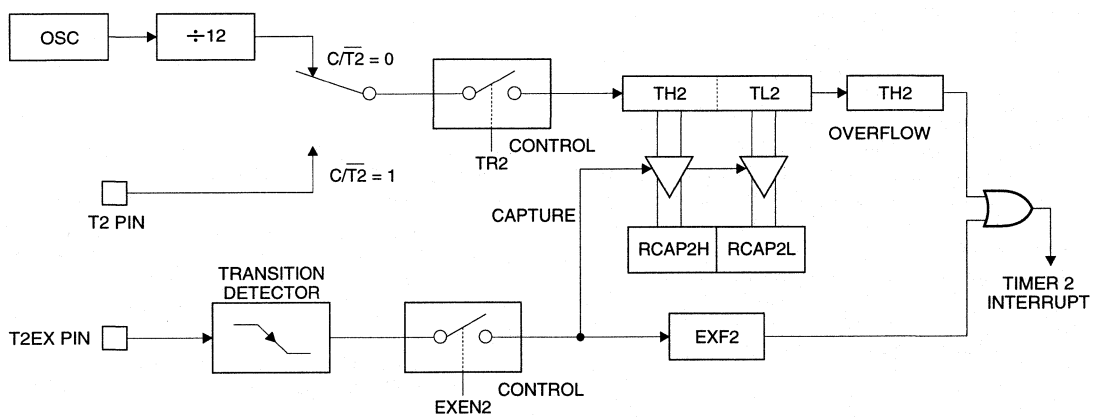
Timer 2 can be programmed to count up or down when configured in its 16-bit auto-reload mode. This feature is invoked by the DCEN (Down Counter Enable) bit located in the SFR T2MOD (see Table 4). Upon reset, the DCEN bit is set to 0 so that timer 2 will default to count up. When DCEN is set, Timer 2 can count up or down, depending on the value of the T2EX pin.

Figure 2 shows Timer 2 automatically counting up when DCEN = 0. In this mode, two options are selected by bit EXEN2 in T2CON. If EXEN2 = 0, Timer 2 counts up to 0FFFFH and then sets the TF2 bit upon overflow. The overflow also causes the timer registers to be reloaded with the 16-bit value in RCAP2H and RCAP2L. The values in RCAP2H and RCAP2L are preset by software. If EXEN2 = 1, a 16-bit reload can be triggered either by an overflow or by a l-to-0 transition at external input T2EX. This transition also sets the EXF2 bit. Both the TF2 and EXF2 bits can generate an interrupt if enabled.

Setting the DCEN bit enables Timer 2 to count up or down, as shown in Figure 3. In this mode, the T2EX pin controls the direction of the count. A logic 1 at T2EX makes Timer 2 count up. The timer will overflow at 0FFFFH and set the TF2 bit. This overflow also causes the 16-bit value in RCAP2H and RCAP2L to be reloaded into the timer registers, TH2 and TL2, respectively.

(continued)

Figure 1. Timer 2 in Capture Mode



Auto-Reload (Up or Down Counter) (Continued)

A logic 0 at T2EX makes Timer 2 count down. The timer underflows when TH2 and TL2 equal the values stored in RCAP2H and RCAP2L. The underflow sets the TF2 bit and causes 0FFFFH to be reloaded into the timer registers.

The EXF2 bit toggles whenever Timer 2 overflows or underflows and can be used as a 17th bit of resolution. In this operating mode, EXF2 does not flag an interrupt.

Figure 2. Timer 2 Auto Reload Mode (DCEN = 0)

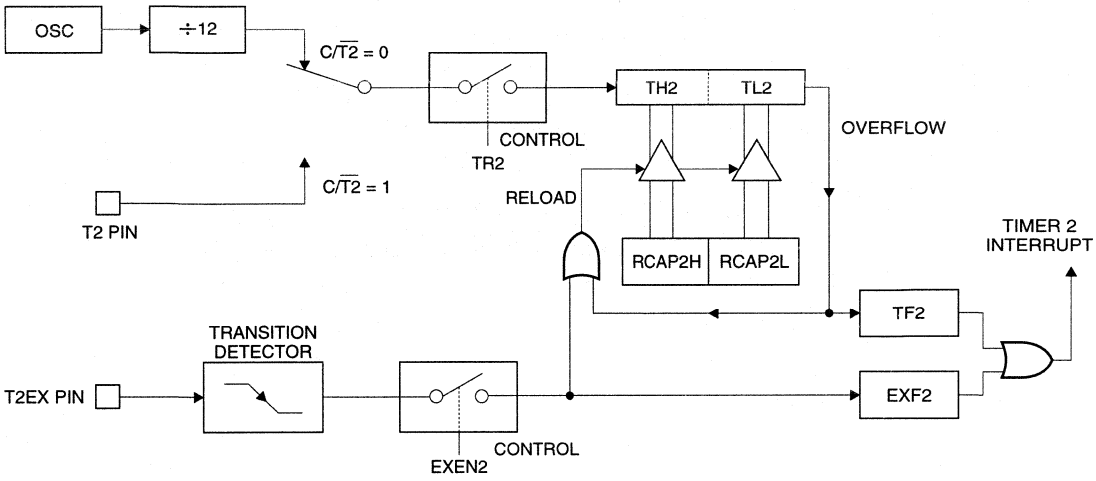


Table 4. T2MOD—Timer 2 Mode Control Register

T2MOD Address = 0C9H — Reset Value = XXXX XX00B

Not Bit Addressable

	—	—	—	—	—	—	T2OE	DCEN
Bit	7	6	5	4	3	2	1	0

Symbol	Function
—	Not implemented, reserved for future use.
T2OE	Timer 2 Output Enable bit.
DCEN	When set, this bit allows Timer 2 to be configured as an up/down counter.

Figure 3. Timer 2 Auto Reload Mode (DCEN = 1)

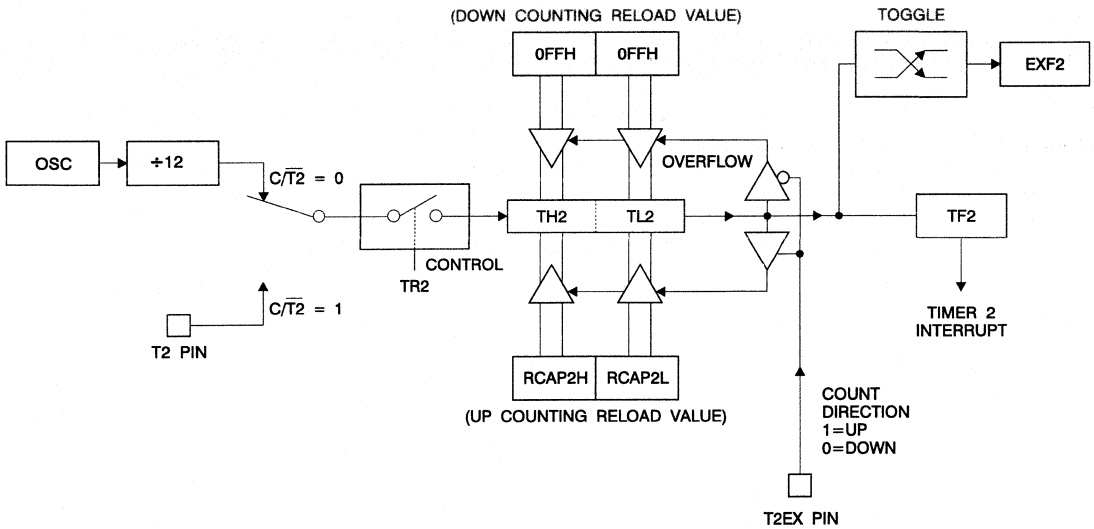
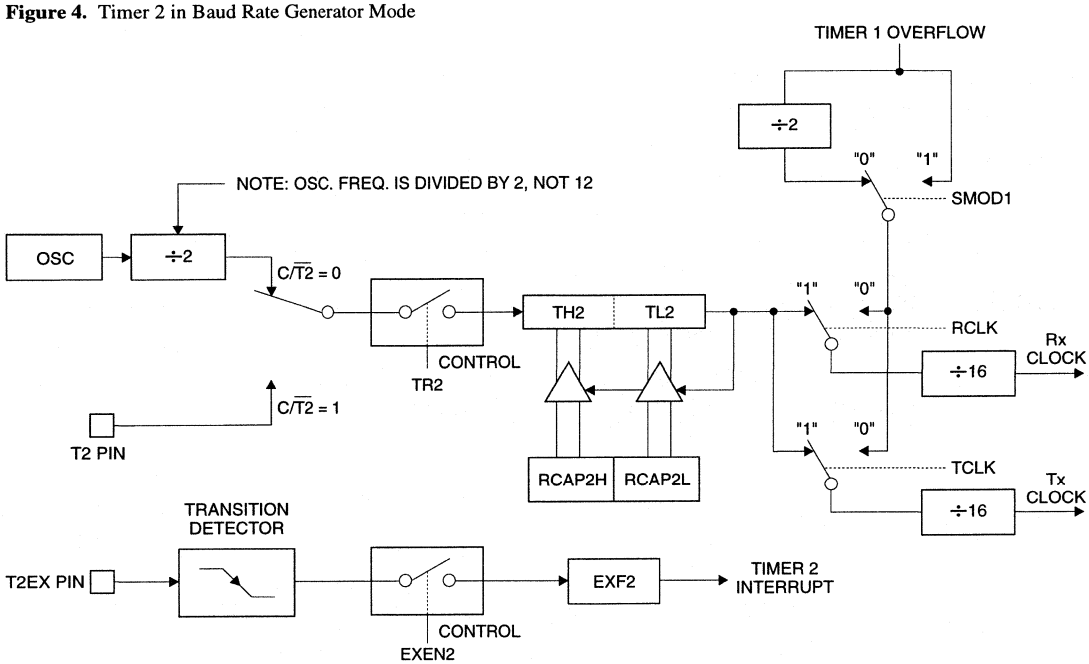


Figure 4. Timer 2 in Baud Rate Generator Mode



Baud Rate Generator

Timer 2 is selected as the baud rate generator by setting TCLK and/or RCLK in T2CON (Table 2). Note that the baud rates for transmit and receive can be different if Timer 2 is used for the receiver or transmitter and Timer 1 is used for the other function. Setting RCLK and/or TCLK puts Timer 2 into its baud rate generator mode, as shown in Figure 4.

The baud rate generator mode is similar to the auto-reload mode, in that a rollover in TH2 causes the Timer 2 registers to be reloaded with the 16-bit value in registers RCAP2H and RCAP2L, which are preset by software.

The baud rates in Modes 1 and 3 are determined by Timer 2's overflow rate according to the following equation.

$$\text{Modes 1 and 3 Baud Rates} = \frac{\text{Timer 2 Overflow Rate}}{16}$$

The Timer can be configured for either timer or counter operation. In most applications, it is configured for timer operation ($C/\overline{T2} = 0$). The timer operation is different for Timer 2 when it is used as a baud rate generator. Normally, as a timer, it increments every machine cycle (at 1/12 the oscillator frequency). As a baud rate generator, however, it increments every state time (at 1/2 the oscillator frequency). The baud rate formula is given below.

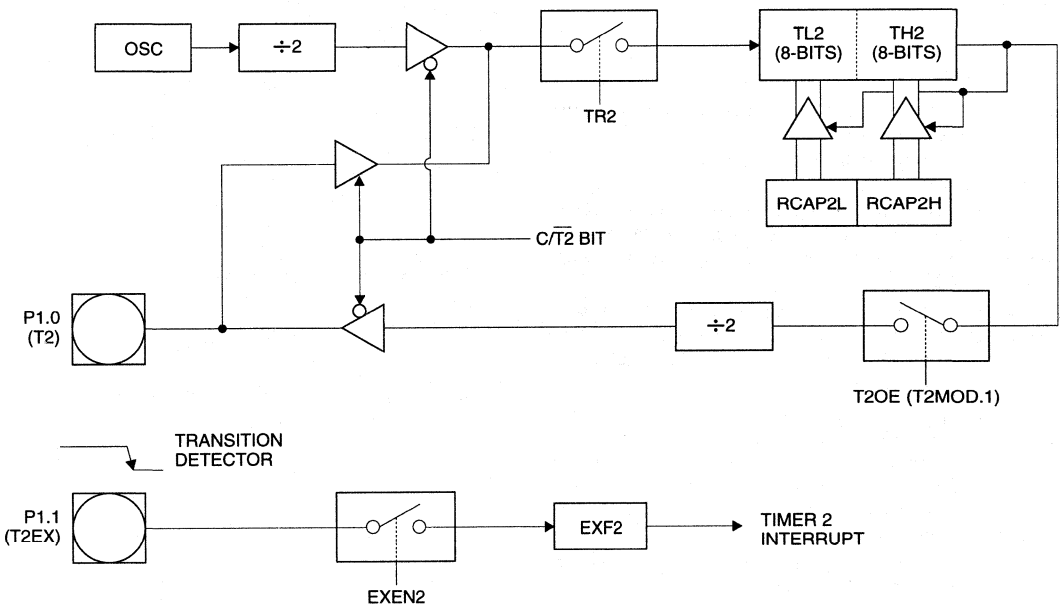
$$\frac{\text{Modes 1 and 3}}{\text{Baud Rate}} = \frac{\text{Oscillator Frequency}}{32 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

where (RCAP2H, RCAP2L) is the content of RCAP2H and RCAP2L taken as a 16-bit unsigned integer.

Timer 2 as a baud rate generator is shown in Figure 4. This figure is valid only if RCLK or TCLK = 1 in T2CON. Note that a rollover in TH2 does not set TF2 and will not generate an interrupt. Note too, that if EXEN2 is set, a 1-to-0 transition in T2EX will set EXF2 but will not cause a reload from (RCAP2H, RCAP2L) to (TH2, TL2). Thus when Timer 2 is in use as a baud rate generator, T2EX can be used as an extra external interrupt.

Note that when Timer 2 is running (TR2 = 1) as a timer in the baud rate generator mode, TH2 or TL2 should not be read from or written to. Under these conditions, the Timer is incremented every state time, and the results of a read or write may not be accurate. The RCAP2 registers may be read but should not be written to, because a write might overlap a reload and cause write and/or reload errors. The timer should be turned off (clear TR2) before accessing the Timer 2 or RCAP2 registers.

Figure 5. Timer 2 in Clock-Out Mode



Programmable Clock Out

A 50% duty cycle clock can be programmed to come out on P1.0, as shown in Figure 5. This pin, besides being a regular I/O pin, has two alternate functions. It can be programmed to input the external clock for Timer/Counter 2 or to output a 50% duty cycle clock ranging from 61 Hz to 3 MHz at a 12 MHz operating frequency.

To configure the Timer/Counter 2 as a clock generator, bit C/$\overline{T2}$ (T2CON.1) must be cleared and bit T2OE (T2MOD.1) must be set. Bit TR2 (T2CON.2) starts and stops the timer.

The clock-out frequency depends on the oscillator frequency and the reload value of Timer 2 capture registers (RCAP2H, RCAP2L), as shown in the following equation.

$$\text{Clock–Out Frequency} = \frac{\text{Oscillator Frequency}}{4 \times [65536 - (\text{RCAP2H, RCAP2L})]}$$

In the clock-out mode, Timer 2 roll-overs will not generate an interrupt. This behavior is similar to when Timer 2 is used as a baud-rate generator. It is possible to use Timer 2 as a baud-rate generator and a clock generator simultaneously. Note, however, that the baud-rate and clock-out frequencies cannot be determined independently from one another since they both use RCAP2H and RCAP2L.

UART

The UART in the AT89LV52 operates the same way as the UART in the AT89LV51.

Interrupts

The AT89LV52 has a total of six interrupt vectors: two external interrupts ($\overline{INT0}$ and $\overline{INT1}$), three timer interrupts (Timers 0, 1, and 2), and the serial port interrupt. These interrupts are all shown in Figure 6.

Each of these interrupt sources can be individually enabled or disabled by setting or clearing a bit in Special Function Register IE. IE also contains a global disable bit, EA, which disables all interrupts at once.

Note that Table 5 shows that bit position IE.6 is unimplemented. In the AT89LV51, bit position IE.5 is also unimplemented. User software should not write 1s to these bit positions, since they may be used in future AT89 products.

Timer 2 interrupt is generated by the logical OR of bits TF2 and EXF2 in register T2CON. Neither of these flags is cleared by hardware when the service routine is vectored to. In fact, the service routine may have to determine whether it was TF2 or EXF2 that generated the interrupt, and that bit will have to be cleared in software.

The Timer 0 and Timer 1 flags, TF0 and TFI, are set at S5P2 of the cycle in which the timers overflow. The values are then polled by the circuitry in the next cycle. However, the Timer 2 flag, TF2, is set at S2P2 and is polled in the same cycle in which the timer overflows.

Table 5. Interrupt Enable (IE) Register

(MSB)							(LSB)
EA	—	ET2	ES	ET1	EX1	ET0	EX0

Enable Bit = 1 enables the interrupt.

Enable Bit = 0 disables the interrupt.

Symbol	Position	Function
EA	IE.7	Disables all interrupts. If EA = 0, no interrupt is acknowledged. If EA = 1, each interrupt source is individually enabled or disabled by setting or clearing its enable bit.
—	IE.6	Reserved.
ET2	IE.5	Timer 2 interrupt enable bit.
ES	IE.4	Serial Port interrupt enable bit.
ET1	IE.3	Timer 1 interrupt enable bit.
EX1	IE.2	External interrupt 1 enable bit.
ET0	IE.1	Timer 0 interrupt enable bit.
EX0	IE.0	External interrupt 0 enable bit.
User software should never write 1s to unimplemented bits, because they may be used in future AT89 products.		

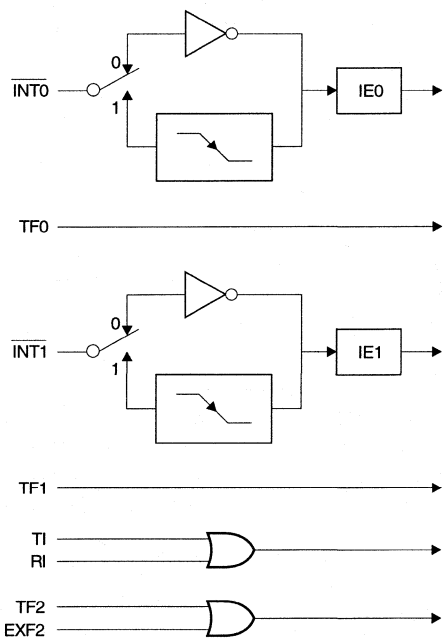


Figure 6. Interrupt Sources

Oscillator Characteristics

XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier that can be configured for use as an on-chip oscillator, as shown in Figure 7. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven, as shown in Figure 8. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

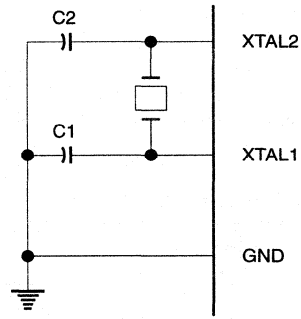
In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

Note that when idle mode is terminated by a hardware reset, the device normally resumes program execution from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when idle mode is terminated by a reset, the instruction following the one that invokes idle mode should not write to a port pin or to external memory.

Power Down Mode

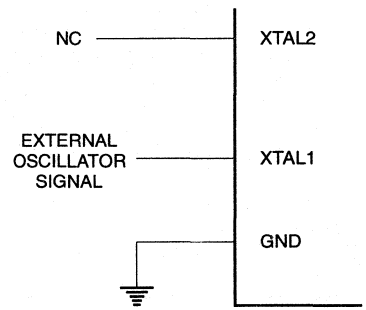
In the power down mode, the oscillator is stopped, and the instruction that invokes power down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power down mode is terminated. The only exit from power down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Figure 7. Oscillator Connections



Notes: C1, C2 = 30 pF ± 10 pF for Crystals
= 40 pF ± 10 pF for Ceramic Resonators

Figure 8. External Clock Drive Configuration



Status of External Pins During Idle and Power Down

Mode	Program Memory	ALE	$\overline{PSEN}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power Down	Internal	0	0	Data	Data	Data	Data
Power Down	External	0	0	Float	Data	Data	Data

Program Memory Lock Bits

The AT89LV52 has three lock bits that can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the following table.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value and holds that value until reset is activated. The latched value of $\overline{EA}$ must agree with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features.
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash memory is disabled.
3	P	P	U	Same as mode 2, but verify is also disabled.
4	P	P	P	Same as mode 3, but external execution is also disabled.

Programming the Flash

The AT89LV52 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low voltage programming mode provides a convenient way to program the AT89LV52 inside the user's system, while the high-voltage programming mode is compatible with conventional third party Flash or EPROM programmers.

The AT89LV52 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12 V	V_{PP} = 5 V
Top-Side Mark	AT89LV52 xxxx yyww	AT89LV52 xxxx-5 yyww
Signature	(030H)=1EH (031H)=62H (032H)=FFH	(030H)=1EH (031H)=62H (032H)=05H

The AT89LV52 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89LV52, the address, data and control signals should be set up according to the Flash programming mode table and Figures 9 and 10. To program the AT89LV52, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12 V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

$\overline{Data}$ Polling: The AT89LV52 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written data on PO.7. Once the write cycle has been completed, true data is valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate $\overline{BUSY}$. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all 1s. The chip erase operation must be executed before the code memory can be reprogrammed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H,

(continued)

Programming the Flash (Continued)

031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 52H indicates 89LV52
(032H) = FFH indicates 12 V programming
(032H) = 05H indicates 5 V programming

Programming Interface

Every code byte in the Flash array can be written, and the entire array can be erased, by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

Flash Programming Modes

Mode		RST	$\overline{PSEN}$	ALE/$\overline{PROG}$	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V(1)	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L	(2)	H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L		H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Notes: 1. The signature byte at location 032H designates whether V_{PP} = 12 V or V_{PP} = 5 V should be used to enable programming.

2. Chip Erase requires a 10 ms $\overline{PROG}$ pulse.

Figure 9. Programming the Flash Memory

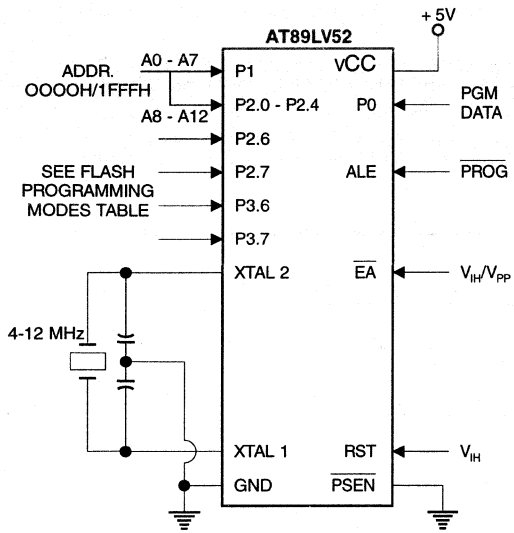
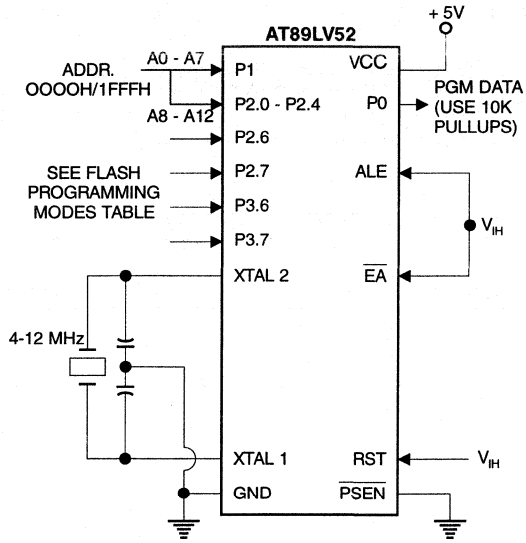


Figure 10. Verifying the Flash Memory



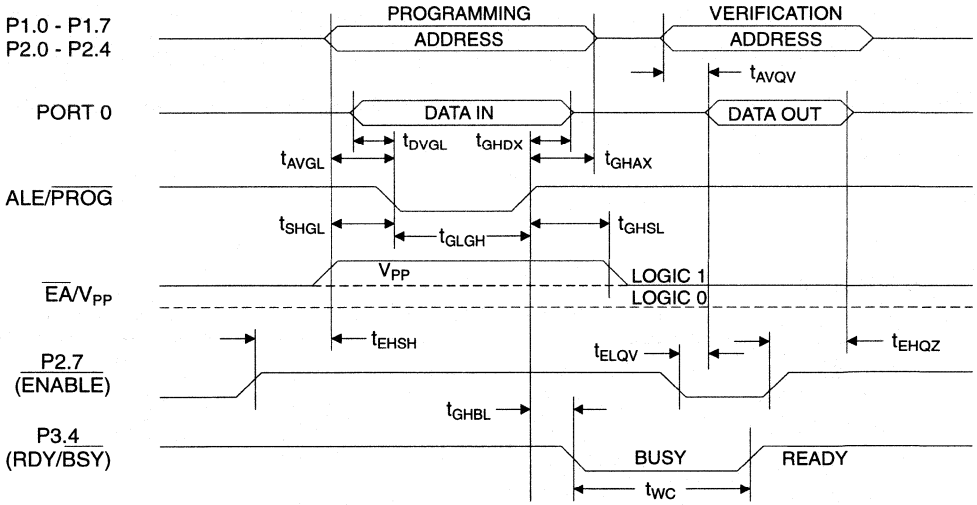
Flash Programming and Verification Characteristics

T_A = 21°C to 27°C, V_{CC} = 5.0 ± 10%

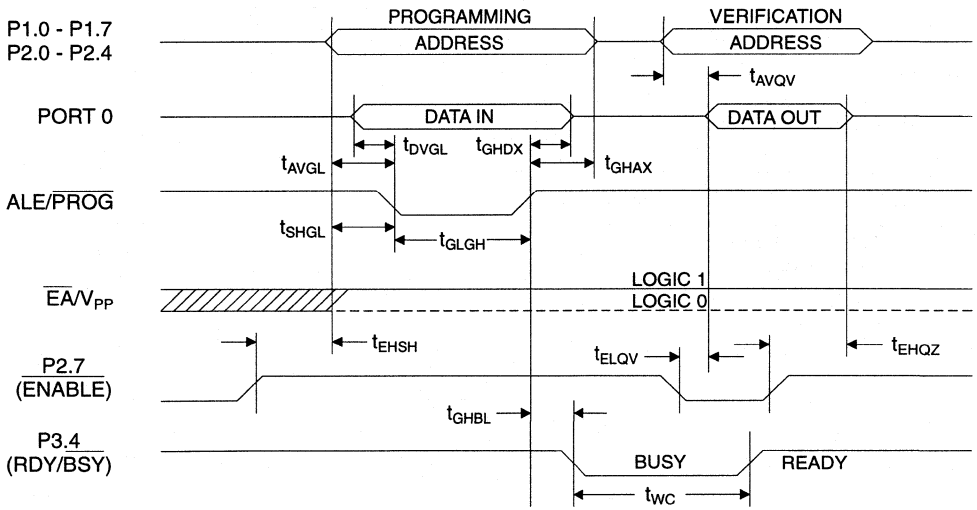
Symbol	Parameter	Min	Max	Units
V_{PP}(1)	Programming Enable Voltage	11.5	12.5	V
I_{PP}(1)	Programming Enable Current		25	µA
1/t_{CLCL}	Oscillator Frequency	4	12	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	48t_{CLCL}		
t_{GHAX}	Address Hold After $\overline{PROG}$	48t_{CLCL}		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	48t_{CLCL}		
t_{GHDX}	Data Hold After $\overline{PROG}$	48t_{CLCL}		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	48t_{CLCL}		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}(1)	V_{PP} Hold After $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		48t_{CLCL}	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		48t_{CLCL}	
t_{EHQV}	Data Float After $\overline{ENABLE}$	0	48t_{CLCL}	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

Flash Programming and Verification Waveforms - High Voltage Mode



Flash Programming and Verification Waveforms - Low Voltage Mode



Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0 V to +7.0 V
Maximum Operating Voltage	6.6 V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. Characteristics

The values shown in this table are valid for T_A = -40°C to 85°C and V_{CC} = 5.0 V ± 20%, unless otherwise noted.

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low Voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC}-0.1	V
V_{IL1}	Input Low Voltage ($\overline{EA}$)		-0.5	0.2 V_{CC}-0.3	V
V_{IH}	Input High Voltage	(Except XTAL1, RST)	0.2 V_{CC}+0.9	V_{CC}+0.5	V
V_{IH1}	Input High Voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC}+0.5	V
V_{OL}	Output Low Voltage(1) (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low Voltage(1) (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High Voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High Voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5 V ± 10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45 V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2 V		-650	µA
I_{LI}	Input Leakage Current (Port 0, EA)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pulldown Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		25	mA
		Idle Mode, 12 MHz		6.5	mA
	Power Down Mode(2)	V_{CC} = 6 V		100	µA
		V_{CC} = 3 V		40	µA

Notes: 1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
Maximum I_{OL} per port pin:10 mA
Maximum I_{OL} per 8-bit port:
Port 0:26 mA
Ports 1,2, 3:15 mA
Maximum total IOL for all output pins:71 mA
If IOL exceeds the test condition, VOL may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power Down is 2 V.

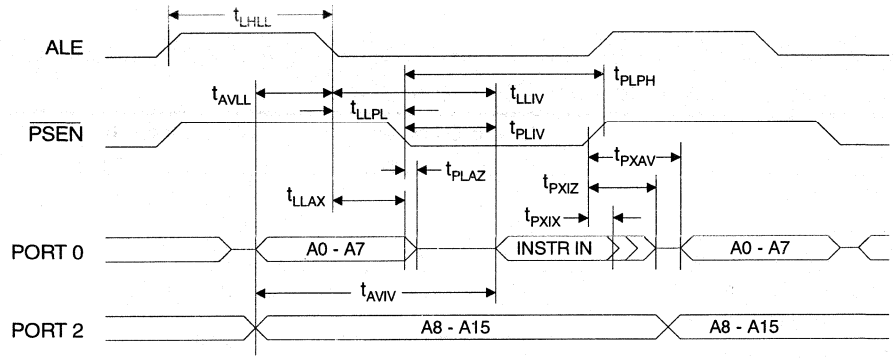
A.C. Characteristics

Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

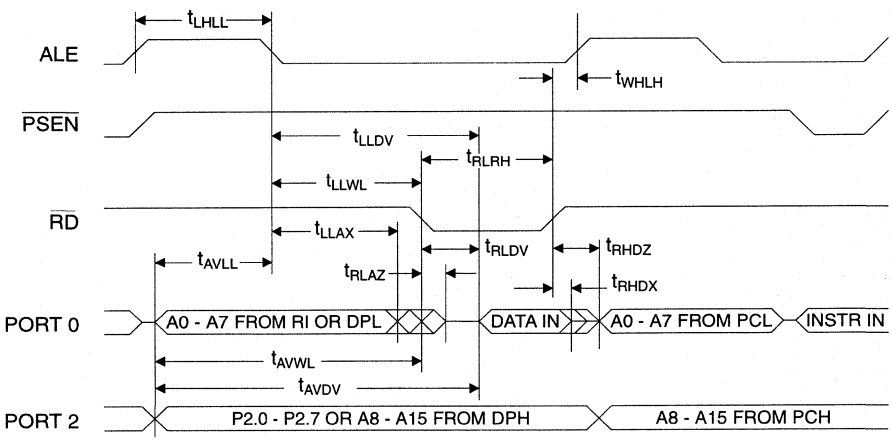
External Program and Data Memory Characteristics

Symbol	Parameter	12 MHz Oscillator		Variable Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	12	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	28		t_{CLCL}-25		ns
t_{LLAX}	Address Hold After ALE Low	48		t_{CLCL}-25		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-25		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-45		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-60	ns
t_{PXIX}	Input Instruction Hold After $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float After $\overline{PSEN}$		59		t_{CLCL}-25	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-80	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold After $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float After $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-30		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-120		ns
t_{WHQX}	Data Hold After $\overline{WR}$	33		t_{CLCL}-25		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-25	t_{CLCL}+25	ns

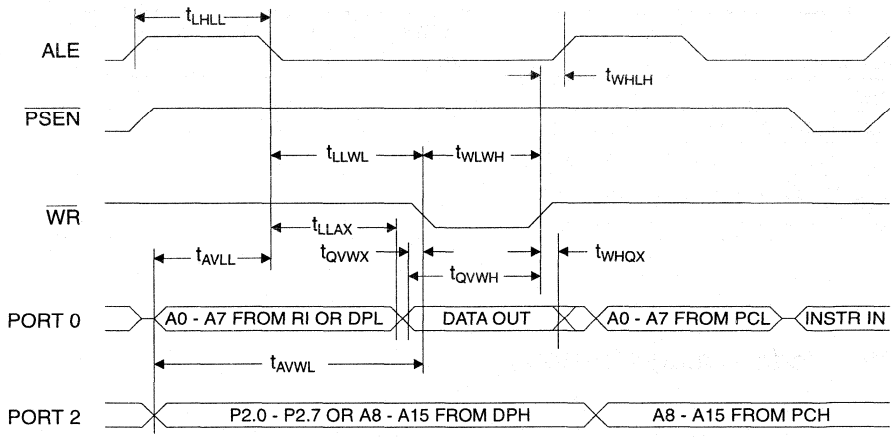
External Program Memory Read Cycle



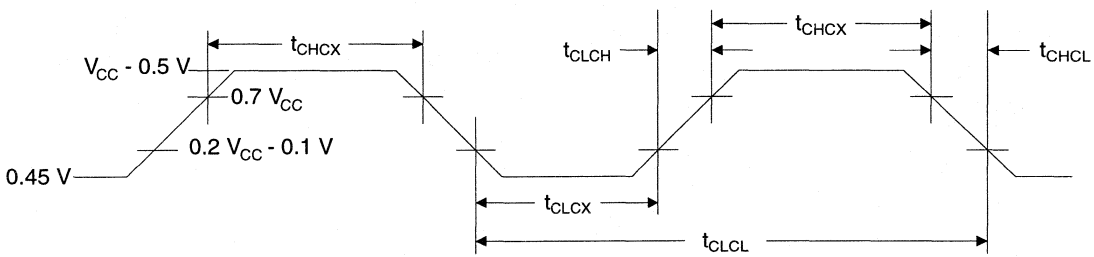
External Data Memory Read Cycle



External Data Memory Cycle



External Clock Drive Waveforms



External Clock Drive

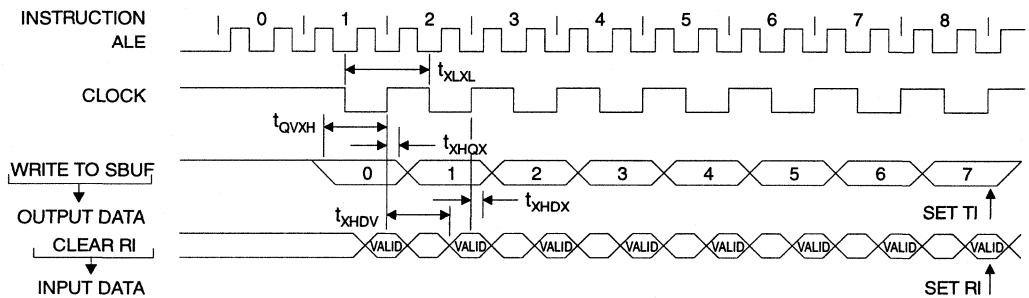
Symbol	Parameter	Min	Max	Units
1/t_{CLCL}	Oscillator Frequency	0	12	MHz
t_{CLCL}	Clock Period	83.3		ns
t_{CHCX}	High Time	20		ns
t_{CLCX}	Low Time	20		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

Serial Port Timing: Shift Register Mode Test Conditions

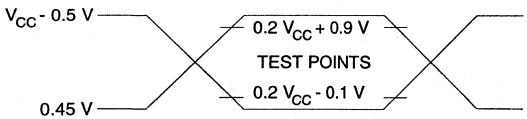
The values in this table are valid for V_{CC} = 5.0 V ± 20% and Load Capacitance = 80 pF.

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold After Clock Rising Edge	50		$2t_{CLCL}$-33		ns
t_{XHDX}	Input Data Hold After Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

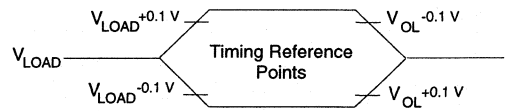


AC Testing Input/Output Waveforms [1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5 V for a logic 1 and 0.45 V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms [1]

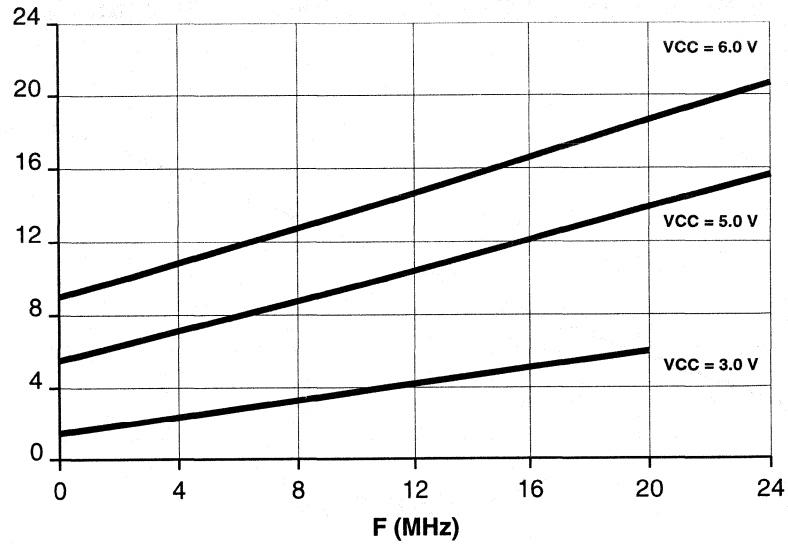


Note: 1. For timing purposes, a port pin is no longer floating when a 100-mV change from load voltage occurs. A port pin begins to float when a 100-mV change from the loaded V_{OH}/V_{OL} level occurs.

AT89LV52

ICC (mA)

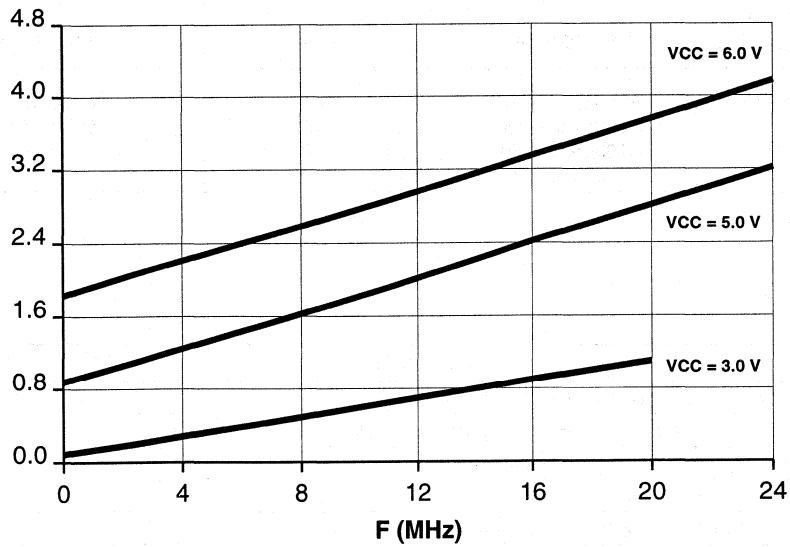
TYPICAL ICC (ACTIVE) at 25° C



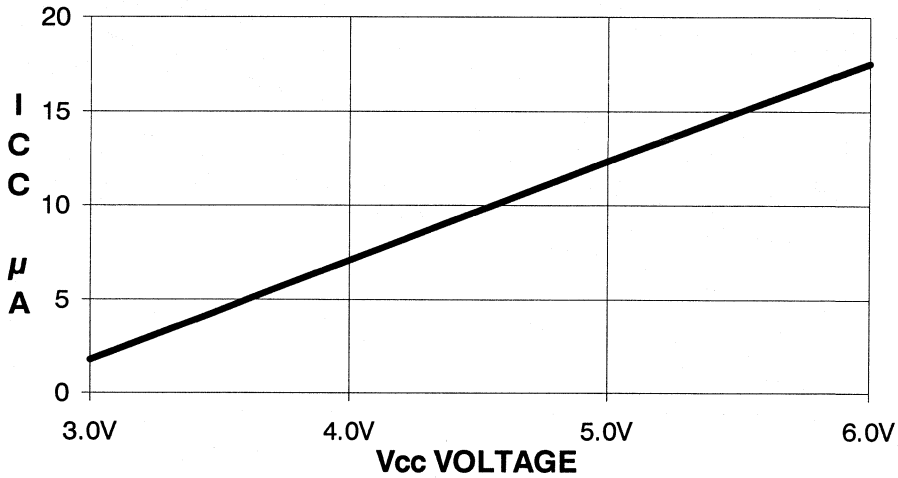
AT89LV52

ICC (mA)

TYPICAL ICC (IDLE) at 25° C



TYPICAL ICC vs. VOLTAGE - POWER DOWN (85°C)



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7 V to 6 V	AT89LV52-12AC AT89LV52-12JC AT89LV52-12PC AT89LV52-12QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
40D6	40 Lead, 0.600" Wide, Non-Windowed, Ceramic Dual Inline Package (Cerdip)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
44L	44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

Features

- Compatible with MCS-51™ Products
- 8 Kbytes of In-System Reprogrammable Downloadable Flash Memory
  - SPI Serial Interface for Program Downloading
  - Endurance: 1,000 Write/Erase Cycles
- 2 Kbytes EEPROM
  - Endurance: 100,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-Level Program Memory Lock
- 256 x 8-Bit Internal RAM
- 32 Programmable I/O Lines
- Three 16-Bit Timer/Counters
- Nine Interrupt Sources
- Programmable UART Serial Channel
- SPI Serial Interface
- Low Power Idle and Power Down Modes
- Interrupt Recovery From Power Down
- Programmable Watchdog Timer
- Dual Data Pointer

8-Bit Microcontroller with 8 Kbytes Downloadable Flash

3

Advance Information

Description

The AT89S8252 is a low-power, high-performance CMOS 8-bit microcomputer with 8 Kbytes of Downloadable Flash programmable and erasable read only memory and 2 Kbytes of EEPROM. The device is manufactured using Atmel's high density nonvolatile memory technology and is compatible with the industry standard 80C51 instruction set and pinout. The on-chip Downloadable Flash allows the program memory to be reprogrammed in-system through an SPI serial interface or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Downloadable Flash on a monolithic chip, the Atmel AT89S8252 is a powerful microcomputer which provides a highly flexible and cost effective solution to many embedded control applications.

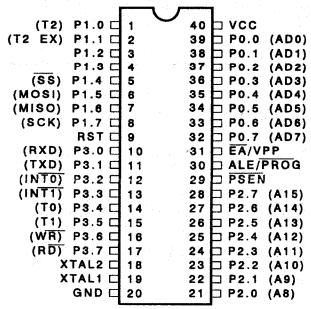
The AT89S8252 provides the following standard features: 8 Kbytes of Downloadable Flash, 2 Kbytes extended endurance EEPROM, 256 bytes of RAM, 32 I/O lines, programmable watchdog timer, two Data Pointers, three 16-bit timer/counters, a seven-vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator, and clock circuitry. In addition, the AT89S8252 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port, and interrupt system to continue functioning. The Power Down Mode saves the RAM contents but freezes the oscillator, disabling all other chip functions until the next interrupt or hardware reset.

