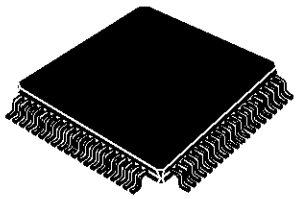
SGS-THOMSON MICROELECTRONICS

ST7548

PCMCIA AND PC GENERAL PURPOSE INTERFACE

ADVANCE DATA

- PCMCIA AND PC GENERAL PURPOSE INTERFACE (MODEM, ISDN, MULTIMEDIA)
- ExCA™ COMPATIBLE (see Note 1)
- 494 BYTES INTERNAL RAM WITH BUS ARBITRATION
- PCMCIA CONFIGURATION REGISTERS (R0, R1)
- CONFIGURABLE I/O DECODER
- I^2C EEPROM INTERFACE (256 BYTES)
- 16C450 UART WITH 16 WORDS FIFO ON DATA
- CONFIGURABLE CLOCK GENERATOR
- I/O - MCU - MEMORY EXTENSION 8 BITS PORT
- HIGH THROUGHPUT :
  - 115200bps THROUGH UART
  - 2MBYTES THROUGH RAM
- 3.3V OR 5V SUPPLY (±10%)
- LOW POWER
- SMALL OUTLINE TQFP80 PACKAGE
- ALSO AVAILABLE WITH COMPLETE MODEM CHIP SET



TQFP80
(Plastic Package)

ORDER CODE : ST7548CQFP

GENERAL DESCRIPTION

The ST7548 is a flexible interface for the PCMCIA and PC buses mainly dedicated to telecommunications. It is fully in accordance with PCMCIA standard and is designed to be coupled to a modem. It includes a UART, a 494 bytes internal memory, an I^2C interface allowing chip initialization from an external EEPROM, and a programmable clock generator.

It also includes a configurable port which can be used either as I/O or memory extension or as an interface to a wide range of microprocessors. In this case, using the internal memory, high speed data transfers can be achieved between the PC and the application.

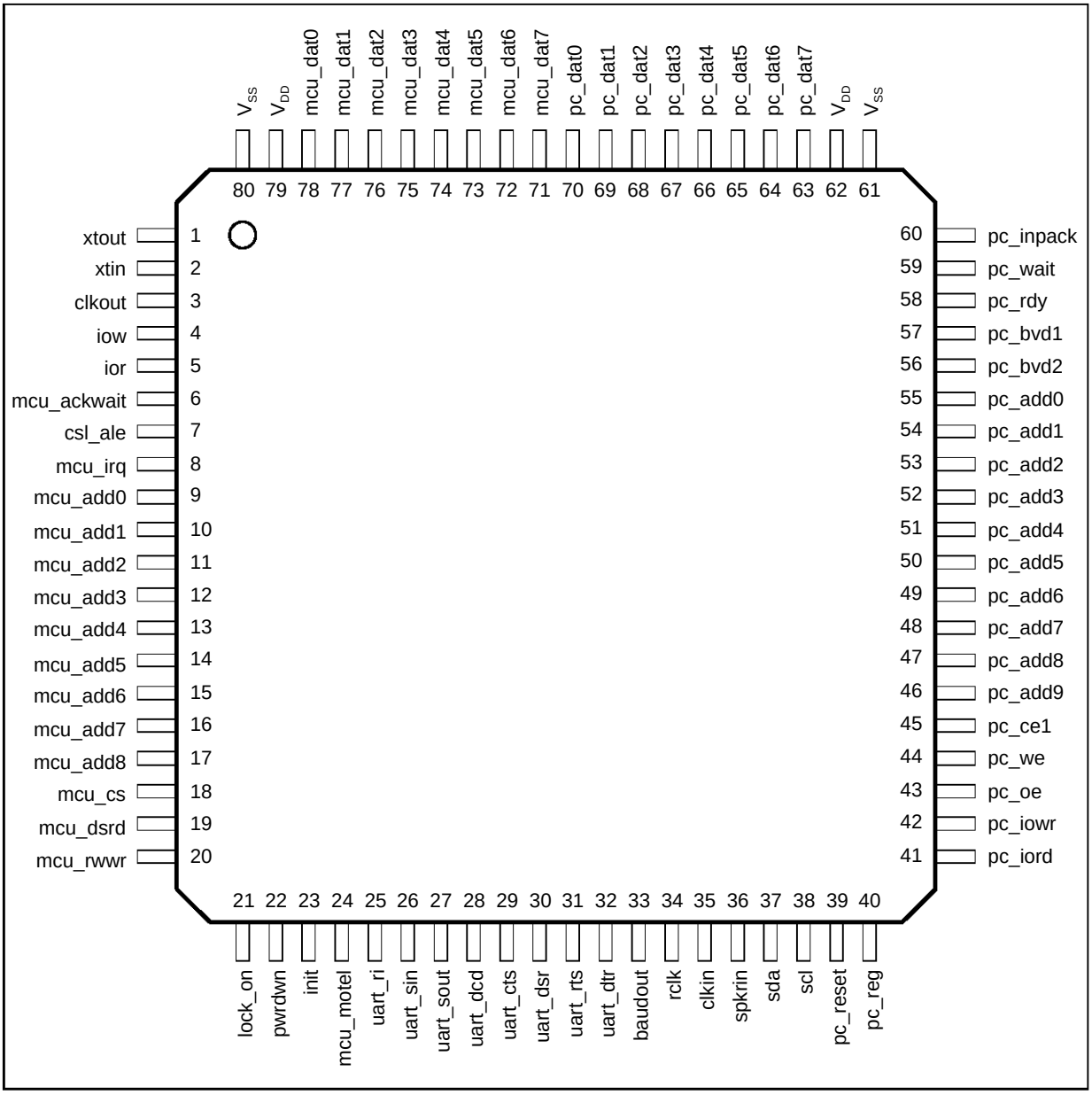
It is realised in an ultra-flat TQFP80 package suitable for PCMCIA cards.

Note : 1. ExCa is a trademark of Intel Corporation.

CONTENTS

CONTENTS (continued) **Page**

I - PIN CONNECTIONS



II - PIN DESCRIPTION

Name	Pin	Type	Function
POWER PINS (ALL THE POWER AND GROUND PINS MUST BE CONNECTED)			
V_{DD1}	62	Power	DC Supply
V_{DD2}	79	Power	DC Supply
V_{SS1}	61	Ground	DC Ground
V_{SS2}	80	Ground	DC Ground
PC address bus			
PC_ADD_0	55	I	PCMCIA Address Bus Bit 0 (LSB)
PC_ADD_1	54	I	PCMCIA Address Bus Bit 1
PC_ADD_2	53	I	PCMCIA Address Bus Bit 2
PC_ADD_3	52	I	PCMCIA Address Bus Bit 3
PC_ADD_4	51	I	PCMCIA Address Bus Bit 4
PC_ADD_5	50	I	PCMCIA Address Bus Bit 5
PC_ADD_6	49	I	PCMCIA Address Bus Bit 6
PC_ADD_7	48	I	PCMCIA Address Bus Bit 7
PC_ADD_8	47	I	PCMCIA Address Bus Bit 8
PC_ADD_9	46	I	PCMCIA Address Bus Bit 9 (MSB)
PC DATA BUS			
PC_DAT_0	70	I/O	PCMCIA DATA Bus Bit 0 (LSB)
PC_DAT_1	69	I/O	PCMCIA DATA Bus Bit 1
PC_DAT_2	68	I/O	PCMCIA DATA Bus Bit 2
PC_DAT_3	67	I/O	PCMCIA DATA Bus Bit 3
PC_DAT_4	66	I/O	PCMCIA DATA Bus Bit 4
PC_DAT_5	65	I/O	PCMCIA DATA Bus Bit 5
PC_DAT_6	64	I/O	PCMCIA DATA Bus Bit 6
PC_DAT_7	63	I/O	PCMCIA DATA Bus Bit 7 (MSB)
PC CONTROL SIGNALS			
PC_CE1	45	I Pull-up	Chip-select. Active low.
PC_WE	44	I Pull-up	Memory Write Enable. Active low.
PC_OE	43	I Pull-up	Memory Output Enable. Active low.
PC_IOWR	42	I Pull-up	I/O Write Enable. Active low.
PC_IORD	41	I Pull-up	I/O Read Enable. Active low.
PC_REG	40	I Pull-up	Common (1) or Attribute (0) Memory Indicator
PC_RESET	39	I, Trigger	Hard Reset. Active high.
PC_INPACK	60	O	I/O Read Indicator (0 indicates that an I/O read access is being performed)
PC_WAIT	59	O	Wait Signal. Active low.
PC_RDY	58	O	Ready/Interrupt Active Low : In all modes, indicates when low that the circuit Is under initialization (CIS being read or written). If no EEPROM is connected, PC_RDY remains low after reset until the MCU writes something in the PROGN register. In I/O modes(2), transmits the interrupt coming from the UART, the I/O port or the RAM.
PC_BVD1	57	O	In memory modes : always 1. In I/O modes : if RINGEN = 1, reproduces UART_RI. if RINGEN = 0, always 1.
PC_BVD2	56	O	In memory modes : always 1. In I/O modes : if AUDIO = 1, reproduces SPKRIN. if AUDIO = 0, always 1.

II - PIN DESCRIPTION

Name	Pin	Type	Function
MCU ADDRESS BUS			
MCU_ADD_0	9	I	MCU Address Bus Bit 0 (LSB)
MCU_ADD_1	10	I	MCU Address Bus Bit 1
MCU_ADD_2	11	I	MCU Address Bus Bit 2
MCU_ADD_3	12	I	MCU Address Bus Bit 3
MCU_ADD_4	13	I	MCU Address Bus Bit 4
MCU_ADD_5	14	I	MCU Address Bus Bit 5
MCU_ADD_6	15	I	MCU Address Bus Bit 6
MCU_ADD_7	16	I	MCU Address Bus Bit 7
MCU_ADD_8	17	I	MCU Address Bus Bit 8 (MSB)
MCU - I/O - MEMORY EXTENSION DATA BUS(3)			
MCU_DAT_0	78	I/O	MCU, I/O or Memory Extension Data Bus Bit 0 (LSB)
MCU_DAT_1	77	I/O	MCU, I/O or Memory Extension Data Bus Bit 1
MCU_DAT_2	76	I/O	MCU, I/O or Memory Extension Data Bus Bit 2
MCU_DAT_3	75	I/O	MCU, I/O or Memory Extension Data Bus Bit 3
MCU_DAT_4	74	I/O	MCU, I/O or Memory Extension Data Bus Bit 4
MCU_DAT_5	73	I/O	MCU, I/O or Memory Extension Data Bus Bit 5
MCU_DAT_6	72	I/O	MCU, I/O or Memory Extension Data Bus Bit 6
MCU_DAT_7	71	I/O	MCU, I/O or Memory Extension Data Bus Bit 7 (MSB)
MCU - I/O - MEMORY EXTENSION CONTROL SIGNALS			
MCU_CS	18	I	Chip Select. Active low
MCU_DSRD	19	I	Data Strobe or Read Enable, depending on MCU_MOTEL (active low)
MCU_RWWR	20	I	Read-Write or Write Enable, depending on MCU_MOTEL (active low)
MCU_MOTEL	24	I	Motorola (1) or Intel (0) Like Bus Style
MCU_ACKWAIT	6	O open collector	Wait or Acknowledge depending on MCU-MOTEL (active low)
CSL_ALE	7	I/O	In memory extension mode : output, active low. Indicates (when 0) an access to the memory extension. In I/O extension mode : output, active low. Indicates (when 0) that the I/O port address has been recognized. In other modes : address latch enable input active high. It is used when the MCU data bus is multiplexed to validate the 8 lsb's of the address on MCU_DAT. The address is latched inside the circuit when CSL_ALE goes low. When the MCU bus is not multiplexed, CSL_ALE must be connected to 0.
MCU_IRQ	8	I/O open collector	In memory extension mode : output, not used. In I/O extension mode : interrupt input active low, transmitted to the PC on the PC_RDY pin. In other modes : interrupt output towards the MCU, active low.
IOW	4	O	In memory extension mode : reproduces PC_WE. In I/O extension mode, reproduces PC_IOW when the I/O port address is recognized (otherwise takes value 1). In other modes : remains at 1.
IOR	5	O	In memory extension mode : reproduces PC_OE. In I/O extension mode, reproduces PC_IOR when the I/O port address is recognized (otherwise takes value 1). In other modes : remains at 1.

II - PIN DESCRIPTION

Name	Pin	Type	Function
UART RELATED PINS			
UART_RI	25	I	Ring Input : can be reproduced on PC_BVD1
UART_SIN	26	I	Serial Input
UART_SOUT	27	O	Serial Output
UART_DCD	28	I	Data Carrier Detect
UART_CTS	29	I	Clear To Send
UART_DSR	30	I	Data Set Ready
UART_RTS	31	O	Request To Send
UART_DTR	32	O	Data Terminal Ready
BAUDOUT	33	O	UART Transmit Clock
RCLK	34	I	Receive Clock for the UART
SERVICE PORTS			
INIT	23	O	Initialization signal provided by the chip for use on the PCMCIA card. It is active high when the card receives a hard or a soft reset.
PWRDWN	22	O	Reproduces Bit PWRDWN of Register R1. active high.
LOCK_ON	21	I	Two consecutive rising transitions on this signal put the chip in stand-by : CLKOUT receives MCU_DIV output instead of the internal clock, unless between these two transitions, stand-by procedure is cancelled. Stand-by procedure is cancelled when any of UART_RI or UART_SOUT goes low, or when any of PWRDWN or PC_RESET goes high.
CLKIN	35	I	Alternate Clock Input. (possibly delivered by a data-pump modem). It is either this clock (when Bit 3 of PROGN register is '1'), or the internal oscillator clock (when Bit 3 of PROGN register is '0') which will be used by UART_DIV to produce the CLOCK for the internal UART.
CLKOUT	3	O	Output Clock Available for a MCU. It is either the internal oscillator clock, or the clock coming from MCU_DIV in stand-by mode when it is desired to reduce the MCU's power consumption.
XTIN	2	I	Oscillator Input. Can be connected either on an external quartz to use the internal oscillator, or on an external oscillator output. The clock on this pin is the one which is used internally by the circuit to operate its internal logic and state machines. The clock used by the UART can come from XTIN or from CLKIN after a division to get the proper frequency.
XTOUT	1	O	Internal Oscillator Output. When the internal oscillator is used, to be connected on the other pin of the quartz crystal.
SPKRIN	36	I	Speaker Input. This input can be reproduced on PC_BVD2 output in I/O mode when AUDIO Bit is one.
I^2C LINE (PORTS RELATED TO THE EXTERNAL EEPROM)			
SDA	37	I/O open collector	I^2C Bus Data Line. An external pull-up resistor is mandatory on this pin.
SCL	38	I/O open collector	I^2C Bus Clock Line. Although this pin is inout, the circuit only uses this Pin as an out Pin. For this reason, one cannot modify the I^2C line frequency by forcing SCL low from outside the chip.

Notes :
1. Memory mode : mode for which, depending on the value of bits UE, MODE0, MODE1 in R0 register, only common or attribute memory accesses are allowed (i.e., no I/O access is allowed).
2. I/O mode : mode for which, depending on the value of bits UE, MODE0, MODE1 in R0 register, only attribute memory and I/O accesses are allowed (i.e., no common memory access is allowed).
3. A number of pins can be used in different ways either to connect a MCU or to connect a memory extension or an I/O extension. These pins are MCU_DAT[7..0], MCU_IRQ, CSL_ALE, IOW and IOR. This corresponds to three different behaviours for the circuit.

III - BLOCK DIAGRAMS

The here under schematic shows the main functions and data flows of the chip.

The CIS (Card Information Structure in PCMCIA terminology) is a memory and registers area which describes the PCMCIA card configuration. During chip initialization, it is loaded from an external EEPROM through the I^2C channel (this is the normal behaviour when such a memory is present) or from the MCU (if no E2PROM is connected ; in this case, loading the CIS is under the MCU's responsibility).

The RAM allows high speed transfers between the PC and the MCU.

The UART may exchange informations with another compatible UART.

When no MCU is necessary, it is possible to connect instead a peripheral requiring an 8 bits parallel interface, or an 8 bits memory extension. There is only one set of ports for the MCU, the peripheral and the memory extension. Therefore, only one of them may be connected at a time on the chip.