The Downloadable Flash can be changed a single byte at a time and is accessible through the SPI serial interface. Holding RESET active forces the SPI bus into a slave input mode and allows the program memory to be Written-from or Read-to unless Lock Bit 3 has been activated.

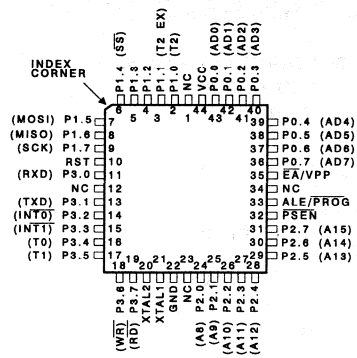
0401A

Pin Configurations

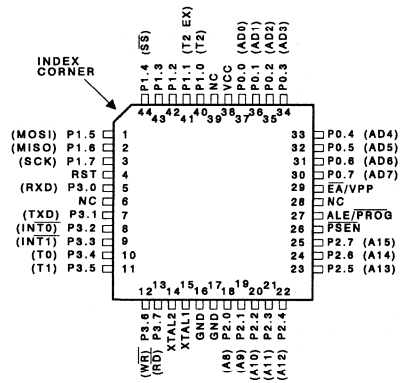
PDIP/Cerdip



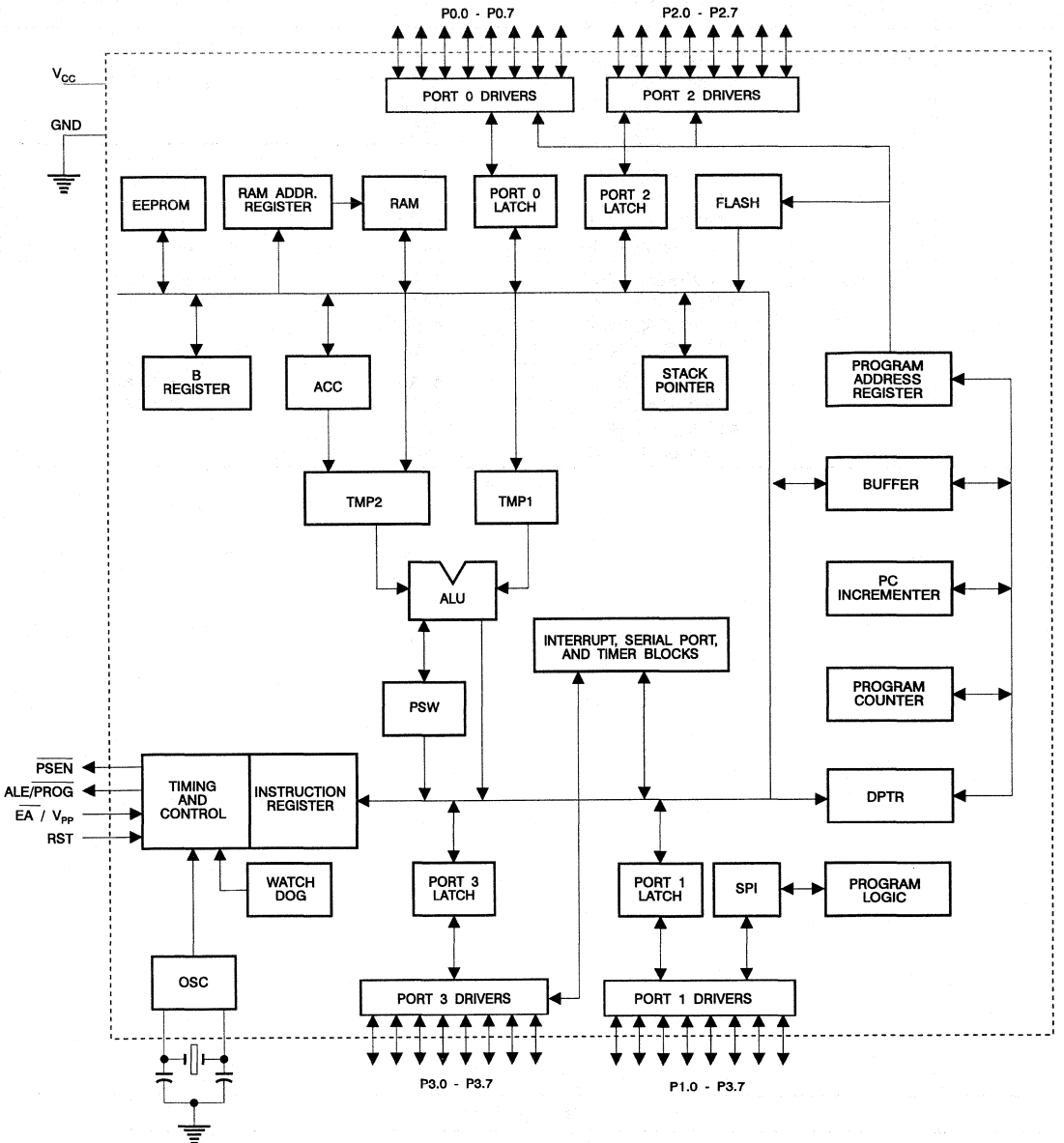
PLCC/LCC



PQFP/TQFP



Block Diagram



Ordering Information

Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	2.7 V ± 10%	AT89S8252-12AC AT89S8252-12JC AT89S8252-12PC AT89S8252-12QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89S8252-12AI AT89S8252-12JI AT89S8252-12PI AT89S8252-12QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)
24	5 V ± 20%	AT89S8252-24AC AT89S8252-24JC AT89S8252-24PC AT89S8252-24QC	44A 44J 40P6 44Q	Commercial (0°C to 70°C)
		AT89S8252-24AI AT89S8252-24JI AT89S8252-24PI AT89S8252-24QI	44A 44J 40P6 44Q	Industrial (-40°C to 85°C)

Package Type	
44A	44 Lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
40P6	40 Lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44 Lead, Plastic Gull Wing Quad Flatpack (PQFP)

Contents

Section 4 Microcontroller Application Notes

Using a Personal Computer to Program the AT89C51/C52/LV51/LV52/C1051/C2051

Introduction

This application note describes a personal computer-based programmer for the AT89C51/C52/LV51/LV52/C1051/C2051 Flash-based Microcontrollers. The programmer supports all flash memory microcontroller functions, including code read, code write, chip erase, signature read, and lock bit write. When used with the AT89C51/C52/LV51/LV52, code write, chip erase, and lock bit write may be performed at either five or twelve volts, as required by the device.

Devices sporting a "-5" suffix are intended for operation at five volts, while devices lacking the suffix operate at the standard twelve volts.

The programmer connects to an IBM PC-compatible host computer through one of the host's parallel ports. Required operating voltages are produced by an integral power supply and external, wall-mounted transformer.

Software

Software for the programmer is available by downloading it from the Atmel BBS at 408-436-4309.

The programmer is controlled by software running on the host. The AT89C51/C52 and C1051/C2051 have dedicated control programs, which were written in Microsoft C. Programs dedicated to the AT89LV51/LV52 do not exist; these devices are supported by the programs for the AT89C51/C52, respectively. In the text below, all references to the AT89C51/C52 may be assumed to apply to the AT89LV51/LV52 as well.

All programmer control programs are invoked from the DOS command line by entering the program name followed by "LPT1" or "LPT2" to specify parallel port one or two, respectively. If the parallel port is not specified, the program will respond with an error message. The control programs are menu-driven, and provide the following functions:

Chip Erase

Clear code memory to all ones. The successful operation of this function is not automatically verified.

Program from File

Write the contents of the specified file into device memory. The user is prompted for the file name, which may require path and extension.

The file is expected to contain binary data; hex files are not accepted. The first byte in the file is programmed into the first location in the device. Successive bytes are programmed into successive locations until the last location in the device has been programmed or until the data in the file has been exhausted.

Programming occurs regardless of the existing contents of device memory; a blank check is not automatically performed. After programming, the contents of device memory are not automatically verified against the file data.

Each programmed location in the device receives the maximum programming time specified in the data sheet. This is done because timing is enforced by software; the programming status information provided by DATA* polling and RDY/BSY* is not utilized.

The control program provides no visual indication that programming is in progress. The main menu is redisplayed when programming is complete.

Verify against File

Compare the contents of code memory against the contents of the specified file. The user is prompted for the file name, which may require path and extension.

The file is expected to contain binary data; hex files are not accepted. The first byte in the file is compared to the first location in the device. Successive bytes are compared to successive locations until the last location in the device has been compared or until the data in the file has been exhausted.

Locations which fail to compare are displayed by address, with the expected and actual byte contents. If there are no compare failures, nothing is displayed.

Save to File

Copy the contents of device memory to the specified file. The user is prompted for the file name, which may require path and extension. The number of bytes in the resulting file is the same as the number of memory locations in the device.

Blank Check

Verify that the contents of device memory are all ones. Only pass or fail is reported; the addresses and contents of failing locations are not displayed.

Read Signature

Read and display the contents of the signature bytes. The number of signature bytes and their expected contents varies between devices. Refer to the device data sheet for additional information.

Write Lock Bit 1
Write Lock Bit 2
Write Lock Bit 3

Set the indicated lock bit. Note that the AT89C1051/C2051 contain only two lock bits, while the AT89C51/LV51 and AT89C52/LV52 contain three lock bits. The state of the lock bits cannot be verified by direct observation.

Exit

Quit the programmer control program.

System Dependency

The control programs for the AT89C51 and AT89C52 come in two flavors: host system-dependent and host system-independent. System-dependency results from the use of software timing loops to enforce required delays, the duration of which will vary between host systems running at different speeds. The code provided was tested on an 80386-based system running at 33 MHz, and may require modification for use on other systems. This method was chosen for its simplicity.

Host system-independence is achieved by using the Programmable Interval Timer embedded in the system hardware to enforce time delays independent of system speed. The timer is reconfigured when the control program is invoked and restored to its original state before the program terminates. In order to guarantee that the program is not exited before the timer configuration is restored, the CTRL-C and CTRL-BREAK keys are disabled. This means that the program cannot be aborted except by specifying the exit option at the main menu or by rebooting the system.

The timer control code is provided as an 8086 assembly language module, which is linked with the compiled control program. The granularity of the timer is 0.838 microseconds, but the minimum practical delay is system- and software-dependent. The timer code ensures that the delay produced will not be of shorter duration than requested.

The control programs provided for the AT89C1051/C2051 are system independent.

Programmer

The programmer circuitry (see Figures 1 and 2) consists of the host interface and switchable power supplies. The signal sequencing and timing required for programming is generated by the host under software control. A 40-pin ZIF socket is provided for programming the AT89C51/C52; the 20-pin ZIF socket accommodates the AT89C1051/C2051. Note that the power and ground connections and bypass capacitors required by the TTL devices are not shown on the schematic.

Power for the programmer circuitry and the AT89C51/C52/C1051/C2051 is provided by a fixed five volt supply. A second supply provides either five or twelve volts, selectable, for use during programming. The addition of a transistor to the output of the variable supply provides a third level, ground, for use when programming the AT89C1051/C2051.

The resistor values utilized in the variable power supply circuit were determined using the equations presented in the LM317 voltage regulator data sheet. Power supply ramp rates are accommodated by the host software. For 5 V-Vpp programming, the devices must be ordered from the factory as an AT89CXXX-5 (not available with the AT89C1051/2051).

The programmer is connected to the host with a 25-conductor ribbon cable. To minimize the effect on signal integrity, the length of the cable should be as short as possible, preferably not exceeding three feet.

Parallel Interface

The original parallel interface provided by IBM was probably not intended to support bidirectional data transfers. However, due to the way in which the interface was implemented, bidirectional transfers are possible. Over the years, many products have appeared which exploit this capability.

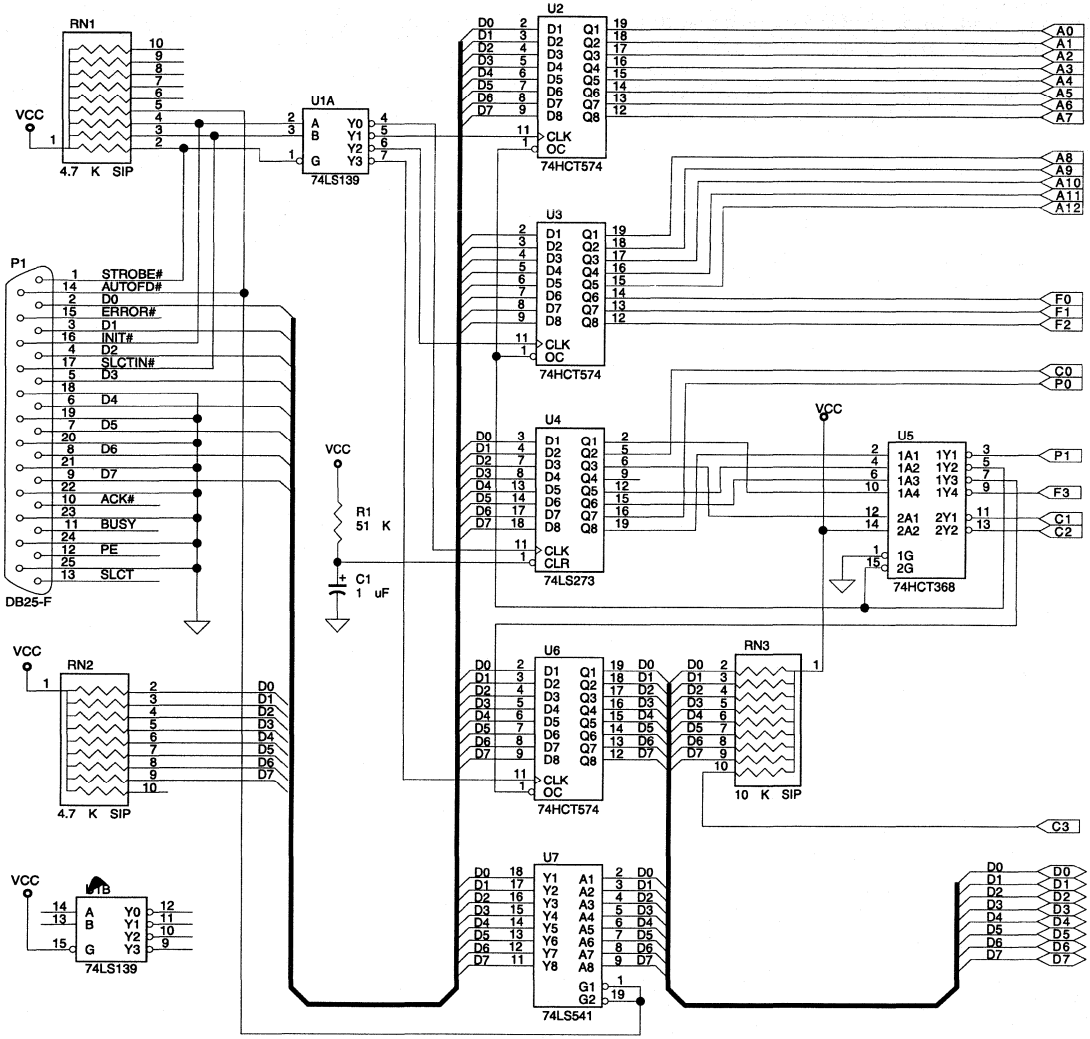
Unfortunately, many system and interface card manufacturers have not faithfully cloned the IBM design, resulting in bus contention when the peripheral attempts to drive return data into the interface. Usually the peripheral drivers can overpower the interface drivers and the peripheral works, though this is not considered a good design practice.

Most parallel interfaces are now implemented in a single chip, such as the 82C411 or 16C452. These chips allow their output drivers to be disabled under software control, providing true bidirectional operation. The programmer software automatically enables bidirectional operation when used with parallel interfaces utilizing the 82C411, 16C452, or similar chips.

Note that these chips also possess a mode control pin which must be at the correct level to enable the directional control feature. As a result, parallel interfaces utilizing these chips cannot be assumed to be bidirectional.

If the programmer writes devices, but fails to verify, or the signal levels at the interface don't meet TTL specifications, the parallel interface may be incompatible with the programmer. A design is provided (see Figure 1) for a parallel interface which supports bidirectional operation and is compatible with the programmer. This design is simple, requiring only six ICs. The interface can be strapped to appear as LPT1 (addresses 378-37F hex) or LPT2 (278-27F hex) and will be recognized by the POST when the host system is powered up. Due to its simplicity, the parallel interface cannot be used as a printer interface.

Figure 1. AT89C Series Programmer Interface



Note: 0.1 µF bypass caps on all ICs.

Figure 2. Power Supply for AT89C Series Programmer

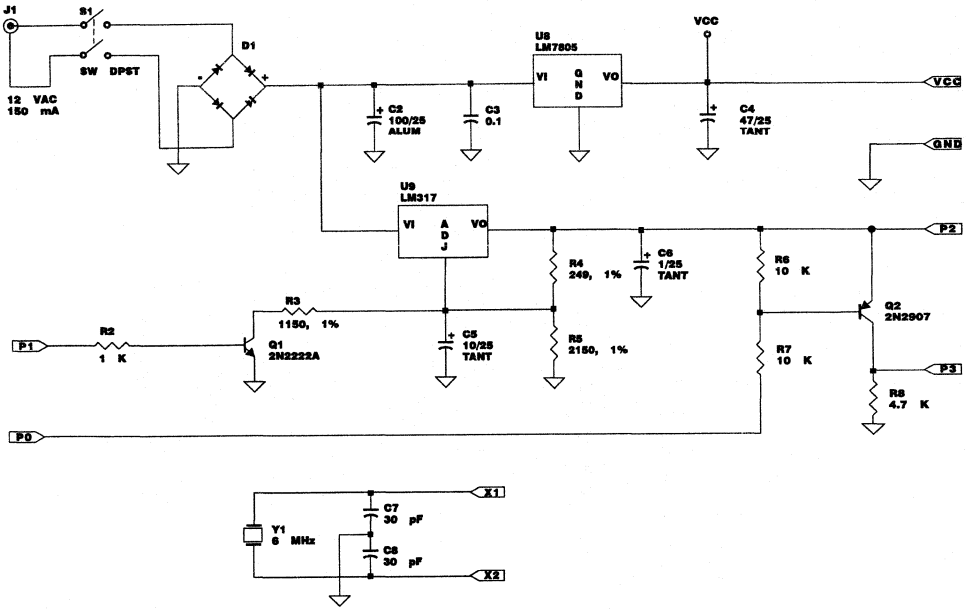
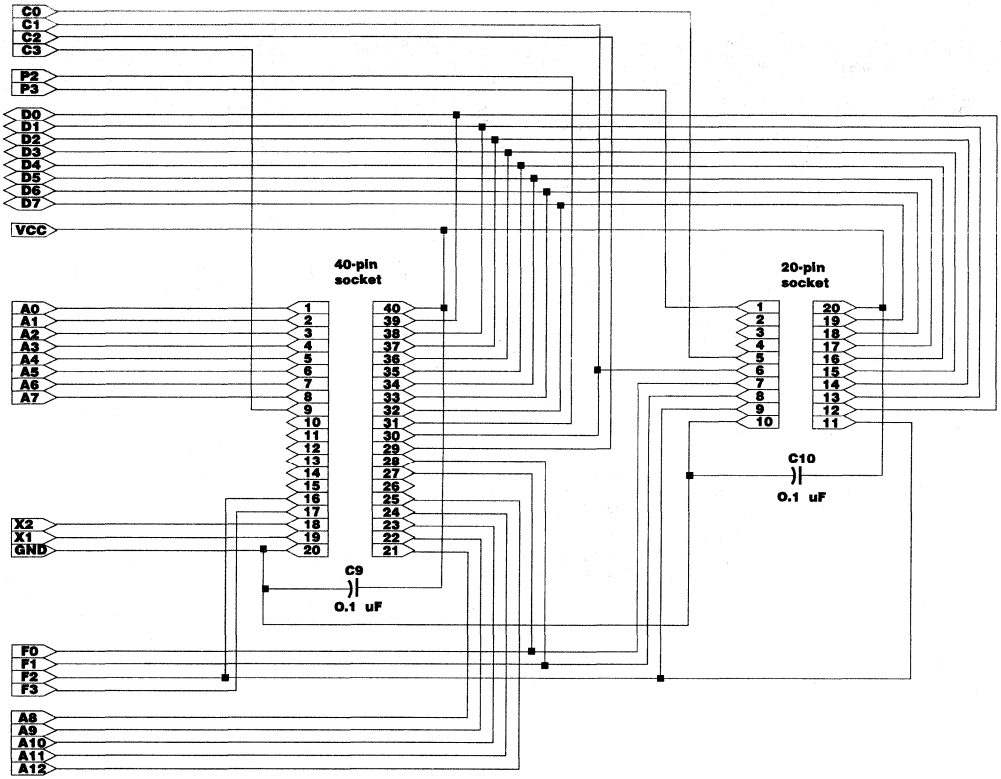


Figure 3. AT89C Series Programmer Socket Wiring



AT89C51 In-Circuit Programming

8-Bit Microcontroller with Flash

Application Note

4

This application note illustrates the in-circuit programmability of the Atmel AT89C51 Flash-based microcontroller. Guidelines for the addition of in-circuit programmability to AT89C51 applications are presented along with an application example and the modifications to it required to support in-circuit programming. A method is then shown by which the AT89C51 microcontroller in the application can be reprogrammed remotely, over a commercial telephone line. The circuitry described in this application note supports five volt programming only, requiring the use of an AT89C51-XX-5. The standard AT89C51 requires 12 volts for programming.

The software for this application may be obtained by downloading from Atmel's BBS: (408) 436-4309.

General Considerations

Circuitry added to support AT89C51 in-circuit programming should appear transparent to the application when programming is not taking place.

EA#/VPP must be held high during programming. In applications which do not utilize external program memory, this pin may be permanently strapped to V_{CC}. Applications utilizing external program memory require that this pin be held low during normal operation.

RST must be held active during programming. A means must be provided for overriding the application reset circuit, which typically asserts RST only briefly after power is applied.

PSEN# must be held low during programming, but must not be driven during normal operation.

ALE/PROG# is pulsed low during programming, but must not be driven during normal operation.

During programming, AT89C51 I/O ports are used for the application of mode select, addresses and data, possibly requiring that the controller be isolated from the application circuitry. How this is done is application dependent and will be addressed here only in general terms.

Port Used for Input

During programming, the controller must be isolated from signals sourced by the application circuitry. A buffer with three-state outputs might be inserted between the application circuitry and the controller, with the buffer outputs three-stated when programming is enabled. Alternately, a multiplexer might be used to select between signal sources, with signals applied to the controller by either the application circuitry or the programmer circuitry.

Port Used for Output

No circuit changes are required if the application circuitry can tolerate the state changes which occur at the port during programming. If the prior state of the application circuitry must be maintained during programming, a latch might be inserted between the controller and the application circuitry. The latch is enabled during programming, preserving the state of the application circuitry.

An Application Example

The AT89C51 application shown in Figure 1 is an implementation of a moving display. This application was selected for its simplicity and ability to show graphically the results of in-circuit reprogramming. The text to be displayed is programmed into the controller as part of its firmware, and cannot be changed without reprogramming the device.

The displayed text is presented in one of two modes selected by the four-position DIP switch. In the first mode, one character at a time enters the display from the right and moves quickly to the left through each element of the display to its final position in the assembled message. In the second mode, the message moves through the display, from right to left, with the display acting as a window onto the message. This mode is familiar as the method often used in displays of stock prices.

The output consists of four DL1414T, four-digit, 17-segment alphanumeric displays with integral decoders and drivers. This yields 16 total display elements, each capable of displaying digits 0-9, the upper case alphabet, and some punctuation characters. The displayable character codes are ASCII 20H-5FH.

A power-on reset circuit and six megahertz crystal oscillator complete the application. Neither external program memory nor external data memory is used.

Modifications to the Application to Support In-Circuit Programming

Figure 2 shows the application modified for in-circuit programming.

It is assumed that the programmer, when inactive, will neither drive nor excessively load the application.

Since the application does not use external program memory, EA#/VPP on the controller is connected to V_{CC}. This meets the requirement for programming.

The reset circuit has been modified by the addition of two transistors, which allow RST on the controller to be forced high by the programmer.

PSEN# and ALE/PROG#, unused in the basic application, are under the direct control of the programmer.

Programming requires programmer access to all of the four AT89C51 I/O ports, as documented in the data sheet. The programmer is connected directly to those controller pins which are unused by the application, while access to pins used by the application requires special treatment, as explained in the following paragraphs.

The least significant four bits of the address generated by the programmer are multiplexed onto port one of the controller with the data from the DIP switch. Note that the four resistors added at the switch are not required in the basic application, since the AT89C51 provides internal pull-ups on port one.

During the normal operation of the application, controller ports zero and two provide data and control signals (respectively) to the displays. During programming and program verification, the programmer asserts control of port zero and part of port two. The programmer is connected to ports zero and two without buffering, since, when inactive, its presence does not affect the normal operation of the application.

A transparent latch has been added between port two of the controller and the display control inputs. The latch holds the display control signals inactive during programming, which eliminates erratic operation of the displays due to programmer activity on ports zero and two. No isolation of the display data inputs is required, since data applied to the inputs is ignored when the control signals are inactive.

The AT89C51 reset circuit, input multiplexer and output latch are controlled by a single signal generated by the programmer. During programming, reset is asserted, the multiplexer switches inputs, and the latch freezes the display control lines.

To ensure that the display control lines are in a known state before they are latched, an AT89C51 external interrupt is used to allow the programmer to signal the application before asserting reset. The application firmware responds to the interrupt by displaying a message and deactivating the display control lines.

After programming, when reset is deasserted, the controller ports are high as the latch becomes transparent. Since the display control inputs are inactive high, the display contents are not disturbed until the new program writes the display.

Although not essential to this application, it might be imperative in some applications that the state of the peripheral circuitry not be disturbed during programming.

The Programmer

The programmer (Figure 3) generates the addresses, data and control signals necessary to program the AT89C51 embedded in the application.

The programmer circuitry consists of an AT89C51 and an RS-232 level translator. The controller runs at 11.0592 megahertz, which allows the serial port to operate at a number of standard baud rates. A Maxim MAX232 line driver/receiver produces RS-232 levels at the serial interface while requiring only a five volt supply.

Many of the signals generated by the programmer are connected directly, without buffering, to the AT89C51 in the application. These signals, when inactive, are not three-stated, but are pulled high. The AT89C51 has internal pull-ups of approximately three KOhms on ports one, two and three. Because port zero does not have internal pull-ups, external pull-ups of ten KOhms have been added to permit proper operation of program verification mode. The sample application operates correctly in this environment. If required for compatibility with an application, programmer signals may be buffered with three-state buffers similar to the 74xx125.

The AT89C51 in the programmer does not utilize external program or data memory, which would require sacrificing needed I/O pins. This requires that program code and I/O buffers be kept small enough to fit in on-chip memory.

Remote Programming Over a Commercial Telephone Line

The programmer and display application described previously are connected to a phone line via a modem at a remote site. Using a personal computer with a modem, a user can upload a new program containing a new message, which is programmed into the AT89C51 embedded in the application. When programming is complete, the application executes the new program, which displays the new message.

Local Station

The local station in the test configuration consists of an IBM PC AT-class computer connected to a Hayes-compatible, Prometheus 1200 baud modem. The modem was selected because it was inexpensive and available. A faster modem may be used if desired, although once the file transmission time is reduced below one minute, further reductions in transmission time do not further reduce connect time charges. A possible advantage to higher transmission speeds is the automatic error detection and correction available in some high speed modems.

Procomm Plus version 2.01, a commercial data communications package, is used to configure the modem, set up commu-

nications parameters, and establish a link with the remote modem. Procomm Plus includes a macro language called ASPECT, which allows the user to write and compile scripts which implement custom file transfer protocols. A simple ASPECT script was written to read the contents of a program file and upload it to the remote programmer.

The file transfer protocol (FTP) implemented is a simple send-and-wait, packet-oriented protocol. The transmit and receive modes of the FTP are illustrated by the flowcharts in figures 4 and 5, respectively. The transmitter sends each packet without flow control and waits for a response. The programmer (the receiver) reads and dissects the packet while calculating a checksum. If the calculated checksum is valid, the programmer acknowledges the packet by sending an ACK. If the checksum is in error, the programmer negatively acknowledges the packet by sending a NAK. Upon receipt of an ACK, the transmitter sends the next packet. If the transmitter receives a NAK, it resends the same packet. Transmission proceeds in this manner until the entire file has been transferred.

The programmer might respond to a packet by sending a CAN, which indicates that a non-recoverable error has occurred and that the transmitter should immediately abort the file transfer. If the programmer fails to respond to a packet within a limited period of time, the transmitter will resend the same packet. The transmitter will continue to resend the same packet until a valid response is received or until the allowed number of attempts is exceeded, at which time the file transfer is aborted.

After each packet is received and validated by the programmer, the data contained in the packet is programmed into the AT89C51 controller in the application. After programming, the data is read back from the controller and verified against the received packet data. Successful verification indicates successful programming, causing the programmer to send ACK to the transmitter. If programming fails, the programmer sends CAN to signal the transmitter to abort the file transfer.

The simplicity of the FTP reduces the amount of AT89C51 program memory used in the programmer. The send-and-wait nature of the FTP allows inter-packet delays due to AT89C51 program and erase times to be easily absorbed. Support for program verification is transparent, requiring no explicit command or result codes, or additional data transfers.

The files which are uploaded to the programmer are created with the tools in the Intel MCS-51 Software Development Package for the IBM PC. Included in the package are the MCS-51 Macro Assembler, MCS-51 Relocator and Linker, and a useful utility, OH. OH converts an absolute 8051 object file to an equivalent ASCII hexadecimal object file.

The records in the hex file produced by the OH utility serve, unchanged, as the packets in the FTP described above; no service fields need to be added. The colon which begins each record serves as the packet signature field. The load address field serves as the packet sequence number. A checksum is provided as the last field in each record. Since seven-bit ASCII coding is utilized, the eighth bit of each byte is available to be used for parity checking.

Because the AT89C51 in the programmer does not utilize external data memory, necessary packet buffering must be done using internal RAM. Limited memory precludes the use of conventional FTPs which utilize packets of 128 bytes and larger. The hex packet format used in this application limits packet data fields to 16 or fewer entries, requiring little memory for buffering.

The ready availability of a utility for creating the packetized program file, combined with small packet size and adequate error checking, makes the hex packet format a near ideal solution for this application. A disadvantage is the use of ASCII, which requires each program data byte to be expressed as two hex characters. This demands that nearly twice as many bytes be transferred as might otherwise be required. This is not a severe limitation, however, since typical file transfer times are less than one minute. Overall, the simplicity of the custom FTP/hex packet format implementation outweighs the drawbacks.

Remote Station

The remote station in the test configuration consists of the display application and programmer circuits, described previously, connected to a Hayes-compatible, Prometheus 1200 baud modem. During normal operation, the application executes its internal program while the modem and programmer monitor the phone line for incoming calls.

After a call has been detected and a connection established, the programmer forces the application to suspend execution of its program. The new program is then downloaded and programmed into the AT89C51 embedded in the application. When programming is complete, the application is allowed to begin execution of its new program, and the programmer returns to monitoring the phone line for the next call.

The programmer powers up with its programming control outputs inactive, allowing the application to run normally. After configuring the modem to answer incoming calls, the programmer puts itself to sleep. The programmer will not disturb the application until a new program is to be downloaded.

The programmer controls the modem by sending ASCII command strings over the serial interface, to which the modem responds with Hayes-style ASCII numeric codes. The software is designed for use with Hayes-compatible modems, which includes the Prometheus ProModem 1200 used here.

The serial interface, through which the programmer connects to the modem, supports two handshaking signals, DTR and DSR. On power up, the programmer asserts DTR, to which the modem responds by asserting DSR. If the modem should fail to respond to any command, including the command to hang up, the programmer deasserts DTR, which forces the modem to drop the line.

The modem monitors the phone line while the programmer sleeps, waiting for an incoming call. When a call is detected, the modem answers and attempts to establish communication with the caller. If a connection is established, the modem sends a code to the programmer, waking it up. The programmer verifies the connect code and begins polling for a valid packet header.

Incoming packets must arrive fewer than thirty seconds apart, or the modem drops the line (hangs up) and the programmer returns to sleep, waiting for the next call. If the caller hangs up, the thirty second period must expire before another call will be answered. Calls incoming during the reset delay period are ignored.

If a valid packet header is received prior to the expiration of the reset delay period, the programmer will attempt to read and validate the incoming packet. At any time during packet reception, an invalid character, parity error or time-out during character reception will cause the partial packet to be declared invalid and discarded.

Two packet types are defined: data and end-of-file. A data packet contains five fields in addition to the packet header, one of which is a variable length data field. The data field contains program data to be written into the AT89C51 controller in the application. The load address field contains the address at which the data is to be written. The end-of-file packet contains the same fields as the data packet, except that the data field is empty. This packet type has special meaning to the programmer, as explained below.

Any packet which contains an invalid record type, record length or checksum is invalid. Program data accumulated during the processing of an invalid packet is discarded. The programmer sends a NAK to the transmitter to signal reception of an invalid packet and resumes polling for a valid packet header.

Receipt of the first valid data packet causes the programmer to interrupt the application controller. The controller responds to the interrupt by abandoning execution of its usual program and displaying a message indicating that programming is taking place. If this is the first valid data packet since power was applied or an end-of-file packet was received, the programmer asserts the control signals necessary to erase the program memory in the application controller. The programmer then places the controller in programming mode.

The first and subsequent valid data packets are dissected as they are received and the data which they contain is programmed into the application controller at the address indicated in the packet load address field. After programming, the data is read back from the controller and verified against the received packet data. Successful verification indicates that programming was successful, causing the programmer to send ACK to the transmitter. The programmer then resumes polling for a valid packet header, subject to the thirty second reset delay.

If programming fails, the programmer sends CAN to signal the transmitter to abort the file transfer. The modem drops the line and the programmer returns to sleep, waiting for the next call. The application controller is left in programming mode, preventing it from executing the incomplete or invalid program which it contains.

It is important to note that invalid packets are NEVER programmed into the application controller. To do so would require that the program memory in the controller be completely erased before the error could be corrected, causing the non-recoverable loss of all previous program data.

Upon receipt of an end-of-file packet, the programmer returns its control outputs to the inactive, power on state, allowing the application controller to begin execution of the new program. The programmer then resumes polling for a valid packet header, subject to the thirty second reset delay.

If a valid packet is received prior to the expiration of the thirty second delay, another programming cycle begins, which can only be terminated by the reception of a valid end-of-file packet.

If the reset delay expires prior to the reception of a valid end-of-file packet, the modem will drop the line and the programmer will return to sleep, waiting for the next call. In this case, the application controller is left in programming mode, preventing it from executing its program. To return the application to normal operation, another call must be received, and a valid program file uploaded, terminated by an end-of-file packet.

Setting Up the Hardware

Local Station

Connect the IBM PC to the ProModem 1200 through one of the system COM ports. Connect the modem to an analog telephone line and set the modem switches as indicated below.

Switch settings:

1	ON
2	ON
3	OFF
4	ON
5	OFF
6	ON
7	OFF
8	OFF
9	OFF
10	OFF

Remote Station

Connect the display application/programmer to the second ProModem 1200 through the programmer serial port. Connect the modem to an analog telephone line and set the modem switches as indicated below.

Turn the modem on and apply power to the display application/programmer. The application will begin executing its program, if it contains one. The programmer will initialize the modem, as shown by the activity on the modem status indicators.

Switch settings:

1	ON
2	ON
3	ON
4	OFF
5	ON
6	ON
7	ON
8	OFF
9	OFF
10	OFF

Installing and Configuring Procomm Plus, Version 2.01

Install Procomm Plus as instructed in the User Manual. When prompted to specify the modem in use, select 'Prometheus ProModem 1200' from the list.

Run Procomm Plus and create a dialing directory entry for the remote station. The baud rate must be set to 1200, parity to EVEN, number of data bits to 7, number of stop bits to 1, plex to HALF.

Enter the Setup utility (ALT-S). Select 'PROTOCOL OPTIONS', then 'EXTERNAL PROTOCOL OPTIONS' from the menus and modify the entry for 'EXTERNAL PROTOCOL 1' as indicated below.

EXTERNAL PROTOCOL 1:

A - NAME:	<any name>
B - TYPE:	ASPECT
C - UPLOAD COMMAND:	ATX.ASX

NOTE: 'ATX.ASX.' is the filename of the compiled ASPECT script to be associated with External Protocol 1.

Save the changes and exit the Setup utility.

Creating a Hex File

The hex files which are uploaded to the programmer are created with the tools in the Intel MCS-51 Software Development Package for the IBM PC. In the example below, the 8051 assembler source file is called 'TEST.ASM'.

Assemble the source file 'TEST.ASM' and create the object file 'TEST.OBJ':

```
ASM51 TEST.ASM
```

Link and locate the object file 'TEST.OBJ' and create the absolute object file 'TEST.ABS':

```
RL51 TEST.OBJ TO TEST.ABS
```

Convert the absolute object file 'TEST.ABS' to the hex file 'TEST.HEX':

```
OH TEST.ABS TO TEST.HEX
```

The resulting file, 'TEST.HEX' is ready to be uploaded.

NOTE: ASM51 is version 2.3; RL51 is version 3.1; OH is version 1.1.

Uploading a Hex File

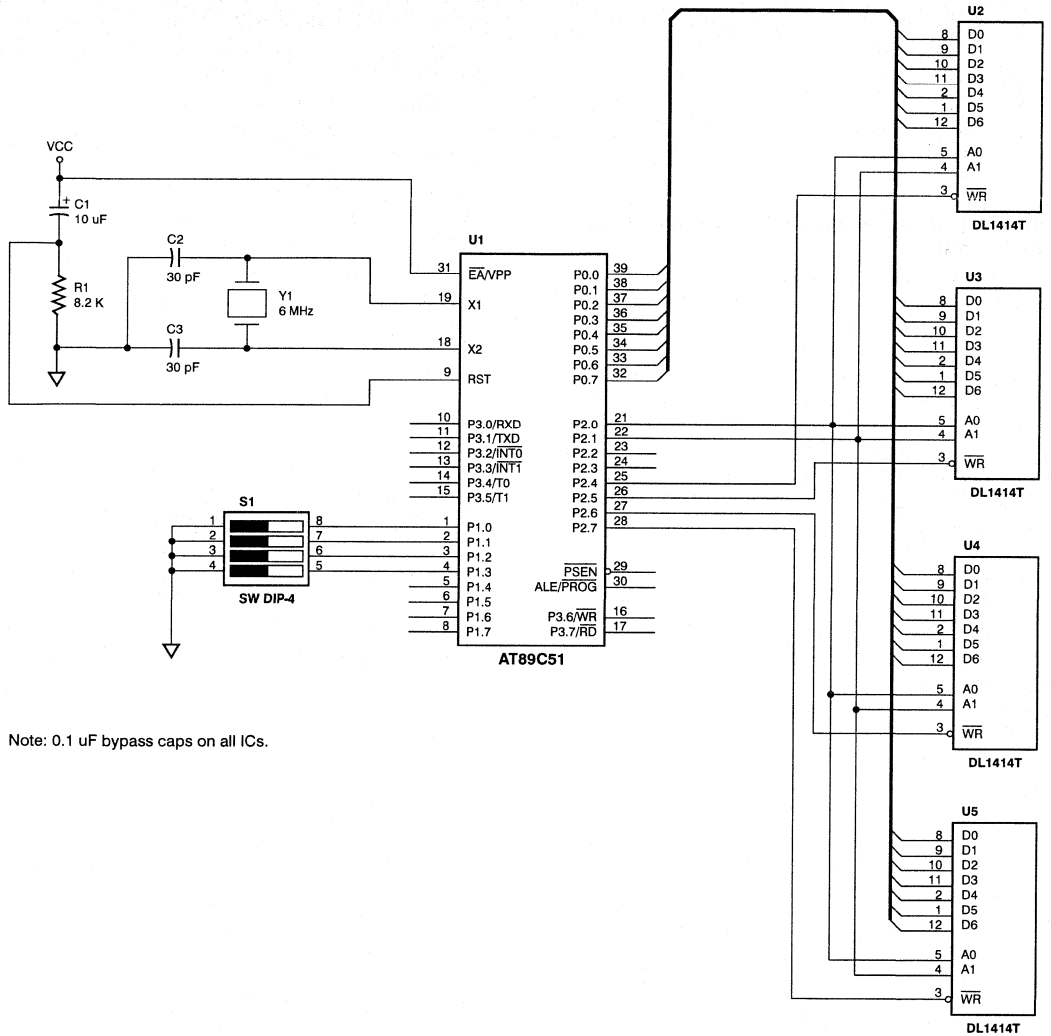
Run Procomm Plus and use the proper dialing directory entry to dial the remote station.

After the connection with the remote station is established, press the 'PgUp' key and select '1' (External Protocol 1) from the menu of upload protocols. This will execute the ASPECT script associated with External Protocol 1.

When prompted, enter the name of the file to be uploaded, including the extension and path, if required.

When the upload is complete, press ALT-H to hang up and ALT-X to exit Procomm Plus and return to DOS.

Figure 1. AT89C51 Moving Display Application Example



Note: 0.1 uF bypass caps on all ICs.

Figure 2. AT89C51 Moving Display Application Modified for In-Circuit Programming

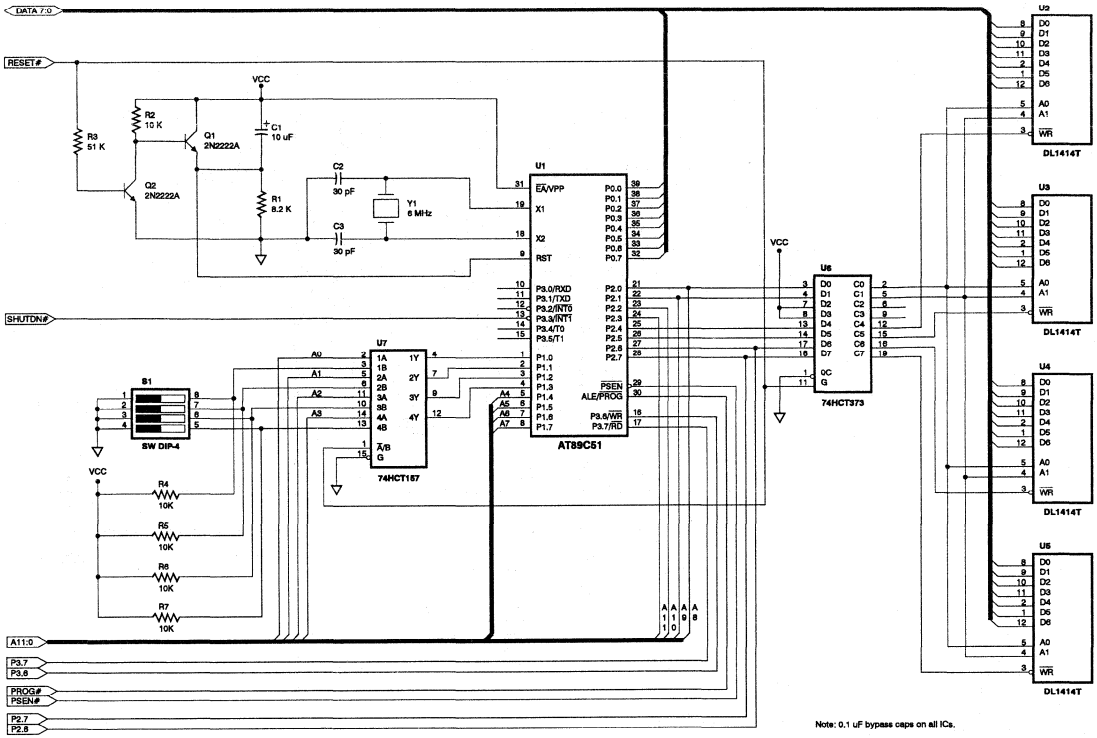


Figure 3. AT89C51 Programmer

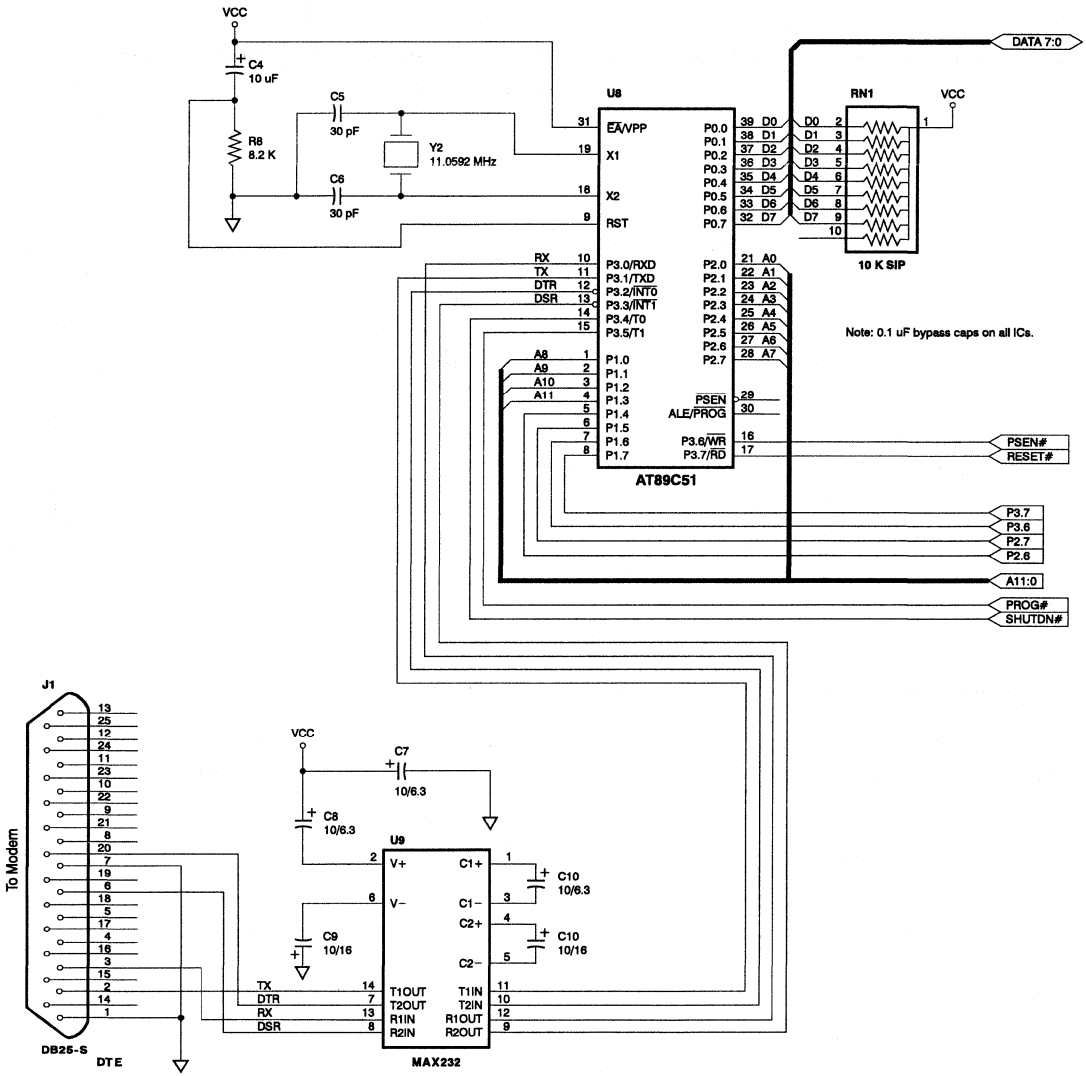


Figure 4. FTP Transmit Mode

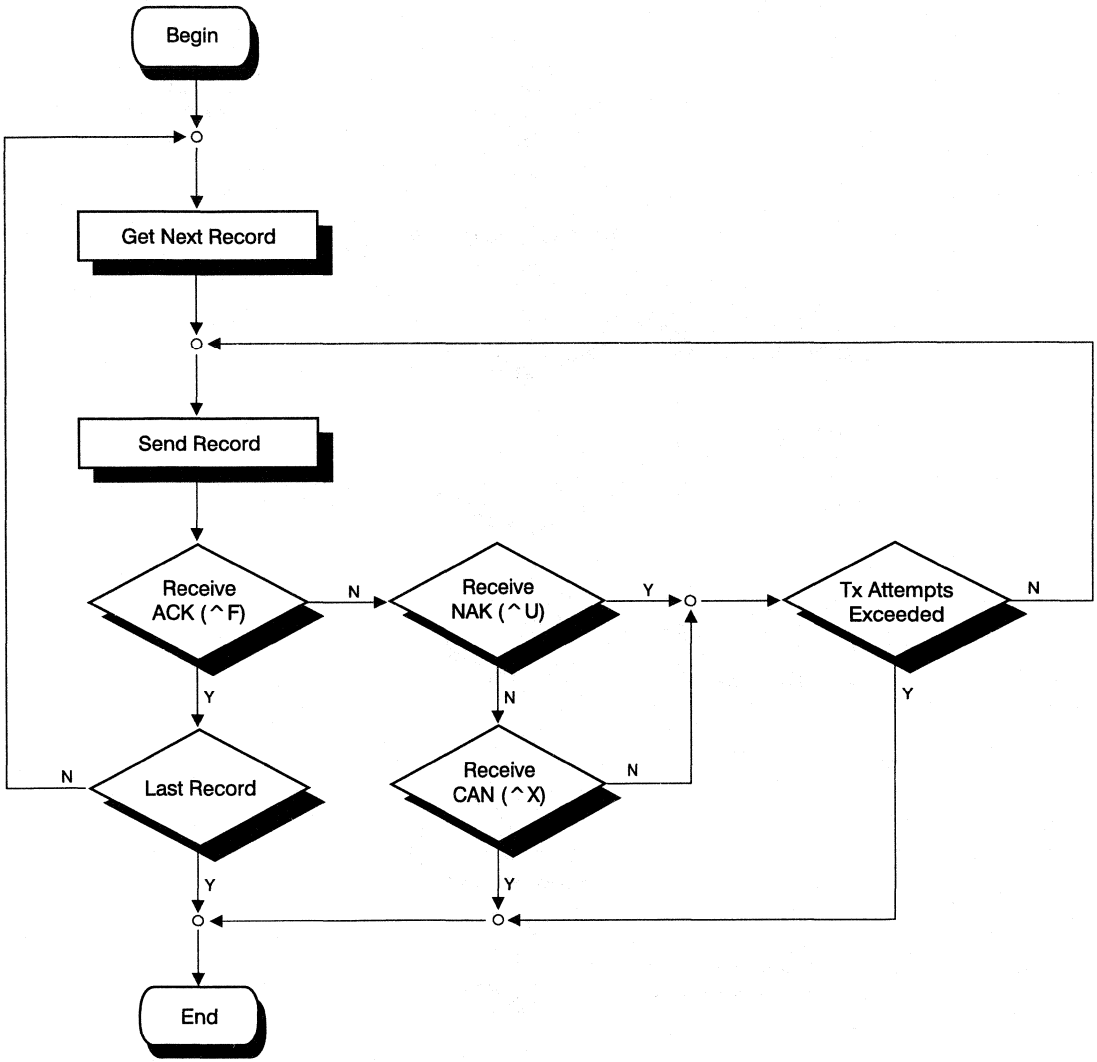
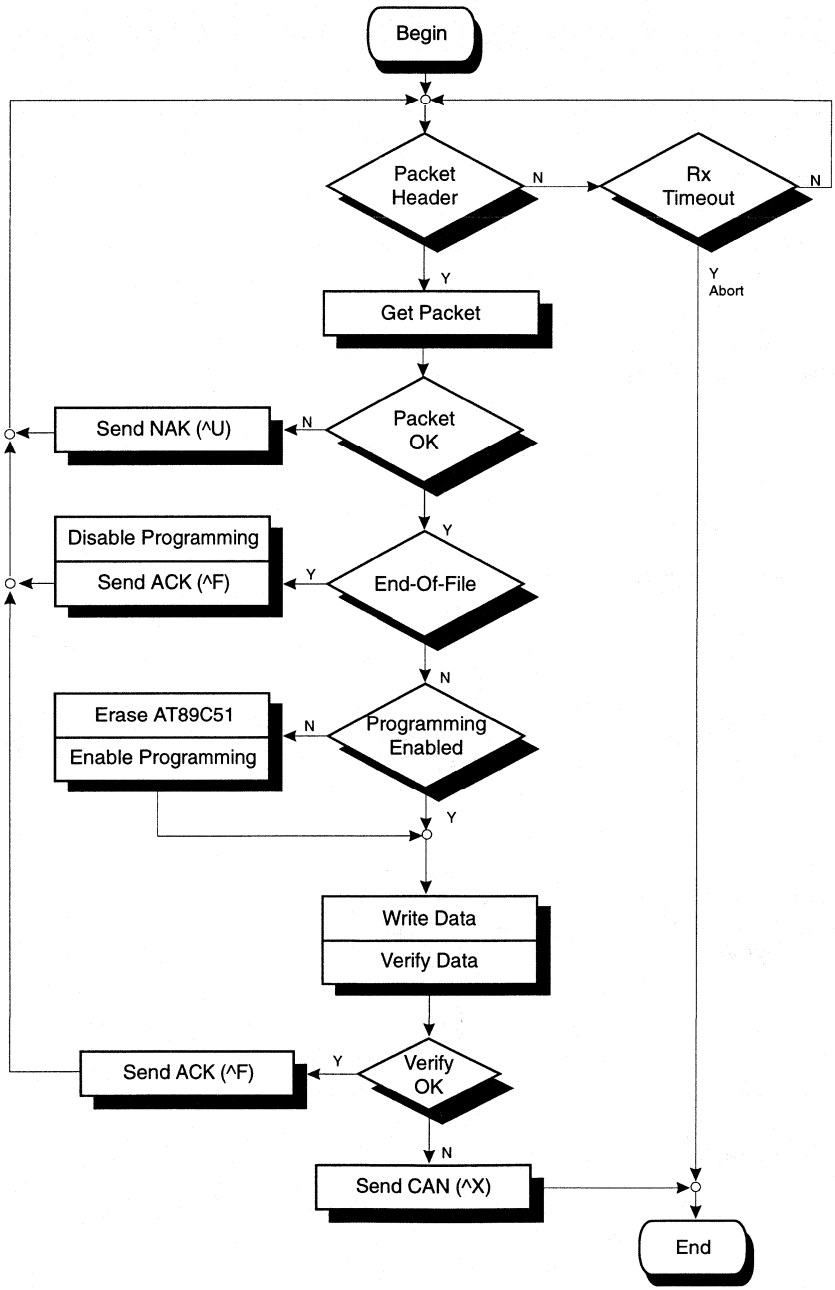


Figure 5. FTP Receive Mode



Appendix I: Intel Hex File Definition

Hexadecimal object file format (Intel hex) is produced by most 80C51 assembler products.

Each record in the file contains the following fields:

<:><rec length><load address><rec type><data><checksum>

The colon is the record header.

The record length field consists of two hex digits, and represents the number of entries in the data field. OH outputs records containing 16 or fewer data field entries.

The load address field consists of four hex digits, and indicates the absolute address at which the data in the data field is to be loaded.

The record type field consists of two hex digits, which are always zero in data records.

The data field contains from one to 16 pairs of hex digits.

The last two hex digits are a checksum on the record length, load address, record type, and data fields. The sum of the binary equivalents of these fields and the checksum itself is zero.

Each record in the file is terminated by a carriage return and line feed.

A type one record marks the end of the file. The record always contains the following value: ':00000001FF'.

Controlling FPGA Configuration with a Flash-Based Microcontroller

Introduction

SRAM-based FPGAs like the Atmel AT6000 series come more and more into use because of the many advantages they offer. Their reconfigurability allows the user to implement more gates in his application than the FPGA actually has, simply by loading the gates as needed into the FPGA. This is also called "Cache Logic™." For an efficient use of cache logic, the FPGA must meet the following requirements: partial reconfigurability, a fast reconfiguration process and full architectural symmetry.

The FPGA can control and change its configuration itself, but this can also be done in a very elegant way by a microcontroller. After the configuration process or in-between two configuration cycles it can be used for other purposes and is not lost for the application. The different options for space-saving realization, design protection or for fast, flexible reconfiguration are shown in this application note. The microcontroller used here is the Atmel AT89C51 which is fully compatible to the industry standard i8031.

Configuration Data Transfer between the FPGA and the Microcontroller

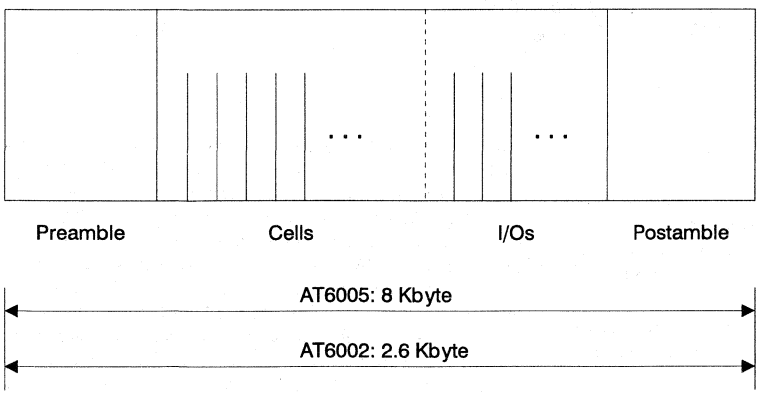
The amount of information that makes up the configuration information for the FPGA is called a bitstream. It is a file stored somewhere in a memory section. Figure 1 shows how this bitstream is structured:

The bitstream begins with a token, the preamble, which indicates the beginning of the header section that contains global information concerning the whole configuration cycle. This is followed by the configuration information for the core cells and by the I/O configuration. The postamble indicates the end of the bitstream.

These data are simply some data bytes that are sent one after another to the FPGA as a text file is sent to a printer. This is done in a serial or parallel fashion by implementing a transfer protocol that is not much different from the transfer protocol of a printer port.

The connections between an FPGA and a microcontroller for serial data transport are

Figure 1. Bitstream Structure



0508A

Figure 2. Connecting an Atmel FPGA with the AT89C51.

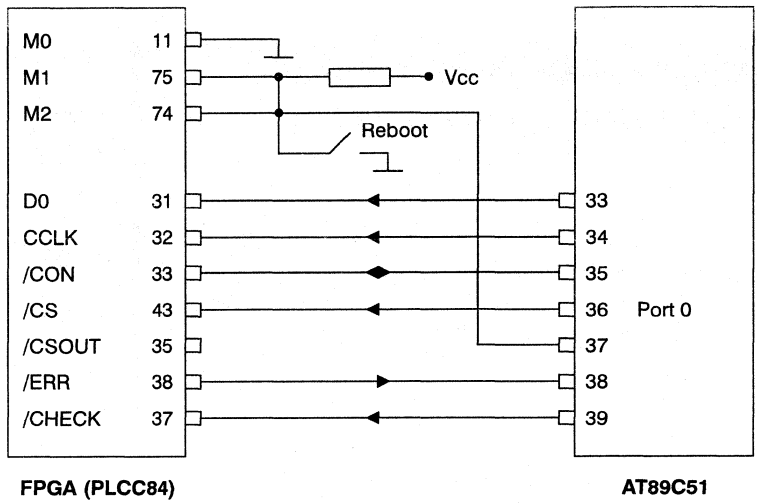
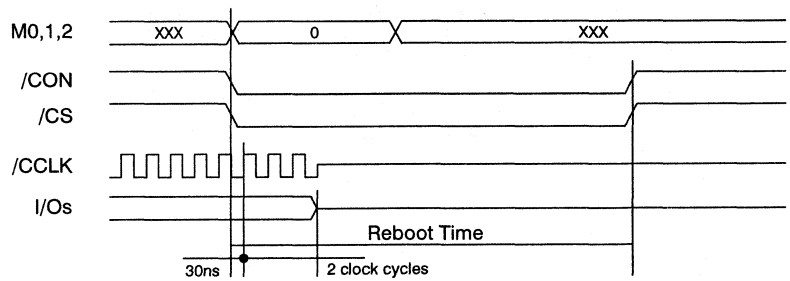


Figure 3. FPGA Reset Timing



shown in Figure 2. Only 7 lines are used enabling full control of the configuration cycle.

The microcontroller can force a complete reset of the whole FPGA by applying the mode 0 to the mode control lines. The state entered by the FPGA is the same state it will enter after power-up. All I/Os are in tristate mode. The exact timing is shown in Figure 3.

After triggering the reboot cycle, the FPGA will pull the /CON line to low. The microcontroller can check this line and detect the end of the cycle.

After a reboot cycle /CS and /CON are held high for two clock cycles. The microcontroller can then select an FPGA through the /CS line for configuration and start the configuration cycle by pulling /CON low. With each clock pulse on the CCLK line one bit of the bitstream transfers to the FPGA. The first transferred bit is the LSB of the preamble, the last transferred bit is the MSB of the postamble. To ensure the correct function of the internal state machine of the FPGA, 24 clock pulses are applied before and after a configuration cycle, for correctly entering the standby state between two configurations. These clock pulses are applied before /CON is pulled low and after /CON goes high.

Figure 4. FPGA Configuration Cycle Timing

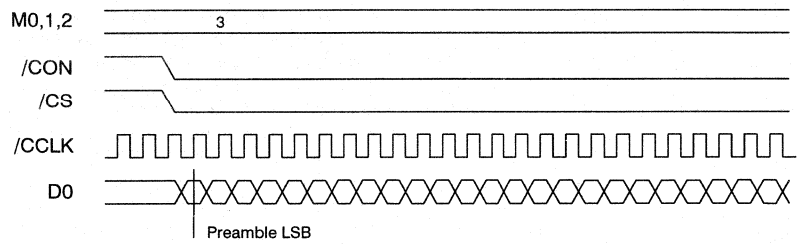
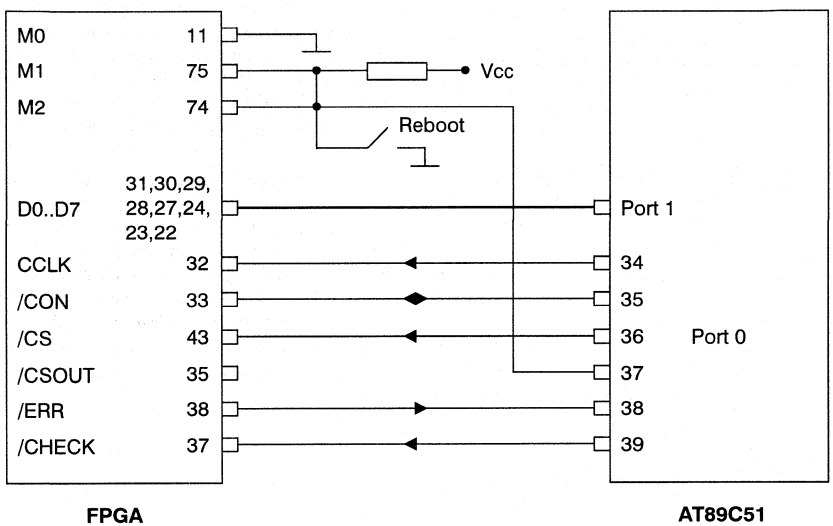


Figure 5. Parallel Data Transfer



With the transition from low to high the FPGA indicates that the configuration cycle has ended. The exact timing is shown in Figure 4.

This is the configuration mode 3 that is used for configuring several cascaded FPGAs. In this case all lines are brought to all FPGAs in parallel, except for the /CS lines. These connect to the /CSOUT pins of the preceding FPGAs in the chain. Only one bitstream that contains several configurations is needed to serve all FPGAs. The postambles are replaced by preambles to separate them. When the first FPGA in the chain sees a new preamble instead of a postamble, it will activate ist /CSOUT pin. The next FPGA in the chain is ready to accept the new configuration information, and so on.

Another possibility is to connect all /CS lines to the microcontroller so that it can select the next FPGA to be configured. All other lines are brought to all FPGAs in parallel. In this case normal bitstreams are sufficient, not the special bitstream that is explicitly generated and contains several preambles as described above.

If an error occurs during configuration, the microcontroller can detect this through the /ERR line. When the error is detected from the FPGA, e.g., an invalid preamble, this line will be pulled low to indicate the error to the microcontroller. When using several FPGAs the error lines can all be connected since this is an open collector output that can be WIRE-OR'ed. The microcontroller asserts the /CHECK line to determine whether the FPGA

is reconfigured. It also compares the bitstream it receives with the information that is already stored in the configuration memory within the FPGA. In the case of a difference the /ERR line is asserted.

If only reconfiguration of the FPGA without wanting error detection, the number of lines are reduced to three. /CS is tied to GND so that the FPGA is always selected. The microcontroller only has to provide the signals /CON, the configuration clock CCLK, and the data line. This solution uses the least board space.

With parallel data transfer, as shown in Figure 5, 8 data lines instead of one are used to transfer one data byte instead of one data bit per clock cycle. The configuration time is therefore much shorter. For these data lines, the normal data bus of the microcontroller is used, and the data transfer is controlled through the /WR signal of the controller. Reconfiguration of the FPGA is just like writing to an external RAM.

For a system where reconfiguration of the FPGA makes part of the regular system function this might be the most flexible solution. In this mode 6 as illustrated in the data book several FPGAs are cascaded as well. In this case, all lines except for /CS are brought to all FPGAs in parallel. /CS is connected to /CSOUT of the preceding FPGA in the chain to configure several FPGAs with one bitstream. The microcontroller can control the /CS pins to select the FPGA to be reconfigured as in a memory-mapped approach. The exact timing is shown in Figure 6.

Options for Storing Configuration Data

For storing the configuration data within the system there are different possibilities, each having its advantages and disadvantages.

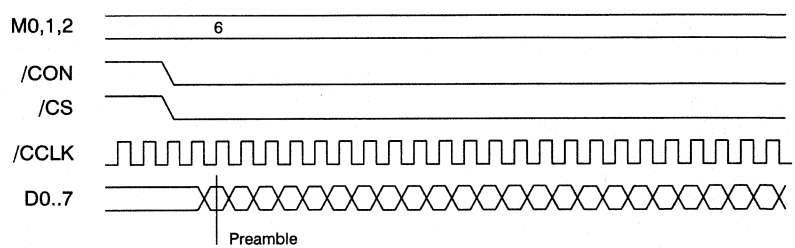
The first possibility consists of using the internal memory of a controller to store the configuration information. At first glance this might look like a waste but it offers distinctive advantages that enable certain applications. The microcontroller AT89C51 has an internal memory of 4 Kbytes, so that a full configuration of the AT6002 and an additional 1.4 Kbytes of program are stored. In this case only two chips make up the whole system which saves board space. The configuration data is also protected by the lock bits of the controller preventing it from being reverse-engineered.

Another possibility is to store the data in a parallel flash or EPROM memory that is handled by the microcontroller. If a flash memory is used, the configurations are changed through the serial link of the microcontroller simply by downloading a new configuration into the flash memory. Depending on the size of the memory, many configurations are stored in the system. For example, for a 5000 gate FPGA, the AT6005, requires only 8 Kbytes of configuration data. The address space of an AT89C51 controller is 64 Kbytes of data memory. This amounts to eight full configurations (or 40000 gates) or, more partial configurations where the bitstream is much smaller depending on how much is changed in a cycle.

The software controlling the FPGA configuration resides within the internal memory of the microcontroller and only needs to know the start address of the bitstreams. It can detect the end itself by searching for the postamble. Another possibility might be a control program that after power-up searches the whole data memory for preambles (for bitstreams). Then the controller receives the command to download the second configuration into the third FPGA by its serial interface, and downloads this configuration into the FPGA, and so on. To control this approach, careful system planning is necessary in order not to destroy a working function.

When using parallel memories instead of serial memories, the configuration is done very fast. Serial memories are more space-efficient, but slower. In space-sensitive applications they can be a solution. There are serial memories that are connected to an FPGA directly. They allow only one configuration, and they are costly. When using a microcontroller, standard serial memories are used that are cheaper and store more than one configuration. For example, if an AT24C64 serial memory with 1^2C bus interface is used, more than 3 full configurations for the AT6002 is stored.

Figure 6. FPGA Select Timing



Software Examples

Provided that a configuration cycle has finished with the FPGA releasing the /CON line, a subroutine for transferring a configuration bitstream to a FPGA is relatively simple. In the case of parallel data transfer it is seen as the following:

```
config:     MOV DPTR, #200H              ;start address bitstream
loopl:      MOVX A, @DPTR                ;load byte of bitstream
            MOV P1, A                    ;output at port 1
            SETB P0.5                    ;one clock pulse high
            CLR P0.5                     ;clock low
            INC DPTR                     ;next byte
            JNB P0.1, error              ;ERR is low?
            JNB P0.4, loopl              ;if CON is low, continue
            RET                          ;configuration finished
error:      do something. . .
```

In the case of serial data transfer a byte of the bitstream has to be converted from parallel to serial, but this is done with ROTATE instructions:

```
            RRC A                        ;first bit to carry
            MOV P1.6, C                  ;output carry as data bit
            RRC A                        ;second bit to carry
            MOV P1.6, C                  ;output carry as data bit
            RRC A                        ;third bit to carry
            MOV P1.6, C                  ;output carry as data bit
            . . .
```

A similar approach is applied in the case of a serial memory containing the configuration bitstreams. The microcontroller assembles them to a byte. When configuring in serial mode, this bit is handed over directly to the FPGA.

When storing several configurations in the address space of a microcontroller, it is complicated and error-prone to change the start addresses of the bitstreams within the microcontroller program. Another possibility is shown in the following code example: a table of 8 start addresses is created and the microcontroller searches the address space for bitstreams. Each time he finds a beginning, the start address is stored in the table. With this approach several configurations are stored at variable locations. In this example it is assumed that every bitstream begins with the preamble and the control register content is 00 hex (this is the second byte of the bitstream). This program is simply to show the principles applying. It is also assumed that an AT89C51 with 4 Kbytes internal memory is used and the bitstreams are stored starting with address 1024 (0400 hex).

```
CONTABLE    db 0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
                                         ;space for 8 16-bit pointers
NUM_CONFIG  db 0                         ;number of configuration
PREAMBLE    EQU #10110010B
POSTAMBLE   EQU #01001101B
CONTROLREG  EQU #00000000B               ;adjust to desired value
detect:     MOV DPTR, #0400H             ;start searching at 0400
            MOV R0, NUM_CONFIG           ;actual config address is held in R0
loopl:      MOV A, @DPTR                 ;load byte
            INC DPTR                     ;next byte
            MOV R1, DPH                  ;load high byte DPTR
            CJNE R1, #10H, cont1         ;continue, if address below #1000H (that is,
                                         ;#4096D for a 89C51 that has 4KB of internal
                                         ;memory)
            AJMP overrun                 ;else stop searching (end of memory)
cont1:      CJNE A, PREAMBLE, loopl      ;preamble found?
            MOV A, DPL,                  ;save DPTR low byte
            MOV @R0, A
            INC R0
            MOV A, DPH                   ;save DPTR high byte
            MOV @R0, A
            INC R0                       ;next pointer location
```

```
loop2:      MOV A, @DPTR                    ;continue searching
            INC DPTR                        ;next byte
            MOV R1, DPH                     ;test end of memory
            CJNE R1, #10H, cont2
            AJMP overrun
cont2:      CJNE A, CONTROLREG, loop2       ;control reg found?
loop3:      MOV A, @DPTR                    ;yes, continue
            INC DPTR                        ;next byte
            CJNE A, POSTAMBLE, loop3        ;postamble found?
            INC DPTR                        ;continue searching
            MOV R1, DPH                     ;text end of memory
            CJNE R1, #10H, cont3
            AJMP overrun
            AJMP loop1                      ;for preamble
overrun     RET                             ;end of subroutine
```

When it is known how many configurations are stored in the memory and where they are, they can be transferred to the FPGA. for example, through commands received with the serial interface of the microcontroller.

A similar program can be written for storing the configuration data in serial memories, but here the bits have to be assembled to bytes first in order to search for the preamble.

Encryption and Security

SRAM-based FPGAs always receive their configuration from the outside. Besides all advantages this offers, e.g., reconfigurability or testability, there can be problems with security and protection of the design. When the configuration is stored in a serial or parallel memory that is read directly by the FPGA, this memory can be copied. In this case no protection is available.

The problem is only half as difficult as it seems because simply copying the configuration is not the whole job. This can only be used to copy the system, but the logic function of the FPGA is very difficult to deduce from the bitstream. The relation between a certain bit in the bitstream and the function it controls is very difficult to determine. Therefore, the circuit realized with the FPGA is very difficult to reverse-engineer.

When using a microcontroller to configure the FPGA, additional security mechanisms are implemented. The bitstream can be encrypted before storing it in the system memory so that the microcontroller decrypts the bitstream before sending it to the FPGA. The key is hidden within the microcontroller or with external means, e.g., a smartcard or identification number.

Even with very basic operations a high degree of security is reached. For example, if all bits of the bitstream are inverted the configuration bitstream is useless for the FPGA. Another way is to exchange some bytes with others through a table. This is a very easy and therefore fast operation that will slightly slow down the configuration process and will result in a high level of protection.

Using the option indicated above (storing the configuration information within the internal memory of the controller) has other advantages. With the lock bits of the controller access to the memory can be inhibited even when the microcontroller is put on a programmer. With the chip-erase function of the Atmel microcontrollers, the whole memory array can be erased in 10ms when the part of the system is accessed, e.g., by opening the case or entering a wrong identification number three times. This also works when the configuration is not stored within the microcontroller, but only the key number is stored.

There are still weak points in the system. These are made up by the data and control lines between the FPGA and the microcontroller. They are sampled with a logic analyzer and the configuration information is extracted from the timing diagram. This is difficult, but not impossible. One needs to know the parts that are used in the system; the right key or identification number, and a running system for analyzing it. Only then the configuration for one given moment is known. It does not infer that the system can be copied. If partial reconfiguration is used, the design can be partitioned in two or more parts. The major part is transferred unencrypted and some few cells of central importance are transferred at another point of time or from the outside. A system that changes itself frequently is much harder to copy or reverse-engineer. Other tricks such as custom-marking the FPGA (so it is thought to be an ASIC), additional power and ground pins help to disguise the identity of the used part. By implementing all these methods, the process of copying the design is complicated, but there is no absolute security.

Conclusion

The following table shows the different options for parallel or serial configurations in conjunction with parallel or serial configuration storage. The given configuration times are for full configuration of an AT6005 without encryption. An AT89C51 microcontroller with a clock frequency of 24 MHz is used. For assessing the necessary board space, it was assumed that the microcontrollers are used with QFP packages and the memories are used in SOIC or TSOP packages.

Connection to FPGA	Serial	Serial	Parallel	Parallel
External Memory	Serial	Parallel	Serial	Parallel
Space Requirements	199mm^2	329mm^2	199mm^2	329mm^2
Configuration Time	93ms	61ms	61ms	30ms

The space requirements are mainly determined by the chosen memory. It is difficult to assess the board space required by parallel or serial wiring. Either one will be determined by application requirements, that is, fast reconfiguration or small space. Configuration time is more dependent on the connection between the controller and the FPGA; the memory connection is not as important.

It is obvious that controlling the configuration of FPGAs with the help of microcontrollers is implemented very easily. When a controller is already in use within the system, only one additional port is required, and some space in the flash memory that might already be in the system as well. Flexibility in the design is increased and additional features can easily be implemented.

Programming Atmel's Family Of Flash Memories

Flash

Application Note

4

Introduction

Atmel offers a diverse family of Flash Memory devices ranging in density from 256 K to 4 Mbits. These devices read and program with a single voltage supply. The nominal supply voltage is 5 V for the AT29Cxxx, 3.3 V for the "low voltage" AT29LVxxx, and 3 V for the "*Battery-Voltage™*" AT29BVxxx Flash family. The entire Flash family is designed to allow users to have one common programming algorithm for all three Flash voltage families. Therefore, upgrading from one density to another and from a higher voltage to a lower voltage device is simplified.

This application note describes the design benefits of Atmel's Flash architecture as well as how the device ID feature is used to adjust for varying densities and supply voltages. In addition, Atmel's Software Data Protection (SDP) feature, which prevents inadvertent writes, is described. An example is given to illustrate the ease with which the programming software can be written to accommodate four different 4 Mbit Flash devices, the AT29C040, AT29LV040, AT29C040A, and AT29LV040A.

Hardware and software have been developed to demonstrate the relevant design issues. The demo uses an AT89C51 Flash-based microcontroller (which has the same pinout and instruction set as an 80C51) as the host processor and a "C" language program for the software. The software automatically adjusts the amount of time required for programming the varying voltage versions of the 4 Mbit Flash devices in addition to accommodating for their different sector sizes.

The AT89C51, a member of Atmel's growing family of Flash microcontroller devices, features 4 Kbytes of in-system reprogrammable Flash memory (see Atmel application note "AT89C51 In-Circuit Programming" for additional information). Current and future versions of Atmel's microcontroller family incorporate from as little as 1 Kbytes of Flash memory to as much as 128 Kbytes, providing many density options for different applications. Other versions will also include special architectures such as a combination of Flash and parallel EEPROM memory on board.

Programming Flash Devices

Unlike Atmel's Flash Memories, previous generations of Flash memories had large kilobyte sectors and required that an entire sector be erased prior to programming. Generally, the sector erase cycle time was hundreds or thousands of milliseconds and could be as long as 30 seconds for the entire memory array. In addition, a separate high voltage supply was required for a write and erase operation. Atmel's Flash family has simplified usage by having only one supply voltage, reducing the sector size, having the programming similar to an SRAM write operation, and decreasing significantly the total programming time.

Small sector sizes reduce the amount of system resources necessary for programming. When only a few bytes in a Flash memory need to be altered, a RAM image of the Flash sector must be created. The RAM must then be altered with the new data, and the image transferred back into the Flash device. Because Atmel's Flash devices have small sector sizes (from 64 to 512 bytes, depending on the memory density), the RAM requirements are much less than those of large sector Flash devices. The latter generally have 4 K to 128 Kbyte sector sizes.

A second advantage of Atmel's Flash is that an entire sector can be updated during a single program operation, instead of the byte-by-byte programming of previous generation Flash memories. This saves significant programming time when updating an entire sector, especially when comparing Atmel's small sector devices with large sector devices. In addition, Atmel's devices do not require a sector erase prior to writing, thus

0510A

saving additional programming time. The maximum sector program time is 10 ms and 20 ms for the AT29Cxxx and AT29LVxxx/AT29BVxxx families respectively.

AT29C040 and AT29C040A Architecture

The AT29C040 provides operation similar to a byte-wide SRAM. The device has eight data lines and 19 address lines. The familiar three input control lines are also present ($\overline{CE}$, $\overline{OE}$, $\overline{WE}$). Read operations are identical to an SRAM, but write operations are somewhat different due to the write cycle time (t_{WC}) requirements of all Flash memories. Flash write operations take several milliseconds to complete, compared to the nanosecond writes of SRAM devices. It should be noted that Atmel's Flash Memories require only a write operation; the erase operation is automatically performed internally in the device.

Data is loaded into the AT29C040 one sector at a time, with each sector consisting of 512 bytes. The sector chosen for modification is defined by the upper order address bits (A9-A18). The entire sector must be loaded during the write operation. Any byte not loaded during the sector load will contain FFH after the write operation has completed. Address lines A0 through A8 define the location of the bytes within a sector. All data must be loaded into the same sector (A9 through A18 must remain constant) and can be randomly loaded within that sector.

The AT29C040A is identical to the AT29C040 except for the sector size and the Device ID Code (the Device ID Code is described later). The AT29C040A has a 256 byte sector (instead of a 512 byte sector) which is defined by address lines A8 through A18; the bytes within the sector are determined by address lines A0 through A7.

Software Data Protection (SDP)

One concern of systems designers when using nonvolatile programmable memories is the possibility of inadvertent write operations that can be caused by noise or by power-up and power-down sequences. Atmel's Flash memories provide a feature called Software Data Protection (SDP) that addresses this issue. The user can enable SDP upon receipt of the device from Atmel, and its usage is highly recommended. Data can be written into a sector with or without SDP enabled. However, once SDP has been enabled, the device requires that all subsequent write operations perform a series of "dummy" write operations before loading the chosen sector with data. The "dummy" writes consist of loading three known data values into three predefined addresses. This three-byte sequence preceding a write operation virtually eliminates the chance of inadvertent write operations. The sequence is described below.

1. Load Data AAH into Address 05555H
2. Load Data 55H into Address 02AAAH
3. Load Data A0H into Address 05555H
4. Load desired sector with data
5. Pause t_{WC} (device write cycle time)
6. The device is returned to standard operating mode

If SDP is enabled, any attempt to write to the device without the three-byte command sequence will start a write cycle. However, no data will actually be written to the device, and during this "write" cycle time (t_{WC}), valid data cannot be read from the Flash.

Product and Manufacturer ID

Atmel's Flash memory devices allow the user to access both device and manufacturer information. This feature allows a system to determine exactly which Flash memory is being used. Once this is known, the host system can choose different algorithms for write operations in order to accommodate for differences in device density, V_{CC} requirements, sector size, and required write cycle time.

Product and manufacturer ID information is determined with the Software Product Identification procedure, which is similar to the Software Data Protection sequence. The sequence is described below.

1. Load Data AAH into Address 05555H
2. Load Data 55H into Address 02AAAH
3. Load Data 90H into Address 05555H
4. Pause t_{WC} (device write cycle time)
5. Read Address 00000H
   Data read is the Manufacturer Code
6. Read Address 00001H
   Data read is the Device ID Code
7. Load Data AAH into Address 05555H
8. Load Data 55H into Address 02AAAH
9. Load Data F0H into Address 05555H
10. Pause t_{WC} (device write cycle time)
11. The device is returned to standard operating mode

The following table uses the 4 Mb Flash as an example to illustrate the pertinent device information than can be determined once the Device ID Code is known.

DEVICE	ID	V_{CC}	SECTOR SIZE	t_{WC}
AT29C040	5B	5.0 V ± 10%	512 bytes	10 ms
AT29C040A	A4	5.0 V ± 10%	256 bytes	10 ms
AT29LV040	3B	3.3 V ± 0.3 V	512 bytes	20 ms
AT29LV040A	C4	3.3 V ± 0.3 V	256 bytes	20 ms
AT29BV040	3B	3.0 V ± 10%	512 bytes	20 ms
AT29BV040A	C4	3.0 V ± 10%	256 bytes	20 ms

Hardware Description

The demo hardware consists of a 12 MHz AT89C51 Flash-based microcontroller with 4 Kbytes of on-board Flash memory. The internal AT89C51 Flash memory is used for boot code, and the external 8K x 8 SRAM and the AT29C040 are mapped as data memory. The AT29C040 is also mapped as program memory to facilitate off-chip program execution. The AT89C51 can only access a maximum of 64 Kbytes of data memory space, while the AT29C040 has 512 Kbytes of storage capacity. To solve this size mismatch, the AT29C040 is bank switched into the AT89C51 data memory map in 8 Kbyte blocks. The bank switching is performed with six general purpose I/O port bits on the AT89C51. The system address map is shown below.

System Address Map

AT89C51 Microcontroller	0000-1FFF	Internal program memory
8 K x 8 Static RAM	2000-3FFF	Data memory
AT29C040 Flash	4000-5FFF	Program and data memory

Software Description

The software (available from Atmel's BBS 408-436-4309) demonstrates how the Device ID Code can be used to allow a single program to work with different Atmel Flash memories. The program uses Atmel's 4 Mb Flash (AT29C040, AT29LV040, AT29C040A, and AT29LV040A) as an example, but the software can be easily adapted to accommodate other device densities.

In order to program the Flash memory, the software must first determine which Flash device is being used. This is accomplished by first putting the device into the Software Product Identification mode by executing a three-byte command sequence (described in the "Product and Manufacturer ID" section of this application note). The program subsequently reads the Device ID Code and executes another three-byte command sequence to return the Flash to the standard operating mode. Using the Device ID Code, the program then determines the appropriate sector size and write cycle time (t_{WC}) for the particular 4 Mb Flash being used.

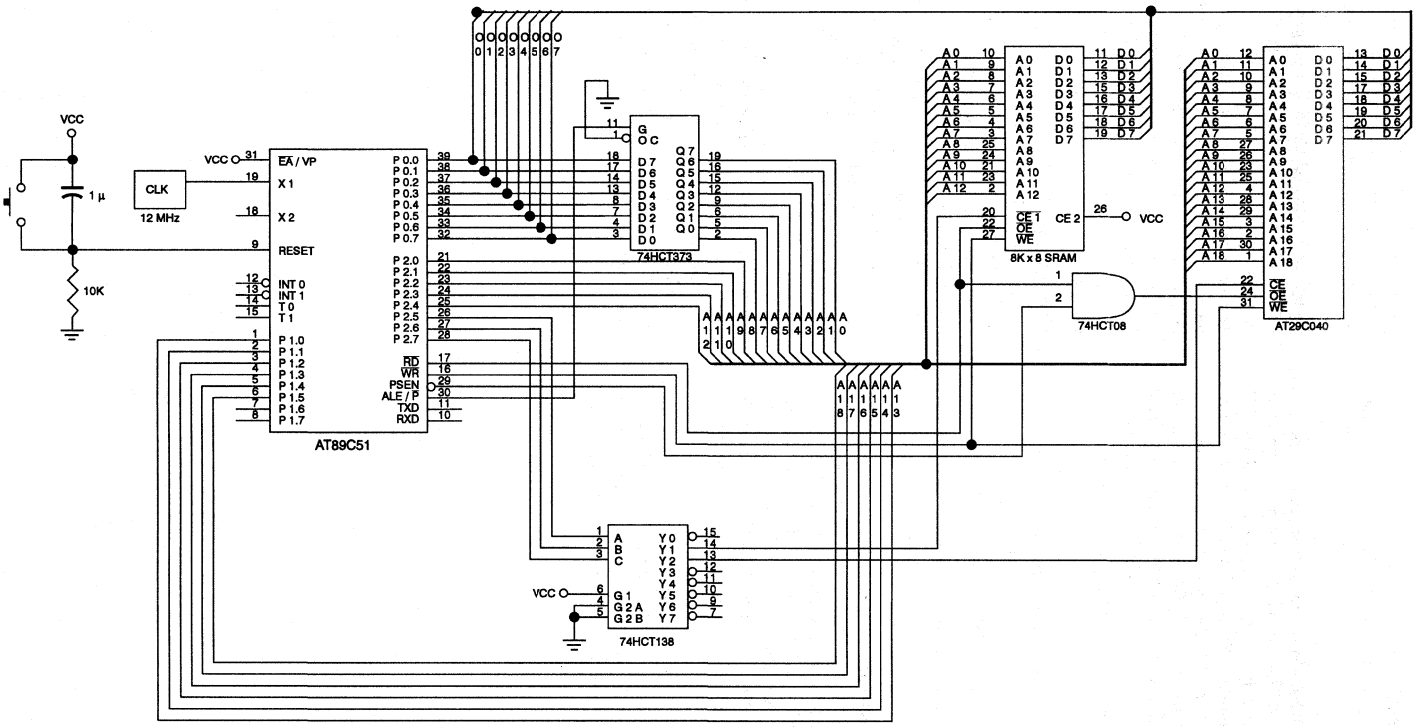
To demonstrate a sector write, the program proceeds to load the SRAM with "dummy" data. After the data has been loaded, the program transfers the data from the SRAM to a predefined sector (within one of the mapped 8 Kbyte blocks) of the 4 Mb Flash. After pausing the required write cycle time (t_{WC}), the sector that was just written is transferred back to the SRAM buffer.