After reset, the RAM memory contains the CIS (loaded from the EEPROM through the I^2C bus, or by the MCU). The RAM is also used to exchange information between the PC and the MCU. Its capacity is 494 bytes (hexadecimal addresses ranging from x"000" to x"1ED"). It is organized in such a way that the CIS does not take all the available room. Thus it is possible to perform exchanges between the PC and the MCU while keeping the CIS available for read.

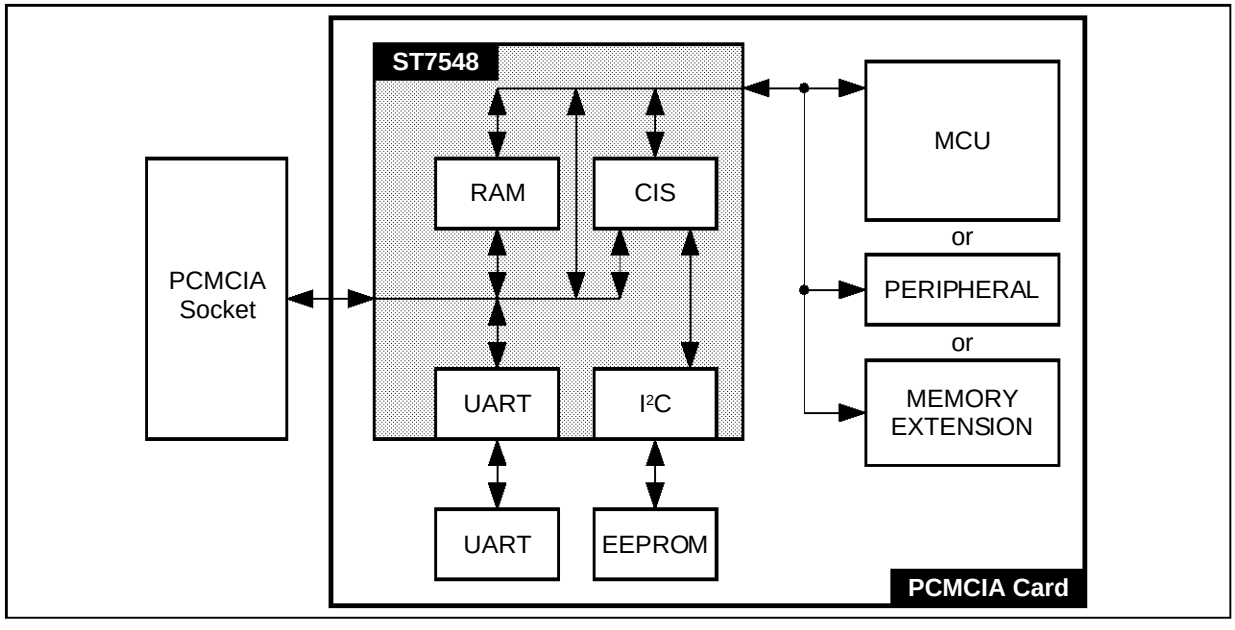
The clocks block produces the necessary clock signals. It contains multiplexers and frequency dividers which can be programmed with the PROGN register. With its internal oscillator, it delivers the external clock CLKOUT which can be used to drive a MCU. It provides the UART with the required clock, building it either from the internal oscillator, or from the external clock input CLKIN. It incorporates programmable dividers which allow to adapt the different clock frequencies.

The registers are used to configurate the chip and to synchronise data transfer between the PC and the MCU.

The circuit considers that they are three masters : the PCMCIA, the MCU and the I^2C interfaces. The registers and the memory can be reached by the masters. The table in paragraph IV.11 "ADDRESSING CAPABILITIES TABLE" shows the access permission. Each device which is supposed to be reached by at least two masters is placed on a common internal bus. An arbitration logic manages the resulting access conflicts. For this reason, accessing one of the registers or memory placed on the common bus makes the related master support WAIT cycles.

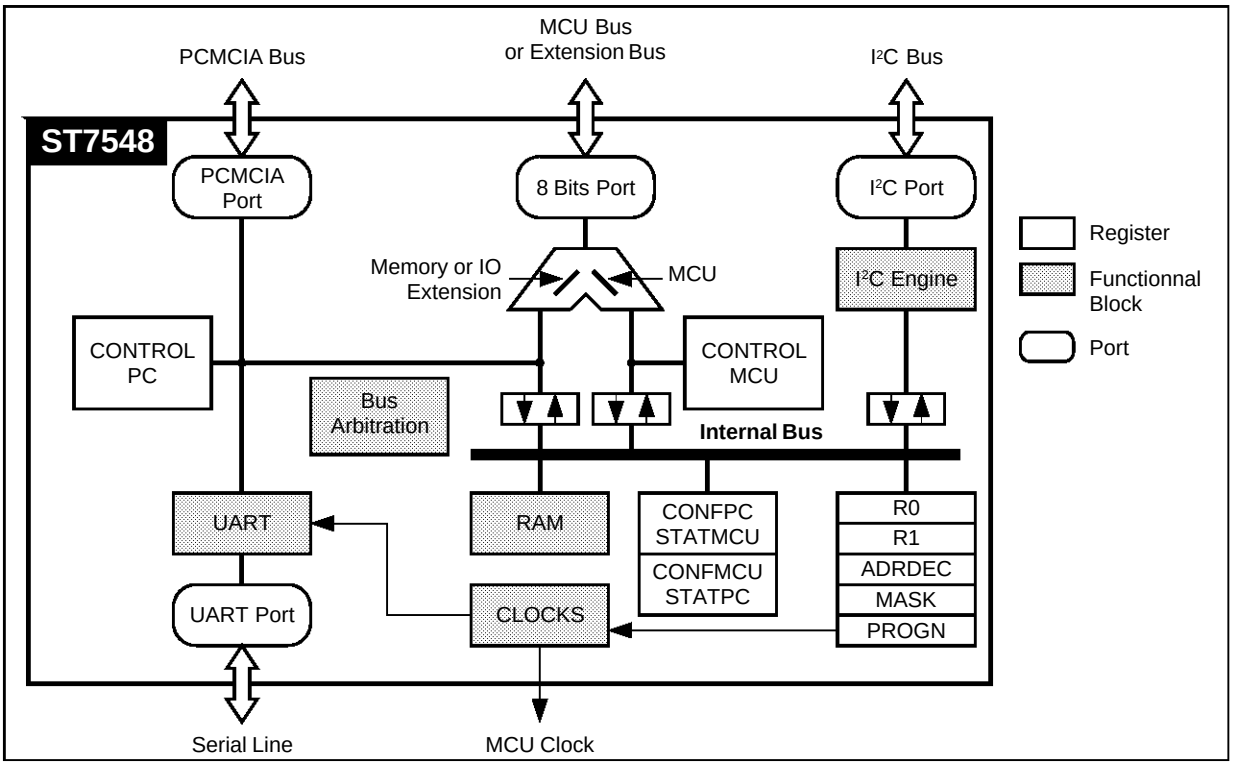
The addresses are given in the table. Two elements can be placed at variable addresses : the UART and the I/O port. For the first one, the address is defined in the R0 register ; for the second one, it is defined in the ADRDEC register, and the recognized address field width is defined in the MASK register.

Figure 1 : Simplified Block Diagram of the Data Flows Allowed by the Chip



III - BLOCK DIAGRAMS (continued)

Figure 2 : Chip Block Diagram



IV - FUNCTIONNAL DESCRIPTION

IV.1 - PCMCIA ADDRESSING MODES

Through a PCMCIA port, a computer may perform its accesses in three different modes : the common memory mode, the attribute memory mode, and the I/O mode.

The usual addressing mode for memory is common memory mode. The PC_REG signal is '1', and the PC_OE and PC_WE signals are used as read/write signals.

The attribute memory mode is mainly used to read configuration information (in registers or memory) of the PCMCIA cards. It differs from the common memory mode in that the PC_REG signal is '0'.

The I/O mode is used to access I/O ports. In this case, the PC_REG is '0' and the read/write signals are PC_IORD and PC_IOWR. The I/O mode allows interrupts to be transmitted to the PCMCIA port. That is why, in the ST7548, the I/O mode can be used not only to access the I/O port, but also to access the memory, thus allowing to synchronize data transfer between the MCU and the PC using interrupts.

In the ST7548, from the PC (or PCMCIA) point of view, there are registers, a memory, an UART, an I/O port and a memory extension port. Each of them cannot be addressed in any mode. For example, the UART and the I/O port can be addressed only in I/O mode. The table in paragraph IV.11 "ADDRESSING CAPABILITIES TABLE" shows in the PCMCIA columns under which conditions the PCMCIA port accesses the different blocks in the circuit, and when the circuit generates a wait state to the PCMCIA interface. It also shows what can be addressed by the MCU (and if a WAIT is sent to the MCU), and what can be reached by the I^2C channel.

IV - FUNCTIONNAL DESCRIPTION (continued)

IV.2 - CLOCKS (Figure 3)

Introduction

The circuit includes an oscillator (XTIN and XTOUT pins) which has been tuned for frequencies ranging from 12 to 24MHz. This oscillator requires an external quartz crystal. It is also possible to operate the circuit at frequencies outside this range, but in this case a quartz crystal cannot be used : an external oscillator must deliver the desired frequency on the XTIN pin. Regardless of its origin, the clock present on XTIN becomes internally CLK, the basic clock of the ST7548. CLK is also named "internal oscillator clock". It is used to operate the internal logic of the chip (state machines and others).

The circuit delivers the CLKOUT clock which relies on the internal oscillator clock. CLKOUT can be used as the basic clock for an MCU placed outside the ST7548. In stand-by, CLKOUT can be slowed down owing to the "MCU_DIV" divider.

The on-chip UART receives the clock which goes out of the frequency divider named "UART_DIV". The input of this divider is either the internal oscillator clock or an external clock connected on CLKIN pin. The division ratios in UART_DIV are well suited to the following clock frequencies : 3.6864MHz, 7.3728MHz, 14.7456MHz, 18.432MHz, 22.1184MHz, 29.4912MHz, 33.1776MHz, 36.864MHz.

The clocks operate in three different ways : normal behaviour, stand-by, and power down.

Normal Behaviour

After reset is released, the clocks run freely, and it is the internal oscillator clock that is sent on UART_DIV. Then, depending on the content of PROGN register bit 3, the UART clock may come from CLK or CLKIN divided in UART_DIV.

The MCU_DIV divider is not used : the multiplexer which follows and delivers CLKOUT uses the CLK or CLK/2 input. Thus, on CLKOUT, it is CLK/2 which is issued when PROGN register bit 7 is '0' (default value), and CLK when this bit is '1'.

Stand-by

What is desired in this mode, is to temporarily reduce the PCMCIA card power consumption, but the card remains able to become operational again at any moment.

Putting the card in stand-by depends upon a MCU action. For this, it sends two consecutive rising edges on the LOCK_ON pin. Doing this causes MCU_DIV to become active. This divider then takes into account the division ratio indicated in the PROGN register. The frequency delivered on CLKOUT is then a sub-multiple of CLK. A PC_RESET (active high) or one of the UART signals UART_RI, UART_SOUT at '0' wake up the circuit and put it back in normal mode. In the same way, if one of these signals occurs between the two rising edges on LOCK_ON, the stand-by procedure is cancelled.

Depending upon the way that the circuit is used (use in conjunction with a 75C501 data pump for example), the MCU may also stop the 75C501. In this case, the 18.432MHz frequency is no longer delivered on CLKIN, and the UART is stopped, unless the user links the UART_DIV input to the local oscillator (by programming accordingly the PROGN register). Switching between CLKIN and the local oscillator clock is done in such a way that no glitch nor shortened cycle happens. The counterpart is that the clock cycle may temporarily be longer.

Waking up After a Stand-by

The circuit goes out of stand-by if a reset happens or if at least one of the UART signals UART_RI or UART_SOUT becomes '0' or if PWRDWN becomes '1'.

If the PC wants to wake up the circuit, it can do it with the UART, simply sending a byte to it.

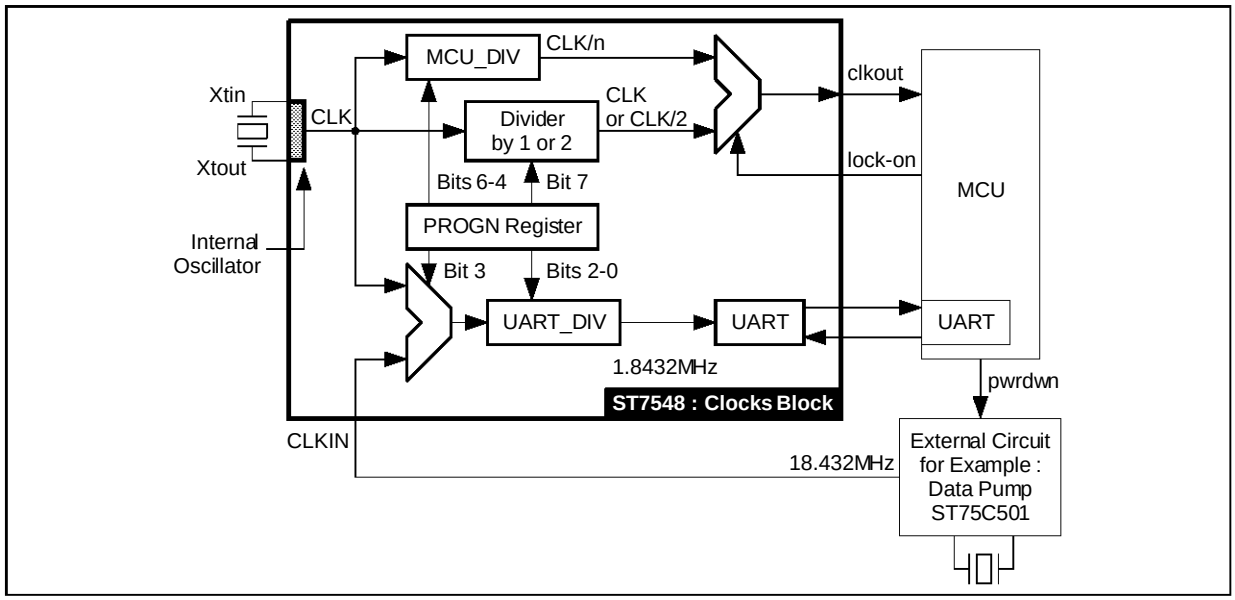
Powerdown Mode

In this mode, the user doesn't intend to use the PCMCIA card on which the circuit is placed. So it is not the UART that will wake it up.

The circuit is placed in powerdown under the PC's responsibility. The PC writes a '1' in R1 register PWRDWN bit. The circuit then no longer delivers any clock neither on CLKOUT nor on the UART, and is no longer able to perform accesses to anything else than the R0 and R1 registers. The circuit will only restart after the PC has written a '0' in the PWRDWN bit.

IV - FUNCTIONNAL DESCRIPTION (continued)

Figure 3 : Theoretical Schematic



IV.3 - UART

The UART is fully compatible with a 16C450 UART. It also provides a transmit and receive FIFO as in a 16C550 UART (see Chapter X.3). Please refer to the documentation on this type of UART to have a detailed description of its behaviour.

The UART may only be used when the UE bit (Uart Enable) in register R0 is set to '1'. The UART address can be chosen between four values depending on the content of R0 register. The three LSB's of the PCMCIA address bus are transmitted to the UART : they are used to select the internal UART registers. The UART thus covers an 8 bytes address range.

The clock which enters the UART is divided by the value held in DLM and DLL UART registers to produce the NBAUD signal. The most common frequency for the UART clock is 1.8432MHz. The table below shows the required divisor value in DLM-DLL to generate typical baud rates for a set of input frequencies (Table 2).

After reset, DLL and DLM are undefined. They are not affected by reset.

Table 1 : UART Registers

Address (Note 1)	DLAB (Note 2)	Register Name	Comment
0	0	RBR Receiver Buffer Register	Read only
0	0	THR Transmitter Holding Register	Write only
1	0	IER Interrupt Enable Register	
2	X	IIR Interrupt Ident Register	Read only
2	X	FCR Fifo Control Register	Write only
3	X	LCR Line Control Register	
4	X	MCR Modem Control Register	
5	X	LSR Line Status Register	Read only (Note 3)
6	X	MSR Modem Status Register	
7	X	SCR Scratch Register	
0	1	DLL Divisor Latch (LSB)	
1	1	DLM Divisor Latch (MSB)	

Notes : 1. The address given in this column refers to the 3 LSBs of the PCMCIA address bus. The complete address of a register is obtained by adding the indicated address and the one of the COM port as defined in the R0 register.
2. DLAB is bit 7 of the UART_LCR register.
3. This register is intended to be read only during normal operation and written during production testing.