Summary

Atmel's Flash Memories are designed to allow all densities and device configurations to be programmed using the same programming algorithm. The user has to simply determine the Device ID Code and set the appropriate sector size and write cycle time. This operation need only be performed once provided the sector size and write cycle information is saved. If only one density or configuration will ever be used, then reading of the Device ID Code can be eliminated, and the sector size and write cycle information can be predefined in the software.

As demonstrated, programming Atmel's Flash is a simple process, similar to loading an SRAM. Architectural and circuit features within the devices minimize software and system overhead while simplifying programming procedures. Atmel's Flash Memories require only about one-tenth of the typical software, buffer memory, and performance overhead of previous generation Flash, thus providing substantial system cost savings.

Atmel AT29C040DA Demo Circuit



Note: If the Flash is to be used as external program memory, then pin 31 ($\overline{EA}$/ Vpp) of the AT89C51 cannot be connected to ground.

Analog-to-Digital Conversion Utilizing the AT89CX051 Microcontrollers

The Atmel AT89C1051 and AT89C2051 microcontrollers feature on-chip Flash, low pin count, wide operating voltage range and an integral analog comparator. This application note describes two low-cost analog-to-digital conversion techniques which utilize the analog comparator in the AT89C1051 and AT89C2051 microcontrollers.

RC Analog-to-Digital Converter

This conversion method offers an extremely low component count at the expense of accuracy and conversion time. In the example presented below, resolution is better than 50 millivolts, accuracy is somewhat less than a tenth of a Volt and conversion time is seven milliseconds or less.

As shown in Figure 1, the RC analog-to-digital conversion method requires only two resistors and a capacitor in addition to the AT89CX051 microcontroller. A microcontroller output (pin 11), which swings from approximately ground to V_{CC}, alternately charges and discharges the capacitor connected to the non-inverting input of the internal comparator (pin 12). The microcontroller measures the time required for the voltage on the capacitor to match the unknown voltage applied to the inverting input of the internal comparator (pin 13). The unknown voltage is a function of the measured time.

The HP5082-7300 LED displays shown in Figure 1 are not required for the conversion, but are utilized by the software to implement a simple two-digit voltmeter. The result of the analog-to-digital conversion is displayed in volts and tenths of a volt on the two displays. The voltmeter application does not utilize the full resolution of the RC conversion software, but serves to demonstrate the method as well as providing a tool for debug.

The waveform for a typical capacitor charge/discharge cycle is shown in Figure 2. The discharge portion of the curve is identical to the charge portion rotated about the line $V_C = V_{CC}/2$. The equations and discussion below apply to the charge portion of the cycle, except where indicated.

The voltage on the capacitor as a function of time is given by the exponential equation:

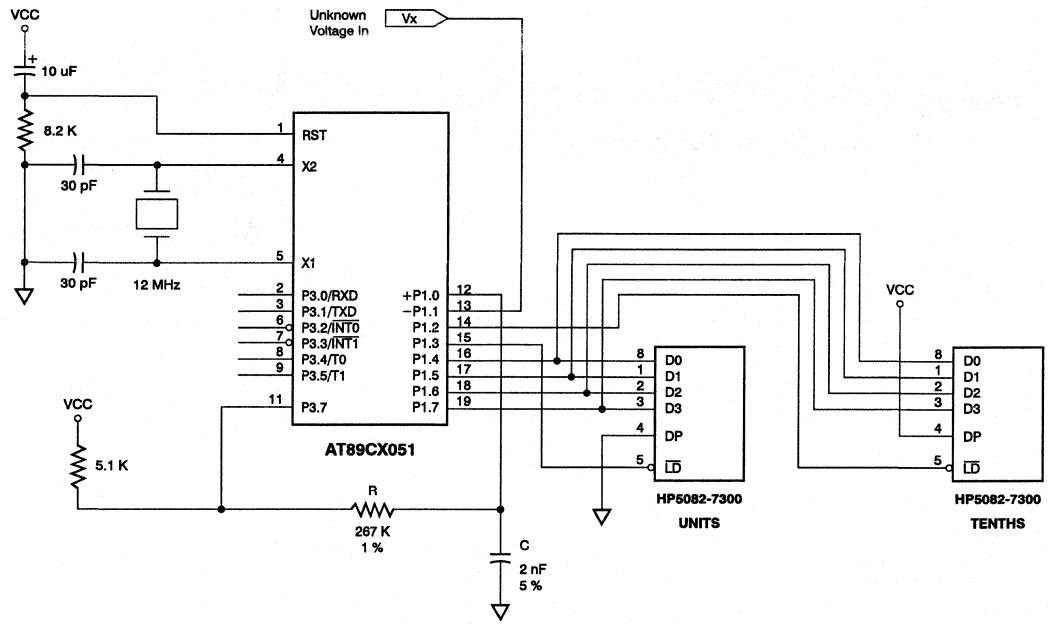
$$V_C = V_{CC}(1-e^{-t/RC}) \qquad (1)$$

where V_C is the voltage on the capacitor at time t, V_{CC} is the supply voltage and RC is the product of the values of the resistor and capacitor. Note that voltage is expressed in Volts, time in seconds, resistance in Ohms and capacitance in Farads. The product RC is also known as the "time constant" of the network and affects the shape of the waveform. The waveform is steepest when capacitor charging or discharging begins and flattens with time.

The first problem with the RC conversion method is the difficulty of solving the exponential equation without utilizing floating point calculations and transcendental functions. On a compressed time scale, the exponential curve appears straight over much of its length, suggesting that it might be approximated by a line. This scheme fails due to the continuous variation in slope over the length of the curve, which produces significant error. It also does not address the problem where the curve rolls off severely near the asymptote at V_{CC}.

The microcontroller need not solve the exponential equation in real time if a lookup table is used to map pre-calculated values to each sampled time interval. This scheme allows the data to be encoded and formatted as required by the application while simplifying the conversion software. Symmetries in the data may be exploited to reduce the size of the table.

Figure 1. Two-Digit Voltmeter



The second problem with the RC conversion method is the substantial error which results from variations in component values. Figure 3 shows an exaggerated view of the variation in the voltage on the capacitor due to variations in the values of the resistor and capacitor. As shown in the figure, the variation in the voltage on the capacitor decreases as the voltage on the capacitor decreases.

The symmetry of the capacitor charge/discharge cycle can be exploited to reduce the effect of variations in component values on conversion accuracy. This is done by utilizing the charge portion of the cycle to measure voltages less than $V_{CC}/2$ and the discharge portion to measure voltages greater than $V_{CC}/2$. The worst case error is reduced to the error at $V_{CC}/2$.

Before component values can be assigned, the time interval at which the comparator output is to be sampled must be determined. The sample interval should be as short as possible to maximize converter resolution and minimize conversion time. The sample interval is limited by the time required to execute the requisite code, which is determined by the clock rate of the microcontroller. In the voltmeter application, the microcontroller operates with a 12 MHz clock, resulting in a sample interval of five microseconds.

The time constant (RC) affects the shape of the capacitor charge/discharge waveform. The value of the time constant must be chosen so that the steepest parts of the waveform are resolvable to the desired resolution. The steepest part of the charge portion of the waveform occurs near the origin, while the steepest part of the discharge portion occurs near V_{CC}. Due to the symmetry of the waveform, the same time constant may be used for measurements made on either portion of the waveform.

Figure 4 shows an expanded view of the relationship between voltage and sample time near the origin. In the figure, ΔV is the desired voltage resolution of the converter and Δt is the sample interval determined previously. The curve labeled 'V_C' represents the voltage on the capacitor, which appears linear at this scale. In the figure, the slope of the curve is ideal, causing sampling to occur near the center of the voltage intervals. The slope of the curve may be less than shown, but may not be greater, or resolution will be lost. Note that the first sample is offset from the origin by 1/2 Δt to center the sample in the first voltage interval.

To obtain the minimum value of the time constant which will produce the required slope at the first sample, solve Equation 1 for RC:

$$RC = -t/\ln(1-V_C/V_{CC}) \qquad (2)$$

Then set ΔV to the minimum desired resolution (0.05-volt), Δt to the sample interval determined previously (five microseconds), and calculate RC at the first sample point, where V_C = 1/2 ΔV and t = 1/2 Δt:

$$RminCmin = \frac{(-1/2)\Delta t}{\ln[1-(1/2(\Delta V))/V_{CC}]} = \frac{-(1/2)(5\cdot10^{-6})}{\ln[1-(1/2)(0.05)/V_{CC}]} \cong 4.99\cdot10^{-4}$$

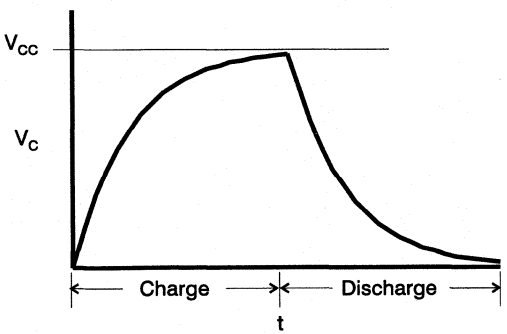
The product of the values of R and C must not be less than the calculated minimum time constant. Utilizing a resistor with a one percent tolerance and a capacitor with a five percent tolerance:

$(Rnom-1\%)(Cnom-5\%) >= 4.99 \cdot 10^{-4}$.

In the voltmeter application, the selected values of R and C are 267 kilohms and 2 nanofarads, respectively, yielding a minimum time constant of approximately $5.02 \cdot 10^{-4}$.

An additional constraint is placed on the value of R. Referring again to Figure 1, note the 5.1 kilohm pullup resistor connected to pin 11 of the microcontroller. This resistor is present to supplement the microcontroller's weak internal pullup, but has the detrimental effect of changing the time constant of the RC network during the charge portion of the capacitor charge/discharge cycle. This produces an asymmetry in the charge/discharge waveform, which contributes to conversion error. To minimize the effect of differences in the capacitor charge and discharge paths, the value of R should be chosen to be much greater than the value of the pullup resistor. In the voltmeter application, the selected value of R is 267 kilohms, which exceeds the value of the pullup resistor by more than an order of magnitude.

Figure 2. Typical Capacitor Charge/Discharge Cycle



The time constant (RC), which is a function of the desired converter resolution, determines the duration of the capacitor charge/discharge cycle. The more time required for the capacitor to charge and discharge, the greater the number of samples required in the measurement loop and the greater the number of entries in the lookup table.

The time required for the capacitor to charge and discharge is approximated by calculating the maximum time for the voltage on the capacitor to rise to within one half of the smallest resolvable voltage interval from the asymptote. For the charge portion of the waveform, the asymptote is at V_{CC}. Due to the symmetry of the waveform, the determined value applies to both the charge and discharge portions of the cycle.

Solving Equation 1 for time yields:

$t = -RC \cdot \ln(1-V_C/V_{CC})$. (3)

Assuming a resolution of 0.05 Volt, the desired capacitor voltage is:

$V_C = V_{CC}-(1/2)(0.05) = V_{CC}-0.025$.

From Equation 3:

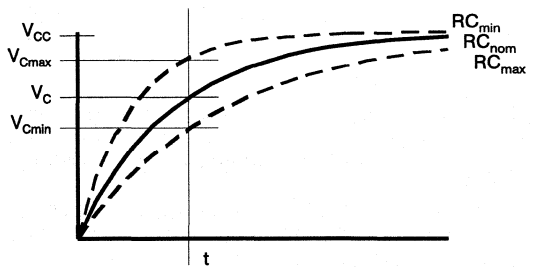
$tmax = -RmaxCmax \cdot \ln(1-(V_{CC}-0.025)/V_{CC})$

$= -(Rnom+1\%)(Cnom+5\%)\ln(0.025/V_{CC})$

$= -(1.01)(267 \cdot 10^3)(1.05)(2 \cdot 10^{-9})\ln(0.025/5.0) \cong 3$ ms.

The minimum number of samples required in the measurement loop is determined by calculating the time required for the voltage on the capacitor to reach $V_{CC}/2$ and dividing the result by the sample interval. The maximum value of the time constant is used in the calculation, since the voltage on the capacitor rises slower when the values of the resistor and capacitor are large. Due to the symmetry of the capacitor charge/discharge waveform, the determined sample count may be used for measurements made during either portion of the cycle.

Figure 3. Capacitor Voltage Variation as a Function of RC Variation



From Equation 3:

$tmax = -RmaxCmax \cdot \ln(1-(1/2)V_{CC}/V_{CC})$

$= -(Rnom+1\%)(Cnom+5\%)\ln(1/2)$

$= -(1.01)(267 \cdot 10^3)(1.05)(2 \cdot 10^{-9})\ln(1/2)$

$\cong 393$ µs.

The minimum number of samples for half the cycle is:

$tmax/\Delta t = (393 \cdot 10^{-6})/(5 \cdot 10^{-6}) = 79$.

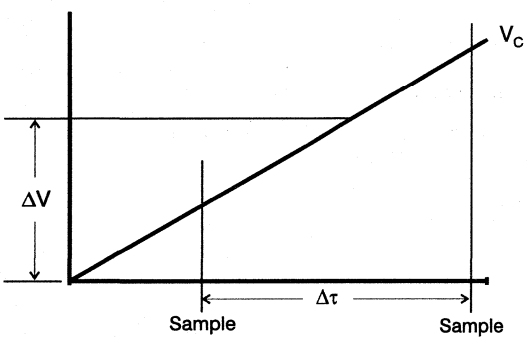
To maximize accuracy, voltages from zero to $V_{CC}/2$ are measured during the charge portion of the capacitor charge/discharge cycle and voltages from V_{CC} to $V_{CC}/2$ are measured during the

discharge portion of the cycle. As a result, the total number of entries in the table is twice the number of samples calculated previously for each half cycle.

The lookup table contains application-specific values corresponding to the calculated voltage at each sample. For each half cycle, the Nth entry in the table corresponds to the voltage at t = (N-1) Δt, where Δt is the sample interval determined previously. For the charge half cycle, the voltage at each sample is calculated by solving Equation 1 for the time elapsed since the capacitor began to charge. For the discharge half cycle, the voltage at each sample is calculated by solving the following equation for the time elapsed since the capacitor began to discharge:

$$V_C = V_{CC} \cdot e^{-t/RC} \quad (4)$$

Figure 4. The Relationship between Voltage and Sample Time near the Origin



The size and contents of the table may vary from application to application depending on the sample interval and conversion resolution. As the resolution increases, the number of entries in the table grows.

In the voltmeter application, with resolution equal to 0.05 Volt, the lookup table contains 158 entries, which is twice the number of samples per half cycle calculated above.

Voltages corresponding to samples taken during the charge half cycle are calculated by replacing 't' with 'N Δt' in Equation 1, where N represents the sample number (0-78). By setting Δt equal to the sample interval of 5 microseconds, R to 267 kilohms, C to 2 nanofarads, and Vcc to 5.00-volts, Equation 1 becomes:

$$V = 5(1-e^{-N\,(.0093633)}).$$

Voltages corresponding to samples taken during the discharge half cycle are calculated by replacing 't' with 'N Δt' in Equation 4, where N represents the sample number (0-78). Using the same values as for the charge half cycle, Equation 4 becomes:

$$V = 5 \cdot e^{-N(.0093633)}).$$

An abbreviated list of the voltages calculated for the capacitor charge/discharge cycle is shown below. The ordering of the voltages, increasing in the first half, decreasing in the second, tracks the voltage on the capacitor and defines the ordering of the table entries.

N = 0 V= 0.000
N = 1 V= 0.047
. .
. .
. .
N = 74 V= 2.499
N = 75 V= 2.523
N = 76 V= 2.546
N = 77 V= 2.569
N = 78 V= 2.591

N = 0 V= 5.000
N = 1 V= 4.953
. .
. .
. .
N = 74 V= 2.501
N = 75 V= 2.477
N = 76 V= 2.454
N = 77 V= 2.431
N = 78 V= 2.409

As shown by the list, the number of samples in each half cycle is greater than required to reach the midrange value of 2.500 Volts. This allows for "fast" cycles which overshoot the nominal midrange value before the last sample is taken in each half cycle. Note that the difference between the calculated voltages at samples N=0 and N=1 is within the desired resolution of 0.050-volt, but the difference in voltage between adjacent samples decreases as N increases. This reflects the non-linear relationship between voltage and time in the circuit.

The calculated voltages shown in the list are not entered into the lookup table, but are used to determine the values of the table entries. In the voltmeter application, the calculated voltages are rounded to tenths of a volt and the result stored in the table in packed-BCD form, two digits per byte. Example: the table entry corresponding to 2.523-volts is 25 hex, which displays as 2.5-volts.

The voltmeter prototype demonstrated accuracy of +/- one count (0.1 Volt), but accuracy of somewhat less than a tenth of a Volt is about the best that can be expected from the RC analog-to-digital conversion method. Even using precision components, variations in component values may contribute an error of +/- 0.104-volt, as shown below.

To calculate the worst case error at V_C = 2.5-volts, first determine the corresponding t at the nominal values of R and C using Equation 3:

$$t = -RnomCnom \cdot \ln(1-V_C/V_{CC})$$
$$= -RnomCnom \cdot \ln(1-2.5/5.0)$$
$$= -RnomCnom \cdot \ln(0.5).$$

Substitute for t in Equation 1 to get minimum Vc:

$$V_{Cmin} = V_{CC}(1-e^{-t/(R_{max}C_{max})})$$
$$= V_{CC}(1-e^{(R_{nom}C_{nom}/R_{max}C_{max})\ln(0.5)})$$
$$= 5(1-e^{\ln(0.5)/(1.01\bullet1.05)})$$
$$\cong 2.399\ V$$

Again, for maximum Vc:

$$V_{Cmax} = V_{CC}(1-e^{-t/(R_{min}C_{min})})$$
$$= V_{CC}(1-e^{(R_{nom}C_{nom}/R_{min}C_{min})\ln(0.5)})$$
$$= 5(1-e^{\ln(0.5)/(0.99\bullet0.95)})$$
$$\cong 2.607\ V$$

The results show a variation of 0.208-volts at 2.5-volts, or a worst case error of +/- 0.104-volts. The worst case conversion error may be further reduced by utilizing components with tighter tolerances. Conversion accuracy and linearity are also affected by the characteristics of the capacitor. The capacitor used in the voltmeter prototype is a polystyrene film type, which not only provides good accuracy, but minimizes error due to dielectric absorption and other effects.

Error sources which have not been examined include: comparator limitations; asymmetries between the charge and discharge portions of the cycle; failure of the voltage on the capacitor to reach ground or V_{CC}; variations in V_{CC}. The contributions to conversion error made by these sources can be expected to increase error to somewhat more than the value due to component tolerances alone.

Successive Approximation Analog-to-Digital Converter

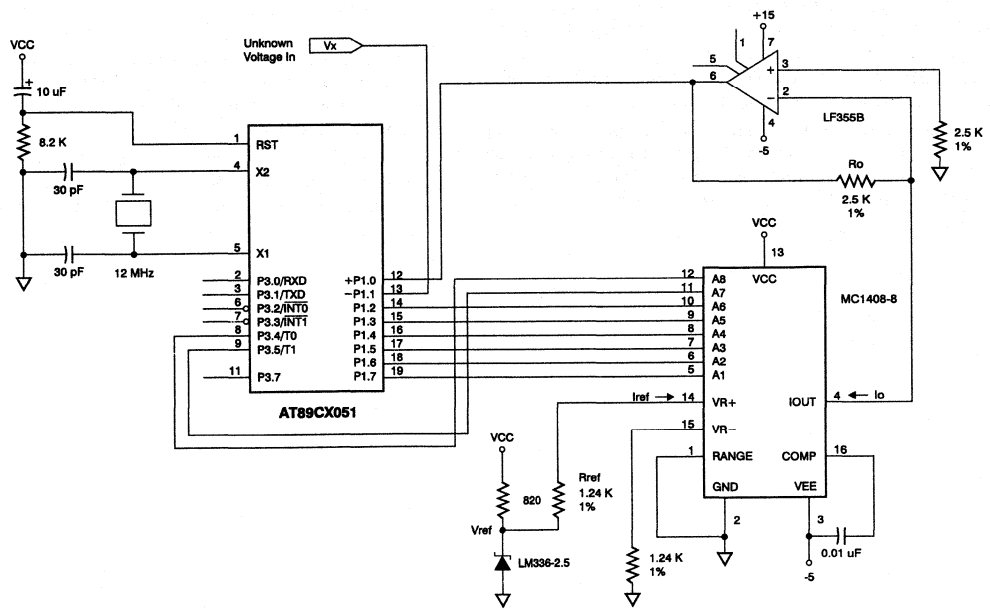
This conversion method offers good resolution and accuracy and a short conversion time at the expense of increased component count.

Successive approximation (SA) ADCs incorporate a digital-to-analog converter (DAC), a comparator and a successive approximation register (SAR). The SAR controls the conversion by performing a search for the binary code which, when fed to the DAC, will produce an output matching the voltage to be converted. The comparator compares the DAC output to the unknown voltage and returns the result to the SAR.

The SAR begins the search with the most significant DAC bit, which controls the widest output variation, and moves toward the least significant bit, causing the DAC output to "zero in" on the unknown value. The result of the trial is the binary code corresponding to the unknown value. In an eight-bit SA converter, only eight iterations are required to find the correct binary code, resulting in relatively fast conversions.

In this application (Figure 5), an AT89CX051 microcontroller with an integral analog comparator performs the SAR function in software, reducing the component count. The DAC selected for the application is an MC1408-8, eight-bit, current output type chosen for its low cost. Seven- and six-bit versions are available as the MC1408-7 and MC1408-6, respectively. The MC1408 series is guaranteed accurate to within +/- 1/2 LSB at 25 degrees C at a full scale output current of 1.992 milliamps. The relative accuracy of the MC1408-8 is better than 0.19%, assuring eight-bit monotonicity and linearity. The DAC has an output settling time of 300 nanoseconds.

Figure 5. Successive Approximation Analog-to-Digital Converter



The DAC contains binary-weighted, current-steering switches which scale an input current by the applied binary code. The input current is derived from an LM336-2.5 precision voltage reference and a series resistor. The scaled current output is converted to a voltage by an LF355B operational amplifier wired as a current-to-voltage (I/V) converter. The LF355B op amp was selected for the I/V converter because of its low input offset voltage and high output slew rate. The voltage output of the I/V converter is fed into the AT89CX051 comparator, where it is compared to the unknown voltage. When the programmed voltage exceeds the unknown voltage the output of the comparator goes high, which is detected by software. A second op amp, wired as a non-inverting, unity gain buffer may be inserted between the unknown voltage source and the input to the AT89CX051 comparator to provide isolation.

The LM336-2.5 reference provides a nominal 2.490-volt output (Vref). The actual voltage may vary from 2.390-volts to 2.590-volts. The reference voltage and temperature coefficient may be trimmed using the method indicated in the LM336-2.5 data sheet. The nominal value of the current reference resistor (Rref) connected to pin 14 of the DAC is 1240 Ohms, yielding a reference current (Iref) of 2.490 V / 1240 Ohms (Vref/Rref) = 2.008 milliamps. The eight-bit binary code applied to the DAC scales Iref by from 0/256 to 255/256, resulting in a current output (Io) of from zero (Iref•0/256) to 2.000 milliamps (Iref •255/256) full scale. Note that the sign of the DAC output current is opposite the sign of the reference (input) current. The output voltage is determined by multiplying the DAC output current (Io) by the value of the I/V converter gain resistor (Ro). Nominal full scale output voltage is 2.000 mA•2500 Ohms (Io F.S.•Ro) = 5.000-volts.

The circuit does not provide adjustments for offset or gain. Offset voltage adjustments should not be required, due to the low offset voltage specification of the LF355B op amp. If the offset voltage must be adjusted, add the offset trim circuit shown in the LF355B data sheet. The gain may be changed by changing the value of the I/V converter gain resistor (Ro).

The resistor connected to the non-inverting input of the op amp should be of the same value as the gain resistor for input bias current balancing. The 1240 Ohm resistor connected to pin 15 of the DAC and the 2500 Ohm resistor connected to pin three of the op amp may be eliminated with only a slight decrease in performance.

The MC1408-8 DAC requires power supplies of +5.0-volts and -5.0 to -15-volts; +/- 5.0-volt supplies were selected to minimize power consumption. The LF355B op amp requires bipolar supplies between +/-5.0-volts and +/-15-volts. -5.0-volts was selected for the negative rail for compatibility with the DAC, but may be replaced with -15-volts, if desired. The positive supply was chosen to be +15-volts to allow the limited output swing of the op amp to reach the five Volt upper input limit of the comparator.

The speed of the A-to-D conversion is limited by the DAC output settling time, the slew rate and settling time of the op amp, the response time and slew rate of the comparator and the time required to execute the successive approximation algorithm. The DAC output settling time and the comparator response time are negligible compared to op amp delays and the time required to execute the SA algorithm, and so may be ignored. The maximum voltage step input to the op amp is five volts, which requires one microsecond to slew and four microseconds to settle (see the LF355B data sheet). This delay is accommodated in the software; consult the listing for additional information. With a 12 MHz processor clock and the resulting one microsecond instruction cycle, an eight-bit conversion can be performed in under 300 microseconds. The unknown input voltage must be held constant for the duration of the conversion.

Obvious disadvantages to the successive approximation analog-to-digital converter presented here are the need for bipolar power supplies and the large number of microcontroller I/O pins required to control the DAC. The +15-volt supply could be eliminated by replacing the LF355B op amp with a single supply, five Volt, functional equivalent with outputs that swing rail-to-rail. The number of microcontroller I/O pins required to control the DAC could be reduced somewhat by substituting a seven or six bit DAC. The parallel input DAC could be replaced with a (more expensive) serial input DAC. Alternately, logic could be added to accept serial data from the microcontroller and present parallel data to the DAC.

The software for this application may be obtained by downloading from Atmel's BBS: (408) 436-4309. Consult the comment block at the beginning of the source code file for detailed information on features and operation.

Interfacing AT24CXX Serial EEPROMs with AT89CX051 Microcontrollers

Serial memory devices offer significant advantages over parallel devices in applications where lower data transfer rates are acceptable. In addition to requiring less board space, serial devices allow microcontroller I/O pins to be conserved. This is especially valuable when adding external memory to low pin count microcontrollers such as the Atmel AT89C1051 and AT89C2051.

This application note presents a suite of software routines which may be incorporated into a user's application to allow an AT89CX051 microcontroller to read and write AT24CXX serial EEPROMs. The software supports all members of the AT24CXX family, and may easily be modified for compatibility with any of the Atmel 8051-code compatible microcontrollers.

Hardware

A typical interconnection between an AT89CX051 microcontroller and an AT24CXX serial EEPROM is shown in Figure 1. As indicated in the figure, up to eight members of the AT24CXX family may share the bus, utilizing the same two microcontroller I/O pins. Each device on the bus must have its address inputs (A0, A1, A2) hard-wired to a unique address. In the figure, the first device recognizes address zero (A0, A1, A2 tied low), while the eighth recognizes address seven (A0, A1, A2 tied high). Not all members of the AT24CXX family recognize all three address inputs, limiting the number of some devices which may be present to less than eight. The exact number of devices of each type which may share the bus is shown in Table 1.

Bidirectional Data Transfer Protocol

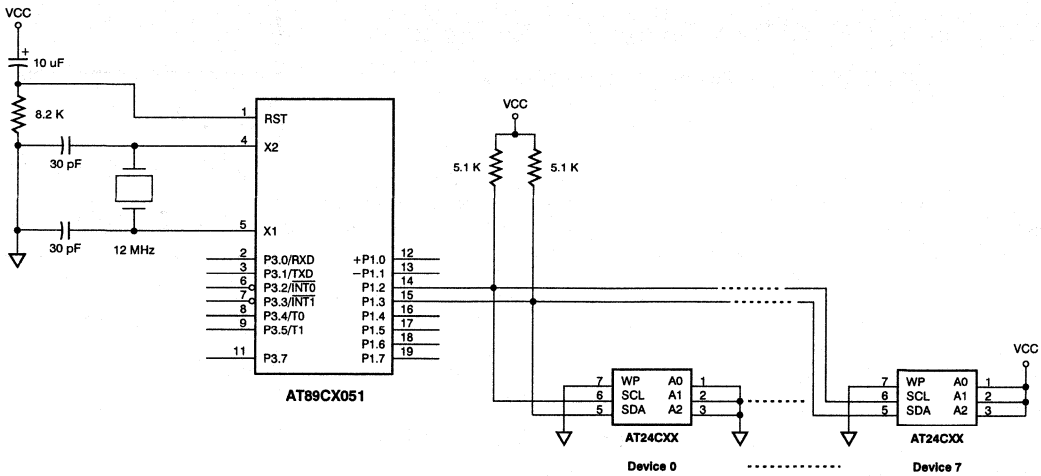
The Bidirectional Data Transfer Protocol utilized by the AT24CXX family allows a number of compatible devices to share a common two-wire bus. The bus consists of a serial clock (SCL) line and a serial data (SDA) line. The clock is generated by the bus master and data is transmitted serially on the data line, most significant bit first, synchronized to the clock. The protocol supports bidirectional data transfers in eight-bit bytes.

In this application, the microcontroller serves as the bus master, initiating all data transfers and generating the clock which regulates the flow of data. The serial devices

Table 1. Atmel's Two-Wire Serial EEPROM Family

Device	Size (Bytes)	Page Size (Bytes)	Max Per Bus	Addresses Used
AT24C01	1 K	8	1	None
AT24C01A	1 K	8	8	A0,A1,A2
AT24C02	2 K	8	8	A0,A1,A2
AT24C04	4 K	16	4	A1,A2
AT24C08	8 K	16	2	A2
AT24C16	16 K	16	1	None
AT24C164	16 K	16	8	A0,A1,A2
AT24C32	32 K	32	8	A0,A1,A2
AT24C64	64 K	32	8	A0,A1,A2

Figure 1. Typical Circuit Cinfiguration



present on the bus are considered slaves, accepting or sending data in response to orders from the master.

The bus master initiates a data transfer by generating a start condition on the bus. This is followed by transmission of a byte containing the device address of the intended recipient. The device address consists of a four-bit fixed portion and a three-bit programmable portion. The fixed portion must match the value hard-wired into the slave, while the programmable portion allows the master to select between a maximum of eight slaves of similar type on the bus.

AT24CXX serial EEPROMs respond to device addresses with a fixed portion equal to '1010' and a programmable portion matching the address inputs (A0, A1, A2). Not all members of the AT24CXX family examine all three address inputs; Table 1 shows which of the three address inputs are valid for each member of the family.

The eighth bit in the device address byte specifies a write or read operation. After the eighth bit is transmitted, the master releases the data line and generates a ninth clock. If a slave has recognized the transmitted device address, it will respond to the ninth clock by generating an acknowledge condition on the data line.

A slave which is busy when addressed may not generate an acknowledge. This is true for the AT24CXX when a write operation is in progress.

Following receipt of the slave's address acknowledgment, the master continues with the data transfer. If a write operation has been ordered, the master transmits the remaining data, with the slave acknowledging receipt of each byte. If the master has ordered a read operation, it releases the data line and clocks in data sent by the slave. After each byte is received, the master generates an acknowledge condition on the bus. The acknowledge is omitted following receipt of the last byte. The master terminates all operations by generating a stop condition on the bus. The master may also abort a data transfer at any time by generating a stop condition.

Refer to the AT24CXX family data sheets for detailed information on AT24CXX device operation and Bidirectional Data Transfer Protocol bus timing.

The software for this application may be obtained by downloading from Atmel's BBS: (408) 436-4309. Consult the comment block at the beginning of the source code file for detailed information on features and operation.

Interfacing AT93CXX Serial EEPROMs with AT89CX051 Microcontrollers

Serial memory devices offer significant advantages over parallel devices in applications where lower data transfer rates are acceptable. In addition to requiring less board space, serial devices allow microcontroller I/O pins to be conserved. This is especially valuable when adding external memory to low pin count microcontrollers such as the Atmel AT89C1051 and AT89C2051.

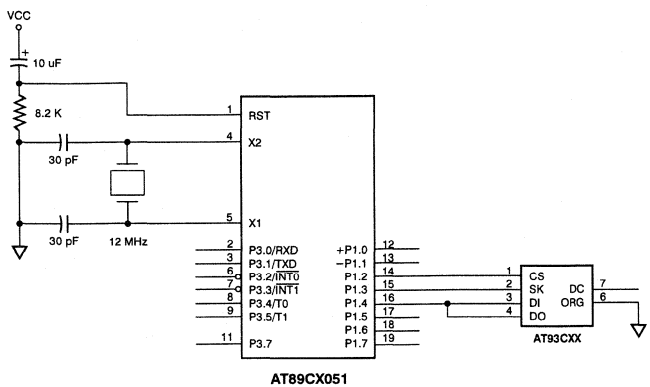
This appplication note presents a suite of software routines which may be incorporated into a user's application to allow AT89CX051 microcontrollers to read and write AT93CXX serial EEPROMs. All seven AT93CXX device functions are supported: read, write, write all, erase, erase all, erase/write enable and erase/write disable. The routines are general purpose, supporting both eight-bit and sixteen-bit accesses to all members of the 93CXX family. In addition, both three-wire and four-wire configurations are supported.

The AT93CXX may be connected to the AT89CX051 microcontroller in either a three-wire (Figure 1) or four-wire (Figure 2) configuration. In the three-wire configuration, the EEPROM serial data in (DI) and serial data out (DO) pins are both connected to the same microcontroller I/O pin, thereby saving a pin. This is possible because the microcontroller I/O pins can be dynamically reprogrammed as input or output.

Note the strapping of the AT93CXX ORG pins shown in Figure 1 and Figure 2. The ORG (internal organization) pin selects eight-bit data when grounded and sixteen-bit data when floating or tied to Vcc. The ORG pin connections shown in the figures are for illustration only; eight-bit or sixteen-bit data may be selected in either the three-wire or four-wire configuration.

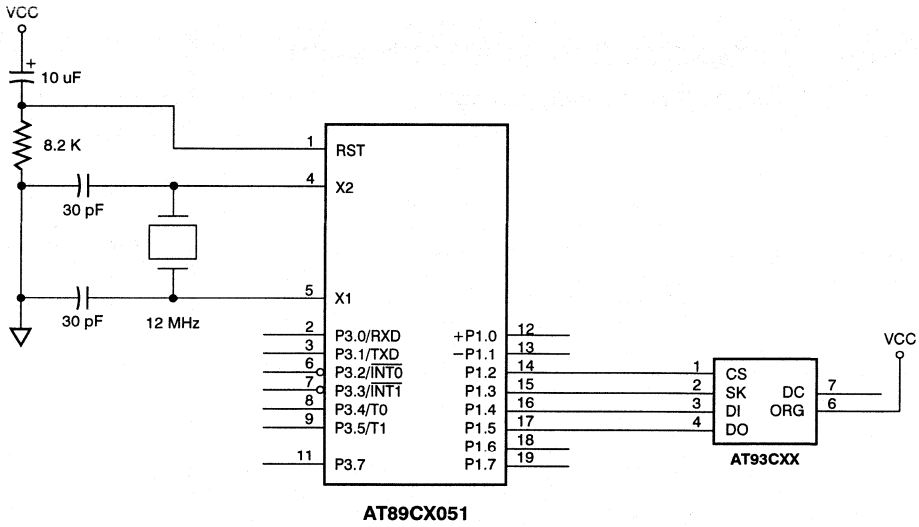
The software for this application may be obtained by downloading from Atmel's BBS: (408) 436-4309. Consult the comment block at the beginning of the source code file for detailed information on features and operation.

Figure 1. Three-Wire Configuration



0490A

Figure 2. Typical Circuit Configuration



Contents

5

Programming Vendors

Atmel Corporation works closely with major suppliers of programming equipment that support our family of Flash microcontrollers.

Atmel has a program in place which certifies the programming vendors, and a complete list can be found on our Bulletin Board at (408) 436-4309. This list will be updated frequently to reflect additional support.

The following is a list of vendors that currently offer support for the Atmel Flash based microcontroller family. If a vendor of your choice does not appear on this list, please contact Atmel at (408) 436-4295, to enroll them in Atmel's certification program.

Programmer	Address	Telephone & Fax	Programmer Model #'s	Software Revision #	Product
ACD	*Larry Wood* 21018 Northeast 92nd St. Redmond, WA 98053	Tel: 206-868-9444 Fax: 206-868-9444			
Advantech	*Jiheng Ha* *Product Marketing* *Jeff Chen* *Business Manager* 4F No.108-3, Ming-Chuan Road Shing-Tien City Taipei, Taiwan, R.O.C.	Tel: 011-886-2-218-4567 Fax: 011-886-2-218-8478 BBS: 011-886-2-218-5434	PC-UPROG from Advantech CM3000 from Logical Devices 9860from American Reliance M1880 from Minato OMPRO-II from Dataman Labtool-48	V1.83 V2.2 V1.0 V1.0	AT89C51 AT89C52 AT89C1051 AT89C2051
Advantest	*Osamu Kasama* 77-1 Miyako, Namekawa-cho Hikigunn, Saitama 355, Japan	Tel: 011-81-493-56-4433 Fax: 011-81-493-57-1092	R4945 VerH00		AT89C51
Advin Systems	*Wing Hui* *VP Engineering* 1050-L East Duane Avenue Sunnyvale, California 94086	Tel: 408-243-7000 Fax: 408-736-2503 Sales: 800-627-2456 BBS: 408-737-9200	PILOT-U40 PILOT-U84 PILOT-145 PILOT-V32	10.75 10.81C	AT89C51 AT89C52 AT89C2051 AT89C1051
Allen Systems	2346 Brandon Road Columbus, Ohio 43221	Tel: 614-488-7122	PB-51/11		AT89C51
American Reliance			9860		
Ando Electronics	*Yoshihiro Homma* 4-19-7 Kamata, Ohta-Ku Tokyo 144, Japan 7617 Standish Place Rockville, Maryland 20855 USA	Tel: 011-81-3-3733-1151 Fax: 011-81-3-3739-7363 USA Tel: 301-294-3365 Fax: 301-294-3359	AF9705 AF9810 AF9704 AF9780		AT89C51
Ashling Micro Systems Inc.	*Michael Healey* Managing Director Plassey Technological Park Limerick, Ireland	Tel: 011-44-1489-577516 Fax: 011-353-61334477 EMAIL: ashling@iol.ie	CTP51	4.6.3	AT89C51 AT89C52
Aval Corp	*Satoru Mikuni* 1-1-2 Manpukuji, Asou-ku, Kawasaki Kanagawa 215, Japan	Tel: 011-81-44-952-1322 Fax: 011-81-44-952-1332	PECKER-50		AT89C51
B & C Microsystems	*Cesar Hernandez* *Application Engineer* 750 North Pastoria Avenue Sunnyvale, CA 94086	Tel: 408-730-5511 Fax: 408-730-5521 BBS: 408-730-2317	PROTEUS (single device)	3.7K 3.7P	AT89C51 AT89C52
BP Microsystems	*Tracy Wilson* *Bill Cates* *Oliver Lazares* *Puarma Murthy* *Manager of Software Support* 1000 N. Post Oak RD., Suite 225 Houston, Texas 77055-7237	Tel: 713-688-2620 Fax: 713-688-0920 Tel: 800-225-2102 Oliver x520 BBS: 713-688-9283 up to 14.4 KBPS	BP-1200/40 EP-1140 BP 1148	2.26 3.03 3.05b 3.08	AT89C51 AT89C52 AT89C2051 AT89C1051
Bytek	*Joe Martiello* 543 North West 77th Street Bocaraton, Florida 33487-1323	Tel: 407-994-3520 Fax: 407-994-3615	EZ Writer Multitrk 4000 Multitrk 2000 Multitrk 1000		AT89C51
CEIBO CEIBO GmbH	*Leor Weinstein* *Victor Waiman* Merkazim Bldg, Ind Zone 5 Maksit Herzelia, 46120, Israel *Roly Schwartzman* 7 Edgestone Ct. Florissant, MO 63033 Menachem Kimron Rheinstr. 32 D-64283 Darmstadt Germany	Tel: 011-972-9-555-387 Fax: 011-972-9-553-297 Tel: 314-830-4089 Fax: 314-830-4083 Tel: 0049-6151-27505 Fax: 0049-6151-28540	MP-51 ET-PIC 4000 ET-PIC 12000 (gang programmer) ET-PIC 5000 ET-PIC 1000 ET-SEPROG ET-PIC 51	2.26 4.14 1.68 3.07 2.1 1.01	AT89C51 AT89C52 AT89C2051 AT89C1051
Celectronic	*Klaus Leistner* *Volker Czmok* Nordlichtrasse 63-65 D-13405 Berlin, Germany	Tel: 011-030-413-6075 Fax: 011-030-49-913-6098	Promicron 1000	5.61	AT89C51 AT89C52

Programmer	Address	Telephone & Fax	Programmer Model #'s	Software Revision #	Product
Data I/O Corp	*Vaclav Klimsa* *Keith Miller (Mgr Mfg interface)* *Mike Durham (Mfg interface)* P.O. Box 97946 10525 Willows Rd NE Redmond, Washington 98073-9746 *Susan Kuenster* Corporate Communications	Technical Assistance 800-247-5700 Service 800-735-6070 Sales 800- 332-8246 Susan: 206-867-6884 Fax: 206-869-7423 Mfg: 206-867-6841 Mike: 206-867-6841 BBS: 206-882-3211 KeepCurrent:206-881-3465 EMail: Durham@mailgwy.Data-IO.COM at CCGATE	UNISTE site 48 UNISTE pinsite UNISTE 48HS 2900 (no 5V support) 3900 UNISITE chipsite Auto site 2500 Chip Lab-48 PSX500 PSX1000	4.3 4.7 3.4 2.4 3.0 3.3 3.3 4.9 4.9	AT89C51 AT89C52 AT89C51 AT89C52 AT89C2051 AT89C1051
Dataman Programmers, LTD	*Gary Comer* Technical Support Manager Station Road Maiden Newton Dorchester, Dorset, DT2 OAE United Kingdom	Tel: 011-44-0-1300-320719 Fax: 011-44-0-1300-321012 BBS: 011-4-4-0-1300-321095 Modem:V.34/V.FC/V.32bis	S4 S4 MCS-51 (module)	1.20 1.21	AT89C51 AT89C52
Datatech Int. Inc.	*Dan Ho* ON319 Stanley St. Winfield, IL 60190 P.O. Box 549 Villa Park, IL 60181	Tel: 708-832-8818 Fax: 708-653-9087	PROMA-3		
Elan Digital Systems Limited	*Julian Hartridge* Device Support Engineer Little Park Farm Rd. Segensworth West Fareham Hants Po15 5SJ UK	Tel: 011-44-489-579-799 Fax: 011-44-189-577-516	6000APS	K2.07	AT89C51
Electronic Engineering Tools	*John Kim* 544 Weddell Dr., Ste. 6 Sunnyvale, CA 94089 *gsh Systemtechnik* Postfach 600511 81025 Munchen	Tel: 408-734-8184 Fax: 408-734-8185 Tel: 089-8343047 Fax: 089-8340448	ALLMAX ALLMAX PLUS PROMAX	1.6 2.0 2.1	AT89C51 AT89C52 AT89C2051
Electronix Corp.	*Jason Clinger* 900 South 300 West Salt Lake City, Utah 84115	Tel: 801-486-2825 Fax: 801-484-9285	X0SI Programmer	1.0	AT89C2051 AT89C1051
EHA- Elekronik	*Walter Hacklander* Hittorfstabe 17 50735 Koln	Tel: 011-0221-7602252 Fax: 011-0221-766923	PROM8952	3.0 3.0	AT89C2051 AT89C1051
Emulation Technology	*Don Krenn* 2344 Walsh Ave., Bldg F Santa Clara, CA 95051	Tel: 408-982-0660 x118 Fax: 408-982-0664 BBS: 408-982-9044	ET-PIC 12000 ET-PIC 10000 ET-PIC 4000 AS-20-20-01S-6(SOIC to DIP programming adaptor)	8.69 2.3 3.0	AT89C51 AT89C52 AT89C2051 AT89C1051
Ertec	*Hr. Nickel* Am Pestalozzining 24 91058 Erlangen	Tel: 011-49-9131-77000 Fax: 011-49-9131-110010			
GTEK, Inc.	*Cal Edmonds* RAM# 1916 399 Highway 90 Bay St. Louis, MS 39520	Tel: 601-467-8048 Fax: 601-467-0935			AT89C51
Hi-Lo System Research Co., LTD	*Grason Kirk* 4F, No. 2, Sec. 5,, Ming-Shen E. Rd. Taipei, Taiwan ROC	Tel: 011-886-2-7640215/7633931 Fax: 011-886-2-7566403/7601559 Internet: 101400.1555@computserve.com	ALL03A ALL07+ PAC-DIP 40	3.30 3.33	AT89C51 AT89C52 AT89C2051 AT89C1051
Hi Tools Inc.	*Ulaf Pfeiffer* 2055 Gateway Place, Suite 400 San Jose, CA 95110	Tel: 408-481-9486 Fax: 408-441-9486			
ICE Technology	*Sheila Boakes* Managing Director - Commercial John Lamb Technical Support Unit 4, Penistone Court Station Buildings Penistone, S. Yorkshire 530 6HG UK	Tel: 011-44-01226-767404 Fax: 011-44-01226-3704.34 BBS: 011-44-01226-761181	Micromaster 1000/1000E Micromaster LV	 3.07 3.07	AT89C51 AT89C52 AT89C1051 AT89C2051
International Microsystems Inc.	*Mason Tom* *Peter A. Schade/ President* 521 Valley Way Milpitas, California 95035	Tel: 408-942-1001 Fax: 408-942-1051	Eprom 1 (single device) M4016 (gang programmer)	2.64 2.74	AT89C51 AT89C52
iSYSTEMS The Tool Company	*Hr. Gerd Punsmann* Einsteinstrasse 5 85221 Dachau *Germany*	Tel: 011-49-08131-25083 Fax: 011-49-08131-14024 BBS: 011-49-8131-1687 Email: 100632.42@compuserve.com	iUP8000 SEPROG	2.5 5.44	AT89C51 AT89C52 AT89C1051 AT89C2051