IV - FUNCTIONNAL DESCRIPTION (continued)

Table 2

Line Baud Rate	NBAUD Frequency	UART Clock = 1.8432MHz	UART Clock = 3.072MHz	UART Clock = 8.00MHz (Note 1)
50	800	2304	3840	10000
75	1200	1536	2560	6667
110	1760	1047	1745	4545
134.5	2152	857	1428	3717
150	2400	768	1280	3333
300	4800	384	640	1667
600	9600	192	320	833
1200	19200	96	160	417
1800	28800	64	107	278
2000	32000	58	96	250
2400	38400	48	80	208
3600	57600	32	53	139
4800	76800	24	40	104
7200	115200	16	27	69
9600	153600	12	20	52
19200	307200	6	10	26
38400	614400	3	5	13
56000	896000	2	** (Note 2)	9
115200	1843200	1	**	**
128000	2048000	**	**	4
256000	4096000	**	**	2

Notes : 1. In this column, most theoretical divisors are decimal numbers. They have been rounded to the nearest integer.
2. ** = No suitable divisor.

IV.4 - I^2C LINE INTERFACE

IV.4.1 - Introduction

An I^2C line has two wires : the SDA wire carries serial data, the SCL wire carries the clock. It is used for a wide range of applications, including some EEPROMs. It allows several master and slave devices to exchange data while efficiently managing access conflicts. Each device present on the bus has a unique address. Data transfers, whatever their direction is, are always initialized by a master. The slave may slow down the clock if it finds it is too fast.

It is a simplified version of this line which is incorporated in the circuit. It doesn't manage accesses conflicts between several masters. In fact, the ST7548 will always be the only master on the line. The EEPROM is the slave. This is quite enough for the application area. The I^2C line is managed by a state machine clocked by the internal oscillator clock. The EEPROM must be wired at address 0. The recommended model is a ST24C02A or compatible.

IV.4.2 - State Machine Behaviour

The line is managed by a state machine. After a hard or soft reset, the state machine will try and read an EEPROM using the "Sequential Random Read" protocol. If it doesn't get any answer, there is no EEPROM, and it stops.

The PC may also ask the ST7548 to copy in the EEPROM data that have previously been placed in the RAM. It is a convenient way of working on the CIS. This operation is launched by writing a '1' in CONTROLPC register bit 7 (VALWREEPROM). The state machine puts this bit back to '0' when the dump is finished. The state machine uses the "Byte Write" protocol to transfer the data to the EEPROM.

IV - FUNCTIONNAL DESCRIPTION (continued)

IV.4.3 - Sequential Random Read Protocol

The state machine sends the device address (the EEPROM address on the I^2C line), followed by the first byte address (an address inside the EEPROM). Then, it sends again the device address and reads the first byte, then the second, and so on, without sending any address again. The EEPROM internally performs address incrementing. When the state machine wants to stop reading, it doesn't acknowledge the last received byte and sends a STOP condition on the line (see Figure 4).

IV.4.4 - Byte Write Protocol

The state machine performs a complete write cycle for each byte, (sends the device address, the byte address, and the byte to write) and repeats this operation as long as there is something to write. The EEPROM may take some time to physically write the data, and hence, not be ready when the next data comes. In this case, it answers no_acknowledge after it has received the device address. The state machine makes a polling on this signal (sends the device address until there is an acknowledge) to ensure that every byte will be written (see Figure 5).

The above sequence is repeated for each byte.

IV.5 - BUS ARBITRATION

This block manages access conflicts which can occur during accesses requiring use of the internal bus. Its function is to register access requests, allow accesses and put in a wait state those that can't be satisfied immediately by sending them wait signals. It is controlled by a state machine which operates with the internal oscillator clock.

Between the MCU and the PCMCIA, if requests are sent at high speed, the access priority is switched after each access to prevent one of them from always taking the bus. When the I^2C line requires access, it has absolute priority : when the CIS is being read or dumped, no bus access is possible for the MCU and the PCMCIA.

Each access to any register or RAM by a MOTOROLA like MCU automatically produces a DTACK like wait signal on the MCU_ACKWAIT pin, even if no bus arbitration took place.

Figure 4

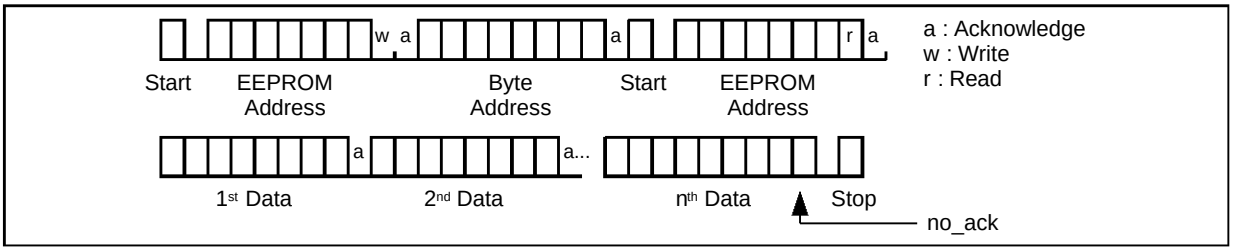
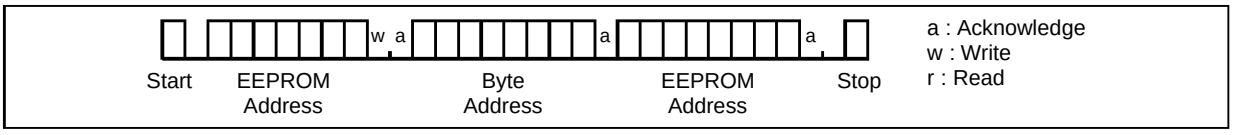


Figure 5



IV - FUNCTIONNAL DESCRIPTION (continued)

IV.6 - REGISTERS

IV.6.1 - Control PC (8 bits, address x"1EE", R/W)

This register is only accessible from the PCMCIA side. Neither the MCU nor the I^2C line can reach it. After a hard or soft reset, the register has all bits at 0.

Bit	Name	Action
7	VALWREEPROM	'1' = initiate a dump of the RAM in the EEPROM (automatically returns to '0' when the dump is done)
6	UART TEST BIT	Must be left at '0'
5	I^2C ESCAPE BIT	Must be left at '0'
4	Not Used	
3	Not Used	
2	Not Used	
1	EIOMEM	Extend I/O Memory Address Space
0	MASK_IT_MCU	at '1', masks the interrupt coming from the MCU

Bit 7 (VALWREEPROM) set at '1' indicates that it is desired to dump a CIS previously written in memory at addresses x"000" to x"0FF" into the EEPROM. This bit is switched back to '0' when the dump is done.

Bit 1 (EIOMEM) : when it is '0' (default value), valid RAM addresses in I/O mode start at x"100" : addresses smaller than x"100" cannot be reached. When set at '1', I/O memory addresses for the RAM start at x"000".

Bit 0 (MASK_IT_MCU) when set at '1' masks the interrupt coming from the MCU.

Bits 5 and 6 (I^2C ESCAPE BIT and UART TEST BIT) are kept for test purposes. After a hard or soft reset, their value is '0'. The user must leave them at '0', otherwise the chip's behaviour would not be in accordance with the specification.

IV.6.2 - Control MCU (8 bits, address x"1EE", R/W)

This register is only accessible from the MCU side. Neither the PCMCIA nor the I^2C line can reach it. After a hard or soft reset, the register has all bits at '0'.

Bit	Name	Action
7	Not Used	
6	Not Used	
5	Not Used	
4	Not Used	
3	Not Used	
2	Not Used	
1	Not Used	
0	MASK_IT_PC	At '1' masks the interrupt coming from the PC

IV.6.3 - Configuration_status (8 bits, address x"1EF")

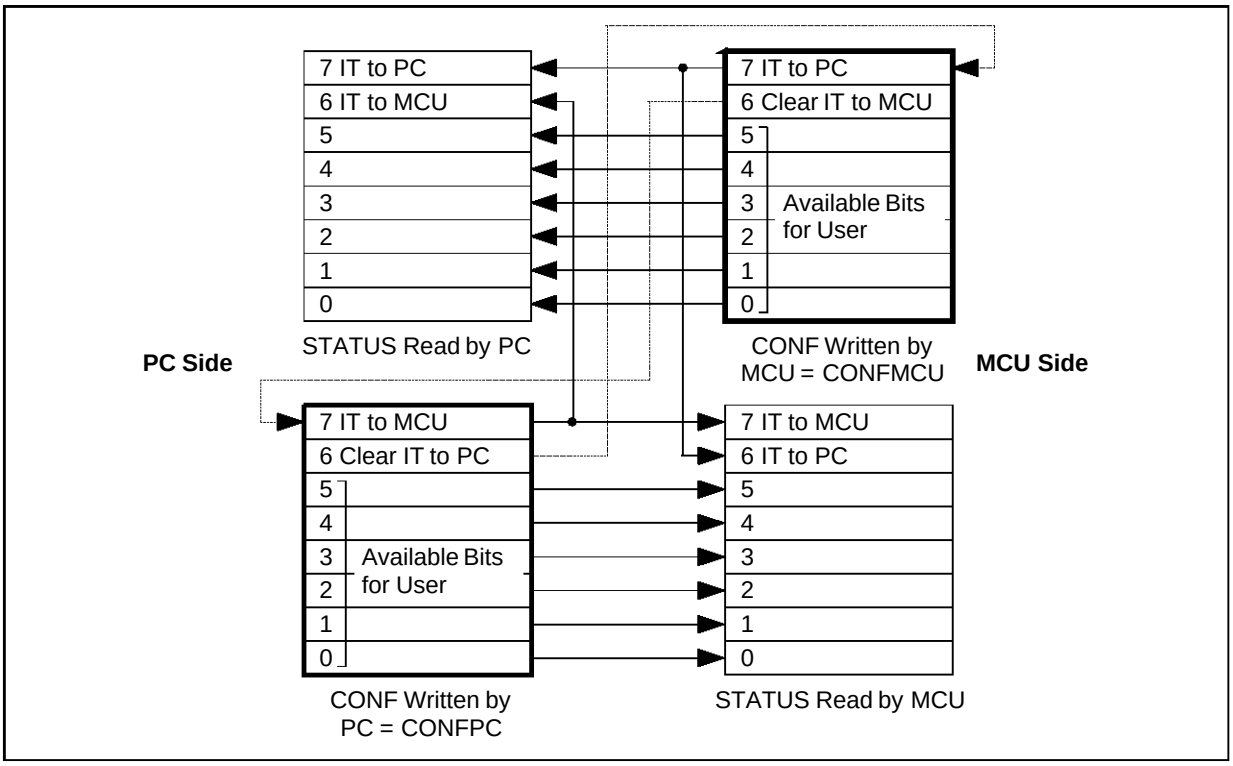
This is a couple of registers which is provided to synchronize data transfer between the PC and the MCU.

Only the configuration registers drawn with bold outline actually exist. The status registers are in fact only the outputs (with a proper wiring) of the configuration registers. Bits 6 and 7 of these registers are active high. (the interrupt is present when a '1' is written on bit 7). Clearing the interrupt which is in bit 7 of one of the CONF registers is done by writing a '1' in bit 6 of the other register. For instance, clearing bit 7 of CONFPC is done by MCU writing a '1' in bit 6 of CONFMCU. It is not necessary to put back bit 6 to '0' after this operation. Erasing the interrupt by writing a '1' in bit 6 will work even if there was already a '1' in this bit, because it is just the fact of writing a '1' which operates, not the fact of having a pulse or an edge on this bit.

The bit range named "available bits for user" can be used freely. It can be used for example to transfer a code telling how many words are to be found, how many transfers there will be, or anything else (see Figure 6).

IV - FUNCTIONNAL DESCRIPTION (continued)

Figure 6



IV.6.4 - R0 (8 bits, address x"1F0", R/W)

This register, accessible by PC and MCU, is initialized by the I^2C channel if an EEPROM is present ; it is a PCMCIA configuration register. It is mainly used to chose the operating mode of the ST7548, seen from the PC side : use with or without an MCU, in I/O or memory, with or without an UART, UART address, interrupts style...

Bit	Name	Action
7	SRESET	Soft reset by PCMCIA active high
6	INTL	Interrupts style (towards PC) : '0' = pulse of at least 5µs, '1' = level
5	UE	Uart Enable : when '1', allows UART use at I/O addresses defined by SEL0 and SEL1
4	Not Used	
3	MODE1	ST7548 operating mode
2	MODE0	ST7548 operating mode
1	SEL1	UART (communication port) selection code
0	SEL0	UART (communication port) selection code

ST7548 operating mode (see also paragraph IV.11 "ADDRESSING CAPABILITIES TABLE") :

MODE1	MODE0	ST7548 Operating Mode
'0'	'0'	Used with an MCU. Memory addressed in common memory
'0'	'1'	Used as memory extension (no MCU)
'1'	'0'	Used as an I/O port (no MCU)
'1'	'1'	Used with a MCU. Memory addressed in I/O mode, allowing interrupts to be sent to the PC.

Communication port selection code :

SEL0	SEL1	Communication Port	Address
'0'	'0'	com1	x"3F8"
'0'	'1'	com2	x"2F8"
'1'	'0'	com3	x"3E8"
'1'	'1'	com4	x"2E8"

During a hard or soft reset (PC_RESET port or SRSET bit at '1'), R0 is reset at '0' except its bit 7 (SRESET) which doesn't turns itself back to'0' when it has been set at '1' : the PC has to turn it back to '0' after a soft reset. Apart from this, for what regards the chip's behaviour, the soft reset has the same action as the hard reset. The circuit waits until the soft or hard reset is finished to start reading the EEPROM.

IV - FUNCTIONNAL DESCRIPTION (continued)

IV.6.5 - R1 (8 bits, address x"1F2", R/W)

This register, accessible by PC and MCU, is initialized by the I^2C channel ; it is a PCMCIA configuration register. It contains all '0' after a hard or soft reset.

Bit	Name	Action
7	Not Used	Leave at '0'
6	Not Used	Leave at '0'
5	Not Used	Leave at '0'
4	RINGEN	Ring Enable : when '1', connects UART_RI to PC_BVD1, otherwise leaves PC_BVD1 at '1'
3	AUDIO	Audio enable : when '1', connects SPKRIN to PC_BVD2, otherwise leaves PC_BVD2 at '1'
2	PWRDWN	Power down active high
1	Not Used	Leave at '0'
0	Not Used	Leave at '0'

During powerdown (when bit PWRDWN is '1'), the circuit prevents from accessing the UART, stops the clock dividers, and doesn't allow I/O and common memory accesses. It only allows attribute memory accesses.

IV.6.6 - ADRDEC (8 bits, address x"1F4", R/W)

This register, accessible by PC and MCU, is initialized by the I^2C channel ; it is a configuration register. It contains all '0' after a hard or soft reset. It is one of the two registers that manage accesses to the I/O port. It defines the base address that the ST7548 will recognize for the I/O port.

Bit	Name	Action
7	A9	I/O port address
6	A8	I/O port address
5	A7	I/O port address
4	A6	I/O port address
3	A5	I/O port address
2	A4	I/O port address
1	A3	I/O port address
0	A2	I/O port address

The I/O port address is decoded by comparing the content of this register with the bits PC_ADD[9..2] on the PC address bus.

Bits 1 and 0 of the address are ignored, so that the decoded address field width is at least 4 bytes. Moreover, depending upon the MASK register content, additional bits in the PC address bus may be ignored, thus making the decoded address range larger. The number of additional bits ignored is given in the MASK register. There may be 0 to 4 additional address bits ignored (PC_ADD[2] to PC_ADD[5]).

IV.6.7 - MASK (8 bits, address x"1F6", R/W)

This register, accessible by PC and MCU, is initialized by the I^2C channel ; it is a configuration register. Its content is all '0' after a hard or soft reset. It is the second register which manages accesses to the I/O port. It has an action on the decoded address field width.

Bit	Name	Action
7	Not Used	
6	Not Used	
5	Not Used	
4	Not Used	
3	Not Used	
2	W2	Decoded address field width code
1	W1	
0	W0	

Address field width code table :

W2	W1	W0	Action	Decoded I/O Address Field Width
'0'	'0'	'0'	PC_ADD[2..5] not masked	4 bytes corresponding to PC_ADD[0..1]
'0'	'0'	'1'	PC_ADD[2] masked PC_ADD[3..5] not masked	8 bytes corresponding to PC_ADD[0..2]
'0'	'1'	'0'	PC_ADD[2..3] masked PC_ADD[4..5] not masked	16 bytes corresponding to PC_ADD[0..3]
'0'	'1'	'1'	PC_ADD[2..4] masked PC_ADD[5] not masked	32 bytes corresponding to PC_ADD[0..4]
'1'	'0'	'0'	PC_ADD[2..5] masked	64 bytes corresponding to PC_ADD[0..5]

IV - FUNCTIONNAL DESCRIPTION (continued)

IV.6.8 - PROGN (8 bits, address x"1F8", R/W)

This register, accessible by PC and MCU, is initialized by the I^2C channel ; it is a configuration register. It contains all '0' after a hard or soft reset. It drives the clock dividers which generate the UART clock and the clock output on port CLKOUT.

Bit	Name	Action
7	NOT_HALF_CLK	Frequency choice for CLKOUT in normal operation : '0' : CLKOUT frequency is half the one on XTIN port. '1' : CLKOUT frequency is the same as the one on XTIN port. This bit is meaningless in stand-by.
6	MCU_DIV(2)	MCU_DIV division ratio code during stand-by for CLKOUT
5	MCU_DIV(1)	
4	MCU_DIV(0)	
3	UART_CLK	Clock input choice for UART_DIV which generates the UART clock
2	UART_DIV(2)	UART_DIV division ratio code for the UART clock
1	UART_DIV(1)	
0	UART_DIV(0)	

- Bit 3 : input clock choice for UART_DIV :
  - '0' → internal oscillator clock (XTIN port).
  - '1' → external clock input port CLKIN.
- Code giving UART_DIV division ratio : report to tables below.

CLKOUT port stand-by division ratio table :

Bit 6	Bit 5	Bit 4	Stand-by Division Ratio
'0'	'0'	'0'	2
'0'	'0'	'1'	4
'0'	'1'	'0'	8
'0'	'1'	'1'	16
'1'	'0'	'0'	32
'1'	'0'	'1'	64
'1'	'1'	'0'	128
'1'	'1'	'1'	256

UART clock division ratio table :

Bit 2	Bit 1	Bit 0	UART Clock Division Ratio	Example : UART Divider Input Clock giving 1.8432MHz on the UART
'0'	'0'	'0'	2	3.6864MHz
'0'	'0'	'1'	4	7.3728MHz
'0'	'1'	'0'	8	14.7456MHz
'0'	'1'	'1'	10	18.432MHz
'1'	'0'	'0'	12	22.1184MHz
'1'	'0'	'1'	16	29.4912MHz
'1'	'1'	'0'	18	33.1776MHz
'1'	'1'	'1'	20	36.864MHz

IV.7 - RAM

The RAM is a 494 bytes memory. It is used to store the CIS and to exchange data between the PCMCIA bus and the MCU bus. The CIS takes only a part of the memory, so that it is possible to keep it visible in the memory while performing data transfer through the RAM. Nevertheless, if a large amount of memory is need for these data transfers, and if the CIS is no longer necessary after it has been read, it is possible to use the CIS area in the RAM to exchange the data. The correspondence between addresses seen by the PC, those seen by the MCU and those seen by the I^2C interface is detailed in a separate paragraph later.

IV.8 - I/O - MCU - MEMORY EXTENSION MULTI-PURPOSE PORT

IV.8.1 - Introduction

The same pins are used to connect either a peripheral, a memory extension or a MCU. This corresponds to different behaviours for the chip. Choosing one configuration must be done depending upon how the chip will be used. It is obviously not possible to switch in real time from one behaviour to another since it would mean that the chip environment changes at the same time. The choice is done after the reset owing to R0 register's content.
The multi-purpose port has 8 pins for data, 9 for the addresses and a few other pins for control signals.

IV - FUNCTIONNAL DESCRIPTION (continued)

IV.8.2 - I/O port

The I/O port is an 8 bits parallel port that the PC may access. It uses the pins MCU_DAT[7..0] to transfer data. The nine address pins are useless and they must be tied to either '0' or '1' to avoid power consumption. The chip generates a chip-select on CSL_ALE and reproduces on IOW and IOR outputs the PC_IOWR and PC_IORD signals coming from the PC when the I/O port address is decoded. The I/O port address is programmable owing to registers ADRDEC and MASK.

IV.8.3 - MCU port

The MCU port allows a parallel 8 bits connection between a MCU and the PC through the ST7548. The MCU may read and write in the RAM and in some registers. The MCU port either works with separate data and address buses, or with a multiplexed address/data bus. The ST7548 automatically detects if the bus is multiplexed. In the multiplexed mode, since the address on the MCU side is on 9 bits, the 8 LSBs of the address are multiplexed with the data and the MSB is carried on the address bus at its normal location.

In addition to the fact of working with a multiplexed or non multiplexed MCU bus, for maximum flexibility, the ST7548 also accepts two styles of control signals for the MCU. One is called "Motorola-like" and the other "Intel-like". Motorola-like style is chosen by wiring the MCU_MOTEL pin to '1', Intel-like style by wiring it to '0'.

- The "Motorola-like" style has the following characteristics :
  - MCU_DSRD pin is interpreted as a data strobe active low.
  - MCU_RWWR pin is interpreted as a level active signal indicating "read" when'1' and "write" when '0'.
  - MCU_ACKWAIT is interpreted as a "dtack" signal which goes low to indicate that the MCU can terminate its memory cycle. The ST7548 always issues a DTACK on MCU_ACKWAIT since it is mandatory for the MCU to achieve its memory cycle.
- The "Intel-like" style has the following characteristics :
  - MCU_DSRD pin is interpreted as a read signal active low.
  - MCU_RWWR pin is interpreted as a write signal active low.
  - MCU_ACKWAIT is interpreted as a "wait" signal which is maintained low by the ST7548 as long as it wants the MCU to wait during its memory cycle. If it doesn't need any cycle extension, the ST7548 will not issue a wait on MCU_ACKWAIT.

IV.8.4 - Memory Extension

Used in the memory extension mode, the ST7548 just extends the 8 bits data bus PC_DAT, and reproduces the read/write signals on its outputs. When the circuit is placed in memory extension mode, the following things occur :

- PC_CE1 is reproduced on pin CSL_ALE (only for common memory accesses whith PC_REG = '1').
- PC_WE is reproduced on pin IOW if there is a common memory write on the chip.
- PC_OE is reproduced on pin IOR if there is a common memory read on the chip.
- PC_DAT* and MCU_DAT* are connected together according to IOW and IOR signals.

The addresses are not transmitted by the circuit : they have to be handled outside the chip.

IV - FUNCTIONNAL DESCRIPTION (continued)

IV.9 - ADDRESSES CORRESPONDENCE

IV.9.1 - Addresses Correspondence in Attribute Memory Mode (Figure 7)

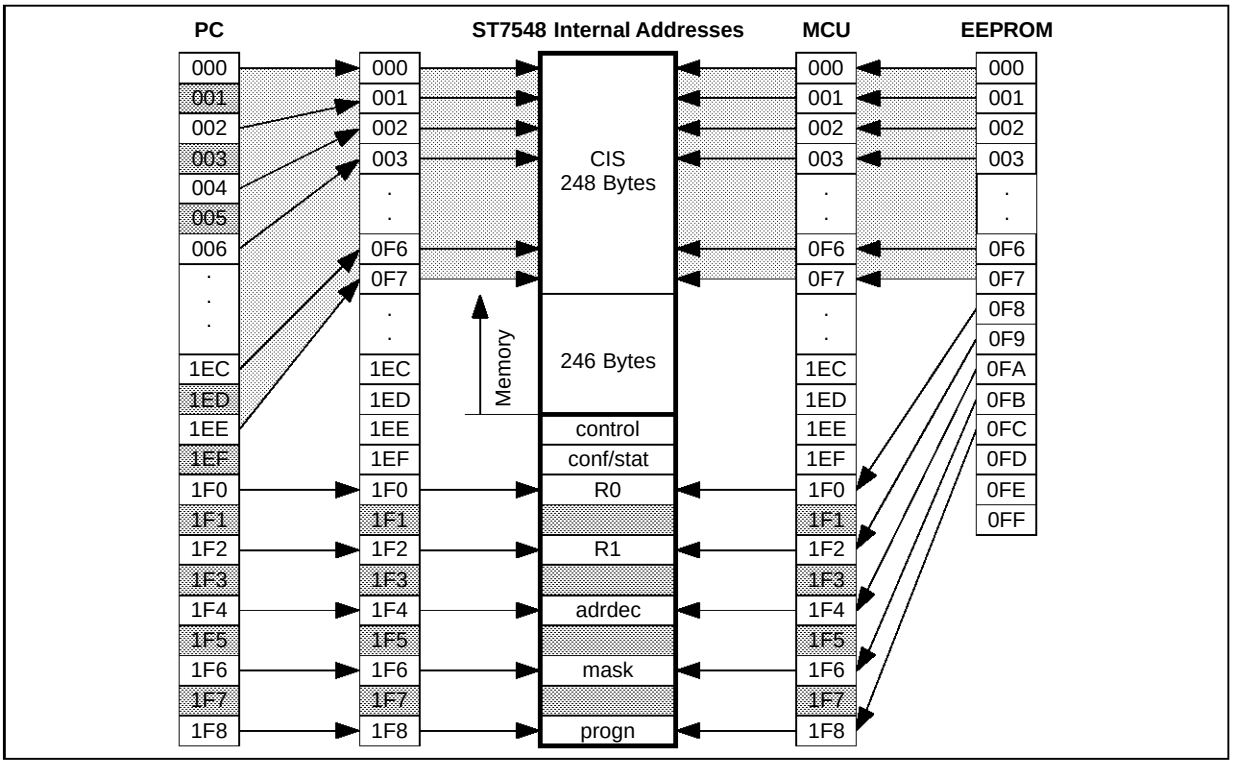
The attribute memory is accessible on the PCMCIA side when PC_CE1 = '0', PC_REG = '0', and with the read/write signals PC_OE et PC_WE. In attribute memory, only even addresses are allowed. One address out of two is thus used. In order to avoid wasting room in the chip's RAM, and to allow the user to keep the CIS in memory while having enough room to perform exchanges between the PCMCIA bus and the MCU, the CIS is "compressed" at the beginning of the RAM. To achieve this, attribute memory addresses seen on the PCMCIA bus are divided by two (shifted one bit right) internally when they are comprised between x"000" and x"1EE". Higher addresses (which correspond to registers) are transmitted without any modification.

When the CIS is loaded in memory by the MCU, since the MCU has total freedom for its addressing, it uses the addresses directly seen by the RAM, between x"00" and x"F7".

When the CIS and the registers are loaded from the EEPROM, it is the opposite operation of that which is performed for the PCMCIA bus. The addresses from x"00" to x"F7" of the EEPROM are copied in the RAM memory from x"00" to x"F7". Then, the following addresses correspond to registers and they are shifted one bit left so that they fall in the registers.

When the chip is used on a true PCMCIA board, the user should put x"00" in the EEPROM at addresses x"F8" and x"F9", because they correspond to registers R0 and R1 which are usually loaded by the PCMCIA bus after the CIS has been loaded. When the chip is used directly with a PC, the user can put directly at these locations, the information he wants to have in R0 and R1.

Figure 7



IV - FUNCTIONNAL DESCRIPTION (continued)

IV.9.2 - Addresses Correspondence in Common Memory (Figure 8)

Common memory is accessible on the PCMCIA side when PC_CE1 = '0', PC_REG = '1', and with the read/write signals PC_OE and PC_WE. In common memory, on the PCMCIA side, even and odd addresses are allowed, and the addresses inside the chip are the same as those transmitted on PC_ADD and MCU_ADD address buses.

Possible addresses on PC_ADD range from x"000" to x"1EF". They correspond to the RAM and to the first two registers: CONTROLPC and CONF/STATUS.

The circuit RAM memory extends from address x"000" to address x"1ED" : it overlaps the CIS area. It is up to the user to destroy or not to destroy the CIS when he uses the memory.

IV.9.3 - Address Correspondence in I/O Memory Mode (Figure 9)

I/O accesses on the PCMCIA side are characterized by PC_CE1 = '0', PC_REG = '0' and the use of PC_IORD and PC_IOWR as read/write signals. I/O addresses start at x"100" and extend to x"1EF". They correspond to the second half of the memory and to the first two registers: CONTROLPC and CONF/STATUS which are preferably used to synchronize exchanges between PC and MCU. The internal addresses exactly correspond to those that are sent by the PC and the MCU.

When the ST7548 RAM is used in I/O memory mode, the circuit filters addresses on the PC side which are outside the range x"100" to x"1EF", except those which are for the UART. This filter is deactivated when the I/O port is used because in this case, the internal memory is not used.

On the MCU side, no filtering is done on accesses : it is up to the user to care.

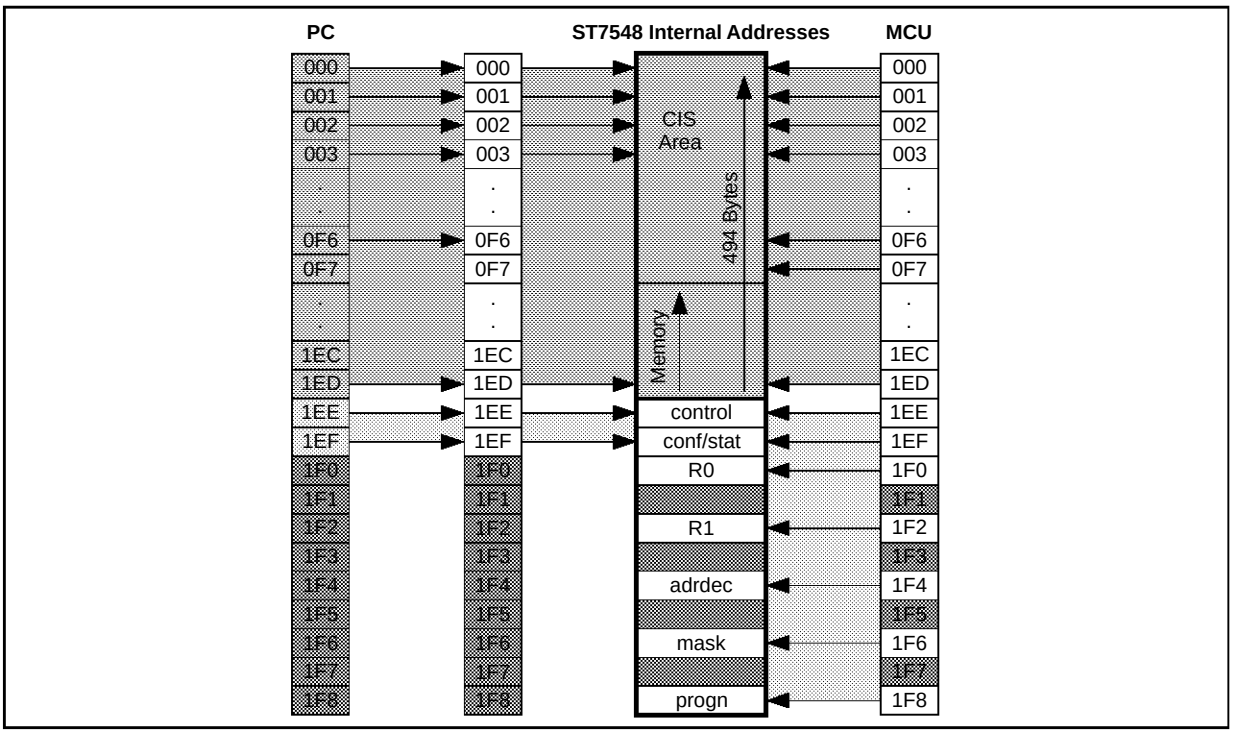
IV.9.4 - How to Dump a CIS in the EEPROM (Figure 10)

The circuit has been provided with a facility to program the EEPROM without the use of an external programming tool : it is possible to write a CIS in common memory mode into the RAM and then to transfer it in the EEPROM automatically. The transfer is initiated just by writing a '1' on bit 7 of CONTROLPC register.