5

Programmer	Address	Telephone & Fax	Programmer Model #'s	Software Revision #	Product
Leap Electronic Co.	6F, No 4, Lane 609, Sec. 5 Chung Hsin Rd. San Chung City Taipei Hsien, Taiwan. R.O.C. P.O. Box 91-249 Taipei	Tel: 011-02-999-1860 Fax: 011-02-999-0015	Leap-U1 Leaper-5	3.12 1.20	AT89C52 AT89C51
Link Instruments	*Bill Lam* *Sales Manager* 369 Passaic Ave., Suite 100 Fairfield, N.J. 07004	Tel: 201-808-8990 Fax: 201-808-8786	CLK-3100 (w/ 875X adapter)	4.55 4.95 4.98 4.98	AT89C51 AT89C52 AT89C1051 AT89C2051
Logical Devices	*David Mot* 264 South West 12th Ave. Deerfield Beach, Florida 33442 *Bill Hall/ Software Engineer* *Jeff Williams* *Conor Mcaleer* 130 Capital Drive, Suite A Golden, Colorado 80401	Tel: 800-331-7766 Tel: 305-428-6868 Fax: 305-428-1811 BBS: 305-428-8014 Tel: 303-279-6868 Fax: 303-279-6869 Email: logdev@henge.com	ALLPRO-88 ALLPRO-88/XR ALLPRO-40 GANGPRO-8 GANGPRO- SII CM3000 Chipmaster 2000	ver2.4 rev 0 2.4 2.5	AT89C51 AT89C52 AT89C2051 AT89C1051
Logical Systems	*Lynn Burko* PO Box 6184 Syracuse, NY 13217	Tel: 315.478-0722 Fax: 315-475-6753	PA51-FC (adapter) Sunshine EW901BN		AT89C51
Magnadata	*HR. Schuster* Hauptstabe 1 61389 Schmitten	Tel: 011-06082-7421615 Fax: 011-06082-7423448			
Micropross	5, Rue Denis Papin 59650 Villeneuve D'Ascq France	Tel: 011-33-204-79040 011-33-204-79369	ROM 3000U		
Mid-Tech Computing Devices	*John Dybowski* *Technical Director* P.O. Box 218 45 Monson Rd. Stafford Springs, Conneticut	Tel: 203-684-2442 Fax: 203-684-2442	2051 Design Center Complete design environment	1.4 1.4	AT89C2051 AT89C1051
MQP Electronics	*Pat Crowe* Technical Director Unit 2 Park Road Centre Malmesbury Wiltsuire SN16 OBX England	Tel: 011-44-0666-825146 Fax: 011-44-1666-825141	200P Adapter AD3 (PLCC pin converter AD45) System 2000 (Gang Programmer) Model S2510	PROMDRIVE V8.10 V8.69	AT89C51 AT89C52 AT89C2051 AT89C1051
Mandeno Granville	*Jim Granville* 128 Grange Rd. Mt. Eden 3 Auckland, New Zealand	Tel: 011-649-6300-558 Fax: 011-649-6301720	ICEP2051	DBGX51 V4.65B	AT89C51 AT89C52 AT89C2051 AT89C1051
MCT PROMA3	*Gilbert* *Atmel Hong Kong*				
Minato Electronics	*Matsatoshi Koyama* 4105 Minami Yamada-cho Kouhoku-ku Yokohama, Kanagawa 223 USA 3628 Madison Ave., Suite 5 North Highlands, CA 95660	Tel: 011-81-45-591-5611 Fax: 011-81-45-591-6451 USA Tel: 916-348-6066 Fax: 916-348-0926	M1880 1890A 1891		AT89C51
Needham Electronics	*Brian Neal* 4630 Beloit Dr. Ste 20 Sacramento, Calif. 95838	Tel: 916-924-8037 Fax: 916-924-8065 BBS: 916-924-8094	EMP-20	V1.45 V1.92 V2.16 V2.16	AT89C51 AT89C52 AT89C2051 AT89C1051
Owen	Ringstr 11 Postfach 1104 D-6798 Kusel Germany	Tel: 011-49-6381-5085	AP-II		
Phyton Inc.	*Issac Grinberg* President 7206 Baypark Way 2nd Floor Brooklyn, NY 11204	Tel: 718-259-3191 Fax: 718-259-3191 compuserve: 73232,251	EBP	3.3	
Prologic Systems	*Frank Deines* 557-p Burbank St. Broomfield, Co 80020	Tel: 303-460-0103 Fax: 303-469-5565	UNIV-DBM	15.50	AT89C52 AT89C51
SMS GmbH	*Martin Hinz* *Product Manager* Im Grund 15 D-88239 Wangen Germany	Tel: 011-49-7522-97280 Fax: 011-49-7522-972850 BBS: 49-7522-9728-88	SPRINT EXPERT SPRINT OPTIMA PLUS 48 MULTISITE	C/93 A95 B/94 B/94 A/95 A/95	AT89C51 AT89LV51 AT89C52 AT89LV52 AT89C2051 AT89C1051

Programmer	Address	Telephone & Fax	Programmer Model #'s	Software Revision #	Product
Stag Programmers LTD.	*Paul Chanel* *Director of Sales* *Silver Court, Watchmead* *Welwyn Garden City* *Herts AL7 1LT U.K.* *Richard Hough* *Applications Engineer* *Chris Humphreys* Engineering Manager	UK: 011-44-1707-332148 Fax: 011-44-1707-371503	Stag Quasar 1040 Stag Quasar 1084 Eclipse (EPU84P PLCC & EPU48D modules)	2.0 21-12-94 4.7.29 4.4.22 4.12.22	AT89C51 AT89C52 AT89C2051 AT89C1051
Sunshine Systems	*James Huang* *Managing Director* RM. 304, 3F, No 2, Lane 137 Sec. 5, Ming Shen E. RD., Taipei, Taiwan, R.O.C.	Tel:011-02-7633732/7660206 /7685370 Fax: 011-02-886-2-7654065	EW901BN Power 100	8.32.26 8.32.26	AT89C51 AT89C52
Sunrise Electronics	*Anh Le* *Director of Engineering* *Joe Petralia* *Sales Manager* 675 Brea Canyon Rd. Suite 6 Walnut, California 91789	Tel: 909-595-7774 Fax: 909-594-7009	T-10 Z-3000 (Gang Programmer)	3.15 1.4 (gang programmer 3.22 3.32 3.32	AT89C51 AT89C52 AT89C2051 AT89C1051
System General	*Alvin Apvan* *Technical Support* *Tim Morse* *Marketing Sales* 1603 A South Main St. Milpitas, Calif. 95035	Tel: 408-263-6667 Fax: 408-262-9220 BBS: 408-262-6438 Sales: 1-800-967-4776	TURPRO-1/FX (v2.15) TURPRO-1 (v2.15) TURPRO-840 (v2.21B) APRO (v1.24)	2.21 2.21 2.01 1.19 2.15 2.21Blevel 1 2.21 Blevel1	AT89C51 AT89C52 AT89C2051 AT89C1051
Strebor	1008 N Nob Hill Dr. America Fork, UT 84003	Tel: 801-756-3605	PLP-SI		
Teradyne	*Dominic Haigh* 2625 Shadelands Dr. Walnut Creek, California 94598	Tel: 510-932-6900 Fax: 510-932-7965 Email: WWWmail- mktg@atb.teradyne.com	Z1800 Series Incircuit Tester		AT89C52 AT89C51
Tribal Microsystems	*Robert Kruger* 44388 S. Grimmer Blvd. Fremont, CA 94538	Tel: 510-623-8859 Fax: 510-623-9925 BBS: 510-623-0430	FLEX-700 UNIVERSAL TUP-400 UNIVERSAL 4 GANG MODULE TUP-51F (PDIP) TUP-51F-PL (PLCC)	3.29 PGM51.EXE 3.33 4.22 3.35	AT89C51 AT89C52 AT89C2051 AT89C1051
Vail Silicon Tools	*Gabby Green* Product Marketing 692 South Military Trail Deerfield Beach, Florida 33442	Tel: 305-570-558880 Fax: 305-428-1811			
Xeltek	*Joo Nam Kim* *2563 Ryder St.* *Santa Clara, CA 95051*	Tel: 408-524-1929 Fax: 408-245-7084 BBS: 408-245-7082	SUPERPRO II	2.1 2.01 2.2 2.2A	AT89C51 AT89C52 AT89C2051 AT89C1051

Third Party Tool Vendors

Atmel Corporation works closely with many "third-party" vendors who provide support tools for our wide variety of 80C51 based microcontroller derivatives.

The following is a non-inclusive list of some of the vendors that offer support for the Atmel AT89C series microcontroller family.

Any system that supports an 80C51-based microcontroller will be compatible with the Atmel AT89C series of microcontrollers.

Microcontroller Third Party Tool Vendors

5

0514A

Table 1. Third Party Tool Vendors

Company	Contact Name	Phone / Fax	Address	Product Name	Description
EMULATOR COMPANIES					
ALLEN SYSTEMS	John Allen	Tel: 612-488-7122	2346 Brandon Road Columbus, OH 43221	Little Byte-51/ PB-51/11	8051-based Single board computer
AMERICAN ARIUM	Jeff Acampora; VP Sales & Mkting	Tel: 714-731-1661	14281 Chambers Road, Tustin, CA 92680		
APPLIWARE ELEKRONIK GmbH	Hr. Heinrich Bals	Tel: 011-49-8061- 90940 BBS: 011-49-8061- 377190 Fax: 011-49-8061- 37298	Westend Strasse - 4 83043 Bad Aibling, Germany	A1CE51	Emulator/Software/ Debugger/ C-Compiler/ Programmer Development Kits/Eval Boards/Modules
ASHLING MICROSYSTEMS, LTD	Michael Healy	Tel: 011-353-61-33-44- 66 Fax: 011-353-61-33- 4477 Email: ashling@iol.ie Tel: 011-49-08202- 1276 Fax: 011-49-08202- 8745	Plassey Technological Park, Limerick, Ireland Waldstrbe 18 86510 Baindlkirch, Germany		Emulator/Software
BRENDES DATENTECHNIK	Hr. Dr. Brendes	Tel: 49-531-506499 Fax: 49-531-506499	Lebacher Strasse 122 38116 Braunschwweig, Germany		Emulator
BVG ELECTRONIC	Hr. Montanus	Tel: 49-8106-5794 Fax: 49-8106-33921	Karl-Bohm-Strasse 137 85598 Baldham, Germany		
CACTUS LOGIC	Joel Lagerquist	Tel: 800-847-1998	180 North Venedo Ave., Pasedena, CA 91107		
CIEBO	Hr. Kimron	Tel: 49-6151-27505 Fax: 49-6151-28540	Rhein Strasse 32 64283 Darmstadt, Germany		Emulator/Software
CEIBO ITD		Tel: 617-863-9927			
CMX Company	Mark Moran	Tel: 508-872-7675 Fax: 508-620-6828	5 Grant St. Ste. C Framingham, MA 01701	STIMGATE	Soft Emulator-new productivity tool for embedded processors
DEEMAX		Tel: 886-35-723311	Taiwan		
DR. KROHN & STILLER	Mr. Maihoefner Dr. Krohn	Tel: 49-89-61-00-0012 Tel: 49-89-61-00-0011 Fax: 49-89-61-00-0099	Ottobrunner Strasse 28, D-82008 Unterhaching, Germany	E8	Emulator/Software
EMBEDDED SYSTEM PRODUCTS, INC	Ron Hodge Rhonda Warzecha	Tel: 713-728-9688 Tel: 800-525-4302 Fax: 713-728-1049 Email: Sales@esphou.com	11501 Chimney Rock, Houston, TX 77035-2900	RTXC RTXCio RTXCfile RTXCnet	Multitasking O.S. Input/Output Subsystem DOS compatible file system Real-time Networking including TCP/IP protocols
EMULATION TECHNOLOGY INC.	Don Krenn	Tel: 800-995-4381 Fax: 408-982-0664	2344 Walsh Ave. Bldg. F Santa Clara, CA 95051		Emulator/Software
ENGELMANN & SCHRADER		Tel: 49-05148-286 Fax: 49-05148-853	Am Fuhrengehege 2 29351 Eldingen, Germany		Emulator/Software
HI-LO SYSTEMS RESEARCH CO., LTD		Tel: 886-2-7640215	Taiwan		Device Programmer

Table 1. Third Party Tool Vendors (continued)

Company	Contact Name	Phone / Fax	Address	Product Name	Description
HITEX SYSTEMENTWICKLUNG GmbH	Mr. Otterstaetter Hr. Christoph	Tel: 49-721-9628-181 Fax: 49-721-9628140	Greschbach Strasse 12, D-76229 Karlsruhe, Germany		Emulator
HUNTSVILLE MICROSYSTEMS, INC	Jim Bell	Tel: 205-881-6005 Fax: 205-882-6701	3322 South Memorial Parkway, Bldg. 500, Huntsville, AL 35801		Emulator/Software
iSYSTEMS	Hr. Punsmann	Tel: 49-08131-25083 Fax: 49-08131-14024	Einstein Strasse 5, 85221 Dachau		Emulator/Software
IWASAKI ELECTRONICS CO, LTD		Tel: 81-3-863-3025	Japan		
KONTRON		Tel: 49-08165-77444 Fax: 49-08165-77128	Oskar-von-Miller-Strasse 85386 Eching, Germany		
LAUTERBACH DATENTECHNIK		Tel: 49-08104-889430 Fax: 49-08104-894349	Fichten Strasse 27 85649 Hofolding, Germany		Emulator
LINDNER ELECTRONICS, INC		Tel: 603-523-9005	PO Box 68, Canaan, NH 03741		Programmer for the AT89C51
MANDENO GRANVILLE	Jim Granville	Tel: 64-9-6300-558	128 Grange Rd, Mt. Eden 3, Aukland, New Zealand	ICEP2051	Development station for the AT89C2051
METALINK CORP	Jack Blankenship	Tel: 602-926-0797 Tel: 49-8091-2046 Fax: 49-8091-2386	325 East Elliot Rd., Suite 23, Chandler, AZ 85225 Westring 2 85614 Kirchseeon, Germany		Emulator
MICRO TIME		Tel: 886-2-88-11791	Taiwan		
MICROTEK INTERNATIONAL	Kevin Jagla	Tel: 503-645-7333 x449	3300 NorthWest 2, 11th Terrace, Hillburough, OR 97124		
MIDTECH COMPUTING DEVICES	John Dybowski	Tel: 203-684-2442 Fax: 203-684-2443	45 Monson Road Stafford Springs, CT	2051 Design Center	AT89C2051 development system
NOHAU CORP NOHAU DANMARK NOHAU ELEKRONIK GmbH GERMANY NOHAU UK LTD NOHAU ELEKTRONIK SWEDEN	Bill Matsumoto	Tel: 408-866-1820 Tel: 43-44-60-10 Fax: 43-44-60-20 Tel: 49-7043-40247 Fax: 49-7043-40521 Tel: 0962-733-140 Fax: 0962-735-408 Tel: 040-92-24-25 Fax: 040-96-81-61	51 East Campbell Ave., Suite 144, Campbell, CA 95008 Goethe Strasse 4 75433 Maulbronn		Emulator
ORION INSTRUMENTS	Ken Schoggins: c/o Macrotorn	Tel: 415-327-8800 Fax: 415-327-9881 Tel: 011-49-89-42080	180 Independent Drive, Menlo Park, CA 94025 Stahlgruberring 2 81804 München, Germany		Emulator
RAINSONNANCE	Roth	Tel: 011-49-8144-1536 Fax: 011-49-8144-1535	Wald Strasse 16 82284 Grafrath, Germany		Emulator/Software
RHOMBUS		Tel: 803-676-0012 Fax: 803-676-0015	P.O. Box 871 Mauldin, SC 29662	2051-PD	AT89C2051 Development System

Table 1. Third Party Tool Vendors (continued)

Company	Contact Name	Phone / Fax	Address	Product Name	Description
SIGNUM SYSTEMS	Jerry Lewandowski	Tel: 805-371-4608 Fax: 805-371-4610	171 East Thousand Oaks Blvd., Suite #202, Thousand Oaks, CA 91360	USP-51	Emulator Base Unit, 20 Mhz, 128K memory, 32K *80 Trace Buffer
	Donald Mull	Tel: 510-353-1616 Fax: 510-353-1618	200 Brown Rd. #26 Fremont, CA 94539		
	Bonacker	Tel: 49-7244-94012 Fax: 49-7244-92128	Rohrackerweg 11 76297 Stutensee, Germany	POD-51	Probe for 8XC51/52, 80C31/32, AT89C51
SOPHIA SYSTEMS AND TECHNOLOGY/ Sales Dept	David Freemire	Tel: 415-493-6700 Fax: 415-493-4648	777 California Ave., Palo Alto, CA 94304		
	Mr. Kohno	Tel: 813-33487001 Fax: 813-3348-2446	NS Bldg., 2-4-1, Nishi-Shinjuku, Tokyo, Japan	IP 2813R	Emulator
	Hr. Santen	Tel: 49-721-377044 Fax: 49-721-377241	Postfach 2928 76016 Karlsruhe, Germany		
SUNSHINE		Tel: 886-2-7660206	Taiwan		Emulator/Software
	C/O Innotron	Tel: 49-20240526 Fax: 49-20240522	Nesselberg Strasse 1 42349 Wuppertal, Germany		
U. S. SOFTWARE	Don Dunstan	Tel: 503-641-8446 Tel: 800-356-7097 Fax: 503-644-2413	4215 Northern West Science Park Drive Portland, OR 97229		Emulator
TRANSFERTECH	Hr. Pogrzeba	Tel: 49-531-890255 Fax: 49-531-890355	Cyriaksring 9a 38118 Braunschweig, Germany	FCM-8051	Emulator Fuzzy- SW
UNILAB	Buddy Baker	Tel: 514-630-4600 Fax: 514-630-4680	235 Place Frantenac Pointe-Claire, PQ, Canada H9R4Z7	UNILAB	8051 Development System
VAIL SILICON TOOLS	Product Marketing: Gabby Green	Tel: 305-570-5580 Fax: 305-428-1811	692 South Military Trail, Deerfield Beach, FL 33442		
WHYMON	Hr. Brugger	Tel: 49-0041-41-852212 Fax: 49-0041-41-85238	CH-5734 Reinach, Germany		Emulator/Software
LOGIC ANALYZER					
HEWLETT- PACKARD	John Marshall PME	Tel: 800-447-3282 Tel: 719-590-5985 Fax: 719-590-5054	1900 Garden of the Gods Road, CSO CO 80901; PO Box 2197, CSO CO 80901-2197	HP1660 Family HP16500 HPE24158	Emulator/Software Logic Analyzer Frame for 8051 Logic Analyzer Family 80C51 Preprocessor
	Hr. Walkamm	Tel: 49-07031-146513 Fax: 49-07031-146429	Schickard Strasse 2 71004 Boblingen, Germany		
	Chuck Small Product Marketing Engineer	Tel: 719-590-2006 Fax: 719-590-5054	1900 Garden of the Gods Road, CSO CO 80901; PO Box 2197, CSO CO 80901-2197	HP647886	
TEKTRONIX, INC	Steve Hass; Technical Support Center	Tel: 800-426-2200 Fax: 503-690-3959	18700 Northwest Walker Rd., Bldg. 94 Dock, Aloha, OR 97006		

Table 1. Third Party Tool Vendors (continued)

Company	Contact Name	Phone / Fax	Address	Product Name	Description
SOFTWARE DEVELOPERS					
2500 A.D. SOFTWARE	Vicky Beske	Tel: 800-843-8144 Fax: 719-395-8206 Tel: 719-395-8683	109 Brookdale Ave., Buena Vista, CO 81211	8051C Compiler	C Compiler Includes: macro cross assembler, linker, librarian, high level simulator/ debugger and object libraries
APPLIWARE	Hr. Heinrich Bals	Tel: 011-49-8061-90940 BBS: 011-49-8061-377190 Fax: 011-49-8061-37298	Westend Strasse - 4 83043 Bad Aibling, Germany	WORKS-51 WORKS PLUS-51	Integrated Devlopment Environment (IDE), HLL Debugger, C-Compiler, Macro assembler, Simulator complete development packs
ARCHIMEDES SOFTWARE	Timothy Hang	Tel: 800-338-1453 Tel: 206-822-6300 Fax: 206-822-8632	303 Park Place Center Suite G Kirkland, WA 98033	C-8051PC 5.0 Sim8051PC 5.0	C-Cross Compiler kit for the MCS-51 family Simulator/debugger for the MCS-51 family
AVOCET SYSTEMS, INC	Tony Taylor	Tel: 800-448-8500 Tel: 207-236-9055 Fax: 207-236-6713	P.O. Box 490 120 Union St., Rockport, MA 04856	AvCase II 8051 Avocet C AVA51 AVS51 AVSIM51	Development pkg includes: C compiler Macro assembler, source level simulator, remote monitor debugger C Compiler/ assembler Macro Assembler C Source level simulator Assembly level simulator
BSO/TASKING	 Shailui Gargeya 	Tel: 800-458-8276 Tel: 617-320-9400 Tel: 49-7152-979910 Fax: 49-7152-9799120	128 Technology Center P.O. Box 9164 Waltham, MA 02254-9164 Norfolk Place, 333 Elm St., Dedham, MA 02026-4530 Steinbeis Strasse 4 71229 Leonberg, Germany		C Compiler Software
BINAR TECHNOLOGY	Bob Oleksiak	Tel: 508-369-9556 Fax: 508-369-9549	463 Autumm Lane, Carlisle, MA 01741; PO Box 541, Carlisle, MA 01741		
BYTE-BOS-INTEGRATED SYSTEMS	Craig Rand	Tel: 800-788-7288 Fax: 714-851-2267	451 Zuni Drive, Delmar, CA 92014		Multitasking O.S.

Table 1. Third Party Tool Vendors (continued)

Company	Contact Name	Phone / Fax	Address	Product Name	Description
CHIP TOOLS, INC	Ken Anderson	Tel: 905-274-6244 Fax: 905-891-2715	1232 Stavebank Rd., Mississauga, Ontario, Canada, L5G2V2	ChipView-SI	Simulator/ ROM Debugger
CIMETRICS TECHNOLOGY	Conray Wharff	Tel: 617-350-7550 Fax: 617-350-7552	55 Temple Place Boston, MA 02111-1300	9-Bit Solution Network	RS-485 Tools/HW/SW
CMX COMPANY	Charles Behrmann President	Tel: 508-872-7675 Fax: 508-620-6828	5 Grant St. Ste C Framingham, MA 01701	CMX-RTX CMX-TINY+ CMX-TINY	Real Time Multitasking O.S.
DRUMLIN		Tel: 818-244-4600 Fax: 818-244-4246	3447 Ocean View Blvd., Glendale, CA 91208		Software Functional Libraries
DUNFIELD DEVELOPMENT SYSTEMS	Dave Dunfield	Tel: 613-256-5820 Fax: 613-256-5821 BBS: 613-256-6289	P.O. Box 31044 Nepan, Ontario, Canada K2B 8S8	EMILY52	Simulator/Emulator C Compiler
EMULATION TECHNOLOGY	Robert Diaz	Tel: 408-982-0660 Fax: 408- 982-0664	2344 Walsh Ave, Bldg. F, Santa Clara, CA 95051		Adapters
EMBEDDED SYSTEM PRODUCT	Ron Hodge	Tel: 713-728-9688 Fax: 713-728-1049	11501 Chimney Rock, Houston, TX 77035-2900		R.T. Kernal/ Operating System
EDI	Milos Kregoheck	Tel: 702-735-4997 Fax: 702-735-8339		Socket Adaptor	
FORTH, INC	Steve Agarwal	Tel: 800-55-FORTH Fax: 310-318-7130	111 North Sepulveda Blvd., Manhattan Beach, CA 90266	Chip Forth	Compiler/ Multi-Tasking Kernel
FRANKLIN SOFTWARE, INC.	Rhett D. Rowan General Manager	Tel: 408-296-8051 Fax: 408-296-8061 BBS: 408-296-8060 24HR Info: 408-296-8056 Sales: 800-880-8051 CompuServe: 75442,1423 Internet: fsinfo@fsinc.com	888 Saratoga Ave., Suite #2, San Jose, CA 95129	A51 C51 DK51 PK51 RTX-51	Macro assembler, complete kit with editor, linker, librarian, and include files. C compiler kit Developers kit Professional Developers kit Real Time Executive
IAR SYSTEMS	Nadin Sahehayed Michael Ohman	Tel: 415-765-5500 Fax: 415-765-5503 Tel: 011-4618-167800 Fax: 011-4618-167838	One Maritime Plaza San Francisco, CA 94111 P.O. Box 23051 S-750 23 Uppsala, Sweden	ICC8051 C58051 Embedded work bench	IAR C-Cross Compiler C-SPY Simulater/Debugger Windows based C-Compiler for 8051
INFORM SOFTWARE CORP	Woongkee Min	Tel: 708-866-1838 Fax: 708-866-1839	1840 Oak Ave., Evenston, IL 60201		Fuzzy S/W Development
INTERMETRICS MICROSYSTEMS SOFTWARE	Marty Stolz Technical Support Scott Tatiel Marketing V.P.	Tel: 617-661-0072 Fax: 617-868-2843 Fax: 617-868-2518	733 Concord Ave., Cambridge, MA 02138	White Smiths	Compiler/Assembler/Linker/Programmer Utilities
IOTA SYSTEMS	Steve Motts	Tel: 702-831-6302 Fax: 702-831-4629	924 Incline Way, Suite N, Incline Villiage, NV 89450		
ITT POMONA	Bob Poirier	Tel: 909-469-2912 Fax: 909-629-3317	1500 East 9th Street, Pomona, CA 91766-3835		Adapters
KEIL ELEKTRONIK		Tel: 49-89-465057 Fax: 49-89-468162	Bretonischeer Ring 15 85630 Grasbrunn, Germany		Software
KEIL SOFTWARE		Tel: 214-735-8052 Fax: 214-735-8055	16990 Dallas Parkway Suite 120 Dallas, TX 75248		
KYLE			Germany		C Compiler
SYNOPSIS LOGIC MODELING	Technical Support	Tel: 800-445-1888 Fax: 503- 690- 6906	19500 Northwest Gibbs Dr., Beaverton, OR 97006		Device Simulation Model for 8051

Table 1. Third Party Tool Vendors (continued)

Company	Contact Name	Phone / Fax	Address	Product Name	Description
MDL Labs(NEW MICROS, INC); interpreter requires 3rd timer	Ray Lavender	Tel: 614-431-2675 Fax: 614-431-2675	1073 Limberlost Ct., Colombus, OH 43235		
MICRO VIEW	Andrew	Tel: 408-356-4221			S/W Debugger/ Code Analyzer
NEW MICROS	Joe Getz	Tel: 214-339-2204 Fax: 214-339-1585	1601 Chalk Hill Rd., Dallas, TX 75212		Evaluation Board
PRODUCTION LANGUAGES CORP (PLC)	JD Hancock Joshua Kuanfung	Tel: 817-599-8363 Tel: 800-525-6289 Fax: 817-599-5098 Internet: plcorp@aol.com	P.O. Box 109 200 Cochran Rd., Weatherford, TX 76086	COMPASS/51	Macro Assembler/Linker/Object/Librarian/ANSI C-compiler, Source Level Debugger/Simulator, ROM moniter, IDE Online Documentation, Programmer's Editer.
QUANTASM CORP	Mike Schmit	Tel: 800-765-8086 Fax: 408-2447268	19672 Stevens Creek Blvd., Suite 397, Cupertino, CA 95014		Assembly Language Flowcharter
REICHMANN MICROCOMPUTER	Hr. Reichmann	Tel: 49-7141-71042 Fax: 49-7141-75312	Planck Strasse 3 71691 Freiberg, Germany	HTC-51	ANSI-C Development package: C compiler, Macro Assembler, Remote Monitor Debugger, Library Source, Tools
3M	Bob Soshay	Tel: 800-225-5373 Fax: 1-800- 325-5329	6801 River Place Blvd., Austin, TX 78726-9000 Electronic Products		Sockets
U.S. SOFTWARE	Don Dunstan	Tel: 800-356-7097 Tel: 503-641-8446 Fax: 503-644-2413	4215 Northerwest Science Park Drive, Portland, OR 97229	SuperTask USNET USFiles GOFAST	Real-time OS and associated tools Real-time networking including TCP/IP protocols DOS compatibile file system Single and double precision floating point library
UNIVERSAL CROSS ASSEMBLERS	Peter Aske	Voice: 506-849-8952 Fax: 506-847-0681	9 Westminster Drive, Quispamsis, NB Canada E2E 2V4	Cross-32 Meta-Assembler	DOS or Windows assembler
WICKENHAUSER	Jurgen Wickenhauser	Tel: 49-721-98849-0 Fax: 49-721-886807	Rastatter Strasse 144 76199 Karlsruhe, Germany		Software

8051 Family In-Circuit Emulator

The EMUL51™-PC is a high performance in-circuit emulator specifically designed to provide an optimal environment for 8051 family microcontroller hardware and software development. The EMUL51™-PC consists of a board which plugs directly into the IBM PC/XT/AT bus. An optional Trace board features advanced trace functioning with sophisticated trigger capabilities. The POD, which plugs into the target system, is connected to the emulator board with a 5 ft. ribbon cable for operating range flexibility. Optionally, an RS-232 box can be used which communicates with the PC at up to 115K baud. Yet another option, the LanICE, allows the EMUL51™-PC t run on workstations such as SUN or HP.

The World's Most Popular 8051 Emulator

Since its introduction 1986, Nohau has delivered over 10,000 EMUL51™-PC emulators. Each emulator is often used in several projects where different 8051 derivatives are needed. Only a change of the probe is required when a new derivative needs emulation support.

Choice of Different User Interfaces

Early in the evolution of the EMUL51™-PC's user interface, it became clear that each customer has different opinions of how they would like the interface to work and what features were important to them. One of the three main user interface choices for the EMUL51™-PC is Microsoft Windows 3.x. The other two are ChipView's Borland keypress compatible and Nohau's original pull-down/command line version.

Hosted on PCs and Workstations

The emulator was designed to be plugged into a full size PC/AT style slot. The optional trace needs a second slot. These same boards can also be supplied in an "RS-232 box" which communicates with the PC over a standard COM port. To use the EMUL51™-PC on XWindows workstations such as SUN or HP, the Nohau LanICE is availagble. Because LanICE uses a high speed (10 Mbit/second) local area network, not only can it be placed far away from the workstation but it maintains the relatively high code loading speed of the Nohau emulators plugged into your PC on your desk top. LanICE also supports personal computers on a network.

Real Time Trace

The EMUL51™-PC offers trace features not found in other emulators. The trace buffer can record up to 256K bus cycles with 64 bits of data. The trace can be operated "on-the-fly" which means that it can be viewed, programmed and retriggered without disturbing program execution. With the trace setup menu you can define what events are to be stored in the trace buffer. The real-time trace can be stopped (triggered) at a selected event or after a combination of multiple events.

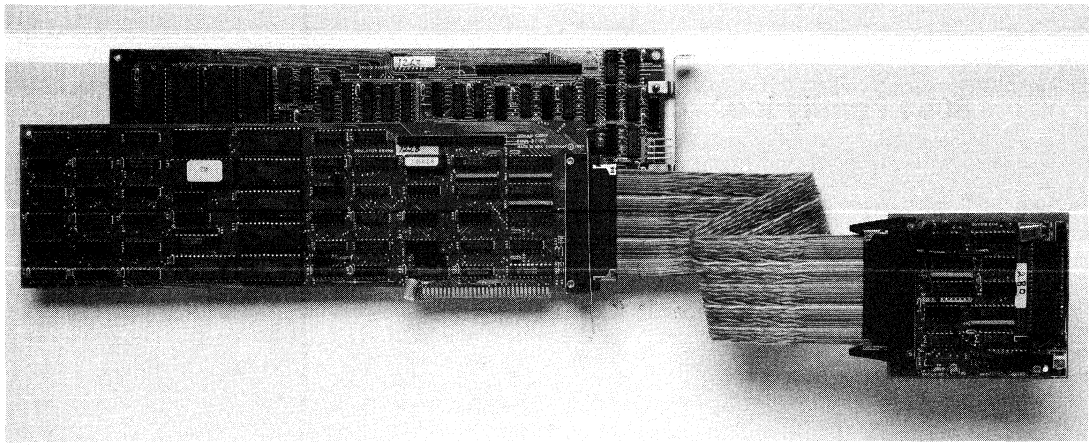
For additional information please contact:

Nohau Corporation
51 E. Campbell Avenue
Campbell, CA 95008
TEL: 408-866-1820
FAX: 408-378-7869

5

0516A

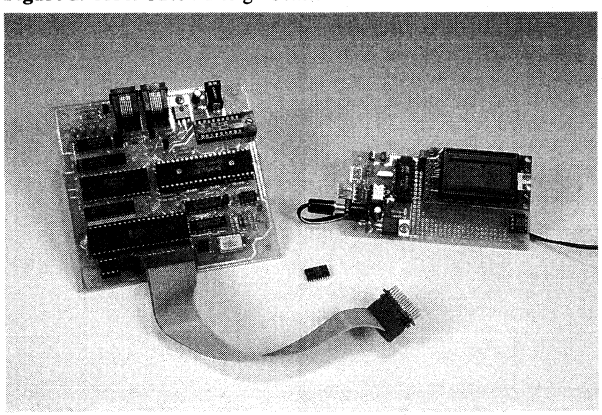
Figure 1. EMUL51



2051 Design Center

Mid-Tech's AT89C2051 Design Center includes everything you need to take your design from simulation, through target degugging, to standalone operation. Specifically designed to support Atmel's 20-pin processor family, the 2051 Design Center delivers features and capabilities normally found only in much more expensive development systems.

Figure 2. AT89C2051 Design Center



Target Debugger Board with Build-in Flash Programmer

Operating under control of a "windowed" PC control program, a high speed serial link provides nearly instantaneous communication to the target system. The result is a friendly, truly interactive, development environment. A ZIF programming site and resident flash algorithms provide full programming support for the AT89C1051 and AT89C2051.

AT89C2051 Family Simulator, Target System Debugger, and Assembler

The simulator runs stand-alone on a PC and also lets you include the target's physical I/O lines, timers, serial port, etc. in your simulation. The target system debugger gives you complete control over the system under develoopment and features a user interface identical to that of the simulator.

Features Include:

- Single-step, multi-step, animate, and high-speed execution modes.
- Simultaneous on-screen displays of program disassembly, data memory, and CPU registers.
- Full screen editors for CPU registers, special function registers, data and program memories.
- Multiple breakpoints are transparent to the user program.
- Serial I/O may be displayed in a window on the PC or can be redirected to the target system.
- Prototyping Board and AT89C2051 Included
- The 2051 Design Center includes an AT89C2051 prototyping board with built-in power supply and large pad-per-hole prototyping area, AT89C2051 processor included.
- Priced at $299.95 for the full-up system.

For additional information please contact:

Mid-Tech Computing Devices
P.O. Box 218
Stafford, CT 06075
TEL: 203-684-2442

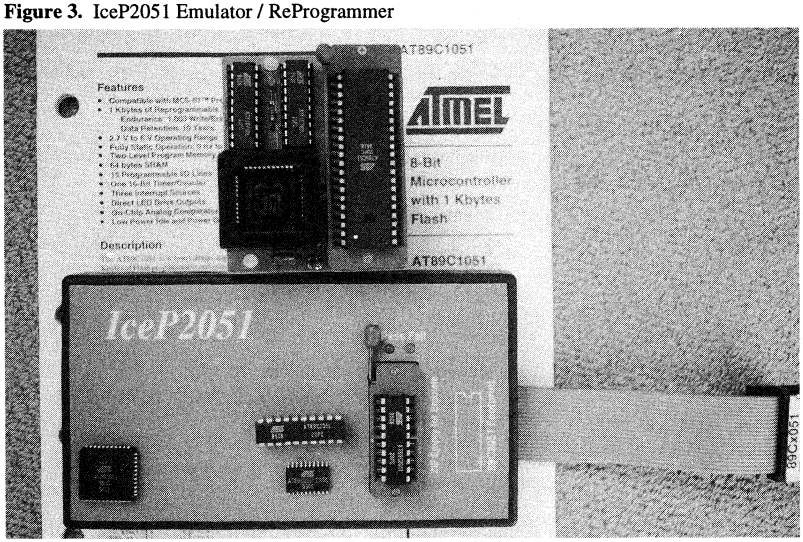
IceP2051 Emulator/ReProgrammer

The IceP2051 Emulator / ReProgrammer is a complete development station for the Atmel Flash AT89C1051 and AT89C2051 microcontrollers. It is a full speed, RAM based ICE, intended for full product developments. Unlike other systems, IceP2051 does not rely on PC Simulation, nor require you to erase/program a chip with each iteration. The Edit Assemble Debug loop takes just a few seconds.

When your code is tested/optimised, the programmer can be used for your volume production.

If you are using the older OTP technology, or a single sourced uC core, there are cross migration tools to assist porting your code.

Figure 3. IceP2051 Emulator / ReProgrammer



The DbgX51 debugger has a multi, scalable window display, and fast 'text editor' style operation. To change any location, just place the cursor, and edit.

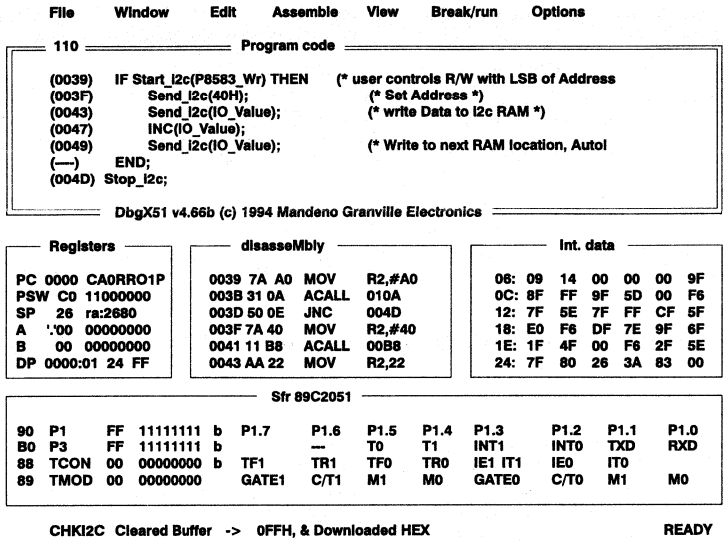
Multiple breakpoints, and Step, Skip, GotoHere debug commands are supported. This example shows DbgX51 / IceP0251 running the optional Modula-2 compiler - code illustrated is the multi i2c library.

Dbg2051.ZIP - Demo of IceP2051 Debugger.

Env2051.ZIP - Demo of IceP2051 environment.

IceP2051.ZIP - Both of the above, full system demo.

Figure 4. Modula-2 compiler - code



IceP2051

- Full ICE + ReProgrammer for the Atmel Flash 20-Pin variants (40.44 With adaptor)
- Real Time, RAM based, Emulation (including the Analog Comparitor)
- Full screen Debug, 'Borland' style interface, Multi Windowed, direct editing
- Full SFR symbolic BYTE.BIT display, for rapid learning Timer Uart Interrupt debug
- Source level Debug, and Mixed language Source debug, allowing .ASM.,C, Modula-2 source codes to be mixed
- Chip Prog step is NOT part of the development loop
- Complete package - Assembler / Linker / Debugger / Editor included
- Optional adaptors for SOL20, and DIP40/PLCC44
- Special cross platform migration support included and added to Assembler

Programmer Features

- FAST RAM based production programmer - 1.2Sec / KByte, (Erase.Vfy.Secure included!)
- Single Key, Erase Program Verify Secure of AT89C1051, AT89C2051, AT89C51, AT89C52 Atmel Flash microcontrollers
- Chips programmed counter
- Mis-socket and incorrect part detection
- Mechanically supported SOL adaptor
- Smart program algorithm, for fastest possible pgm cycle times

For additional information please contact:

Mandeno Granville Electronics Ltd: 80x51 Tools Specialists.
128 Grange Rd
Auckland 3 New Zealand
TEL: 64-9 -6300-558
FAX: 64-9-6301-720

2051-PD Programmer Downloader & I/O Simulator

Features

- Fast, Simple, Download & Run - No Change In Characteristics - No Loss of Features

The 2051-PD offers the simplicity and speed you need for developing small programs. From the simple Command Line to seeing the results of an Intel hex file running on the Target typically takes less than 6 secs. And that includes checksum verification.

In addition, the 2051-PD follows two ideals needed for a smooth transition to stand-alone operation:

- It preserves the true characteristics and special features of the 89C2051 microcontroller
- It allows the code to run in the same memory locations as needed for the stand-alone Target system.