The ST7548 writes the first 256 addresses of its internal RAM in the corresponding addresses in the EEPROM. So, the CIS to dump must have previously been placed in common memory at addresses x"000" to x"0FF".

During the dump in the EEPROM, the signal PC_RDY shows "busy" ('0').

Figure 8



IV - FUNCTIONNAL DESCRIPTION (continued)

Figure 9

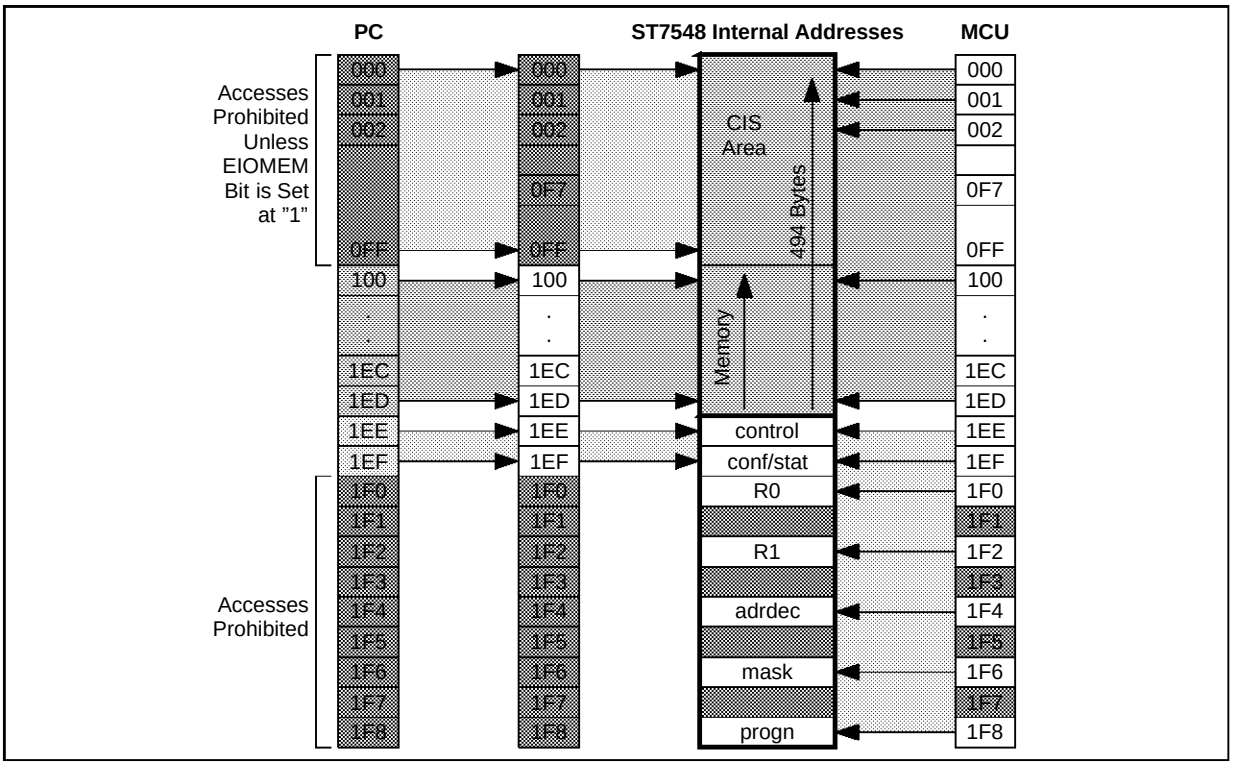
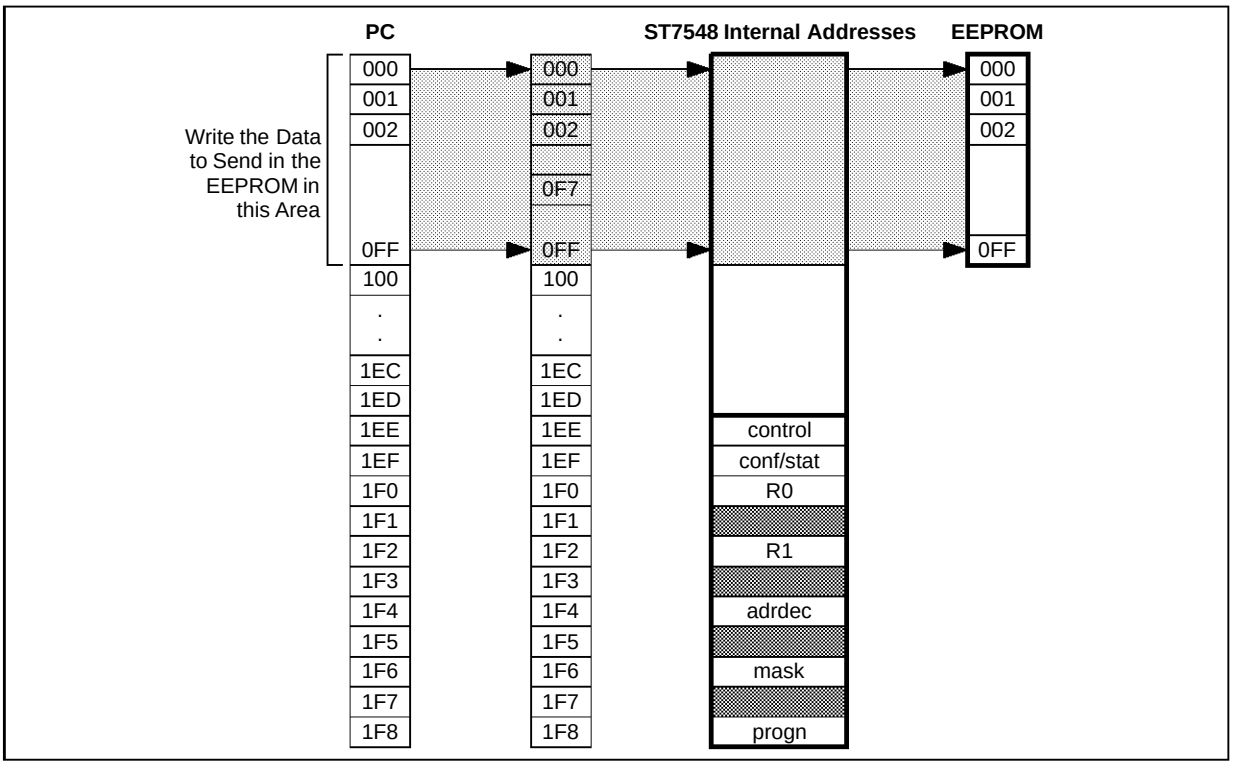


Figure 10



IV - FUNCTIONNAL DESCRIPTION (continued)

IV.10 - INTERRUPTS

IV.10.1 - Interrupts Towards the PC

Interrupts Level

Interrupts towards the PC are active when there is a '0' on the PC_RDY port. INTL bit in register R0 makes it possible to choose between an interrupt active on a level (as long as the interrupt is present, PC_RDY remains at '0') and a pulse interrupt (when the interrupt happens, PC_RDY goes low for at least 5 micro-seconds, and then goes back to '1', independently of the fact that the interrupt has been acquitted or not).

INTL = '0' → pulse
INTL = '1' → level

The PC can only receive interrupts in the modes where I/O accesses are enabled.

Interrupts Origin

Interrupts towards the PC have three possible origins :

- The UART : the UART emits an interrupt when its INTR output is '1'. In order to have this interrupt transmitted to the PC, it is necessary to have the UART validated (bit UE in R0 register at '1') and that the UART output OUT2 be at '0'. Refer to data sheets of 16C550 UARTS to have a detailed description on UART interrupts.
- The I/O port : in this case, interrupts come from a peripheral device. The peripheral sends an interrupt by putting a '0' on MCU_IRQ port. This interrupt is transmitted to the PC when the I/O port is enabled (bit MODE1 at '1' and bit MODE0 at '0' in R0 register). Conditions upon which interrupts appear or disappear on this pin depend upon what is connected on the port.
- The MCU : the ST7548 RAM must be placed in the I/O field of the PC in order to have interrupts enabled (bits MODE1 and MODE0 in register R0 at '1'). In this mode, the RAM is used to exchange data between the PC and a MCU, and interrupts may be used to manage this transfer. The MCU sends an interrupt to the PC by writing a '1' on bit 7 in CONFMCU register. This interrupt reaches the PC if the PC has not masked it (the interrupt is masked with a '1' on bit 0 in CONTROLPC register). To acknowledge this interrupt and make it disappear, the PC must write a '1' on bit 6 in its CONFPC register.

IV.10.2 - Interrupts Towards the MCU

Only the PC may send interrupts to the MCU. The PC sends an interrupt to the MCU by writing a '1' on bit 7 in CONFPC register. The MCU may mask this interrupt by writing a '1' on bit 0 in its CONTROLMCU register. To acknowledge the interrupt and make it disappear, the MCU must write a '1' on bit 6 in its CONFMCU register.

Interrupts operation is exactly symmetrical for the PC and the MCU.

IV - FUNCTIONNAL DESCRIPTION (continued)

IV.11 - ADDRESSING CAPABILITIES TABLE

	Hexa-decimal Address	PCMCIA Addressing Mode				MCU		I^2C	R0 Content		
		common memory pc_ce1=0 pc_reg=1 pc_we pc_oe	attribute memory pc_ce1=0 pc_reg=0 pc_we pc_oe	I/O memory pc_ce1=0 pc_reg=0 pc_iord pc_iowr	Wait	Access	Wait		UE	MODE 1	MODE 0
Memory	0→1ED	yes	no	no	yes	yes	yes	yes	'X'	'0'	'0'
	0→1EE	no	yes						'X'	'X'	'X'
	100(1)→1ED		no	yes					'X'	'1'	'1'
control PC	1EE	yes	no	no	no	no		no	'X'	'0'	'0'
		no		yes					'X'	'1'	'1'
control MCU				no		yes	no		'X'	'X'	'X'
confPC = statMcu	1EF	write				read	yes		'X'	'0'	'0'
		no		write					'X'	'1'	'1'
confMcu = statPC		read		no		write			'X'	'0'	'0'
		no		read	yes				'X'	'1'	'1'
R0	1F0		yes	no		yes		yes	'X'	'X'	'X'
R1	1F2										
Adrdec	1F4										
Mask	1F6										
Progn	1F8										
UART	2E8 2F8 3E8 3F8	no	no	yes	no	no		no	'1'	'X'	'X'
I/O port	000→3FF	no	no	yes	no	no		no	'X'	'1'	'0'
Mem ext	XXX	yes	no	no	no	no		no	'X'	'0'	'1'

Note : 1. The lowest address in I/O memory mode can be reduced to 000 if EIOMEM bit is set to "1" in CONTROLPC.

This table describes the access capabilities for all the circuit, i.e. for the memory, registers, UART, I/O port, and memory extension port. The CONTROLPC and CONTROLMCU registers are seen at the same address on the PC side and on the MCU side, but in fact they are two different registers, and the PC, for example only sees CONTROLPC. The circuit distinguishes them from where the access comes from. It is the same thing for CONFMCU and CONFPC. In addition, the configuration/status registers are actually "configuration" when they are written and "status" when they are read. Their access permission when they are status are written in bold italic. Their access permission when they are configuration are written in regular characters.

Remark about wait : when the MCU is a Motorola or equivalent, a wait (or more precisely a DTACK) is systematically generated, even for an access to the CONTROLMCU register. This is mandatory to allow the MCU to terminate its memory cycle. In this case, the wait duration is as short as possible.

V - ABSOLUTE MAXIMUM RATINGS AND OPERATING VALUES

Symbol	Parameter	Value	Unit
V_{DD}	DC Supply Voltage	-0.5, 6.0	V
V_I	Digital Input Voltage	-0.5, V_{DD} + 0.5	V
I_O	Digital Output Current	4	mA
T_{oper}	Operating Temperature	0, 70	°C
T_{stg}	Storage Temperature	-40, +125	°C
PDmax	Maximum Power Dissipation	100	mW
T_L	Lead Temperature (Soldering, 10s)	+300	°C

VI - DIGITAL INTERFACE (V_{DD} = 5V, T_{amb} = 25°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{IL}	Input Low Voltage	- All digital inputs except PC_RESET - XTIN Pin used as an input			0.8 1.5	V V
V_{IH}	Input High Voltage	- All digital inputs except PC_RESET - XTIN Pin used as an input	2.4 3.5			V V
V_{OL}	Output Low Voltage	I_L = 4mA			0.5	V
V_{OH}	Output High Voltage	I_L = 4mA	2.8			V
I_{IL}	Input Low Current	Any digital input, GND < V_{IN} < V_{IL}	-10		10	µA
I_{IH}	Input High Current	Any digital input, V_{IH} < V_{IN} < V_{DD}	-10		10	µA
I_{OZ}	Output Current in High Impedance	All digital tri-state I/Os without internal pull-up or pull-down resistor	-10		10	µA
V_{HYST}	Trigger Input Hysteresis	PC_RESET only	0.6		1.1	V
VT+	Positive Trigger Voltage	PC_RESET only			2.4	V
VT-	Negative Trigger Voltage	PC_RESET only	0.6			V
I_{LPC}	Low Leak Pull-up Current	V_I = 0V	-250		-100	µA
I_{HPC}	High Leak Pull-up Current	V_I = V_{DD}	-10		+10	µA
I_{LC}	Low Input Leakage Current	XTIN Pin, V_I = 0V	-25		-9	µA
I_{HC}	High Input Leakage Current	XTIN Pin, V_I = V_{DD}	9		25	µA

Never let any unused input not connected. This could cause excessive power dissipation and may damage the component. Unused inputs must be connected either to '0' or '1', unless otherwise specified in the documentation.

VII - CLOCK FREQUENCY (V_{DD} = 5V, T_{amb} = 25°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
CLKIN	CLKIN Max Frequency				36.864	MHz
XTIN	XTIN Max Frequency				36.864	MHz

Note : t_{XTIN} = 1/(XTIN frequency)

VIII - POWER CONSUMPTION AND SUPPLY VOLTAGE (T_{amb} = 25°C unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_{DD}	DC Supply Voltage		4.5	5	5.5	V
I_{CC}	DC Current			20		mA

IX - TIMING CHARACTERISTICS

IX.1 - PCMCIA Bus Timing (Common Memory Cycle)

Read Timing *(see Figure 11)*

Symbol	IEEE	Parameter	Min.	Typ.	Max.	Unit
tdis(OE)	t_{GHQZ}	Output Disable Time from OE			10	ns
tsu(A)	t_{AVGL}	Address Set-up Time	5			ns
th(A)	t_{GHAX}	Address Hold Time	5			ns
tsu(CE)	t_{ELGL}	Card Enable Set-up Time	5			ns
th(CE)	t_{GHEH}	Card Enable Hold Time	5			ns
tv(WT-OE)	t_{GLWTV}	Wait Valid from OE			25	ns
tw(WT) (3)	t_{WTLWTH}	Wait Pulse Width			12	µs
tv(WT)	t_{QVWTH}	Data Set-up for Wait Released	0			ns
ta(OE)		Data Access Time without Wait			25	ns

Write Timing *(see Figure 12)*

Symbol	IEEE	Parameter	Min.	Typ.	Max.	Unit
tsu(A)	t_{AVWL}	Address Set-up Time	5			ns
tsu(D-WEH) (1)	t_{DVWH}	Data Set-up Time for WE	10			ns
th(D)	t_{WMDX}	Data Hold Time	5			ns
trec(WE)	t_{WMAX}	Write Recover Time	5			ns
tsu(CE)	t_{ELWL}	Card Enable Set-up Time	5			ns
th(CE)	t_{GHEH}	Card Enable Hold Time	5			ns
tv(WT-WE)	t_{WLWTV}	Wait Valid from WE			25	ns
tw(WT) (3)	t_{WTLWTH}	Wait Pulse Width			12	µs
tv(WT)	t_{WTHWH}	WE High from Wait Released	0			ns
tv(D-WE) (2)		Data Valid from WE			2 t_{XTIN} -10	ns
tw(WE) (1)		Write Pulse Width	50			ns

Notes : 1. Cycles without wait only.
2. Cycles with wait only.
3. Actual wait width depends upon the operating frequency of the chip, but will be much less than the indicated value for XTIN frequencies between 10MHz and 36.864MHz.

IX - TIMING CHARACTERISTICS (continued)
IX.2 - PCMCIA Bus Timing (Attribute Memory Cycle)

Read Timing *(see Figure 11)*

Symbol	IEEE	Parameter	Min.	Typ.	Max.	Unit
tdis(OE)	t_{GHQZ}	Output Disable Time from OE			10	ns
tsu(A)	t_{AVGL}	Address Set-up Time	5			ns
th(A)	t_{GHAX}	Address Hold Time	5			ns
tsu(CE)	t_{ELGL}	Card Enable Set-up Time	5			ns
th(CE)	t_{GHEH}	Card Enable Hold Time	5			ns
tv(WT-OE)	t_{GLWTV}	Wait Valid from OE			25	ns
tw(WT) (3)	t_{WTLWTH}	Wait Pulse Width			12	µs
tv(WT)	t_{QVWTH}	Data Set-up for Wait Released	0			ns
ta(OE)		Data Access Time without Wait			25	ns

Write Timing *(see Figure 12)*

Symbol	IEEE	Parameter	Min.	Typ.	Max.	Unit
tsu(A)	t_{AVWL}	Address Set-up Time	5			ns
tsu(D-WEH) (1)	t_{DVWH}	Data Set-up Time for WE	10			ns
th(D)	t_{WMDX}	Data Hold Time	5			ns
trec(WE)	t_{WMAX}	Write Recover Time	5			ns
tsu(CE)	t_{ELWL}	Card Enable Set-up Time	5			ns
th(CE)	t_{GHEH}	Card Enable Hold Time	5			ns
tv(WT-WE)	t_{WLWTV}	Wait Valid from WE			25	ns
tw(WT) (3)	t_{WTLWTH}	Wait Pulse Width			12	µs
tv(WT)	t_{WTHWH}	WE High from Wait Released	0		$2\ t_{XTIN}$ -10	ns
tv(D-WE) (2)		Data Valid from WE			40	ns
tw(WE) (1)		Write Pulse Width	50			ns

Notes : 1. Cycles without wait only.
2. Cycles with wait only.
3. Actual wait width depends upon the operating frequency of the chip, but will be much less than the indicated value for XTIN frequencies between 10MHz and 36.864MHz.

IX - TIMING CHARACTERISTICS (continued)

Figure 11 : Read Timing Diagram

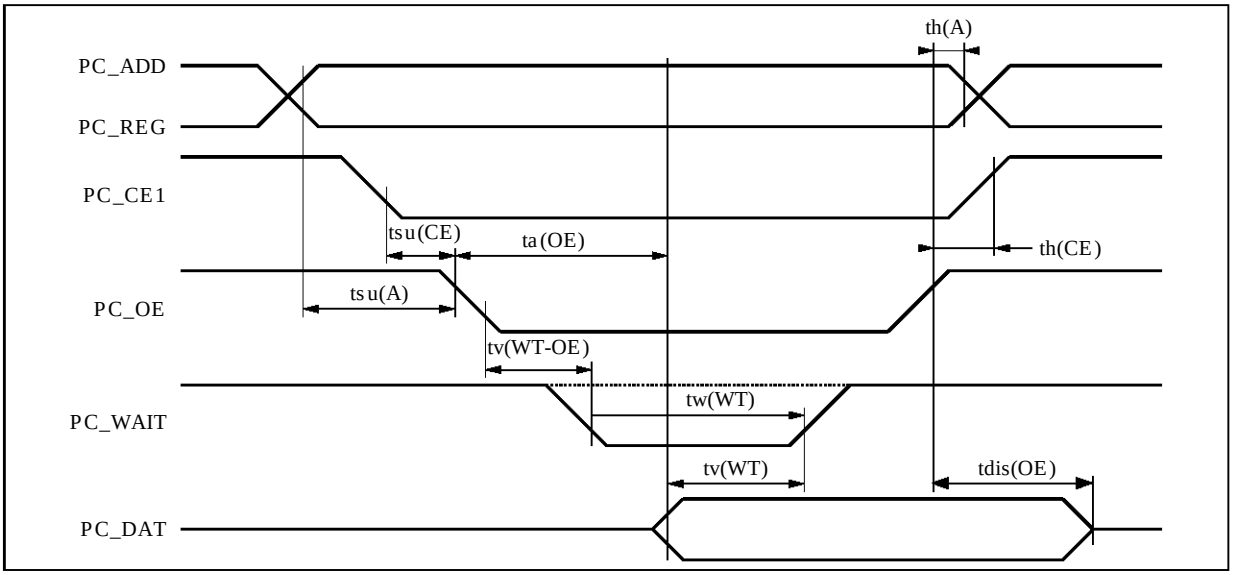
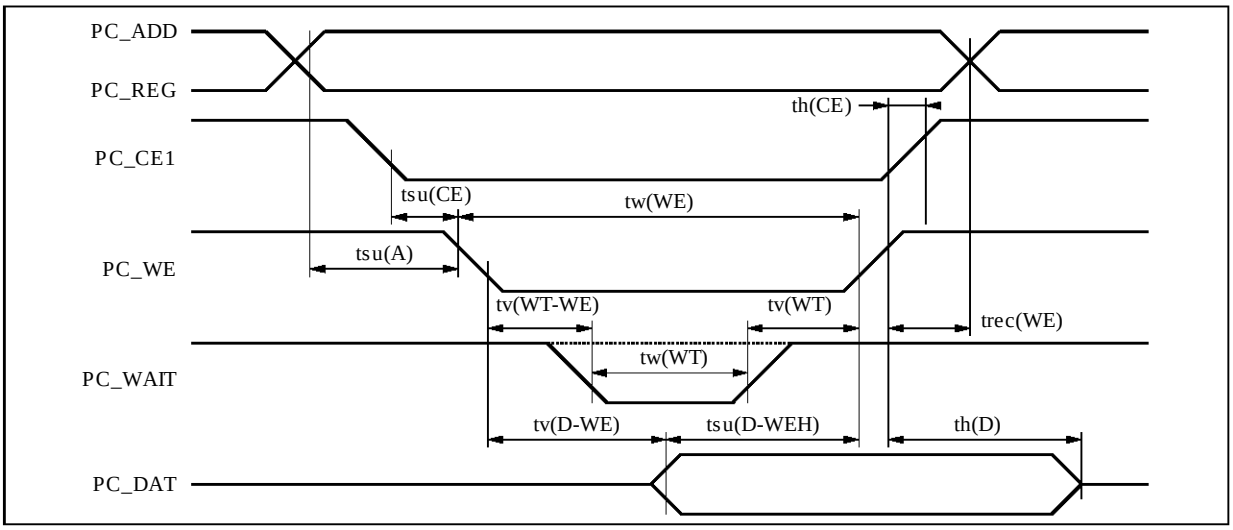
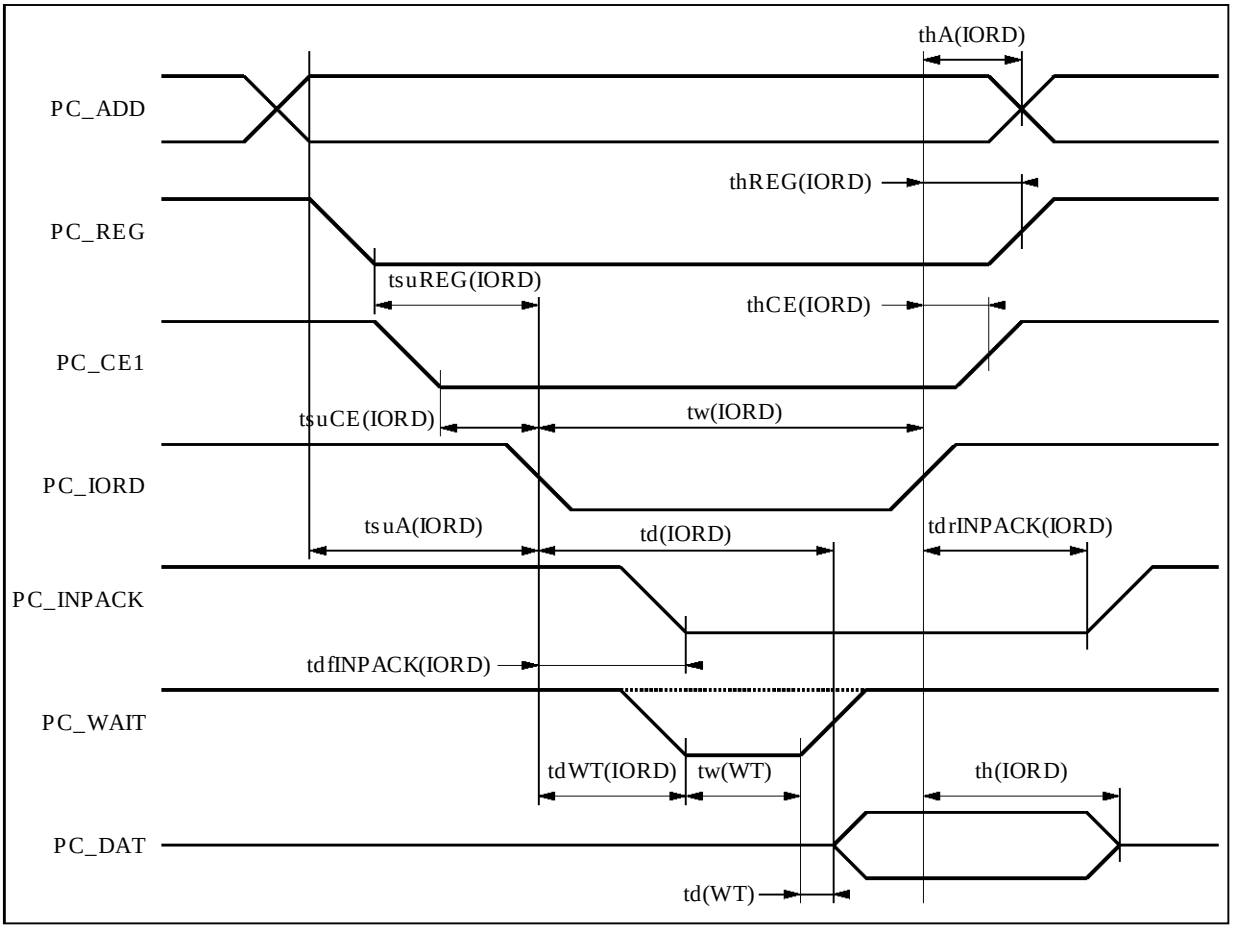


Figure 12 : Write Timing Diagram



IX - TIMING CHARACTERISTICS (continued)
IX.3 - PCMCIA Bus Timing (I/O Cycle)

Figure 15 : Read Timing Diagram

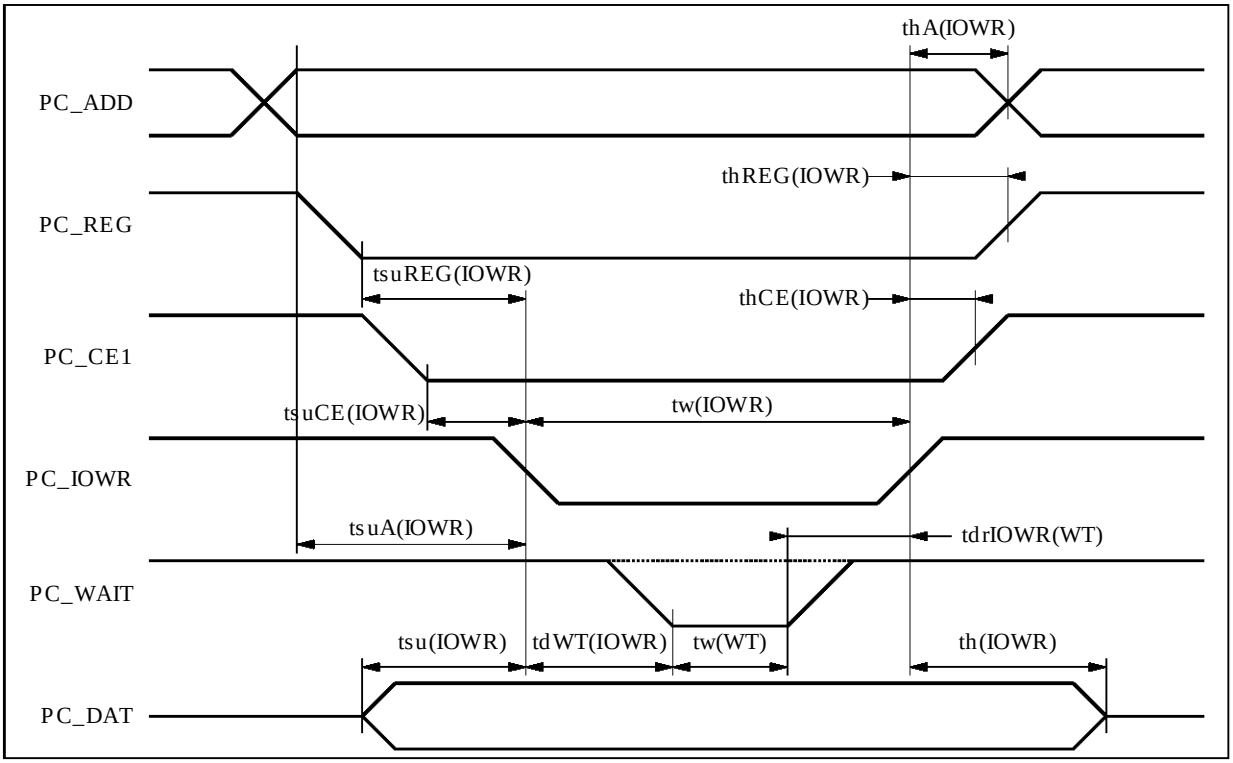


Symbol	IEEE	Parameter	Min.	Typ.	Max.	Unit
th(IORD)	t_{IGHQX}	Data Hold following IORD	0			ns
twIORD	t_{IGLIGH}	IORD Width Time	50			ns
tsuA(IORD)	t_{AVIGL}	Address Set-up before IORD	5			ns
tha(IORD)	t_{IGHAX}	Address Hold following IORD	5			ns
tsuCE(IORD)	t_{ELIGL}	CE Set-up before IORD	5			ns
thCE(IORD)	t_{IGHEH}	CE Hold following IORD	5			ns
tsuREG(IORD)	t_{RGLIGL}	REG Set-up before IORD	5			ns
thREG(IORD)	t_{IGHRGH}	REG Hold following IORD	5			ns
tdfINPACK(IORD)	t_{IGLIAL}	INPACK Delay Falling from IORD	0		15	ns
tdrINPACK(IORD)	t_{IGHIAH}	INPACK Delay Rising from IORD			15	ns
tdWT(IORD)	t_{IGLWTL}	WAIT Delay Falling from IORD			25	ns
td(WT)	t_{WTHQV}	Data Delay from Wait Rising			0	ns
tw(WT) (1)	t_{WTLWTH}	WAIT Width Time			12	µs
td(IORD)		IORD to Data (cycles without wait only)			25	ns

Note : 1. Actual wait width depends upon the operating frequency of the chip, but will be much less than the indicated value for XTIN frequencies between 10MHz and 36.864MHz.