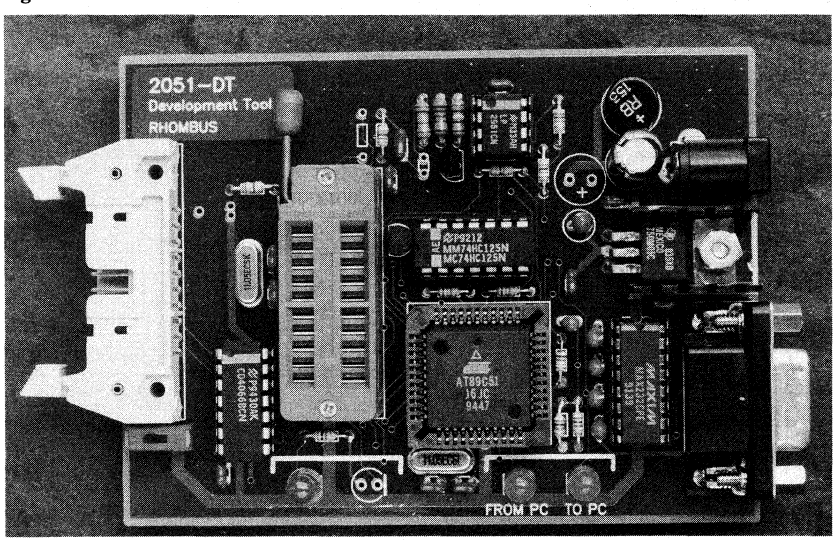
It is PC hosted through a RS232 port, and in turn it runs the Target board through its microcontroller socket. By taking advantage of the 2051's flash memory and ease of re-programming, this arrangement offers the speed and convenience of a ROM Emulator even though working with an internal memory device. In addition, once the download is complete, a Command Line Option allows the same PC port to be automatically switched to the Target for its own use. This is especially useful if the Target board does not require an RS232 interface yet one is desired to assist in the actual development, or production testing.

Another valuable use for this existing connection to the PC, is with the optional 'Dunfield Developments' Simulator Package. This fast PC Simulator (150,000 instr/sec with 386/25) can optionally pass all I/O related instructions to the Target's 2051 for execution. This allows running the application code in a crash-proof PC environment, with all register and memory data readily available, yet still seeing the interaction with the actual Target H/W. The Dunfield S/W package also includes an Assembler, and a Monitor Debugger intended for larger memory versions of the 51. The Simulator on its own is sufficient reason for purchasing this option.

The 2051-PD comes complete with its own S/W for programming and downloading a 6 ft. 9 pin PC serial cable, a 9-25 pin adapter, Target flat cable, and power supply for shipments to the U.S. and Canada.

5

Figure 5. 2051-PD



Terminology definitions for the 2051-PD:

Programmer: The programming and verifying of the internal Flash memory and the setting of lock bits

Downloader: The loading and running of code at full speed on an identical uC connected through the Target socket

I/O Simulator: As for the Downloader, but with the code restricted to those instructions that operate on the uC port pins. All other instructions are simulated by the PC. The I/O code will always execute at less than full speed. Requires the optional Dunfield Simulator Package

Technical

- Measures 4" x 3"
- Accepts 13-20VDC or 10-15VAC
- 3 pin oscillator socket for Xtals or C/Resonators
- RS232 includes RTS (from PC) & CTS (to PC)
- Green Status LED signals PowerUp, Run, Program
- Red LEDs indicate To/From PC
- Cmd/Line options LB1/2, COM#, ConnPt, Vrfy, ChkS, Color

Proto/Evaluation Boards for the 2051

2051 Distinguishing Features

	Evaluate Using	
Relative to other 51s	P1	P2
20mA sink capability		X
Analog Comparator	X	X
Relative to other <=20 Pin uCs		
On board UART	X	X
Multiplication & Division	X	X

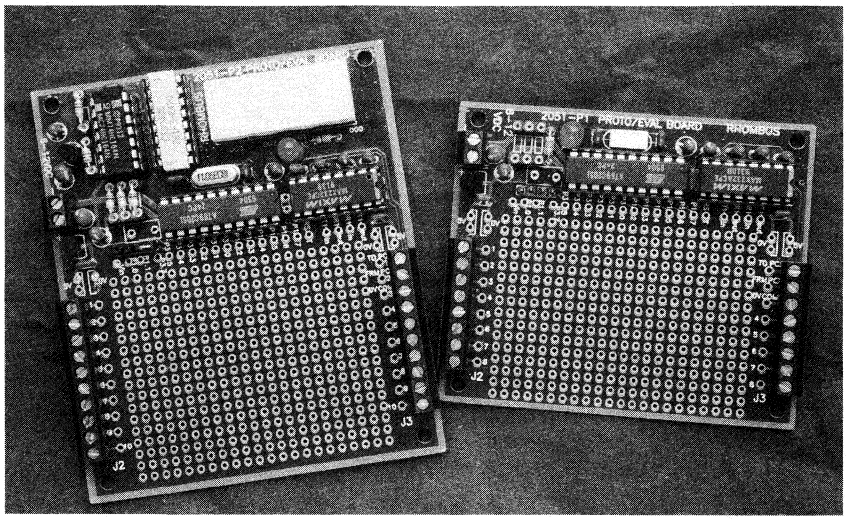
Both the 2051-P1 & P2 demonstrate a 1 capacitor 3 resistor A/D convertor using the analog comparator built into the AT89C2051. Also included are a RS-232 interface, a prototype area with screw terminals, and a precision 3 terminal voltage regulator.

Additionally, the 2051-P2 demonstrates the 20mA sink capability of the AT89C2051 with a high intensity 4 digit display using only 6 I/O pins and consuming only 2% CPU time at 11MHz clock speed.

For additional information please contact:

Rhombus
P.O. Box 871
Mauldin, SC 29662
TEL: 803-676-0012
FAX: 803-676-0015

Figure 6. Proto/Evaluation Boards for the 2051



SCE-51 SingleChip Emulation

Features

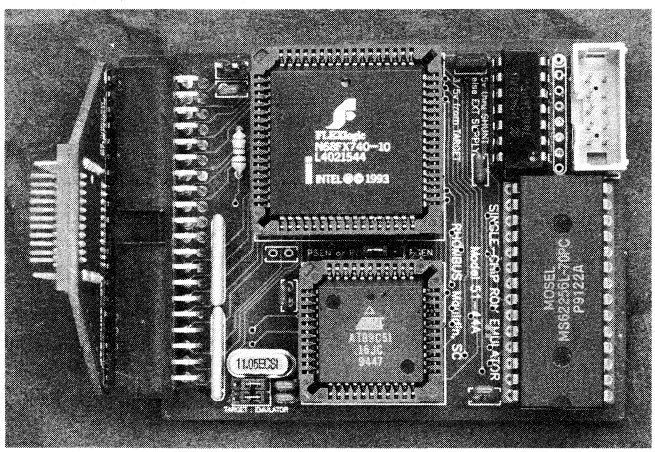
- Creates ROM/RAM Emulator for 51 family single-chip micro-controllers
- Additional memory space available for debug s/w
- Works with all 51 family variants
- No loss of specialized functions

Why SingleChip?

- Gain 18 more I/O pins
- High MHz without concern for timing
- Reduce baord size
- Increase reliability
- Secure proprietary code

5

Figure 7. SCE-51



Until now, low cost development tools such as ROM Emulators and S/W Debuggers, could only be used with 51 Family Microprocessors that were operating in expanded mode and using external memory. That has now changed thatks to SCE-51.

SCE-51 is a 2.4" by 3.2" assembly that plugs directly into the Single-Chip Micro-Controller socket and creates a 32K ROM/RAM Emulator. The extra memory provides sufficient space for both the Application code and Debug S/W. At the same time, SCE-51 maintains the availability of the 18 extra Port pins created by Single-Chip operation. Those two important functions give SCE-51 the most valuable features of a Single-Chip 'ICE', but at a fraction of its cost.

Another plus when compared to an 'ICE' is that one device covers all 40/44 pin variants of the 51 Family. And all specialized functions of each variant are retained. Simply install a PLCC version of the specific 51 to be emulated into SCE-51 and select a PLCC or DIP adapter to suit the Target socket.

Thanks to SCE-51 you no longer pay a premium for Single-Chip development, and no longer need volume production to share those costs. Make your next application Single-Chip with the help of SCE-51.

Technical

- Access time = 15 ns + SRAM access (optional 10 ns)
- TTL and CMOS compatible
- Quiet 4 layer PCB
- Powered by Single-Chip Socket (100 mA at 5 V 25 MHz)
- Includes cable plus choice of 40 DIP or 44 PLCC adapter
- Installed height 3.5" Width 2.4"
- Connects to PC printer port
- Supports Intel HEX file format
- Command line loader with memory map options

Note: The 18 Port Pins normally allocated to expanded memory will be restricted to byte wide Read & Write operations during Emulation.

For additional information please contact:

Rhombus
P.O. Box 871
Mauldin, SC 29662
TEL: 803-676-0012
FAX: 803-676-0015

SOIC to DIP Programming Adapter for AT89C1051/2051

The programming adapter will convert the 20-pin SOIC down to a 20-pin DIP. It is a universal adapter which works on any programmer. This adapter can be ordered as part number AS-20-20-01S-6 from Emulation Technology.

For emulation purposes an adapter is available which enables usage of a 20-pin DIP to a 20-pin SOIC footprint. The surface mount adapter can be ordered as AS-DIP.3-020-5003-1.

For additional information please contact:

Emulation Technology
2344 Walsh Ave, Bldg. F
Santa Clara, CA 95051
TEL: 408-982-0660
FAX: 408-982-0664
BBS: 408-982-9044

Features

- **Compatible with 40-pin MCS-51™ sockets**
- **Compatible with in-circuit-emulators with 40-pin sockets**
- **32 Programmable I/O Lines or 15 Programmable I/O Lines**

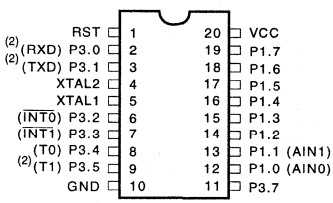
Microcontroller Two Way Adaptor Board

Description[1]

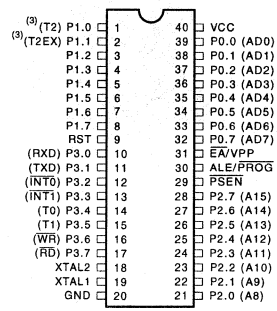
Atmel's 20- to 40-pin adapter board is for use in designing AT89C1051/AT89C2051 systems. The ATABX051 maps the pins on the 20-pin device to the corresponding pin locations in a 40-pin 80C51 footprint. Customers can use this board to adapt existing in-circuit-emulators for use in AT89C1051/AT89C2051 designs (note that a comparator must also be added for full emulation) or to plug in 40-pin 80C51 devices into 20-pin AT89C1051/AT89C2051 sockets. In addition, the board can be used to adapt the 20-pin footprint to an existing 40-pin socket.

Pin Configurations

PDIP/SOIC
20-Pin

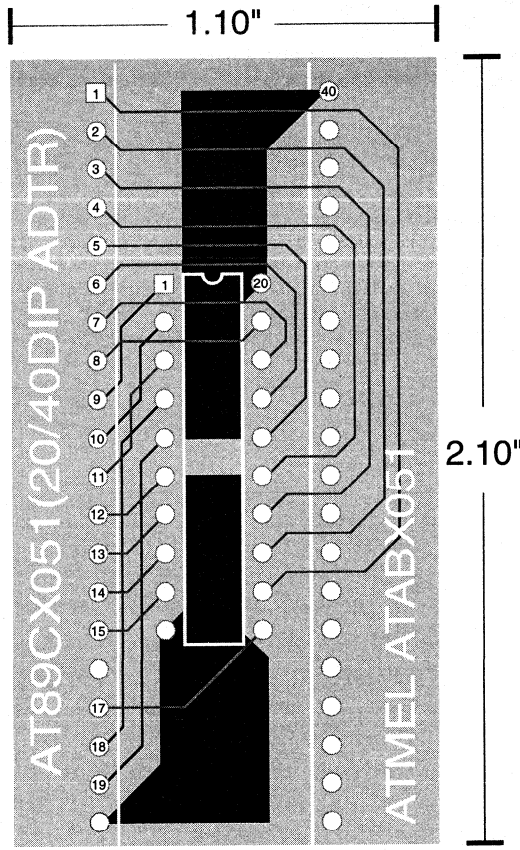


PDIP/Cerdip
40-Pin



Notes:
1. This adapter cannot be used with programmers for programming purposes. Although the pins are defined the same in operation mode, the pin definitions and algorithms are different from the AT89C51 in programming mode.
2. Pins (RxD) P3.0, (TxD) P3.1, and (T1) P3.5 are AT89C2051 pins only, but not for AT89C1051.
3. Pins (T2) P1.0 and (T2EX) P1.1 are AT89C52 pins only.

0376B



Board Characteristics

1. Holes: .040 Diameter; Plated
2. Material: .063 Thick FR4; 10Z Copper
3. Finish: SMOBC
4. Soldermask both sides per artwork

Ordering Information

Contact the lead Atmel Sales Representative for availability and ordering information.

Contents

Microcontroller Abbreviated Cross-Reference Guide

	Intel		Atmel	
PART	MCS-51			
NUMBER	i80C31	no program store	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	i80C51	4 Kbytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	i87C51	4 Kbytes of EPROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	i80C52	8 Kbytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	i87C52	8 Kbytes of EPROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
PACKAGE	D	CERDIP	D	CERDIP
	P	PDIP	P	PDIP
	N	PLCC	J	PLCC
	S	PQFP	Q	PQFQ
	Philips/Signetics		**Atmel**	
PART	PCx80C31	0 bytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
NUMBER	SC80C31	0 bytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	PCx80C51	4 Kbytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	SC80C51	4 Kbytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	SC87C51	4 Kbytes of EPROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	P80C32	0 bytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	P80C52	8 Kbytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	P87C52	8 Kbytes of EPROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	S83C752	2 Kbytes of ROM & 64 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	S87C752	2 Kbytes of EPROM & 64 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	S83C751	2 Kbytes of ROM & 64 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	S87C751	2 Kbytes of EPROM & 64 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	S83C750	1 Kbyte of ROM & 64 bytes of RAM	AT89C1051*	1 Kbyte of FLASH & 64 bytes of RAM
	S87C750	1 Kbyte of EPROM & 64 bytes of RAM	AT89C1051*	1 Kbyte of FLASH & 64 bytes of RAM
PACKAGE			S	SOIC (89C2051 only)
	F	CERDIP	D	CERDIP
	N	PDIP	P	PDIP
	A	PLCC	J	PLCC
	K	LCC	L	LCC
	B	PQFQ	Q	PQFQ
	AMD		**Atmel**	
PART	8751	4 Kbytes of EPROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
NUMBER	87C51	4 Kbytes of EPROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	87C52T2	8 Kbytes of EPROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
PACKAGE	D	CERDIP	D	CERDIP
	P	PDIP	P	PDIP
	J	PLCC	J	PLCC
	L	LCC	L	LCC

6

* Indicates Atmel similar function and not direct replacement/socket compatible

0517A

Microcontroller Abbreviated Cross-Reference Guide (continued)

	Siemens		Atmel	
PART NUMBER	SAB8051	4 Kbytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	SAB8031	0 bytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	SAB8052	8 Kbytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	SAB8032	0 bytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	SABC501-1R	8 Kbytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	SABC501-L	0 bytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
PACKAGE	P	PDIP	P	PDIP
	N	PLCC	J	PLCC
	Matra		**Atmel**	
PART NUMBER	80C31	0 bytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	80C51	4 Kbytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	80C32	0 bytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	80C52	8 Kbytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
PACKAGE	D	CERDIP	D	CERDIP
	P	PDIP	P	PDIP
	S	PLCC	J	PLCC
	R	LCC	L	LCC
	V	PQFQ	Q	PQFQ
	T	TQFP	A	TQFP
	Dallas		**Atmel**	
PART NUMBER	DS5000FP	0 bytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	DS5001FP	0 bytes of ROM & 128 bytes of RAM	AT89C51	4 Kbytes of FLASH & 128 bytes of RAM
	DS5000	8 Kbytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	DS5000T	8 Kbytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	DS2250	8 Kbytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM
	DS2250T	8 Kbytes of ROM & 256 bytes of RAM	AT89C52	8 Kbytes of FLASH & 256 bytes of RAM

	Microchip		Atmel	
PART NUMBER	PIC16C54	512 bytes EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16C54A	512 bytes EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16LC54AA	512 bytes EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16CR57A	512 bytes EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16CLR57A	512 bytes EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16C55	512 bytes EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16C56	1 Kbyte EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16C57	2 Kbytes EPROM & 80 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16C71	512 bytes EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16LC71	512 bytes EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16C84	512 bytes EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	PIC16LC84	512 bytes EPROM & 32 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
PACKAGE	JW	CERDIP	S	SOIC
	P	PDIP	P	PDIP
	SP	.3 PDIP	W	DIE
	S	DIE		
	S	SOIC		
	SS	SSOP		

* Indicates Atmel similar function and not direct replacement/socket compatible

Microcontroller Abbreviated Cross-Reference Guide (continued)

PART NUMBER	Zilog		Atmel	
	Z86C08	2 Kbytes of ROM & 124 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86E08	2 Kbytes of ROM & 124 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86C09	2 Kbytes of ROM & 124 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86C19	2 Kbytes of ROM & 124 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86E09	2 Kbytes of ROM & 124 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86C17	2 Kbytes of ROM & 124 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86L06	1 Kbyte of ROM & 124 bytes of RAM	AT89C1051*	1 Kbyte of FLASH & 64 bytes of RAM
	Z86L29	6 Kbytes of ROM & 124 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86C06	1 Kbyte of ROM & 124 bytes of RAM	AT89C1051*	1 Kbyte of FLASH & 64 bytes of RAM
	Z86E09	1 Kbyte of ROM & 124 bytes of RAM	AT89C1051*	1 Kbyte of FLASH & 64 bytes of RAM
	Z86C08	2 Kbytes of ROM & 124 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86E08	2 Kbytes of ROM & 124 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86C30	4 Kbytes of ROM & 236 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86E30	4 Kbytes of ROM & 236 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86C40	4 Kbytes of ROM & 236 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM
	Z86E40	4 Kbytes of ROM & 236 bytes of RAM	AT89C2051*	2 Kbytes of FLASH & 128 bytes of RAM

* Indicates Atmel similar function and not direct replacement/socket compatible

AT89C51 Microcontroller Detailed Cross-Reference Guide

AMD	Atmel	Part Description	Speed	Pkg	Temp
EPROM	**FLASH**				
P8031AH-18	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
D8031AH-18	AT89C51-20DC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	COM
N8031AH-18	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
P8031AH-15	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
D8031AH-15	AT89C51-20DC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	COM
N8031AH-15	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
P8031AH	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
D8031AH	AT89C51-20DC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	COM
N8031AH	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
ID8031AHB	AT89C51-20DI	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	IND
D8751H	AT89C51-20DC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	COM
R8751H	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
ID8751H	AT89C51-20DI	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	IND
ID8751HB	AT89C51-20DI	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	IND
P80C51BH	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
D80C51BH	AT89C51-20DC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	COM
N80C51BH	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
P80C51BH-1	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
D80C51BH-1	AT89C51-20DC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	COM
N80C51BH-1	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
P80C31BH	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
D80C31BH	AT89C51-20DC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	COM
N80C31BH	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
P80C31BH-1	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
D80C31BH-1	AT89C51-20DC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	CERDIP	COM
N80C31BH-1	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM

Intel	Atmel	Part Description	Speed	Pkg	Temp
UV EPROM	**FLASH**				
D87C51	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
D87C51-2	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
TD87C51	AT89C51-20PI	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	IND
D87C51-1	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
D87C51-20	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
OTP					
N87C51	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
N87C51-2	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
P87C51	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
P87C51-2	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
S87C51	AT89C51-20QC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PQFP	COM
S87C51-2	AT89C51-20QC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PQFP	COM
TN87C51	AT89C51-20JI	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	IND
TP87C51	AT89C51-20PI	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	IND
N87C51-1	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
P87C51-1	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
S87C51-1	AT89C51-20QC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PQFP	COM
N87C51-20	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
P87C51-20	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
S87C51-20	AT89C51-20QC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PQFP	COM
MROM					
N80C51BH	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
N80C51BH-2	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
P80C51BH	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
P80C51BH-2	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
S80C51BH	AT89C51-20QC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PQFP	COM
S80C51BH-2	AT89C51-20QC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PQFP	COM
TN0C51BH	AT89C51-20JI	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	IND
TP80C51BH	AT89C51-20PI	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	IND
N80C51BH-1	AT89C51-20JC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	COM
P80C51BH-1	AT89C51-20PC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	COM
S80C51BH-1	AT89C51-20QC	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PQFP	COM
TN80C51BH-1	AT89C51-20JI	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PLCC	IND
TP80C51BH-1	AT89C51-20PI	8 BIT MICROCONTROLLER 4 KB FLASH	20 MHz	PDIP	IND

AT89C51 Microcontroller Detailed Cross-Reference Guide (continued)

Matra	Atmel	Part Description		Speed	Pkg	Temp
ROM	**FLASH**					
A P 80C51 F -1	AT89C51-20PA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	AUTO
A P 80C51 F -S	AT89C51-20PA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	AUTO
A P 80C51 F	AT89C51-20PA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	AUTO
A S 80C51 F -1	AT89C51-20JA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	AUTO
A S 80C51 F -S	AT89C51-20JA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	AUTO
A S 80C51 F	AT89C51-20JA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	AUTO
A T 80C51 F -1	AT89C51-20AA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	AUTO
A T 80C51 F -S	AT89C51-20AA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	AUTO
A T 80C51 F	AT89C51-20AA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	AUTO
A V 80C51 F -1	AT89C51-20QA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	AUTO
A V 80C51 F -S	AT89C51-20QA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	AUTO
A V 80C51 F	AT89C51-20QA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	AUTO
I P 80C51 F -25	AT89C51-24PI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	IND
I P 80C51 F -1	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
I P 80C51 F -L	AT89LV51-16PI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PDIP	IND
I P 80C51 F -L	AT89LV51-20PI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
I P 80C51 F -S	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
I P 80C51 F	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
I S 80C51 F -25	AT89C51-24JI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PLCC	IND
I S 80C51 F -1	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
I S 80C51 F -L	AT89LV51-16JI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PLCC	IND
I S 80C51 F -L	AT89LV51-20JI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
I S 80C51 F -S	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
I S 80C51 F	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
I T 80C51 F -25	AT89C51-24AI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	TQFP	IND
I T 80C51 F -1	AT89C51-20AI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	IND
I T 80C51 F -L	AT89LV51-16AI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	TQFP	IND
I T 80C51 F -L	AT89LV51-20AI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	IND
I T 80C51 F -S	AT89C51-20AI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	IND
I T 80C51 F	AT89C51-20AI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	IND
I V 80C51 F -25	AT89C51-24QI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PQFP	IND
I V 80C51 F -1	AT89C51-20QI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	IND
I V 80C51 F -L	AT89LV51-16QI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PQFP	IND
I V 80C51 F -L	AT89LV51-20QI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	IND
I V 80C51 F -S	AT89C51-20QI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	IND
I V 80C51 F	AT89C51-20QI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	IND
M D 80C51 F -MB	AT89C51-20DM /883	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	CERDIP	MIL
M D 80C51 F -MB	AT89C51-20DM /883	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	CERDIP	MIL
M R 80C51 F -MB	AT89C51-20LM /883	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	LCC	MIL
M R 80C51 F -MB	AT89C51-20LM /883	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	LCC	MIL
P 80C51 F -25	AT89C51-24PC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	COM
P 80C51 F -1	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
P 80C51 F -L	AT89LV51-16PC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PDIP	COM
P 80C51 F -L	AT89LV51-20PC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
P 80C51 F -S	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
P 80C51 F	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
S 80C51 F -25	AT89C51-24JC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PLCC	COM
S 80C51 F -1	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
S 80C51 F -L	AT89LV51-16JC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PLCC	COM
S 80C51 F -L	AT89LV51-20JC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
S 80C51 F -S	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
S 80C51 F	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
T 80C51 F -25	AT89C51-24AC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	TQFP	COM
T 80C51 F -1	AT89C51-20AC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	COM
T 80C51 F -L	AT89LV51-16AC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	TQFP	COM
T 80C51 F -L	AT89LV51-20AC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	COM
T 80C51 F -S	AT89C51-20AC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	COM
T 80C51 F	AT89C51-20AC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	COM
V 80C51 F -25	AT89C51-24QC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PQFP	COM
V 80C51 F -1	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
V 80C51 F -L	AT89LV51-16QC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PQFP	COM
V 80C51 F -L	AT89LV51-20QC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
V 80C51 F -S	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
V 80C51 F	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM

AT89C51 Microcontroller Detailed Cross-Reference Guide (continued)

Matra	Atmel	Part Description		Speed	Pkg	Temp
A P 80C51 -1	AT89C51-20PA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	AUTO
A P 80C51 -S	AT89C51-20PA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	AUTO
A P 80C51	AT89C51-20PA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	AUTO
A S 80C51 -1	AT89C51-20JA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	AUTO
A S 80C51 -S	AT89C51-20JA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	AUTO
A S 80C51	AT89C51-20JA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	AUTO
A T 80C51 -1	AT89C51-20AA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	AUTO
A T 80C51 -S	AT89C51-20AA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	AUTO
A T 80C51	AT89C51-20AA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	AUTO
A V 80C51 -1	AT89C51-20QA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	AUTO
A V 80C51 -S	AT89C51-20QA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	AUTO
A V 80C51	AT89C51-20QA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	AUTO
I P 80C51 -25	AT89C51-24PI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	IND
I P 80C51 -1	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
I P 80C51 -L	AT89LV51-16PI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PDIP	IND
I P 80C51 -L	AT89LV51-20PI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
I P 80C51 -S	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
I P 80C51	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
I S 80C51 -25	AT89C51-24JI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PLCC	IND
I S 80C51 -1	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
I S 80C51 -L	AT89LV51-16JI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PLCC	IND
I S 80C51 -L	AT89LV51-20JI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
I S 80C51 -S	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
I S 80C51	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
I T 80C51 -25	AT89C51-24AI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	TQFP	IND
I T 80C51 -1	AT89C51-20AI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	IND
I T 80C51 -L	AT89LV51-16AI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	TQFP	IND
I T 80C51 -L	AT89LV51-20AI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	IND
I T 80C51 -S	AT89C51-20AI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	IND
I T 80C51	AT89C51-20AI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	IND
I V 80C51 -25	AT89C51-24QI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PQFP	IND
I V 80C51 -1	AT89C51-20QI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	IND
I V 80C51 -L	AT89LV51-16QI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PQFP	IND
I V 80C51 -L	AT89LV51-20QI	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	IND
I V 80C51 -S	AT89C51-20QI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	IND
I V 80C51	AT89C51-20QI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	IND
M D 80C51 -MB	AT89C51-20DM /883	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	CERDIP	MIL
M D 80C51 -MB	AT89C51-20DM /883	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	CERDIP	MIL
M R 80C51 -MB	AT89C51-20LM /883	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	LCC	MIL
M R 80C51 -MB	AT89C51-20LM /883	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	LCC	MIL
P 80C51 -25	AT89C51-24PC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	COM
P 80C51 -1	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
P 80C51 -L	AT89LV51-16PC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PDIP	COM
P 80C51 -L	AT89LV51-20PC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
P 80C51 -S	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
P 80C51	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
S 80C51 -25	AT89C51-24JC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PLCC	COM
S 80C51 -1	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
S 80C51 -L	AT89LV51-16JC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PLCC	COM
S 80C51 -L	AT89LV51-20JC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
S 80C51 -S	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
S 80C51	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
T 80C51 -25	AT89C51-24AC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	TQFP	COM
T 80C51 -1	AT89C51-20AC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	COM
T 80C51 -L	AT89LV51-16AC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	TQFP	COM
T 80C51 -L	AT89LV51-20AC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	COM
T 80C51 -S	AT89C51-20AC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	COM
T 80C51	AT89C51-20AC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	TQFP	COM
V 80C51 -25	AT89C51-24QC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PQFP	COM
V 80C51 -1	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
V 80C51 -L	AT89LV51-16QC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	16 MHz	PQFP	COM
V 80C51 -L	AT89LV51-20QC	3 V, 8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
V 80C51 -S	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
V 80C51	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM

AT89C51 Microcontroller Detailed Cross-Reference Guide (continued)

Philips	Atmel	Part Description		Speed	Pkg	Temp
UVEPROM	**FLASH**					
SC87C510CK44	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SC87C510CF40	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SC87C51ACK44	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
SC87C51ACF40	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
SC87C510GK44	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SC87C510GF40	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SC87C51AGK44	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
SC87C51AGF40	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
SC87C510PK44	AT89C51-24JC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PLCC	COM
SC87C510PF40	AT89C51-24PC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	COM
OTP						
SC87C510CA44	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SC87C510CN40	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SC87C510CB44	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
SC87C510CA44	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
SC87C510CN40	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
SC87C510GA44	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SC87C510GN40	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SC87C510GB44	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
SC87C51AGA44	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PLCC	IND
SC87C51AGN40	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	IND
SC87C51CPA44	AT89C51-24JC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PLCC	COM
SC87C510CPN40	AT89C51-24PC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	COM
SC87C510CPB44	AT89C51-24QC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PQFP	COM
MROM						
SC80C51B0CA44	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SC80C51B0CN40	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SC80C51B0CB44	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
SC80C51B0CA44	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
SC80C51B0CN40	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
SC80C51B0CGA44	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SC80C51B0CGN40	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
SC80C51B0CGB44	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	IND
SC80C51B0CGA44	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PLCC	IND
SC80C51B0CGN40	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	IND
SC80C51B0CPA44	AT89C51-24JC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PLCC	COM
SC80C51B0CPN40	AT89C51-24PC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	COM
SC80C51B0CPB44	AT89C51-24QC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PQFP	COM
MROM						
PCB8051BH-2WP	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
PCB8051BH-2P	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
PCB8051BH-2H	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
PCB8051BH-3WP	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
PCB8051BH-3P	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
PCB8051BH-3H	AT89C51-20QC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	COM
PCB8051BH-3WP	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
PCB8051BH-3P	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
PCB8051BH-3H	AT89C51-20QI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PQFP	IND
PCB8051BH-3WP	AT89C51-16JA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	AUTO
PCB8051BH-3P	AT89C51-16PA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	AUTO
PCB8051BH-4WP	AT89C51-24JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
PCB8051BH-4P	AT89C51-24PC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	COM
PCB8051BH-4H	AT89C51-24QC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PQFP	COM

AT89C51 Microcontroller Detailed Cross-Reference Guide (continued)

Philips	Atmel	Part Description		Speed	Pkg	Temp
UV EPROM	**FLASH**					
SC87C510CK44	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SC87C510CF40	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SC87C51ACK44	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
SC87C51ACF40	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
SC87C510GK44	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SC87C510GF40	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SC87C51AGK44	AT89C51-20JI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	IND
SC87C51AGF40	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
SC87C510PK44	AT89C51-24JC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PLCC	COM
SC87C510PF40	AT89C51-24PC	8 BIT MICROCONTROLLER	4 KB FLASH	24 MHz	PDIP	COM

Siemens	Atmel	Part Description		Speed	Pkg	Temp
ROM/ROMless	**FLASH**					
SAB8051A-P	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SAB8031A-P	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SAB8051A-16-P	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SAB8031A-16-P	AT89C51-20PC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	COM
SAB8051A-N	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SAB8051A-16-N	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SAB8031A-16-N	AT89C51-20JC	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PLCC	COM
SAB8051A-12-P-T 40/85	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
SAB8051A-10-P-T 40/110	AT89C51-20PA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	AUTO
SAB8031A-12-P-T 40/85	AT89C51-20PI	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	IND
SAB8051A-10-P-T 40/110	AT89C51-20PA	8 BIT MICROCONTROLLER	4 KB FLASH	20 MHz	PDIP	AUTO

AT89C52 Microcontroller Detailed Cross-Reference Guide

AMD	Atmel	Part Description		Speed	Pkg	Temp
EPROM	**FLASH**					
D87C52T2	AT89C52-20DC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	COM
R87C52T2	AT89C52-20LC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	COM
P87C52T2	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
N87C52T2	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
ID87C52T2	AT89C52-20DI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	IND
IR87C52T2	AT89C52-20LI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	IND
IP87C52T2	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
IN87C52T2	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
D87C52T2-1	AT89C52-20DC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	COM
R87C52T2-1	AT89C52-20LC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	COM
P87C52T2-1	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
N87C52T2-1	AT89C52-20DC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	COM
ID87C52T2-1	AT89C52-20DI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	IND
IR87C52T2-1	AT89C52-20LI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	IND
IP87C52T2-1	AT89C52-20DI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	IND
IN87C52T2-1	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
P87C32T2-1	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
N87C32T2-1	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
IP87C32T2-1	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
IN87C32T2-1	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND

Intel	Atmel	Part Description		Speed	Pkg	Temp
MROM	**FLASH**					
P8052AH	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
D8052AH	AT89C52-20DC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	COM
N8052AH	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
P8032AH	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
D8032AH	AT89C52-20DC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	COM
N8032AH	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
EPROM						
P8752BH	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
D8752BH	AT89C52-20DC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	COM
N8752BH	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
R8752BH	AT89C52-20LC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	COM
TD8752BH	AT89C52-20DI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	IND
QP8752BH	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
LD8752BH	AT89C52-20DI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	IND
D87C52	AT89C52-20DC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	COM
N87C52	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
P87C52	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
S87C52	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
TP87C52	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
TD87C52	AT89C52-20DI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	IND
TN87C52	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
LP87C52	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
LD87C52	AT89C52-20DI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	IND
LN87C52	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
P80C52	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
S80C52	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
N80C52	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
TP80C52	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
TN80C52	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
LP80C52	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
LN80C52	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
P80C32	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
S80C32	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
N80C32	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
TP80C32	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
TN80C32	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
LP80C32	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
LN80C32	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
D87C52-20	AT89C52-20DC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	COM

AT89C52 Microcontroller Detailed Cross-Reference Guide (continued)

Intel	Atmel	Part Description		Speed	Pkg	Temp
N87C52-20	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
P87C52-20	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
S87C52-20	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
P80C52-20	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
S80C52-20	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
N80C52-20	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
P80C32-20	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
S80C32-20	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
N80C32-20	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM

Matra	Atmel	Part Description		Speed	Pkg	Temp
ROM	**FLASH**					
A P 80C52 F -1	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A P 80C52 F -S	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A P 80C52 F	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A S 80C52 F -1	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A S 80C52 F -S	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A S 80C52 F	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A T 80C52 F -1	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A T 80C52 F -S	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A T 80C52 F	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A V 80C52 F -1	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
A V 80C52 F -S	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
A V 80C52 F	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
I P 80C52 F -25	AT89C52-24PI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	IND
I P 80C52 F -1	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C52 F -L	AT89LV52-16PI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PDIP	IND
I P 80C52 F -L	AT89LV52-20PI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C52 F -S	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C52 F	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I S 80C52 F -25	AT89C52-24JI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	IND
I S 80C52 F -1	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C52 F -L	AT89LV52-16JI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PLCC	IND
I S 80C52 F -L	AT89LV52-20JI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C52 F -S	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C52 F	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I T 80C52 F -25	AT89C52-24AI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	TQFP	IND
I T 80C52 F -1	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C52 F -L	AT89LV52-16AI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	TQFP	IND
I T 80C52 F -L	ATT89LV52-20AI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C52 F -S	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C52 F	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I V 80C52 F -25	AT89C52-24QI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFPQ FP	IND
I V 80C52 F -1	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C52 F -L	AT89LV52-16QI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PQFP	IND
I V 80C52 F -L	AT89LV52-20QI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C52 F -S	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C52 F	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
M D 80C52 F -MB	AT89C52-20DM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	MIL
M D 80C52 F -MB	AT89C52-20DM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	MIL
M R 80C52 F -MB	AT89C52-20LM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	MIL
M R 80C52 F -MB	AT89C52-20LM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	MIL
P 80C52 F -25	AT89C52-24PC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	COM
P 80C52 F -1	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C52 F -L	AT89LV52-16PC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PDIP	COM
P 80C52 F -L	AT89LV52-20PC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C52 F -S	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C52 F	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
S 80C52 F -25	AT89C52-24JC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	COM
S 80C52 F -1	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
S 80C52 F -L	AT89LV52-16JC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PLCC	COM
S 80C52 F -L	AT89LV52-20JC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM

AT89C52 Microcontroller Detailed Cross-Reference Guide (continued)

Matra	Atmel	Part Description		Speed	Pkg	Temp
S 80C52 F -S	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
S 80C52 F	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
T 80C52 F -25	AT89C52-24AC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	TQFP	COM
T 80C52 F -1	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C52 F -L	AT89LV52-16AC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	TQFP	COM
T 80C52 F -L	AT89LV52-20AC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C52 F -S	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C52 F	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
V 80C52 F -25	AT89C52-24QC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFP	COM
V 80C52 F -1	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C52 F -L	AT89LV52-16QC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PQFP	COM
V 80C52 F -L	AT89LV52-20QC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C52 F -S	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C52 F	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
A P 80C52 -1	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A P 80C52 -S	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A P 80C52	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A S 80C52 -1	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A S 80C52 -S	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A S 80C52	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A T 80C52 -1	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A T 80C52 -S	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A T 80C52	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A V 80C52 -1	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
A V 80C52 -S	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
A V 80C52	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
I P 80C52 -25	AT89C52-24PI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	IND
I P 80C52 -1	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C52 -L	AT89LV52-16PI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PDIP	IND
I P 80C52 -L	AT89LV52-20PI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C52 -S	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C52	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I S 80C52 -25	AT89C52-24JI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	IND
I S 80C52 -1	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C52 -L	AT89LV52-16JI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PLCC	IND
I S 80C52 -L	AT89LV52-20JI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C52 -S	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C52	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I T 80C52 -25	AT89C52-24AI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	TQFP	IND
I T 80C52 -1	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C52 -L	AT89LV52-16AI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	TQFP	IND
I T 80C52 -L	AT89LV52-20AI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C52 -S	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C52	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I V 80C52 -25	AT89C52-24QI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFP	IND
I V 80C52 -1	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C52 -L	AT89LV52-16QI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PQFP	IND
I V 80C52 -L	AT89LV52-20QI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C52 -S	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C52	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
M D 80C52 -MB	AT89C52-20DM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	MIL
M D 80C52 -MB	AT89C52-20DM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	MIL
M R 80C52 -MB	AT89C52-20LM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	MIL
M R 80C52 -MB	AT89C52-20LM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	MIL
P 80C52 -25	AT89C52-24PC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	COM
P 80C52 -1	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C52 -L	AT89LV52-16PC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PDIP	COM
P 80C52 -L	AT89LV52-20PC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C52 -S	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C52	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
S 80C52 -25	AT89C52-24JC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	COM
S 80C52 -1	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
S 80C52 -L	AT89LV52-16JC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PLCC	COM
S 80C52 -L	AT89LV52-20JC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
S 80C52 -S	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM

AT89C52 Microcontroller Detailed Cross-Reference Guide (continued)

Matra	Atmel	Part Description		Speed	Pkg	Temp
S 80C52	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
T 80C52 -25	AT89C52-24AC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	TQFP	COM
T 80C52 -1	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C52 -L	AT89LV52-16AC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	TQFP	COM
T 80C52 -L	AT89LV52-20AC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C52 -S	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C52	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
V 80C52 -25	AT89C52-24QC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFP	COM
V 80C52 -1	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C52 -L	AT89LV52-16QC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PQFP	COM
V 80C52 -L	AT89LV52-20QC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C52 -S	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C52	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
ROMless						
A P 80C32 F -1	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A P 80C32 F -S	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A P 80C32 F	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A S 80C32 F -1	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A S 80C32 F -S	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A S 80C32 F	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A T 80C32 F -1	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A T 80C32 F -S	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A T 80C32 F	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A V 80C32 F -1	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
A V 80C32 F -S	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
A V 80C32 F	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
I P 80C32 F -25	AT89C52-24PI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	IND
I P 80C32 F -1	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C32 F -L	AT89LV52-16PI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PDIP	IND
I P 80C32 F -L	AT89LV52-20PI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C32 F -S	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C32 F	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I S 80C32 F -25	AT89C52-24JI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	IND
I S 80C32 F -1	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C32 F -L	AT89LV52-16JI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PLCC	IND
I S 80C32 F -L	AT89LV52-20JI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C32 F -S	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C32 F	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I T 80C32 F -25	AT89C52-24AI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	TQFP	IND
I T 80C32 F -1	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C32 F -L	AT89LV52-16AI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	TQFP	IND
I T 80C32 F -L	AT89LV52-20AI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C32 F -S	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C32 F	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I V 80C32 F -25	AT89C52-24QI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFP	IND
I V 80C32 F -1	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C32 F -L	AT89LV52-16QI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PQFP	IND
I V 80C32 F -L	AT89LV52-20QI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C32 F -S	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C32 F	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
M D 80C32 F -MB	AT89C52-20DM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	MIL
M D 80C32 F -MB	AT89C52-20DM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	MIL
M R 80C32 F -MB	AT89C52-20LM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	MIL
M R 80C32 F -MB	AT89C52-20LM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	MIL
P 80C32 F -25	AT89C52-24PC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	COM
P 80C32 F -1	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C32 F -L	AT89LV52-16PC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PDIP	COM
P 80C32 F -L	AT89LV52-20PC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C32 F -S	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C32 F	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
S 80C32 F -25	AT89C52-24JC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	COM
S 80C32 F -1	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
S 80C32 F -L	AT89LV52-16JC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PLCC	COM
S 80C32 F -L	AT89LV52-20JC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
S 80C32 F -S	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM

AT89C52 Microcontroller Detailed Cross-Reference Guide (continued)

Matra	Atmel	Part Description		Speed	Pkg	Temp
S 80C32 F	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
T 80C32 F -25	AT89C52-24AC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	TQFP	COM
T 80C32 F -1	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C32 F -L	AT89LV52-16AC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	TQFP	COM
T 80C32 F -L	AT89LV52-20AC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C32 F -S	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C32 F	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
V 80C32 F -25	AT89C52-24QC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFP	COM
V 80C32 F -1	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C32 F -L	AT89LV52-16QC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PQFP	COM
V 80C32 F -L	AT89LV52-20QC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C32 F -S	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C32 F	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
A P 80C32 -1	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A P 80C32 -S	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A P 80C32	AT89C52-20PA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	AUTO
A S 80C32 -1	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A S 80C32 -S	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A S 80C32	AT89C52-20JA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	AUTO
A T 80C32 -1	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A T 80C32 -S	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A T 80C32	AT89C52-20AA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	AUTO
A V 80C32 -1	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
A V 80C32 -S	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
A V 80C32	AT89C52-20QA	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	AUTO
I P 80C32 -25	AT89C52-24PI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	IND
I P 80C32 -1	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C32 -L	AT89LV52-16PI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PDIP	IND
I P 80C32 -L	AT89LV52-20PI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C32 -S	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I P 80C32	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
I S 80C32 -25	AT89C52-24JI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	IND
I S 80C32 -1	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C32 -L	AT89LV52-16JI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PLCC	IND
I S 80C32 -L	AT89LV52-20JI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C32 -S	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I S 80C32	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
I T 80C32 -25	AT89C52-24AI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	TQFP	IND
I T 80C32 -1	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C32 -L	AT89LV52-16AI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	TQFP	IND
I T 80C32 -L	AT89LV52-20AI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C32 -S	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I T 80C32	AT89C52-20AI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	IND
I V 80C32 -25	AT89C52-24QI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFP	IND
I V 80C32 -1	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C32 -L	AT89LV52-16QI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PQFP	IND
I V 80C32 -L	AT89LV52-20QI	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C32 -S	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
I V 80C32	AT89C52-20QI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	IND
M D 80C32 -MB	AT89C52-20DM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	MIL
M D 80C32 -MB	AT89C52-20DM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	MIL
M R 80C32 -MB	AT89C52-20LM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	MIL
M R 80C32 -MB	AT89C52-20LM /883	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	MIL
P 80C32 -25	AT89C52-24PC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	COM
P 80C32 -1	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C32 -L	AT89LV52-16PC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PDIP	COM
P 80C32 -L	AT89LV52-20PC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C32 -S	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P 80C32	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
S 80C32 -25	AT89C52-24JC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	COM
S 80C32 -1	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
S 80C32 -L	AT89LV52-16JC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PLCC	COM
S 80C32 -L	AT89LV52-20JC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
S 80C32 -S	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
S 80C32	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM

AT89C52 Microcontroller Detailed Cross-Reference Guide (continued)

Matra	Atmel	Part Description		Speed	Pkg	Temp
T 80C32 -25	AT89C52-24AC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	TQFP	COM
T 80C32 -1	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C32 -L	AT89LV52-16AC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	TQFP	COM
T 80C32 -L	AT89LV52-20AC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C32 -S	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
T 80C32	AT89C52-20AC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	TQFP	COM
V 80C32 -25	AT89C52-24QC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFP	COM
V 80C32 -1	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C32 -L	AT89LV52-16QC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	16 MHz	PQFP	COM
V 80C32 -L	AT89LV52-20QC	3 V, 8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C32 -S	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
V 80C32	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM

Philips	Atmel	Part Description		Speed	Pkg	Temp
ROMless	**FLASH**					
P80C32 EBPN	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P80C32 EBAA	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
P80C32 EBBB	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
P80C32 EFPN	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
P80C32 EFAA	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
P80C32 EFBB	AT89C52-24PC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	COM
P80C321 BPN	AT89C52-24JC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	COM
P80C321 BAA	AT89C52-24QC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFP	COM
P80C321 FPN	AT89C52-24PI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	IND
P80C321 FAA	AT89C52-24JI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	IND
ROM						
P80C52 EBPN	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P80C52 EBAA	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
P80C52 EBBB	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
P80C52 EFPN	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
P80C52 EFAA	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
P80C52 EFBB	AT89C52-24PC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	COM
P80C52 BPN	AT89C52-24JC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	COM
P80C52 BAA	AT89C52-24QC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFP	COM
P80C52 FPN	AT89C52-24PI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	IND
P80C52 FAA	AT89C52-24JI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	IND
EPROM						
P87C52 EBPN	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
P87C52 EBAA	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
P87C52 EBBB	AT89C52-20QC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PQFP	COM
P87C52 EFPN	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
P87C52 EFAA	AT89C52-20JI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	IND
P87C52 EFBB	AT89C52-24PC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	COM
P87C52 BPN	AT89C52-24JC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	COM
P87C52 BAA	AT89C52-24QC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PQFP	COM
P87C52 FPN	AT89C52-24PI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PDIP	IND
P87C52 FAA	AT89C52-24JI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	PLCC	IND
P87C52 EBFFA	AT89C52-20DC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	COM
P87C52 EBLKA	AT89C52-20LC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	COM
P87C52 EFFFA	AT89C52-20DI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	CERDIP	IND
P87C52 EFLKA	AT89C52-20LI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	LCC	IND
P87C521 BFFA	AT89C52-24DC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	CERDIP	COM
P87C521 BLKA	AT89C52-24LC	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	LCC	COM
P87C521 FFFA	AT89C52-24DI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	CERDIP	IND
P87C521 FLKA	AT89C52-24LI	8 BIT MICROCONTROLLER	8 KB FLASH	24 MHz	LCC	IND
ROM/ROMless						
SAB 8032B -P	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
SAB 8032B -N	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
SAB 8032B -P-T40/48	AT89C52-20PI	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	IND
SAB 8032B -16-P	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
SAB 8032B -16-N	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
SAB 8032B -20-P	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM
SAB 8032B -20-N	AT89C52-20JC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PLCC	COM
SAB 8052B -P	AT89C52-20PC	8 BIT MICROCONTROLLER	8 KB FLASH	20 MHz	PDIP	COM

AT89C2051[1] Microcontroller Detailed Cross-Reference Guide

Philips/Signetics	Atmel	Part Description		Speed	Pkg	Temp
EPROM	**FLASH**					
S87C752-1F28	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S87C752-2F28	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S87C752-4F28	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S87C752-5F28	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S87C752-1N28	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S87C752-2N28	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S87C752-4N28	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S87C752-5N28	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S87C752-1A28	AT89C2051-24SC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	SOIC	COM
S87C752-2A28	AT89C2051-24SI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	SOIC	IND
S87C752-4A28	AT89C2051-24SC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	SOIC	COM
S87C752-5A28	AT89C2051-24SI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	SOIC	IND
S87C752-6A28	AT89C2051-16SA	8 BIT MICROCONTROLLER	2 KB FLASH	16 MHz	SOIC	AUTO
S87C752-6F28	AT89C2051-16PA	8 BIT MICROCONTROLLER	2 KB FLASH	16 MHz	PDIP	AUTO
S87C752-6N28	AT89C2051-16PA	8 BIT MICROCONTROLLER	2 KB FLASH	16 MHz	PDIP	AUTO
ROM						
S83C752-1N28	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S83C752-2N28	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S83C752-4N28	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S83C752-5N28	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S83C752-1A28	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S83C752-2A28	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S83C752-4A28	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S83C752-5A28	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S83C752-6A28	AT89C2051-16SA	8 BIT MICROCONTROLLER	2 KB FLASH	16 MHz	SOIC	AUTO
S83C752-6F28	AT89C2051-16PA	8 BIT MICROCONTROLLER	2 KB FLASH	16 MHz	PDIP	AUTO
S83C752-6N28	AT89C2051-16PA	8 BIT MICROCONTROLLER	2 KB FLASH	16 MHz	PDIP	AUTO
EPROM	**FLASH**					
S87C751-1F24	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S87C751-2F24	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S87C751-4F24	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S87C751-5F24	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S87C751-1N24	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S87C751-2N24	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S87C751-4N24	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S87C751-5N24	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S87C751-1A28	AT89C2051-24SC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	SOIC	COM
S87C751-2A28	AT89C2051-24SI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	SOIC	IND
S87C751-4A28	AT89C2051-24SC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	SOIC	COM
S87C751-5A28	AT89C2051-24SI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	SOIC	IND
ROM						
S83C751-1N24	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S83C751-2N24	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S83C751-4N24	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S83C751-5N24	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S83C751-1A28	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S83C751-2A28	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND
S83C751-4A28	AT89C2051-24PC	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	COM
S83C751-5A28	AT89C2051-24PI	8 BIT MICROCONTROLLER	2 KB FLASH	24 MHz	PDIP	IND

Contents

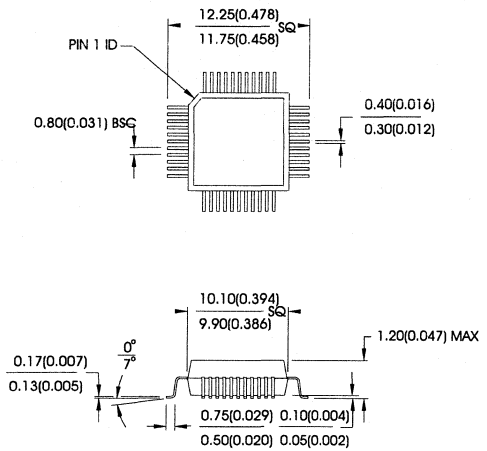
7

Package Drawings

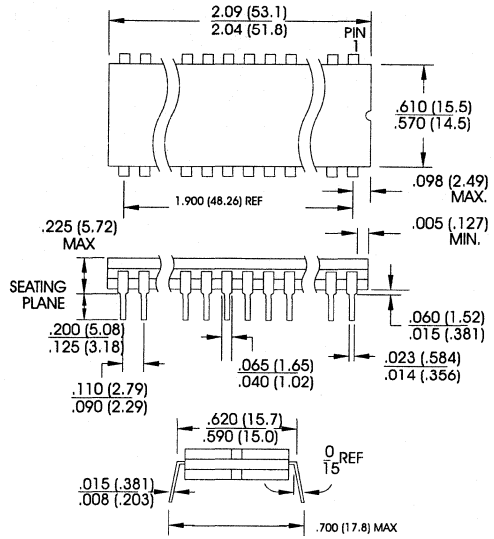
Each Atmel data sheet includes an Ordering Information Section which specifies the package types available. This section provides size specifications and outlines for all package types.[1]

Note: 1. Dimensions shown do not include lead plating or mold flash.

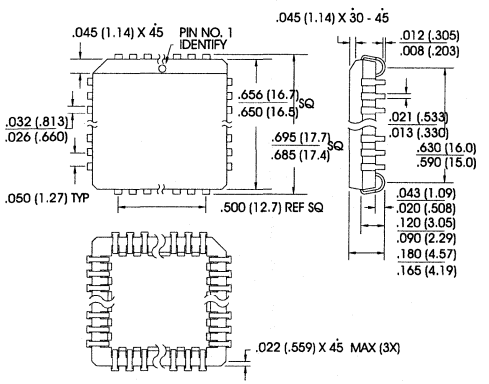
44A, 44 Lead, Thin (1.0 mm) Plastic Gull Wing Quad Flat Package (TQFP)
Dimensions in Inches and (Millimeters)



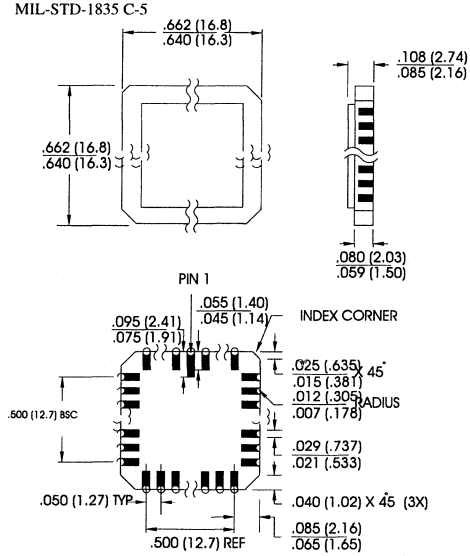
40D6, 40 Lead, 0.600" Wide, Non-Windowed, Cermic Dual Inline Package (Cerdip)
Dimensions in Inches and (Millimeters)
MIL-STD-1835 D-5 CONFIG A



44J, 44 Lead, Plastic J-Leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC OUTLINE MO-047 AC

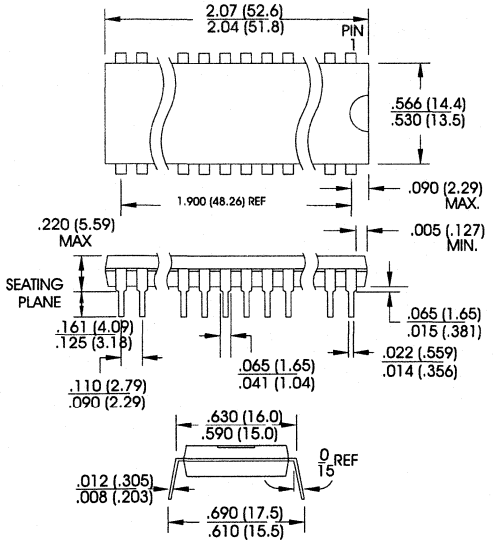


44L, 44 Pad, Non-Windowed, Ceramic Leadless Chip Carrier (LCC)
Dimensions in Inches and (Millimeters)*
MIL-STD-1835 C-5

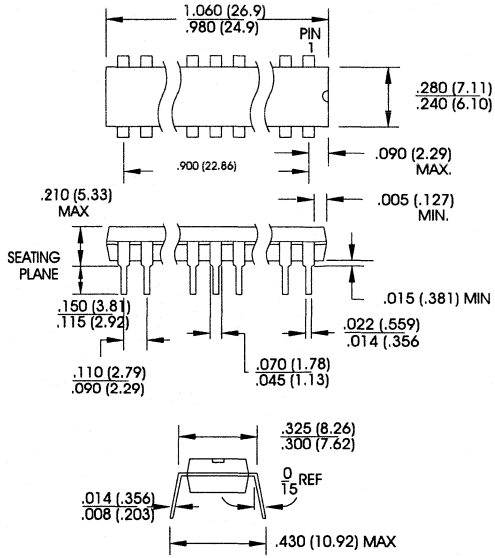


*Ceramic lid standard unless specified.

40P6, 40 Lead, 0.600" Wide,
Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)

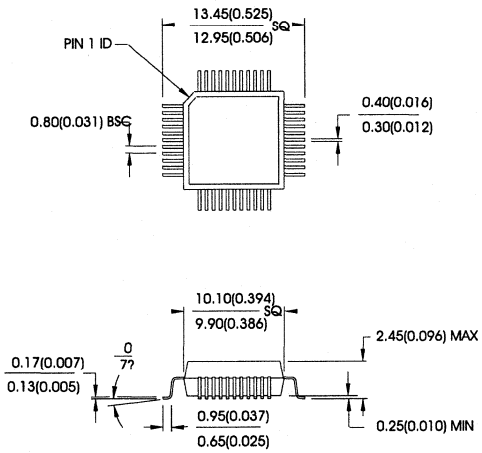


20P3, 20 Lead, 0.300" Wide,
Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)

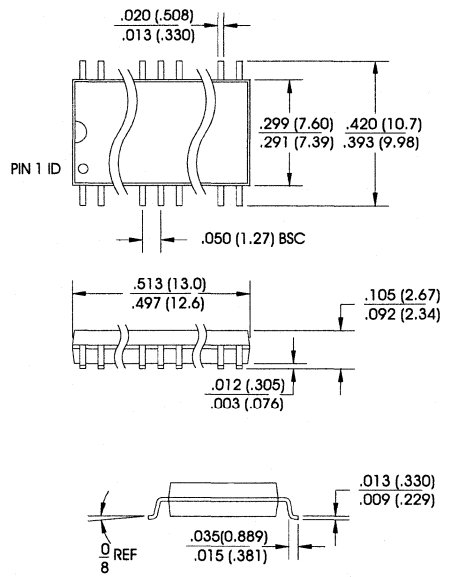


*Controlling dimension: millimeters

44Q, 44 Lead, Plastic Gull Wing Quad Flat
Package (PQFP)
Dimensions in Inches and (Millimeters)



20S, 20 Lead, 0.300" Wide,
Plastic Gull Wing Small Outline (SOIC)
Dimensions in Inches and (Millimeters)



7

Contents

Programmable Logic Devices

Part Number	Packages	Speeds	Description	Availability
Flash-Based				
ATF16V8B	20-Pin	7.5-25 ns	8 FFs, 8 I/O Pins, Standard Power	Now
ATF16V8BQ,BQL	20-Pin	10-25 ns	8 FFs, 8 I/O Pins, Quarter Power, Low Power	Now
ATF16V8C	20-Pin	5-7 ns	8 FFs, 8 I/O Pins, Standard Power	4Q-95
ATF16V8CZ	20-Pin	10-15 ns	8 FFs, 8 I/O Pins, Zero Power	4Q-95
ATF20V8B	24-Pin, 28-Pin	7.5-25 ns	8 FFs, 8 I/O Pins, Standard Power	Now
ATF20V8BQ,BQL	24-Pin, 28-Pin	10-25 ns	8 FFs, 8 I/O Pins, Quarter Power, Low Power	Now
ATF22V10B	24-Pin, 28-Pin	7.5-25 ns	10 FFs, 10 I/O Pins, Standard Power	Now
ATF22V10BL,BQ,BQL	24-Pin, 28-Pin	10-25 ns	10 FFs, 10 I/O Pins, Quarter Power, Low Power	Now
ATF22V10BL,BQ,BQL	24-Pin, 28-Pin	10-25 ns	10 FFs, 10 I/O Pins, Quarter Power, Low Power	Now
ATF1500,L	44-Pin	7.5-15 ns	32 FFs, 32 I/O Pins, Standard Power, Low Power	Now
Low Voltage				
ATF16LV8C	20-Pin	10-15 ns	8 FFs, 8 I/O Pins, Low Voltage	4Q-95
ATF16LV8CZ	20-Pin	15-25 ns	8 FFs, 8 I/O Pins, Low Voltage, Zero Power	4Q-95
AT22LV10,L	24-Pin, 28-Pin	20-30 ns	10 FFs, 10 I/O Pins, Standard & Low Power	Now
ATLV750B,BL	24-Pin, 28-Pin	10-15 ns	20 FFs, 10 I/O Pins, Standard & Low Power	4Q-95
ATLV2500B,BL	40-Pin, 44-Pin	15-20 ns	48 FFs, 24 I/O Pins, Standard & Low Power	4Q-95
5-Volt, EPROM-Based				
AT22V10,L	24-Pin, 28-Pin	15-30 ns	10 FFs, 10 I/O Pins, Standard & Low Power	Now
AT22V10B	24-Pin, 28-Pin	7.5-10 ns	10 FFs, 10 I/O Pins, Standard Power	Now
ATV750,L	24-Pin, 28-Pin	20-30 ns	20 FFs, 10 I/O Pins, Standard & Low Power	Now
ATV750B,BL	24-Pin, 28-Pin	7.5-25 ns	20 FFs, 10 I/O Pins, Standard & Low Power	Now
ATV750BQ,BQL	24-Pin, 28-Pin	15-25 ns	20 FFs, 10 I/O Pins, Quarter Power, Low Power	4Q95
ATV2500H,L	40-Pin, 44-Pin	25-35 ns	48 FFs, 24 I/O Pins, Standard & Low Power	Now
ATV2500B,BL	44-Pin	12-20 ns	48 FFs, 24 I/O Pins, Standard & Low Power	Now
ATV2500BQ,BQL	40-Pin, 44-Pin	20-25 ns	48 FFs, 24 I/O Pins, Quarter Power, Low Power	Now
ATV5000,L	68-Pin	25-35 ns	128 FFs, 52 I/O Pins, Standard & Low Power	Now
ATV5100,L	68-Pin	25-35 ns	128 FFs, 52 I/O Pins, Standard & Low Power	Now

Cache Logic™ FPGAs

Part Number	Registers	Usable Gates	Frequency	Description	Availability
AT6002	1.024	2K-4K	250 MHz	96 I/O Pins, 5V, Very Low Power	Now
AT6003	1,600	3K-6K	250 MHz	120 I/O Pins, 5V, Very Low Power	Now
AT6005	3,136	5K-10K	250 MHz	108 I/O Pins, 5V, Very Low Power	Now
AT6010	6,400	10K-20K	250 MHz	204 I/O Pins, 5V, Very Low Power	Now
Low Voltage					
AT6002LV	1.024	2K-4K	250 MHz	96 I/O Pins, 3V, Very Low Power	4Q-95
AT6003LV	1,600	3K-6K	250 MHz	120 I/O Pins, 3V, Very Low Power	4Q-95
AT6005LV	3,136	5K-10K	250 MHz	108 I/O Pins, 3V, Very Low Power	Now
AT6010LV	6,400	10K-20K	250 MHz	204 I/O Pins, 3V, Very Low Power	1Q-96

FPGA Serial Configuration E^2PROMS

Part Number	Memory Size	Description	Availability
AT17C65	65,536 x 1	65K FPGA Configuration E^2PROM	Now
AT17C128	131,072 x 1	128K FPGA Configuration E^2PROM	Now
AT17C256	262,144 x 1	256K FPGA Configuration E^2PROM	4Q-95

System Serial Configuration E^2PROMS

Part Number	Memory Size	Description	Availability
AT34C64	65,536 x 1	64K System Configuration E^2PROM	4Q-95
AT34C128	131,072 x 1	128K System Configuration E^2PROM	4Q-95
AT34C256	262,144 x 1	256K System Configuration E^2PROM	1Q-96

Gate Arrays

Part Number	Gates	Description	Availability
ATL60 Series	4K-1120K	0.6-Micron CMOS Gate Array, 3.3-Volt & 5.0-Volt Operation, 16 Versions with Various Pin & Gate Counts	Now
ATL80 Series	2K-600K	0.8-Micron CMOS Gate Array, 3.3-Volt & 5.0-Volt Operation, 12 Versions with Various Pin & Gate Counts	Now
ATLV Series	2K-35K	1.0-Micron CMOS Gate Array, 1.0-Volt & 3.3-Volt Operation, 8 Underlayers with Various Pin & Gate Counts	Now

Logic

Part Number	Speeds	Description	Availability
AT40281	16-40 MHz	80386SX PC/AT Core Logic Controller, with Posted-Write Cache	Now
AT40283	16-33 MHz	80386SX PC/AT Core Logic Controller	Now
AT40285	16-40 MHz	80386SX/486SLC/486SLC2 PC/AT Core Logic Controller	Now
AT40391B	25-40 MHz	80386DX PC/AT System & Cache Controller	Now
AT40392	25-50 MHz	80386DX PC/AT Memory Controller	Now
AT40410	25-50 MHz	ISA/PCI/VL PC/AT Core Logic Chipset	Now
AT40493	25-50 MHz	80486 PC/AT System & Cache Controller	Now
AT40495	25-50 MHz	80486 PC/AT System & Cache Controller	Now

Secure Memory ICs

Part Number	Memory Size	Description	Availability
AT88SC101	1024 x 1	1K Serial E^2PROM with Security, 1 Memory Zone, 1024 Bits	Now
AT88SC102	1024 x 1	1K Serial E^2PROM with Security, 2 Memory Zones, 512 Bits Each	Now
AT88SC103	1536 x 1	1K Serial E^2PROM with Security, 3 Memory Zones, 512 Bits Each	Now
AT88SC200	2048 x 1	2K Serial E^2PROM with Gate Array	Now
RF ID ASICs	Up to 16K x 1	Analog, Digital & Memory on Single-Chip ASIC	Now

Flash PEROMs

Part Number	Organization	Speeds	Description	Availability
Battery-Voltage™ **(2.7V to 3.6V)**				
AT29BV010A	128K x 8	200-350 ns	1-Mbit, 2.7-Volt Read and 2.7-Volt Write Flash PEROM	Now
AT29BV020	256K x 8	250-350 ns	2-Mbit, 2.7-Volt Read and 2.7-Volt Write Flash PEROM	Now
AT29BV040A	512K x 8	250-350 ns	4-Mbit, 2.7-Volt Read and 2.7-Volt Write Flash PEROM	Now
Low Voltage (3V to 3.6V)				
AT29LV256	32K x 8	150-250 ns	256K, 3-Volt Read and 3-Volt Write Flash PEROM	Now
AT29LV512	64K x 8	200-250 ns	512K, 3-Volt Read and 3-Volt Write Flash PEROM	Now
AT29LV010A	128K x 8	200-250 ns	1-Mbit, 3-Volt Read and 3-Volt Write Flash PEROM	Now
AT29LV1024	64K x 16	150-250 ns	1-Mbit, 3-Volt Read and 3-Volt Write Flash PEROM	Now
AT29LV020	256K x 8	200-250 ns	2-Mbit, 3-Volt Read and 3-Volt Write Flash PEROM	Now
AT29LV040A	512K x 8	200-250 ns	4-Mbit, 3-Volt Read and 3-Volt Write Flash PEROM	Now
Standard Voltage (5V)				
AT29C256	32K x 8	70-250 ns	256K, 5-Volt Read and 5-Volt Write Flash PEROM	Now
AT29C257	32K x 8	70-250 ns	256K, 5-Volt Read and 5-Volt Write Flash PEROM	Now
AT29C512	64K x 8	70-200 ns	512K, 5-Volt Read and 5-Volt Write Flash PEROM	Now
AT29C1024	64K x 16	70-200 ns	1-Mbit, 5-Volt Read and 5-Volt Write Flash PEROM	Now
AT29C010A	128K x 8	70-200 ns	1-Mbit, 5-Volt Read and 5-Volt Write Flash PEROM	Now
AT29C020	256K x 8	100-200 ns	2-Mbit, 5-Volt Read and 5-Volt Write Flash PEROM	Now
AT29C040A	512K x 8	120-250 ns	4-Mbit, 5-Volt Read and 5-Volt Write Flash PEROM	Now

Serial E^2PROMs

Part Number	Organization	V	Description	Availability
AT24C01	128 x 8	1.8, 2.5, 2.7, 5.0 V	1K, 2-Wire Bus Serial E^2PROM, Non-Cascadable	Now
AT24C21	128 x 8	2.5 - 5.0 V	1K, 2-Wire Bus Serial E^2PROM, Dual Mode, Plug & Play Operation	Now
AT24C01A	128 x 8	1.8, 2.5, 2.7, 5.0 V	1K, 2-Wire Bus Serial E^2PROM	Now
AT24C02	256 x 8	1.8, 2.5, 2.7, 5.0 V	2K, 2-Wire Bus Serial E^2PROM	Now
AT24C04	512 x 8	1.8, 2.5, 2.7, 5.0 V	4K, 2-Wire Bus Serial E^2PROM	Now
AT24C08	1024 x 8	1.8, 2.5, 2.7, 5.0 V	8K, 2-Wire Bus Serial E^2PROM	Now
AT24C16	2048 x 8	1.8, 2.5, 2.7, 5.0 V	16K, 2-Wire Bus Serial E^2PROM	Now
AT24C164	2048 x 8	1.8, 2.5, 2.7, 5.0 V	16K, 2-Wire Bus Serial E^2PROM with Cascadable Feature	Now
AT24C32	4096 x 8	1.8, 2.5, 2.7, 5.0 V	32K, 2-Wire Bus Serial E^2PROM with Cascadable Feature	Now
AT24C64	8192 x 8	1.8, 2.5, 2.7, 5.0 V	64K, 2-Wire Bus Serial E^2PROM with Cascadable Feature	Now
AT25C01	128 x 8	1.8, 2.7, 5.0 V	1K, SPI Bus Serial E^2PROM, Supports SPI Mode 1	Consult Factory
AT25C02	256 x 8	1.8, 2.7, 5.0 V	2K, SPI Bus Serial E^2PROM, Supports SPI Mode 1	Consult Factory
AT25C04	512 x 8	1.8, 2.7, 5.0 V	4K, SPI Bus Serial E^2PROM, Supports SPI Mode 1	Consult Factory
AT25010	128 x 8	1.8, 2.7, 5.0 V	1K, SPI Bus Serial E^2PROM, Supports SPI Mode 0 and 3	Now
AT25020	256 x 8	1.8, 2.7, 5.0 V	2K, SPI Bus Serial E^2PROM, Supports SPI Mode 0 and 3	Now
AT25040	512 x 8	1.8, 2.7, 5.0 V	4K, SPI Bus Serial E^2PROM, Supports SPI Mode 0 and 3	Now
AT93C46	64 x 16 / 128 x 8	1.8, 2.5, 2.7, 5.0 V	1K, 3-Wire Bus Serial E^2PROM	Now
AT93C46A	64 x 16	1.8, 2.5, 2.7, 5.0 V	1K, 3-Wire Bus Serial E^2PROM	Now
AT93C56	128 x 16 / 256 x 8	2.5, 2.7, 5.0 V	2K, 3-Wire Bus Serial E^2PROM	Now
AT93C57	128 x 16 / 256 x 8	2.5, 2.7, 5.0 V	2K, 3-Wire Bus Serial E^2PROM with Special Address	Now
AT93C66	256 x 16 / 512 x 8	2.5, 2.7, 5.0 V	4K, 3-Wire Bus Serial E^2PROM	Now
AT59C11	64 x 16 / 128 x 8	2.5, 2.7, 5.0 V	1K, 4-Wire Bus Serial E^2PROM	Now
AT59C22	128 x 16 / 256 x 8	2.5, 2.7, 5.0 V	2K, 4-Wire Bus Serial E^2PROM	Now
AT59C13	256 x 16 / 512 x 8	2.5, 2.7, 5.0 V	4K, 4-Wire Bus Serial E^2PROM	Now

Parallel E^2PROMs

Part Number	Organization	Speeds	Description	Availability
High Speed				
AT28HC64B	8K x 8	55-120 ns	64K E^2PROM with 64-Byte Page, Software Data Protection	Now
AT28HC256	32K x 8	70-120 ns	256K E^2PROM with 64-Byte Page & Software Data Protection	Now
AT28HC256E	32K x 8	70-120 ns	256K E^2PROM with Extended Endurance, Standard & Low Power	Now
AT28HC256F	32K x 8	70-120 ns	256K E^2PROM with Fast Write, Standard & Low Power	Now
***Battery-Voltage™* (2.7V to 3.6V)**				
AT28BV16	2K x 8	250-300 ns	16K E^2PROM, 2.7-Volt	Now
AT28BV64	8K x 8	300 ns	64K E^2PROM, 2.7-Volt	Now
Low Voltage (3.0V to 3.6V)				
AT28LV64B	8K x 8	200-300 ns	64K E^2PROM with 64-Byte Page & Software Data Protection, 3.0-Volt	Now
AT28LV256	32K x 8	200-300 ns	256K E^2PROM with 64-Byte Page & Software Data Protection, 3.0-Volt	Now
AT28LV010	128K x 8	200-250 ns	1-Mbit E^2PROM with 128-Byte Page & Software Data Protection, 3.0-Volt	Now
Standard Voltage (5V)				
AT28C16	2K x 8	150-250 ns	16K E^2PROM	Now
AT28C16E	2K x 8	150-250 ns	16K E^2PROM with Extended Endurance & Fast Write	Now
AT28C17	2K x 8	150-250 ns	16K E^2PROM with Ready/$\overline{\text{Busy}}$	Now
AT28C17E	2K x 8	150-250 ns	16K E^2PROM with Ready/$\overline{\text{Busy}}$ & Extended Endurance & Fast Write	Now
AT28C64	8K x 8	120-350 ns	64K E^2PROM	Now
AT28C64E	8K x 8	120-350 ns	64K E^2PROM with Extended Endurance & Fast Write	Now
AT28C64X	8K x 8	150-450 ns	64K E^2PROM without Ready-$\overline{\text{Busy}}$	Now
AT28C64B	8K x 8	150-250 ns	64K E^2PROM with 64-Byte Page & Software Data Protection	Now
AT28C256	32K x 8	150-350 ns	256K E^2PROM with 64-Byte Page & Software Data Protection	Now
AT28C256E	32K x 8	150-350 ns	256K E^2PROM with Extended Endurance	Now
AT28C256F	32K x 8	150-350 ns	256K E^2PROM with Fast Write & Software Data Protection	Now
AT28C010	128K x 8	120-250 ns	1-Mbit E^2PROM with 128-Byte Page & Software Data Protection	Now
AT28C010E	128K x 8	120-250 ns	1-Mbit E^2PROM with 128-Byte Page & Extended Endurance & Software Data Protection	Now
AT28C040	512K x 8	150-250 ns	4-Mbit E^2PROM with 256-Byte Page & Software Data Protection	Now

Flash Memory Cards

Part Number	Organization	V	Description	Availability
AT5FC001	1 Mbyte	5.0 V	PCMCIA Compatible Flash Memory Card	Now
AT5FC002	2 Mbyte	5.0 V	PCMCIA Compatible Flash Memory Card	Now
AT5FC004	4 Mbyte	5.0 V	PCMCIA Compatible Flash Memory Card	Now
AT5FC008	8 Mbyte	5.0 V	PCMCIA Compatible Flash Memory Card	Now

EPROMs

Part Number	Organization	Speeds	Description	Availability
Battery-Voltage ™ **(2.7V)**				
AT27BV010	128K x 8	90-150 ns	1-Mbit, 2.7-Volt to 3.6-Volt EPROM	Now
AT27BV020	256K x 8	120-150 ns	2-Mbit, 2.7-Volt to 3.6-Volt EPROM	Now
AT27BV040	512K x 8	150 ns	4-Mbit, 2.7-Volt to 3.6-Volt EPROM	Now
Low Voltage (3 to 5.5V)				
AT27LV256R	32K x 8	150-250 ns	256K 3-Volt EPROM	Now
AT27LV512R	64K x 8	150-250 ns	512K 3-Volt EPROM	Now
AT27LV1024	64K x 16	150-250 ns	1-Mbit, 3-Volt EPROM	Now
AT27LV010	128K x 8	150-250 ns	1-Mbit, 3-Volt EPROM	Now
AT27LV020	256K x 8	150-300 ns	2-Mbit, 3-Volt EPROM	Now
AT27LV4096	256K x 16	200-300 ns	4-Mbit, 3-Volt EPROM	Now
AT27LV040	512K x 8	200-300 ns	4-Mbit, 3-Volt EPROM	Now
AT27LV080	1024K x 8	250-300 ns	8-Mbit, 3-Volt EPROM	4Q-96
Standard Voltage (5V)				
AT27C256R	32K x 8	45-200 ns	256K EPROM	Now
AT27C512R	64K x 8	45-200 ns	512K EPROM	Now
AT27C1024	64K x 16	55-200 ns	1-Mbit EPROM	Now
AT27C010,L	128K x 8	45-200 ns	1-Mbit EPROM, Standard & Low Power	Now
AT27C020	256K x 8	70-200 ns	2-Mbit EPROM	Now
AT27C4096	256K x 16	85-200 ns	4-Mbit EPROM	Now
AT27C040	512K x 8	80-200 ns	4-Mbit EPROM	Now
AT27C080	1024K x 8	100-200 ns	8-Mbit EPROM	Now

North American Sales Offices

NORTHWEST

2125 O'Nel Drive
San Jose, CA 95131
TEL (408) 436-4270
FAX (408) 436-4314

NORTHEAST

300 Granite Street, #106
Braintree, MA 02184
TEL (617) 849-0220
FAX (617) 848-0012

135 Michael Cowpland Dr., #203
Kanata, Ontario K2M 2E9
Canada
TEL (613) 599-5338
FAX (613) 599-5337

MID-ATLANTIC

101 Carnegie Center, #205
Princeton, NJ 08540
TEL (609) 520-0606
FAX (609) 520-9175

SOUTHEAST

809 Spring Forest Road, #600
Raleigh, NC 27609
TEL (919) 850-9889
FAX (919) 850-9894

NORTH CENTRAL

1721 Moon Lake Blvd., #430
Hoffman Estates, IL 60194
TEL (708) 310-1200
FAX (708) 310-1650

SOUTH CENTRAL

11782 Jollyville Rd.
Austin, TX 78759
TEL(512)219-4050
FAX(512)219-4051

17304 Preston Road, Suite 720
Dallas, TX 75252
TEL (214) 733-3366
FAX (214) 733-3163

SOUTHWEST

8101 Kaiser, Suite 140
Anaheim Hills, CA 92808
TEL (714) 282-8080
FAX (714) 282-0500

International Sales Offices

UNITED KINGDOM

Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686677
FAX (44) 1276-686697

FINLAND

Atmel OY
Sinikalliontie 5
02630 Espoo
Finland
TEL (358) 0-5023026
FAX (358) 0-5023126

FRANCE

Atmel Southern Europe
55 Avenue Diderot
94100 St. Maur Des Fosses
Paris, France
TEL (33) 1-48855522
FAX (33) 1-48855596

GERMANY

Atmel GmbH
Ginnheimer Strasse 45
D-60487 Frankfurt 90
Germany
TEL (49) 69-7075910
FAX (49) 69-7075912

Atmel GmbH
Niederlassung Sud
Litzdorfer Strasse 11
D-83064 Raubling
Germany
TEL (49) 8034-9127
FAX (49) 8034-9330

HONG KONG

Atmel Asia, Ltd.
Room 1219, Chinachem Golden Plaza
77 Mody Road, Tsimshatsui East
Kowloon
Hong Kong
TEL (852) 27219778
FAX (852) 27221369

ITALY

Ufficio di Milano
Centro Direzionale Colleoni
Palazzo Andromeda 3
20041 Agrate Brianza
Italy
TEL (39) 39 605 69 55
FAX (39) 39 605 69 69

JAPAN

Atmel Japan K.K.
Thomas Bldg., 16-1
Nihonbashi Hakosaki-Cho
Chuo-Ku, Tokyo 103
Japan
TEL (81) 3-5641-0211
FAX (81) 3-5641-0217

KOREA

Atmel Korea, Ltd.
6F, Norsan Bldg., 106-8
Guro 5--Dong, Guro-Ku
Seoul, Korea (152-055)
TEL (82) 2-8396341
FAX (82) 2-8396343

SINGAPORE

Atmel Singapore PTE., Ltd.
6001 Beach Road
Golden Mile Tower #21-01
Singapore 0719
TEL (65) 2999-212
FAX (65) 2910-955

SWEDEN

Atmel Sweden
P.O. Box 142
S-19422 Upplands Vasby
Sweden
TEL (46) 8-590-74910
FAX(46) 8-590-74910

TAIWAN

Atmel Taiwan Ltd.
FL 15-4, No. 83, Sec. 1
Nan-Kan Road
Lu Chu Hsiang, Taoyuan Hsien
Taiwan, R.O.C.
TEL (886) 3-3229133
FAX (886) 3-3229131

Alabama

ALL AMERICAN SEMICONDUCTOR
4900 University Square
Suite 34
Huntsville, AL 35816
TEL (205) 837-1555
FAX (205) 837-7733

ARROW/SCHWEBER ELECTRONICS
1015 Henderson Road
Hunstville, AL 35816
TEL (205) 837-6955
FAX (205) 721-1581

INSIGHT ELECTRONICS
4835 University Square
Suite 19
Huntsville, AL 35818
TEL (205) 830-1222
FAX (205) 830-1225

MARSHALL INDUSTRIES
3313 Memorial Parkway South
Huntsville, AL 35801
TEL (205) 881-9235
FAX (205) 881-1490

MILGRAY/HUNTSVILLE
5021 Bradford Drive
Suite 202
Huntsville, AL 35805
TEL (205) 722-9709
FAX (205) 722-0161

PIONEER STANDARD ELECTRONICS
4835 University Square
Suite 5
Huntsville, AL 35816
TEL (205) 837-9300
FAX (205) 837-9358

Arizona

ARROW/SCHWEBER ELECTRONICS
2415 West Erie Drive
Tempe, AZ 85282
TEL (602) 431-0030
FAX (602) 431-9555

INSIGHT ELECTRONICS
1515 West University
Suite 103
Tempe, AZ 85281
TEL (602) 829-1800
FAX (602) 967-2658

MARSHALL INDUSTRIES
9831 S. 51st Street
Suite C107-109
Phoenix, AZ 85044
TEL (602) 496-0290
FAX (602) 893-9029

PIONEER STANDARD ELECTRONICS
1438 West Broadway
Suite B-140
Tempe, AZ 85282
TEL (602) 350-9335
FAX (602) 350-9376

California

ALL AMERICAN SEMICONDUCTOR
230 Devcon Drive
San Jose, CA 95112
TEL (408) 441-1300
FAX (408) 437-8970

26010 Mureau Road, # 120
Calabasas, CA 91302
TEL (818) 878-0555
FAX (818) 878-0533

10805 Holder Street, Suite 100
Cypress, CA 90630
TEL (714) 229-8600
FAX (714) 229-8603

5060 Shoreham Place, Suite 115
San Diego, CA 92122
TEL (800) 382-3441
FAX (619) 458-5866

ARROW/SCHWEBER ELECTRONICS
6 Cromwell St., Suite 100
Irvine, CA 92718
TEL (714) 587-0404
FAX (714) 454-4206

Malibu Canyon Business Park
26677 West Agoura Road
Calabasas, CA 91302
TEL (818) 880-9686
FAX (818) 880-4687

9511 Ridgehaven Court
San Diego, CA 92123
TEL (619) 565-4800
FAX (619) 279-0862

1180 Murphy Avenue
San Jose, CA 95131
TEL (408) 441-9700
FAX (408) 453-4810

INSIGHT ELECTRONICS
4333 Park Terrace Dr., Suite 101
Westlake Village, CA 91316
TEL (818) 707-2101
FAX (818) 707-0321

2 Venture Plaza, Suite 340
Irvine, CA 92718
TEL (714) 727-3291
FAX (714) 727-1804

9980 Huennekens Street
San Diego, CA 92121
TEL (619) 587-1100
FAX (619) 587-1380

1295 Oakmead Parkway
Sunnyvale, CA 94086
TEL (408) 720-9222
FAX (408) 720-8390

JAN DEVICES
6925 Canby Avenue, Building 109
Reseda, CA 91335
TEL (818) 757-2000
FAX (818) 708-7436

MARSHALL INDUSTRIES
9320 Telstar Avenue
El Monte, CA 91731
TEL (818) 307-6000
FAX (818) 307-6173

3039 Kilgore Avenue
Rancho Cordova, CA 95670
TEL (916) 635-9700
FAX (916) 635-6044

336 Los Coches Street
Milpitas, CA 95035
TEL (408) 942-4600
FAX (408) 262-1224

One Morgan
Irvine, CA 92718
TEL (714) 859-5050
FAX (714) 581-5255

26537 Agoura Road
Calabasas, CA 91302
TEL (818) 878-7000
FAX (818) 880-6846

5961 Kearny Villa
San Diego, CA 92123
TEL (619) 627-4184
FAX (619) 627-4163

MILGRAY ELECTRONICS, INC.
2880 Zanker Road, Suite 102
San Jose, CA 95134
TEL (408) 456-0900
FAX (408) 456-0300

275 E. Hillcrest Dr., Suite 145
Thousand Oaks, CA 91360
TEL (805) 371-9399
FAX (805) 371-9317

MILGRAY/ORANGE COUNTY
25 Mauchly, Suite 329
Irvine, CA 92718
TEL (714) 753-1282
FAX (714) 753-1682

0412B

6835 Flanders Drive, Suite 300
San Diego, CA 92121
TEL (619) 457-7545
FAX (619) 457-9750

PIONEER STANDARD ELECTRONICS

5126 Clareton Drive, Suite 160
Agoura Hills, CA 91301
TEL (818) 865-5800
FAX (818) 865-5814

9449 BalboaAvenue, Suite 114
San Diego, CA 92123
TEL (619) 560-1318
FAX (619) 514-7799

217 Technology Drive
Suite 110
Irvine, CA 92718
TEL (714) 753-5090
FAX (714) 753-5074

333 River Oaks Pkwy
San Jose, CA 95134
TEL (408) 954-9100
FAX (408) 954-9113

ZEUS ELECTRONICS

6 Cromwell St., Suite 100
Irvine, CA 92718
TEL (714) 581-4622
(800) 52-HI-REL
FAX (714) 454-4355

6276 San Ignacio Avenue, Suite E
San Jose, CA 95119
TEL (408) 629-4789
(800) 52-HI-REL
FAX (408) 629-4792

Colorado

ARROW/SCHWEBER ELECTRONICS

61 Inverness Dr. East, Suite 105
Englewood, CO 80112
TEL (303) 799-0258
FAX (303) 799-0730

INSIGHT ELECTRONICS

384 Inverness Drive South
Suite 105
Englewood, CO 80112
TEL (303) 649-1800
FAX (303) 649-1818

MARSHALL INDUSTRIES

12351 North Grant
Thornton, CO 80241
TEL (303) 451-8383
FAX (303) 457-2899

MILGRAY/COLORADO

5650 D T C Parkway
Suite 202
Englewood, CO 80111
TEL (303) 721-7702
FAX (303) 721-7803

PIONEER TECHNOLOGIES GROUP

5600 Green Wood Plaza Blvd.
Suite 200
Englewood, CO 80111
TEL 303-773-8090
FAX 303-773-8194

Connecticut

ARROW/SCHWEBER ELECTRONICS

860 N. Main Street Extension
Wallingford, CT 06492
TEL (203) 265-7741
FAX (203) 265-7988

MARSHALL INDUSTRIES

20 Sterling Drive
Barnes Industrial Park North
P.O. Box 200
Wallingford, CT 06492-0200
TEL (203) 265-3822
FAX (203) 284-9285

MILGRAY/CONNECTICUT

326 West Main Street
Milford, CT 06460
TEL (203) 878-5538
FAX (203) 878-6970

PIONEER STANDARD ELECTRONICS

Two Trap Falls
Shelton, CT 06484
TEL (203) 929-5600
FAX (203) 929-9791

Florida

ALL AMERICAN SEMICONDUCTOR

16115 N.W. 52nd Avenue
Miami, FL 33014
TEL (305) 621-8282
FAX (305) 620-7831

14450 46th Street
Shelter 116
Clearwater, FL 34622
TEL (813) 532-9800
FAX (813) 538-5567