IX - TIMING CHARACTERISTICS (continued)
IX.3 - PCMCIA Bus Timing (I/O Cycle) (contined)

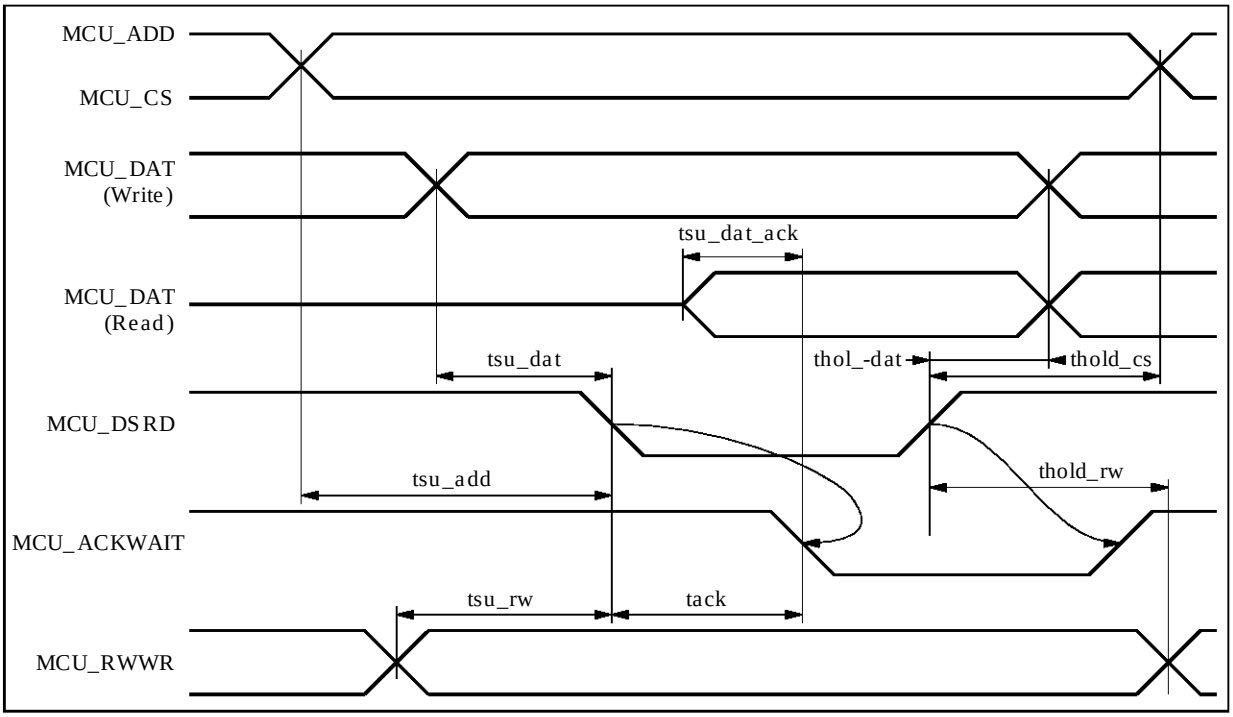
Figure 16 : Write Timing Diagram



Symbol	IEEE	Parameter	Min.	Typ.	Max.	Unit
tsu(IOWR)	t_{DVIWL}	Data Set-up before IOWR	60			ns
th(IOWR)	t_{IWHDX}	Data Hold following IOWR	5			ns
twIOWR	t_{IWLIWH}	IOWR Width Time	50			ns
tsuA(IOWR)	t_{AVIWL}	Address Set-up before IOWR	5			ns
thA(IOWR)	t_{IWHAX}	Address Hold following IOWR	5			ns
tsuCE(IOWR)	t_{ELIWL}	CE Set-up before IOWR	5			ns
thCE(IOWR)	t_{IWHEH}	CE Hold following IOWR	5			ns
tsuREG(IOWR)	t_{RGLIWL}	REG Set-up before IOWR	5			ns
thREG(IOWR)	t_{IWHRGH}	REG Hold following IOWR	5			ns
tdWT(IOWR)	t_{IWLWTL}	WAIT Delay Falling from IOWR			25	ns
tw(WT)	t_{WTLWTH}	WAIT Width Time			6 t_{XTIN}	ns
tdrIOWR(WT)	t_{WTHIWL}	IOWR High from WAIT High	0			ns

IX - TIMING CHARACTERISTICS (continued)
IX.4 - MCU Bus Timing
IX.4.1 - Motorola, Not Multiplexed

Figure 17 : MCU_MOTEL is Tied to '1'

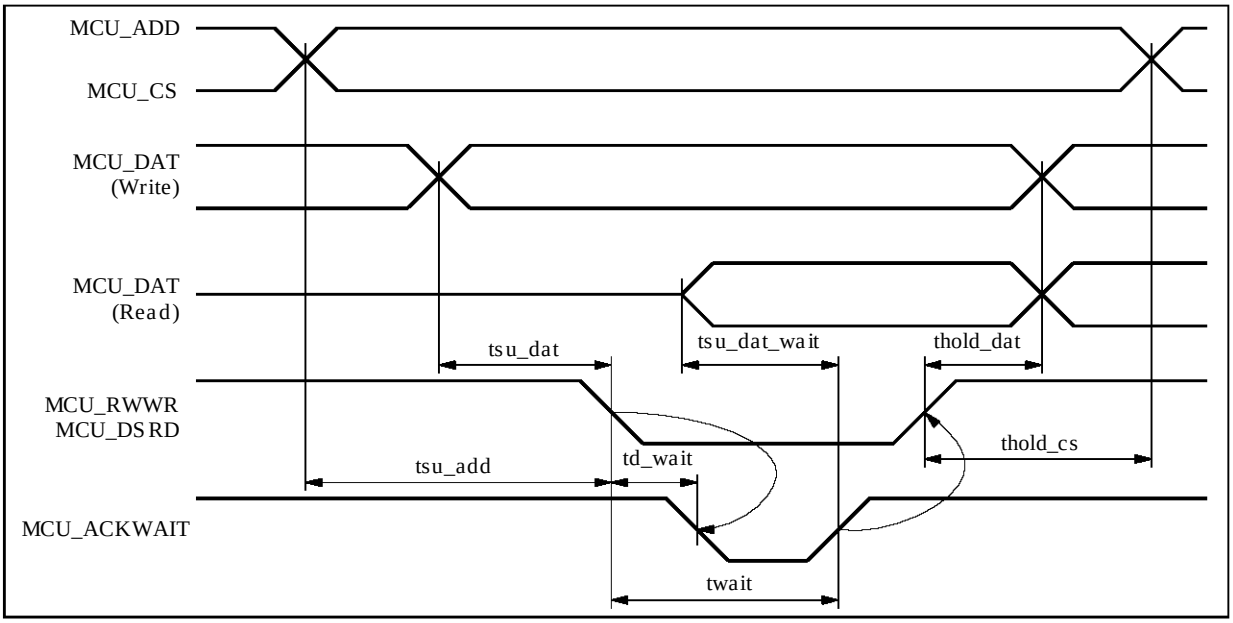


The first edge on MCU_ACKWAIT indicates to the MCU that it can terminate its memory cycle.

Symbol	Parameter	Min.	Typ.	Max.	Unit
tsu_dat_ack	Data Read Set-up Time	20			ns
tsu_dat	Data Write Set-up Time	0			ns
thold_dat	Data Hold Time	5			ns
tsu_add	Address Set-up Time	5			ns
tsu_rw	Read/Write Set-up Time	5			ns
tack	Ack Delay Time			6 t_{XTIN}	ns
thold_cs	Chip Select Hold Time from Data Strobe	5			ns
thold_rw	Read/Write Hold Time from Data Strobe	5			ns

IX - TIMING CHARACTERISTICS (continued)
IX.4 - MCU Bus Timing (continued)
IX.4.2 - Intel, Not Multiplexed

Figure 18 : MCU_MOTEL is Tied to '0'

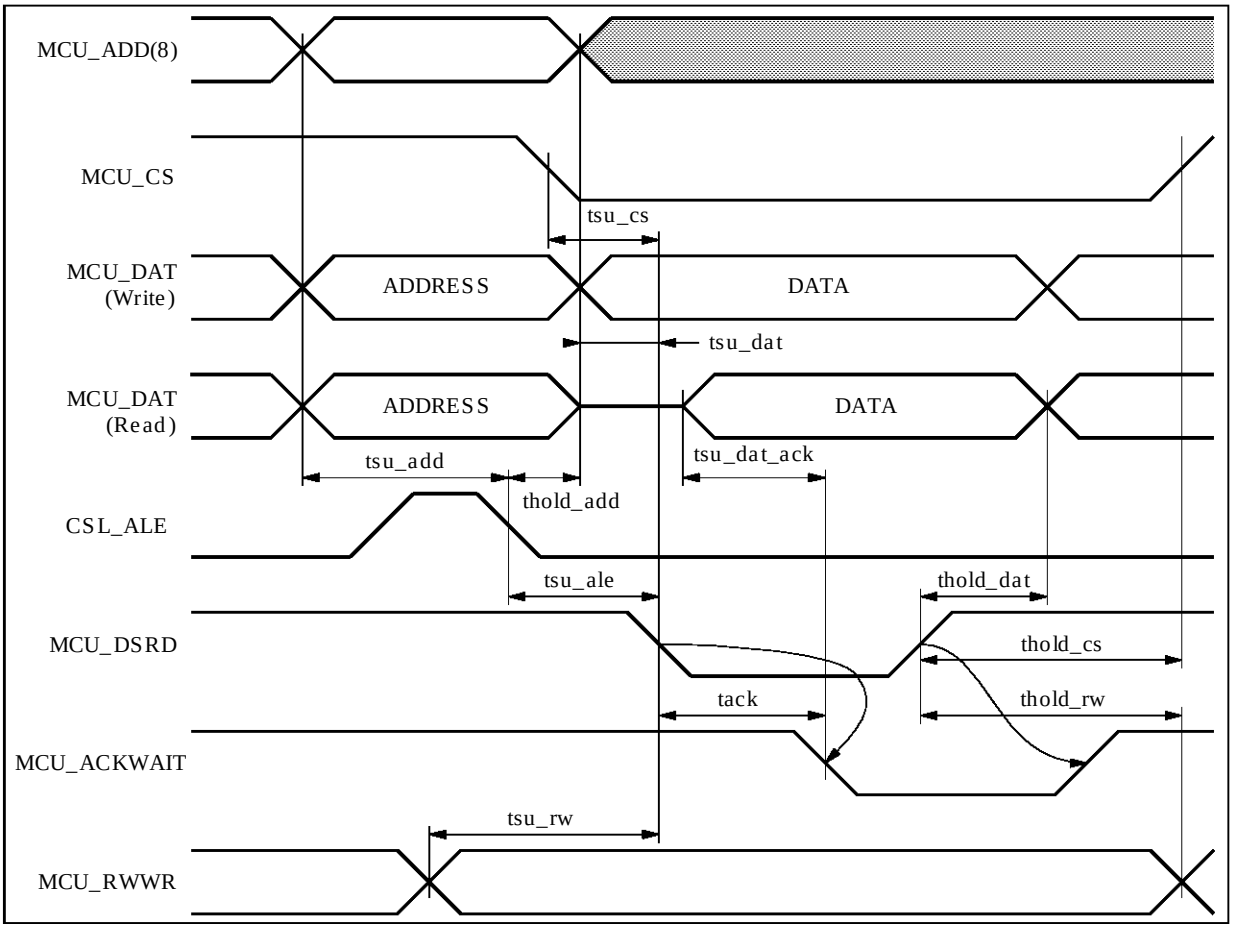


The second edge on MCU_ACKWAIT indicates to the MCU that it may terminate its memory cycle. If the access doesn't require to extend the memory cycle, MCU_ACKWAIT remains at '1'.

Symbol	Parameter	Min.	Typ.	Max.	Unit
tsu_dat	Data Write Set-up Time	0			ns
thold_dat	Data Hold Time	5			ns
tsu_dat_wait	Data Read Set-up Time	20			ns
tsu_add	Address Set-up Time	5			ns
td_wait	Wait Delay Time			25	ns
twait	Wait Width Time			6 t_{XTIN}	ns
thold_cs	Chip Select Hold Time from Data Strobe	5			ns

IX - TIMING CHARACTERISTICS (continued)
IX.4 - MCU Bus Timing (continued)
IX.4.3 - Motorola, Multiplexed

Figure 19 : MCU_MOTEL is Tied to '1'

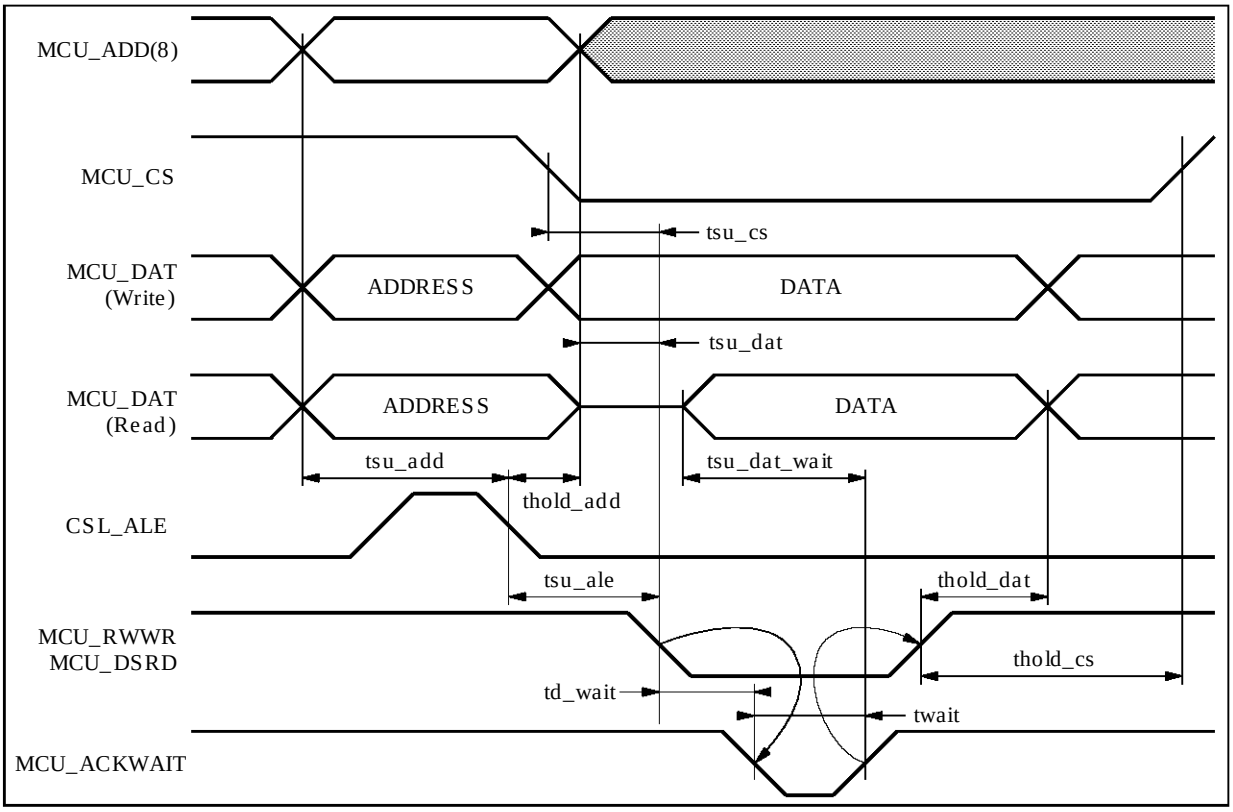


The first edge on MCU_ACKWAIT indicates to the MCU that it can terminate its memory cycle.