1400 East Newport Center Drive
Suite 205
Deerfield Beach, FL 33442
TEL (305) 429-2800
FAX (305) 429-0391

ARROW/SCHWEBER ELECTRONICS

400 Fairway Dr.
Deerfield Beach, FL 33441
TEL (305) 429-8200
FAX (305) 428-3991

Bldg. D, Suite 3101, 3102, 3103
37 Skyline Dr.
Lake Mary, FL 32746
TEL (407) 333-9300
FAX (407) 333-9320

MARSHALL INDUSTRIES

2700 W. Cypress Creek Road
Suite D114
Ft. Lauderdale, FL 33309
TEL (305) 977-4880
FAX (305) 977-4887

380 S. Northlake Boulevard
Suite 1024
Altamonte Springs, FL 32701
TEL (407) 767-8585
FAX (407) 767-8676

2840 Scherer Drive, Suite 410
St. Petersburg, FL 33716
TEL (813) 573-1399
FAX (813) 573-0069

MILGRAY/FLORIDA

755 Rinehart Road, Suite 100
Lake Mary, FL 32746
TEL (407) 321-2555
FAX (407) 322-4225

PIONEER STANDARD ELECTRONICS

674 S. Military Trail
Deerfield Beach, FL 33442
TEL (305) 428-8877
FAX (305) 481-2950

337 S. Northlake Boulevard
Suite 1000
Altamonte Springs, FL 32701
TEL (407) 834-9090
FAX (407) 834-0865

ZEUS ELECTRONICS

37 Skyline Drive
Bldg. D, Suite 3101
Lake Mary, FL 32746
TEL (407) 333-3055
(800) 52-HI-REL
FAX (407) 333-9681

Georgia

ARROW/SCHWEBER ELECTRONICS

4250 E River Green Parkway
Duluth, GA 30136
TEL (404) 497-1300
FAX(404) 476-1493

INSIGHT ELECTRONICS
3005 Breckinridge Blvd.
Suite 201A
Duluth, GA 30138
TEL (404) 717-8566
FAX (404) 717-8588

MARSHALL INDUSTRIES
5300 Oakbrook Parkway
Suite 140
Norcross, GA 30093
TEL (404) 923-5750
FAX (404) 923-2743

MILGRAY/ATLANTA
3000 Northwoods Parkway
Suite 115
Norcross, GA 30071
TEL (404) 446-9777
FAX (404) 446-1186

PIONEER STANDARD ELECTRONICS
4250C Rivergreen Parkway
Duluth, GA 30136
TEL (404) 623-1003
FAX (404) 623-0665

Illinois

ALL AMERICAN SEMICONDUCTOR
1930 N. Thoreau, Suite 200
Schaumburg, IL 60173
TEL (708) 303-1995
FAX (708) 303-1996

ARROW/SCHWEBER ELECTRONICS
1140 W. Thorndale Ave.
Itasca, IL 60143
TEL (708) 250-0500
FAX (708) 250-0916

INSIGHT ELECTRONICS
1365 Wiley Road
Suite 142
Schaumburg, IL 60173
TEL (708) 885-9700
FAX (708) 885-9701

MARSHALL INDUSTRIES
50 East Commerce Drive, Unit 1
Schaumburg, IL 60173
TEL (708) 490-0155
FAX (708) 490-0569

MILGRAY/CHICAGO
Kennedy Corporate Center 1
1530 E. Dundee Road, Suite 310
Palatine, IL 60067-8319
TEL (708) 202-1900
FAX (708) 202-1985

PIONEER STANDARD ELECTRONICS
2171 Executive Drive
Suite 200
Addison, IL 60101
TEL (708) 495-9680
FAX (708) 495-9831

ZEUS ELECTRONICS
1140 W. Thorndale Avenue
Itasca, IL 60143
TEL (708) 595-9730
(800) 52-HI-REL
FAX (708) 595-9896

Indiana

ARROW/SCHWEBER ELECTRONICS
7108 Lakeview Parkway West Drive
Indianapolis, IN 46268
TEL (317) 299-2071
FAX (317) 299-2379

MARSHALL INDUSTRIES
6990 Corporate Drive
Indianapolis, IN 46278
TEL (317) 297-0483
FAX (317) 297-2787

MILGRAY/INDIANA
5226 Elmwood Avenue
Indianapolis, IN 46203
TEL (317) 781-9997
FAX (317) 781-6970

PIONEER STANDARD ELECTRONICS
9350 N. Priority Way, W. Drive
Indianapolis, IN 46240
TEL (317) 573-0880
FAX (317) 573-0979

Kansas

ARROW/SCHWEBER ELECTRONICS
9801 Legler Road
Lenexa, KS 66219
TEL (913) 541-9542
FAX (913) 752-2612

MARSHALL INDUSTRIES
10413 West 84th Terrace
Pine Ridge Business Park
Lenexa, KS 68214
TEL (913) 492-3121
FAX (913) 492-6205

MILGRAY/KANSAS CITY
6400 Glenwood, Suite 313
Overland Park, KS 66202
TEL (913) 236-8800
FAX (913) 384-6825

Maryland

ALL AMERICAN
14636 Rothged Drive
Rockville, MD 20850
TEL (800) 426-0420
FAX (301) 251-8574

ARROW/SCHWEBER ELECTRONICS
9800J Patuxent Woods Drive
Columbia, MD 21046
TEL (301) 596-7800
FAX (301) 596-7821

INSIGHT ELECTRONICS
6925 Oakland Mills Road, Suite D
Columbia, MD 21045
TEL (410) 381-3131
FAX (410) 381 3141

MARSHALL INDUSTRIES
9130B Guilford Road
Columbia, MD 21046-1803
TEL (301) 470-2800
FAX (301) 622-0451

MILGRAY/WASHINGTON
6460 Dobbin Road, Suite D
Columbia, MD 21045
TEL (410) 730-6119
FAX (410) 730-8940

PIONEER STANDARD ELECTRONICS
9100 Gaither Road
Gaithersburg, MD 20877
TEL (301) 921-0660
FAX (301) 921-4255

15810 Gaither Road
Gaithersburg, MD 20877
TEL (301) 921-3822
FAX (301) 921-3858

Massachusetts

ALL AMERICAN
19A Crosby Drive
Bedford, MA 01730
TEL (617) 275-8888
FAX (617) 275-1982

ARROW/SCHWEBER ELECTRONICS
25 Upton Drive
Wilmington, MA 01887
TEL (508) 658-0900
FAX (508) 694-1754

INSIGHT ELECTRONICS
55 Cambridge Street, Suite 301
Burlington, MA 01803
TEL (617) 270-9400
FAX (617) 270-3279

MARSHALL INDUSTRIES
33 Upton Drive
Wilmington, MA 01887
TEL (508) 658-0810
FAX (508) 658-7608

MILGRAY/NEW ENGLAND
Ballardvale Park
187 Ballardvale Street
Wilmington, MA 01887
TEL (508) 657-5900
FAX (508) 658-7989

PIONEER STANDARD ELECTRONICS
44 Hartwell Avenue
Lexington, MA 02173
TEL (617) 861-9200
FAX (617) 863-1547

ZEUS ELECTRONICS
25 Upton Drive
Wilmington, MA 01887
TEL (508) 658-4776
(800) 52-HI-REL
FAX (508) 694-2199

Michigan

ARROW SCHWEBER ELECTRONICS
44720 Helm Street
Plymouth, MI 48170
TEL (313) 455-0850
FAX (313) 455-6656

MARSHALL INDUSTRIES
31067 Schoolcraft
Livonia, MI 48150
TEL (313) 525-5850
FAX (313) 525-5855

PIONEER STANDARD ELECTRONICS
4467 Byron Center Avenue
Wyoming, MI 49509
TEL (616) 534-6074
FAX (616) 534-3922

44190 Plymouth Oaks Drive
Plymouth, MI 48270
TEL (313) 416-2157
FAX (313) 416-2415

Minnesota

ALL AMERICAN SEMICONDUCTOR
7716 Golden Triangle Drive
Eden Prairie, MN 55344
TEL (612) 944-2151
FAX (612) 944-9803

ARROW/SCHWEBER ELECTRONICS
10100 Viking Drive, Suite 100
Eden Prairie, MN 55344
TEL (612) 941-5280
FAX (612) 941-9405

10120 A West 76th Street
Eden Prairie, MN 55344
TEL (612) 946-4800

INSIGHT ELECTRONICS
5353 Gamble Drive
Suite 330
St. Louis Park, MN 55416
TEL (612) 525-9999
FAX (612) 525-9998

MARSHALL INDUSTRIES
14800 28th Avenue, North
Suite 175
Minneapolis, MN 55447
TEL (612) 559-2211
FAX (612) 559-8321

PIONEER STANDARD ELECTRONICS
7625 Golden Triangle
Eden Prairie, MN 55344
TEL (612) 944-3355
FAX (612) 944-3794

Missouri

ARROW/SCHWEBER ELECTRONICS
2380 Schuetz Road
St. Louis, MO 63146
TEL (314) 567-6888
FAX (314) 567-1164

MARSHALL INDUSTRIES
514 Earthcity Expressway
Suite 131
Earthcity, MO 63045
TEL (314) 770-1749
FAX (314) 770-1486

PIONEER STANDARD ELECTRONICS
111 West Port Plaza
Suite 625
St. Louis, MO 63146
TEL (314) 542-3077
FAX (314) 542-3078

New Jersey

ARROW/SCHWEBER ELECTRONICS
43 Route 46 East
Pine Brook, NJ 07058
TEL (201) 227-7880
FAX (201) 227-2064

4 East Stow Road, Unit 11
Marlton, NJ 08053
TEL (609) 596-8000
FAX (609) 596-9632

MARSHALL INDUSTRIES
101 Fairfield Road
Fairfield, NJ 07004
TEL (201) 882-0320
FAX (201) 882-0095

158 Gaither Drive
Mt. Laurel, NJ 08054
TEL (609) 234-9100
FAX (609) 778-1819

MILGRAY/DELAWARE VALLEY
523 Fellowship Road
Suite 275
Mt. Laurel, NJ 08054
TEL (609) 778-1300
FAX (609) 778-7669

MILGRAY/NEW JERSEY
3799 Route 46 East
Suite 303
Parsippany, NJ 07054
TEL (201) 335-1766
FAX (201) 335-2110

PIONEER STANDARD ELECTRONICS
14A Madison Road
Fairfield, NJ 07006
TEL (201) 575-3510
FAX (201) 575-3454

New York

ALL AMERICAN SEMICONDUCTOR
275B Marcus Boulevard
Hauppauge, NY 11788
TEL (516) 981-3935
FAX (516) 434-9394

ARROW/SCHWEBER ELECTRONICS
120 Commerce Street
Hauppauge, NY 11788
TEL (516) 231-1000
FAX (516) 231-1072

3375 Brighton-Henrielts Townline Road
Rochester, NY 14623
TEL (716) 427-0300
FAX (716) 427-0735

MARSHALL INDUSTRIES
100 Marshall Drive
Endicott, NY 13760
TEL (607) 785-2345
FAX (607) 785-5546

1250 Scottsville Road
Rochester, NY 14624
TEL (716) 235-7620
FAX (716) 235-0052

3505 Veterans Memorial Highway
Suite L.
Ronkonkoma, NY 11779
TEL (516) 737-9300
FAX (516) 737 9580

MILGRAY/NEW YORK
77 Schmitt Boulevard
Farmingdale, NY 11735
TEL (516) 391-3000
FAX (516) 420-0685

MILGRAY/UPSTATE NEW YORK
One Corporate Place, Suite 200
1170 Pittsford Victor Road
Pittsford, NY 14534
TEL (716) 381-9700
FAX (716) 381-9495

PIONEER STANDARD ELECTRONICS
1249 Upper Front, Suite 201
Binghamton, NY 13901
TEL (607) 722-9300
FAX (607) 722-9562

840 Fairport Park
Fairport, NY 14450
TEL (716) 381-7070
FAX (716) 381-5955

One Penn Plaza #2032
New York, NY 10119
TEL (212) 631-4700
FAX (212) 971-0374

60 Crossways Park West
Woodbury, NY 11797
TEL (516) 921-8700
FAX (516) 921-2143

ZEUS ELECTRONICS
100 Midland Avenue
Port Chester, NY 10573
TEL (914) 937-7400
(800) 52-HI-REL
FAX (914) 937-2553

North Carolina

ARROW/SCHWEBER ELECTRONICS
5240 Greens Dairy Road
Raleigh, NC 27604
TEL (919) 876-3132
FAX (919) 878-9517

MARSHALL INDUSTRIES
5224 Greens Dairy Road
Raleigh, NC 27604
TEL (919) 878-9882
FAX (919) 872-2431

MILGRAY/RALEIGH
2925 Huntleigh Drive
Suite 101
Raleigh, NC 27604
TEL (919) 790-8094
FAX (919) 872-8851

PIONEER STANDARD ELECTRONICS
2200 Gateway Center Blvd.
Suite 215
Morrisville, NC 27560
TEL (919) 460-1530
FAX (919) 460-1540

Ohio

ARROW/SCHWEBER ELECTRONICS
8200 Washington Village Dr. Suite A
Centerville, OH 45458
TEL (513) 435-5563
FAX (513) 435-2049

6573 E Cochran Road
Solon, OH 44139
TEL (216) 248-3990
FAX (216) 248-1106

INSIGHT ELECTRONICS
9700 Rockside Road
Suite 105
Valley View, OH 44125
TEL (216) 487-2522
FAX (216) 487-3412

MARSHALL INDUSTRIES
30700 Bainbridge Road, Unit A
Solon, OH 44139
TEL (216) 248-1788
FAX (216) 248-2312

3520 Park Center Drive
Dayton, OH 45414
TEL (513) 898-4480
FAX (513) 898-9363

MILGRAY/CLEVELAND
6155 Rockside Road, Suite 206
Cleveland, OH 44131
TEL (216) 447-1520
FAX (216) 447-1761

PIONEER STANDARD ELECTRONICS
2385 Edison Blvd.
Twinsburg, OH 44087
TEL (216) 487-5500
FAX (216) 487-0256

4800 East 131st Street
Cleveland, OH 44105
TEL (216) 587-3600
FAX (216) 587-3906

4433 Interpoint Boulevard
Dayton, OH 45424
TEL (513) 236-9900
FAX (513) 236-8133

100 Old Wilson Bridge Road
Suite 105
Worthington, OH 43085
TEL (614) 848-4854
FAX (614) 848-4889

Oklahoma

ARROW/SCHWEBER ELECTRONICS
12111 E. 51st Street, Suite 101
Tulsa, OK 74146
TEL (918) 252-7537
FAX (918) 254-0917

PIONEER STANDARD ELECTRONICS
9717 E. 42nd Street, Suite 105
Tulsa, OK 74146
TEL (918) 665-7840
FAX (918) 665-1891

Oregon

ALMAC/ARROW ELECTRONICS
1885 N.W. 169th Place
Beaverton, OR 97006
TEL (503) 629-8090
FAX (503) 645-0611

ALL AMERICAN/PORTLAND
1815 N.W. 169th Place
Suite 6025
Beaverton, OR 97006
TEL (800) 531-3334
FAX (503) 531-3695

INSIGHT ELECTRONICS
8705 S.W. Nimbus Avenue
Suite 200
Beaverton, OR 97005
TEL (503) 644-3300
FAX (503) 641-4530

MARSHALL INDUSTRIES
9705 S.W. Gemini Drive
Beaverton, OR 97005
TEL (503) 644-5050
FAX (503) 646-8256

MILGRAY/OREGON
8705 S.W. Nimbus Ave., Suite 260
Beaverton, OR 97008
TEL (503) 626-4040
FAX (503) 641-0650

PIONEER TECHNOLOGIES GROUP
8905 Southwest Nimbus
Suite 160
Beaverton, OR 97008
TEL 503-626-7300
FAX 503-626-5300

Pennsylvania

ARROW/SCHWEBER ELECTRONICS
2681 Mosside Blvd., Suite 204
Monroeville, PA 15146
TEL (412) 856-9490
FAX (412) 856-9507

PIONEER STANDARD ELECTRONICS
500 Enterprise Road
Horsham, PA 19044
TEL (215) 674-4000
FAX (215) 674-3107

259 Kappa Drive
Pittsburgh, PA 15238
TEL (412) 782-2300
FAX (412) 963-8255

Tennessee

ARROW ELECTRONICS
3865 South Perkins Road
Memphis, TN 38118

Texas

ALL AMERICAN SEMICONDUCTOR
11210 Steeplecrest, Suite 206
Houston, TX 77065
TEL (713) 955-1993
FAX (713) 955-2215

ALL AMERICAN/DALLAS
1771 International Parkway
Suite 101
Richardson, TX 75081
TEL (800) 541-1435
FAX (214) 437-0353

AMIGA SALES & MARKETING C/O JAN DEVICES
12342 Hunters Chase Blvd.
Suite 3127
Austin, TX 78729
TEL (818) 757-2005
FAX (818) 708-7436

ARROW/SCHWEBER ELECTRONICS
Braker Center III, Bldg. M1
11500 Metric Blvd., Suite 160
Austin, TX 78758
TEL (512) 835-4180
FAX (512) 832-9875

3220 Commander Drive
Carrollton, TX 75006
TEL (214) 380-6464
FAX (214) 248-7208

19416 Park Row, Suite 190
Westgate Center, Bldg. B
Houston, TX 77084
TEL (713) 647-6868
FAX (713) 492-8722

INSIGHT ELECTRONICS, INC.
1778 Plano Road, Suite 320
Richardson, TX 75081
TEL (214) 783-0800
FAX (214) 680-2402

11500 Metric Boulevard
Suite 215
Austin, TX 78758
TEL (512) 719-3090
FAX (512) 719-3091

10777 Westheimer, Suite 1100
Houston, TX 77042
TEL (713) 260-9614
FAX (713) 260-9602

MARSHALL INDUSTRIES
8504 Cross Park Drive
Austin, TX 79764
TEL (512) 837-1991
FAX (512) 832-9810

Corporate Square Tech
Center III
1551 North Glenville Drive
Richardson, TX 75081
TEL (214) 705-0600
FAX (214) 705-0675

10681 Haddington Drive
Suite 160
Houston, TX 77043
TEL (713) 467-1666
FAX (713) 467-9805

MILGRAY/AUSTIN
11824 Jollyville Road
Suite 103
Austin, TX 78759
TEL (512) 331-9961
FAX (512) 331-1070

MILGRAY/DALLAS
16610 North Dallas Parkway
Suite 1300
Dallas, TX 75248
TEL (214) 248-1603
FAX (214) 248-0218

MILGRAY/HOUSTON
12919 S.W. Freeway, Suite 130
Stafford, TX 77477
TEL (713) 240-5360
FAX (713) 240-5404

PIONEER STANDARD ELECTRONICS
1826-D Kramer Lane
Austin, TX 78758
TEL (512) 835-4000
FAX (512) 835-9829

13765 Beta Road
Dallas, TX 75244
TEL (214) 386-7300
FAX (214) 490-6419

10530 Rockley Road
Houston, TX 77099
TEL (713) 495-4700
FAX (713) 495-5642

8200 Interstate 10 West
Suite 705
San Antonio, TX 78230
TEL (512) 377-3440
FAX (512) 378-3626

ZEUS ELECTRONICS
3220 Commander Drive
Carrollton, TX 75006
TEL (214) 380-4330
(800) 52-HI-REL
FAX (214) 447-2222

Utah

ARROW/SCHWEBER ELECTRONICS
1946 W. Parkway Blvd.
Salt Lake City, UT 84119
TEL (801) 973-6913
FAX (801) 972-0200

ALL AMERICAN/UTAH
4455 South 700 East
Suite 301
Salt Lake City, UT 84107
TEL (800) 682-8313
FAX (801) 261-3885

INSIGHT ELECTRONICS
545 East 4500 South
Suite E110
Salt Lake City, UT 84107
TEL 801-288-9043
FAX 801-288-9195

MARSHALL INDUSTRIES
2355 S. 1070 West, Suite D
Salt Lake City, UT 84119
TEL (801) 973-2288
FAX (801) 973-2296

MILGRAY/UTAH
310 E. 4500 South, Suite 110
Murray, UT 84107
TEL (801) 261-2999
FAX (801) 261-0880

Washington

ALMAC/ARROW ELECTRONICS
14360 S.E. Eastgate Way
Bellevue, WA 98007
TEL (206) 643-9992
FAX (206) 649-9709

INSIGHT ELECTRONICS
12002 115th Avenue N.E.
Kirkland, WA 98034
TEL (206) 820-8100
FAX (206) 821-2976

MARSHALL INDUSTRIES
11715 N. Creek Parkway South
Suite 112
Bothell, WA 98011
TEL (206) 466-5747
FAX (206) 486-6964

PIONEER STANDARD ELECTRONICS
2800 156th Avenue SE, Suite 100
Bellevue, WA 98007
TEL (206) 644-7500
FAX (206) 644-7300

Wisconsin

ARROW/SCHWEBER ELECTRONICS
200 North Patrick Blvd.
Brookfield, WI 53045
TEL (414) 792-0150
FAX (414) 792-0156

INSIGHT ELECTRONICS
10855 West Potter Road
Suite 14
Wauwatosa, WI 53222
TEL (414) 258-5338
FAX (414) 258-5360

MARSHALL INDUSTRIES
Crossroads Corporate Center 1
20900 Swenson Drive, Suite 150
Waukesha, WI 53186
TEL (414) 797-8400
FAX (414) 797-8270

PIONEER STANDARD ELECTRONICS
120 Bishops Way, Suite 163
Brookfield, WI 53005
TEL (414) 784-3480
FAX (414) 784-8207

Canada

ARROW/SCHWEBER ELECTRONICS
1100 St. Regis Blvd.
Dorval, Quebec, Canada H9P2T5
TEL (514) 421-7411
FAX (514) 421-7430

8544 Baxter Place
Burnaby, BC, Canada V5A4T8
TEL (604) 421-2333
FAX (604) 421-5030

36 Antares Drive, Unit 100
Nepean, Ontario, Canada K2E7W5
TEL (613) 226-6903
FAX (613) 723-2018

1093 Meyerside Drive
Mississauga, Ontario, Canada L5T1M4
TEL (905) 670-7769
FAX (905) 670-7781

MARSHALL INDUSTRIES
148 Brunswick Boulevard
Pointe Claire, Quebec H9R 5P9
Canada
TEL (514) 694-8142
FAX (514) 694-6989

6285 Northam Dr.
Mississauga, Ontario L4V 1X5
Canada
TEL (905) 465-1771
FAX (905) 612-1988

MILGRAY/TORONTO
2783 Thamesgate Drive
Mississauga, Ontario L4T 1G5
Canada
TEL (905) 678-0958
FAX (905) 678-1213

MILGRAY/MONTREAL
6600 Trans Canada, Suite 209
Pointe Claire, Quebec H9R-4S2
Canada
TEL (514) 426-5900
FAX (514) 426-5836

PIONEER STANDARD ELECTRONICS
560 1212-31 Avenue NE
Calgary, Alberta T2E 7S8
Canada
TEL (403) 291-1988
FAX (403) 291-0740

148 York Street, Suite 209
London, Ontario N6A 1A9
Canada
TEL (519) 672-4666
FAX (519) 672-3528

3415 American Drive
Mississauga, Ontario L4V 1T6
Canada
TEL (905) 405-8300
FAX (905) 405-6425

223 Colonnade Road, Unit 12
Nepean, Ontario K2E 7K3
Canada
TEL (613) 226-8840
FAX (613) 226-6352

10711 Cambie Road, Suite 170
Richmond, B.C. V6X 3G5
Canada
TEL (604) 273-5575
FAX (604) 273-2413

Place Iberville IV
2954 Blvd. Laurier, Suite 100
Ste-Foy, Quebec G1V 4T2
Canada
TEL (418) 654-1077
FAX (418) 654-2958

520 McCaffrey Street
Ville St. Laurent, Quebec H4T 1N1
Canada
TEL (514) 737-9700
FAX (514) 737-5212

Alabama

ELECTRONIC MARKETING ASSOC.

7501 S. Memorial Parkway
Suite 106
Huntsville, AL 35802
TEL (205) 880-8050
FAX (205) 880-8054

Arizona

COMPASS MARKETING

11801 North Tatum Boulevard
Suite 101
Phoenix, AZ 85028
TEL (602) 996-0635
FAX (602) 996-0586

California

PROMERGE SALES, INC.

100 Century Center Court
Suite 710
San Jose, CA 95112
TEL (408) 467-0600
FAX (408) 467-0610

PROLINE TECHNOLOGIES

P.O. Box 1326
Healdsburg, CA 95448
TEL (707) 431-2937
FAX (707) 431-1809

HARPER & STRONG

2798 Junipero Avenue
Signal Hill, CA 90806
TEL (310) 424-3030
FAX (310) 424-6622

SILICON TECHNICAL SALES, INC.

140 Lomas Santa Fe Drive
Suite 203
Solana Beach, CA 92075
TEL (619) 793-3330
FAX (619) 793-4188

Colorado

THORSON ROCKY MOUNTAIN

7108 D South Alton Way, Suite A
Englewood, CO 80112
TEL (303) 773-6300
FAX (303) 773-6302

Connecticut

DELTA-CONN TECHNICAL SALES

One Prestige Drive, 2nd Floor
Suite 206
Meriden, CT 06450
TEL (203) 634-8558
FAX (203) 238-1240

Florida

COMPONENT DESIGN MARKETING

800 Corporate Drive, Suite 230
Ft. Lauderdale, FL 33334
TEL (305) 492-1160
FAX (305) 492-1167

1900 S.W. 85th Avenue
North Lauderdale, FL 33068
TEL (305) 726-5444
FAX (305) 726-5155

2318 Stag Run Boulevard
Clearwater, FL 34625
TEL (813) 725-4894
FAX (813) 796-7252

7616 Southland Boulevard
Suite 103
Orlando, FL 32809
TEL (407) 240-3903
FAX (407) 240-4305

4502 West Elm Street
Tampa, FL 33614
TEL (813) 886-9721
FAX (813) 888-7816

Calle Hucar 38 (Bajos)
BO Sabanetas
Mercedita 00715
Puerto Rico
TEL (809) 844-3840
FAX (809) 844-3915

Georgia

ELECTRONIC MARKETING ASSOC.

5855 Jimmy Carter Blvd.
Suite 190
Norcross, GA 30071
TEL (404) 448-1215
FAX (404) 446-9363

Iowa

DY-TRONIX, INC.

23 Twixt Town Road N.E.
Cedar Rapids, IA 52402
TEL (319) 377-8275
FAX (319) 377-9163

Illinois

PHASE II MARKETING

2220 Hicks Road, Suite 206
Rolling Meadows, IL 60008
TEL (708) 577-9401
FAX (708) 577-9491

Indiana

CORRAO MARSH, INC.

6211 Stoney Creek Drive
Fort Wayne, IN 46825
TEL (219) 482-2725
FAX (219) 484-7491

3063 West U.S. 40
Greenfield, IN 46140
TEL (317) 462-4446
FAX (317) 462-6568

MILLENNIUM TECHNICAL SALES

6640 Broadway Street
Indianapolis, IN 46220
TEL (317) 257-0828
FAX (317) 257-0837

Kansas

DY-TRONIX, INC.

5001 College Boulevard, Suite 106
Leawood, KS 66211
TEL (913) 339-6333
FAX (913) 339-9449

1999 Amidon, Suite 322
Wichita, KS 67203
TEL (316) 838-0884
FAX (316) 838-2645

Maryland

AVTEK ASSOCIATES, INC.

10632 Little Patuxent Parkway,
Suite 220
Columbia, MD 21044
TEL (410) 740-5100
FAX (410) 740-5103

Massachusetts

CTC ASSOCIATES, INC.

12 Southwest Park
Westwood, MA 02090
TEL (617) 320-1818
FAX (617) 320-8282

Michigan

TRILOGY MARKETING, INC.

691 N. Squirrel Road, Suite 110
Auburn Hills, MI 48326
TEL (810) 377-4900
FAX (810) 377-4906

Minnesota

PSI

8000 Town Line Avenue S.
Suite 206
Bloomington, MN 55438
TEL (612) 944-8545
FAX (612) 944-6249

Missouri

DY-TRONIX, INC.

3407 Bridgeland Drive
Bridgeton, MO 63044
TEL (314) 291-4777
FAX (314) 291-3861

Nevada

PROLINE TECHNOLOGIES

P.O. Box 1326
Healdsburg, CA 95448
TEL (707) 431-2937
FAX (707) 431-1809

New Jersey

NORTH EAST COMPONENTS

19 Spear Road, Suite 205
Ramsey, NJ 07446
TEL (201) 825-0233
FAX (201) 934-1310

TRITEK SALES, INC.

1 Mall Drive, Suite 410
Cherry Hill, NJ 08002
TEL (609) 667-0200
FAX (609) 667-8741

New Mexico

COMPASS MARKETING

4100 Osuna Road, Suite 109
Albuquerque, NM 87109
TEL (505) 344-9990
FAX (505) 345-4848

New York

EMPIRE TECHNICAL ASSOC.

29 Fennell Street, Suite A
Skaneateles, NY 13152
TEL (315) 685-5703
FAX (315) 685-5979

349 West Commercial Street
Suite 2920
E. Rochester, NY 14445
TEL (716) 381-8500
FAX (716) 381-0911

North Carolina

ELECTRONIC MARKETING ASSOC.

6600 Six Forks Road, Suite 201
Raleigh, NC 27615
TEL (919) 847-8800
FAX (919) 848-1787

9407 King Falls Dr.
Charlotte, NC 28210
TEL (704) 544-9948
FAX (704) 544-9941

Ohio

MILLENNIUM TECHNICAL SALES

3165 Linwood Road
Cincinnati, OH 45208
TEL (513) 871-2424
FAX (513) 871-2524

6631 Commerce Parkway
Suite K
Dublin, OH 43017
TEL (614) 793-9545
FAX (614) 793-0256

6519 Wilson Mills Road
Mayfield Village, OH 44143
TEL (216) 461-3500
FAX (216) 461-1335

Oklahoma

QUAD STATE SALES
& MARKETING

110 W. Commercial Street
Suite 210
Broken Arrow, OK 74013
TEL (918) 258-7723
FAX (918) 258-7653

Oregon

ELECTRONIC SOURCES, INC.

6850 SW 105th, Suite B
Beaverton, OR 97008
TEL (503) 627-0838
FAX (503) 627-0238

Pennsylvania

MILLENNIUM TECHNICAL SALES

505 Bayberry Lane
Inperial, PA 15126
TEL (412) 695-7661
FAX (412) 695-7870

Texas

QUAD STATE SALES
& MARKETING

8310 Capital of Texas Hwy, North
Suite 365
Austin, TX 78731
TEL (512) 346-7002
FAX (512) 346-3601

12160 Abrams Road, Suite 406
Dallas, TX 75243
TEL (214) 669-8567
FAX (214) 669-8834

10565 Katy Fwy, Suite 212
Houston, TX 77024
TEL (713) 467-7749
FAX (713) 467-5942

Utah

THORSON ROCKY MOUNTAIN

5505 South 900 East
Suite 140
Salt Lake City, UT 84117
TEL (801) 264-9665
FAX (801) 264-9881

Washington

ELECTRONIC SOURCES, INC.

1603 116th Ave, N.E., Suite 115
Bellevue, WA 98004
TEL (206) 451-3500
FAX (206) 451-1038

Wisconsin

PHASE II MARKETING

205 Bishop's Way
Suite 220
Brookfield, WI 53005
TEL (414) 797-9986
FAX (414) 797-9935

Canada

CLARK-HURMAN ASSOCIATES

78 Donegani, Suite 200
Pointe Claire, Quebec H9R 2V4
Canada
TEL (514) 426-0453
FAX (514) 426-0455

308 Palladium Drive, Suite 200
Kanata, Ontario K2V 1A1
Canada
TEL (613) 599-5626
FAX (613) 599-5707

16 Regan Road, Units 39, 40, 41
Brampton, Ontario L7A 1C1
Canada
TEL (905) 840-6066
FAX (905) 840-6091

Australia

GEC ELECTRONICS DIVISION

38 South Street
Rydalmere, N.S.W. 2116
Australia
TEL (61) 2898-7422
FAX (61) 2638-1798

Austria

CODICO

Muhlgasse 86-88
A-2380
Perchtoldsdorf/A
Austria
TEL (43) 1-863-3050
FAX (43) 1-863-0598

Belgium

ALCOM ELECTRONICS BV

Singel 3
2550 Kontich
Belgium
TEL (32) 3-4583033
FAX (32) 3-4583126

Brazil

COLGIL, INC.

Rua Marques de Itu, 306, Conj. 94
CEP 01223-000
Sao Paulo
Brazil
TEL (55) 11-223-6954
FAX (55) 11-223-4989

Headquarters
New York
United States
TEL (212) 832-1340
FAX (212) 826-3623

Tucuman 1567, Officina 14
Buenos Aires
Argentina
TEL (54) 1-46-2128
FAX (54) 1-46-2128

HASTEC

Rua Ferreira Do Alentecj, 90/92
CEP 04728
Sao Paulo
Brazil
TEL (55) 11-522-1799
FAX (55) 11-522-5366

Bulgaria

CODICO

Blvd. Bulgaria 86 / Apt. 9
BG 5500 Lovetsch
Bulgaria
TEL (359) 68-44812
FAX (359) 68-44812

Denmark

MER-EL A/S

Ved Klaedebo 18
Post Boks 219
DK-2970 Horsholm
Denmark
TEL (45) 42-571000
FAX (45) 42-572299

Finland

OY BEXAB FINN-CRIMP AB

P.O. Box 51
SF-02631 Espoo
Finland
TEL (358) 0-61352690
FAX (358) 0-61352655

France

MICRO PUISSANCE

Immeuble Femto
1 Avenue de Norvege
Z.A. de Courtaboeuf B.P. 79
91943 Les Ulis Cedex
France
TEL (33) 1-69071211
FAX (33) 1-69076712

NEWTEK

8 Rue De L'Estrerel
Silic 583
94663 Rungis Cedex
France
TEL (33) 1-46872200
FAX (33) 1-46878049

Germany

INELTEK GMBH

Hauptstrasse 45
89522 Heidenhiem
Germany
TEL (49) 7321-93850
FAX (49) 7321-938595

INELTEK MITTE

Stehnweg 2
63500 Seligenstadt
Germany
TEL (49) 6182-5066
FAX (49) 6182-65953

INELTEK MURNAU

Am Fugsee 21
82418 Murnau-Riedhausen
Germany
TEL (49) 8841-47775
FAX (49) 8841-2660

INELTEK NORD

Billstrasse 30
20539 Hamburg 26
Germany
TEL (49) 78942-274
FAX (49) 78942-220

Greece

MICRELEC LTD.

339 Thivon Street
GR 12244 Aegaleo
Athens, Greece
TEL (30) 1-5395042-4
FAX (30) 1-5390269

Hong Kong

TLG ELECTRONICS, LTD.

Room 1404, Hang Shing Bldg.
363-373, Nathan Road
Yanumatei, Kowloon
Hong Kong
TEL (852) 2388-7613
FAX (852) 2783-0198

WES TECH ELECTRONICS, LTD.

Room 1601, Star Centre
433-451, Castle Peak Road
Kwai Chung, N.T.
Hong Kong
TEL (852) 2418-9818
FAX (852) 2429-2355

Hungary

CODICO KFT

Ostrom Utca 23-25
1015 Budapest
TEL (36) 1-156-63-30
FAX (36) 1-156-43-76

India

ORIOLE SERVICES
Post Box No. 9275
5 Kurla Industrial Estate
Ghatkopar, Bombay 400 086
India
TEL (022) 5119940
FAX (022) 5115810

Ireland

ZEC SERVICES

Valleymount
Blessington
Co. Wicklow
Ireland
TEL (353) 456-4259
FAX (353) 467-4075

Israel

ISAMTEK LTD.

53 Herzel Street
P.O. Box 13902
Netanya 42-137
Israel
TEL (972) 9-826445
FAX (972) 9-826447

8

0410B

Italy

LASI ELETTRONICA S.P.A.

Viale Fulvio Testi 280
20126 Milano
Italy
TEL (39) 2-66101370
FAX (39) 2-66101385

NEWTEK ITALIA SPA

Via Tonoli 1
20145 Milano
Italy
TEL (39) 2-33105308
FAX (39) 2-33103694

Japan

MCM JAPAN LTD.

Santower Bldg.
2-11-22, Sangenjyaya
Setagaya-Ku, Tokyo 154
Japan
TEL (81) 3-3487-8477
FAX (81) 3-3487-8825

Tachikawa Sales Office
Tachikawa Center Bldg.
2-22-20, Akebono Cho
Tachikawa
Tokyo 190
Japan
TEL (81) 425-22-8600
FAX (81) 425-23-8603

TAKACHIHO KOHEKI CO., LTD.

1-2-8, Yotsuya
Shinjuku-Ku
Tokyo 160
Japan
TEL (81) 3-3355-6696
FAX (81) 3-3357-5034

Fukoku Seimei Bldg.
2-4, Komatsubara-Cho
Kita-Ku, Osaka 530
Japan
TEL (81) 6-313-0671
FAX (81) 6-313-3380

RYOYO ELECTRO CORP.

Konwa Bldg., 1-12-22
Tsukiji, Chuo-Ku
Tokyo 104
Japan
TEL (81) 3-3546-5011
FAX (81) 3-3546-5044

Osaka Branch
Nissin Syokuhin Bldg., 4-1-1
Nishi Nakajima, Yodogawa-ku
Osaka 532
Japan
TEL (81) 6-301-1221
FAX (81) 6-302-1002

UNI ELECTRONICS, INC.

P.O. Box 68, Sumitomo Bldg.
2-6-1 Nishi Shinjukn
Shinjuku-ku
Tokyo 163
Japan
TEL (81) 3-3347-8878
FAX (81) 3-3347-8808

Osaka Branch
5-13-9, Mitejima
Nishi Yodogawa-ku
Osaka 555
Japan
TEL (81) 6-473-8429
FAX (81) 6-473-5513

Korea

I & C MICROSYSTEMS CO. LTD

801, 8th Floor, Bethel Building
324-1, Yang Jae-Dong
Seocho-Ku
Seoul, Korea
TEL (82) 2-577-9131
FAX (82) 2-577-9130

SHINHWA CORPORATION

2F, The Christian Literature,
Society of Korea Bldg.
169-1, Samsung-Dong,
Kangnam-Ku
Seoul, Korea
TEL (82) 2-554-6431
FAX (82) 2-554-7649

UNIQUEST KOREA

Suite 1110, Daejong Bldg.
143-48, Samsung-Dong,
Kangnam-Ku
Seoul, Korea
TEL (82) 2-562-8805
FAX (82) 2-562-6646

Mexico

ADELSA

Club Cuicacalli #66
CTO Cronistas
Zona Azul CD Satelite
Naucalpan De Juarez
C.P. 53100 EDO de Mexico
Mexico
TEL (525) 374-0981
FAX (525) 374-0997

Netherlands

ALCOM ELECTRONICS BV

Essebaan 1
2908 LG Capelle Aan Den Yssel
Netherlands
TEL (31) 10-4519533
FAX (31)10-4586482

New Zealand

APEX ELECTRONICS LTD.

175 Vivian Street
Wellington, New Zealand
TEL (64) 4-3853404
FAX (64) 4-3853483

GEC ELECTRONICS

5 Reliable Way
Penrose Auckland
New Zealand
TEL (64) 9-526-0107
FAX (64) 9-525-7923

Norway

NC NORDCOMP AS

PO Box 190
2020 Skedsmokorset
Norway
TEL (47) 63-879330
FAX (47) 63-879000

Poland

CODICO

Ul Bora Komorowskiego 6/17
86-300 Grudziadz
Poland
TEL (48) 51-23223
FAX (48) 51-23223

Portugal

ANATRONIC-PORTUGAL S.A.

Rua Padre Antonio Vieira, No 31-B
Povoa De Santo Adriao - 2675 Odivelas
Portugal
TEL (351) 1-9376267
FAX (351) 1-9371834

Romania

CODICO

BD Decebal NR 11 BLS 14
sc 2, Apt. 41, Sect. 3
Bucharest
TEL (40) 1-3210431
FAX (40) 1-3210431

Singapore

NUCLEUS ELECTRONICS PTE LTD.

6001 Beach Road #18-03
Golden Mile Tower
Singapore 0719
TEL (65)297-6883
FAX (65) 297-6882

SINGASOF COMPUTERS PTE., LTD.

2 Kallang Pudding Road #09-04
Mactech Industrial Bldg.
Singapore 1334
TEL (65) 747-3012
FAX (65) 747-5279

South Africa

ADVANCED SEMICONDUCTOR DEVICES (PTY), LTD.

P.O. Box 3853
Rivonia 2128
South Africa
TEL (27) 11-444-2333
FAX (27) 11-444-1706

Spain

ANATRONIC S.A.

AVDA. De Valladolid, 27
28008 Madrid
Spain
TEL (34) 1-542-44-55
FAX (34) 1-559-69-75

Bailen, 176. Entresuelo 1o
08037 Barcelona
Spain
TEL (34) 3-458-19-06
FAX (34) 3-458-71-28

Las Mercedes 25 3o Dpto. 1
48930 Las Arenas-Vizcaya
Spain
TEL (34) 4-463-60-66
FAX (34) 4-463-42-35

Sweden

KELTECH COMPONENTS AB

Box 505 Kemistvagen 10 A
S-183 25
Taby
Sweden
TEL (46) 86308590
FAX (46) 87562143

Switzerland

ANATEC AG

Sumpfstrasse 7
CH 6300
ZUG
Switzerland
TEL (41) 42-412441
FAX (41) 42-413124

Taiwan

ALLREACH ENTERPRISE CO., LTD.

6F, No. 50-1, Sec. 1
Hsin Sheng South Road
Taipei, Taiwan
R.O.C.
TEL (886) 2-321-4229
FAX (886) 2-321-5849

APPLIED COMPONENT TECHNOLOGY CORP.

8F, No. 233-1 Pao Chiao Road
Hsin Tien City
Taipei Hsien
Taiwan, R.O.C.
TEL (886) 2-917-0858
FAX (886) 2-917-1895

NEOLEC INTERNATIONAL INC.

3F, Jen Jen Bldg
No. 29, Jen Ai Road, Sec. 3,
Taipei 106
Taiwan, R.O.C.
TEL (886) 2-778-8716
FAX (886) 2-778-6076

PRINCETON TECHNOLOGY CORP.

2F, No. 233-1, Bao Chiao Road
Hsin Tien, Taipei Hsien
Taiwan, R.O.C.
TEL (886) 2-917-8856
FAX (886) 2-917-3836

TOPTREND TECHNOLOGIES CORP.

8F, No. 669, Sec. 5
Chung Hsiao East Road
Taipei, Taiwan
R.O.C.
TEL (886) 2-769-6220
FAX (886) 2-769-6228

Thailand

SEMIKON COMPANY, LTD.

4F, Room 406, Phansak Bldg.
Phetburi Road
138/1 Rajthevee, Bangkok, 10400
Thailand
TEL (662) 215-0760
FAX (662) 215-6857

Turkey

AZTECH ELECTRONIK, LTD.

Kaptanpasa Sok No. 25/2
06700 G.O.P.
Ankara, Turkey
TEL (90) 312-447-0384
FAX (90) 312-447-0387

United Kingdom

GD TECHNIK, LTD.

Tudor House
24 High Street
Twyford, Berks RG10 9AG
England
TEL (44) 734-342277
FAX (44) 734-342896

PRONTO ELECTRONIC SYSTEMS, LTD.

City Gate House
399-425 Eastern Ave.
Gants Hill
Ilford, Essex I92 6LR
England
TEL (44) 81-554-6222
FAX (44) 81-518-3222