Symbol	Parameter	Min.	Typ.	Max.	Unit
tsu_cs	Chip Select Set-up Time	5			ns
tsu_dat	Data Write Set-up Time	0			ns
tsu_add	Address Set-up Time	20			ns
thold_add	Address Hold Time	10			ns
tsu_dat_ack	Data Read Set-up Time	20			ns
tsu_ale	Address Latch Enable Set-up Time	20			ns
thold_dat	Data Hold Time	5			ns
tsu_rw	Read/write Set-up Time	5			ns
tack	Ack Delay Time			6 t_{XTIN}	ns
thold_cs	Chip Select Hold Time from Data Strobe	5			ns
thold_rw	Read/Write Hold Time from Data Strobe	5			ns

IX - TIMING CHARACTERISTICS (continued)
IX.4 - MCU Bus Timing (continued)
IX.4.4 - Intel, Multiplexed

Figure 20 : MCU_MOTEL is Tied to '0'



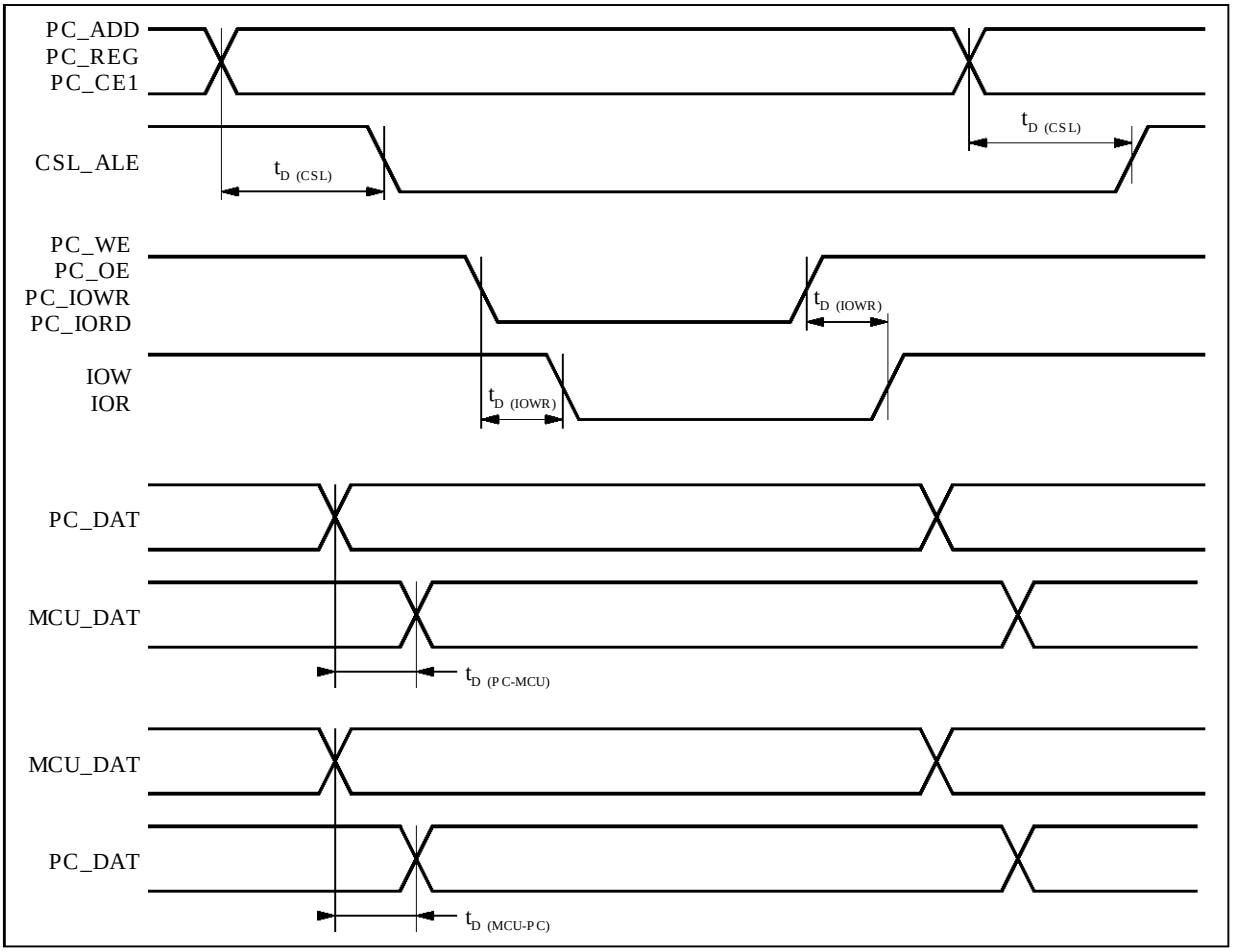
The second edge on MCU_ACKWAIT indicates to the MCU that it may terminate its memory cycle. If the access doesn't require to extend the memory cycle, MCU_ACKWAIT remains at '1'.

Symbol	Parameter	Min.	Typ.	Max.	Unit
tsu_cs	Chip Select Set-up Time	5			ns
tsu_dat	Data Write Set-up Time	0			ns
tsu_add	Address Set-up Time	20			ns
thold_add	Address Hold Time	10			ns
tsu_dat_wait	Data Read Set-up Time	20			ns
tsu_ale	Address Latch Enable Set-up Time	20			ns
thold_dat	Data Hold Time	5			ns
td_wait	Wait Delay Time			25	ns
twait	Wait Width Time			6 t_{XTIN}	ns
thold_cs	Chip Select Hold Time from Data Strobe	5			ns

IX - TIMING CHARACTERISTICS (continued)

IX.5 - Memory Extension and IO Extension Transfer Timing

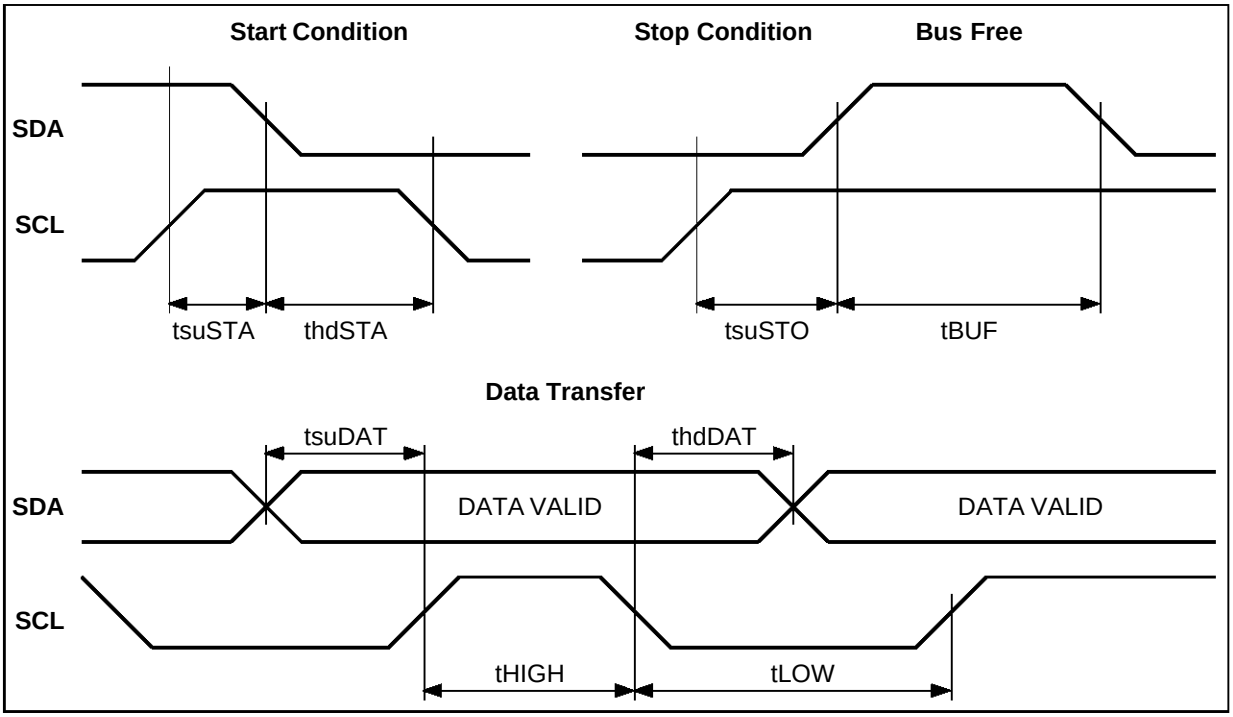
Figure 21 : Transfer Timing Diagram



Symbol	Parameter	Min.	Typ.	Max.	Unit
$t_{D\ (CSL)}$	Chip Select Transfer Delay			20	ns
$t_{D\ (IOWR)}$	Read/Write Transfer Delay			15	ns
$t_{D\ (PC-MCU)}$	Data Transfer Delay from PC to MCU			25	ns
$t_{D\ (MCU-PC)}$	Data Transfer Delay from MCU to PC			25	ns

IX - TIMING CHARACTERISTICS (continued)
IX.6 - I^2C Bus Timing

Figure 22



Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{HIGH}	Clock Pulse Width High	4			µs
t_{LOW}	Clock Pulse Width Low	4.7			µs
tsuSTA	Set-up Time for Start Condition	4.7			µs
thdSTA	Hold Time for Start Condition	4			µs
tsuSTO	Set-up Time for Stop Condition	4.7			µs
t_{BUF}	Time Bus Free before New Transmission	4.7			µs
tsuDAT	Set-up Time for Data	250			ns
thDAT	Hold Time for Data	0			µs

X - APPLICATIONS

X.1 - Introduction

Although it has been designed mainly for telecommunications, the circuit can be used in different ways in a wide range of applications.

For a better understanding of the given examples, we will assume that the following functions are available in the PC or MCU software :

```
PCWMEM(address, data)    /* PC write in common memory mode */
PCRMEM (address)         /* PC read in common memory mode */
PCWATTR (address, data)  /* PC write in attribute memory mode */
PCRATTR (address)        /* PC read in attribute memory mode */
PCWIO (address, data)    /* PC write in I/O memory mode */
PCRIO (address)          /* PC read in I/O memory mode */
MCUW (address, data)     /* MCU write */
MCUR (address)           /* MCU read */
```

X.2 - MCU Connection

X.2.1 - Wiring

The following example shows how to connect a Motorola non multiplexed bus on the chip.
Connect :

- MCU_CS on a logic function that decodes an address space dedicated to the ST7548.
- MCU_MOTEL to '1'.
- CSL_ALE to '0'.
- MCU_DAT on MCU data bus.
- MCU_ADD on MCU address bus.
- MCU_RWWR on MCU RW signal.
- MCU_DSRD on MCU data strobe.
- MCU_ACKWAIT on MCU DTACK.
- MCU_IRQ on a MCU interrupt input line.
- IOW and IOR may be left open.

X.2.2 - Polling

The PC doesn't use interrupts. Therefore it is not necessary to perform accesses in I/O mode.

```
PCWATTR (R0, x"00")              /* set chip in common memory mode */
PCWMEM (CONTROLPC, x"01")        /* PC disables MCU's interrupts */
```

The available bits in CONFPC and CONFMCU can be used as flags to manage the data transfer. On the PC side, the memory can be reached by PCWMEM and PCRMEM functions.

X.2.3 - Interrupts

The PC uses interrupts. It is mandatory to perform accesses in I/O mode.

```
PCWATTR (R0, x"4C")              /* set chip in I/O memory, interrupt = level */
PCWIO (CONTROLPC, x"00")         /* PC enables MCU's interrupts */
...
MCUW (add0, data0)               /* MCU places data in memory */
MCUW (add1, data1)
...
MCUW (CONFMCU, x"80")            /* MCU sends interrupt to PC */
...
PCRIO(add0)                      /* PC reads data from memory */
PCRIO(add1)
...
PCWIO (CONFPC, x"40")            /* PC acknowledges interrupt and makes it disappear*/
```

X - APPLICATIONS (continued)

X.3 - UART Use

The following example assumes that CLKIN is used and that a 18.432MHz frequency is available on this pin.

```
PCWATTR(R0, x"60")          /* put chip in mcu memory mode, enable UART, IT level,
                               UART address = x"3f8" */
PCWATTR(PROGN, x"0B")       /* use clkin, divide by 10 to get 1.8432MHz on the UART */
PCWIO(x"3fb", x"83")        /* set UART DLAB bit at 1,8 bits, no parity */
PCWIO(x"3f8", x"03")        /* program divisor latch to 38400 bauds */
PCWIO(x"3f9", x"00")
PCWIO(x"3fb", x"03")        /* reset UART DLAB bit at 0 to continue */
PCWIO(x"3f8", x"55")        /* send a data */
PCRIO(x"3f8")               /* read a data */
```

Note on the LSR Register use in 16C550 mode :

LSR (Line Status Register) is a byte status giving error status information as parity, framing, break (see 16C550 documentation). This byte status is not provided in accordance with the erronous data byte in the receive FIFO.

In case where the ST7548 is used on a telecommunication Modem/Fax (or an other application) where a microprocessor is used to execute protocol as V.42/MNP... This problem is not important. In fact the microprocessor is in charge to detect errors with the far end equipment and to send to the PC software application only valid data.

In Modem/Fax application the distance between the microprocessor UART and the ST7548 UART is so short that error (parity, framing ...) will never occur.

LSR use is only a problem when the user wants to use UART to design ASYNCHRONOUS serial link (as COM1 on a mother board) due to the cable distance between the ST7548 UART and the far end serial asynchronous part.

X.4 - I/O Extension

X.4.1 - Wiring

The following example shows how to connect a peripheral on the chip. Connect :

- MCU_CS to '1' (it is not used).
- MCU_MOTEL to '1' (it is not used).
- CSL_ALE to the chip select input of your peripheral.
- MCU_DAT on the data line of your peripheral.
- MCU_ADD to '1' (it is not used).
- MCU_RWWR to '1' (it is not used).
- MCU_DSRD to '1' (it is not used).
- MCU_ACKWAIT may be left open.
- MCU_IRQ on a peripheral interrupt output line.
- IOW on the peripheral Write Enable input.
- IOR on the peripheral Read Enable input.

X.4.2 - Operation

```
PCWATTR (R0, x"48")         /* set chip in I/O extension mode, level interrupts*/
PCWATTR (ADRDEC, x"80")     /* peripheral address = x"200" */
PCWATTR (MASK, x"01")       /* peripheral address width = 8 bytes */
PCWIO (x"200", MY_DATA)     /* send a data to the peripheral device */
```

X - APPLICATIONS (continued)

X.5 - Memory Extension

X.5.1 - Wiring

The following example shows how to connect a memory extension on the chip. Connect :

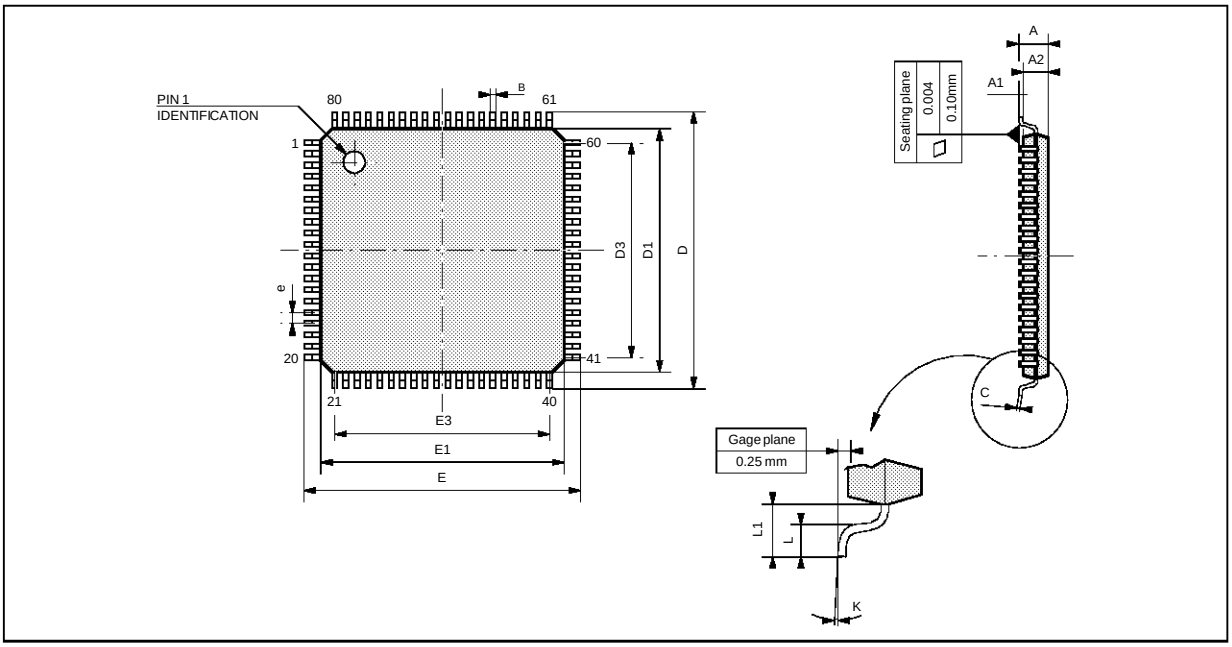
- MCU_CS to '1' (it is not used).
- MCU_MOTEL to '1' (it is not used).
- CSL_ALE to the chip select input of your memory.
- MCU_DAT on the data line of your memory.
- MCU_ADD to '1' (it is not used).
- MCU_RWWR to '1' (it is not used).
- MCU_DSRD to '1' (it is not used).
- MCU_ACKWAIT may be left open.
- MCU_IRQ to '0'.
- IOW on the memory extension Write Enable input.
- IOR on the memory extension Read Enable input.
- PC_ADD to the memory extension address bus.

X.5.2 - Operation

```
PCWATTR (R0, x"04")          /* set chip in memory extension mode */
PCWMEM (x"123", MY_DATA) /* send a data to the memory extension */
PCRMEM (x"123")              /* send a data to the memory extension */
```

XI - PACKAGE MECHANICAL DATA

80 PINS - PLASTIC PACKAGE



Dimensions	Millimeters			Inches		
	Min.	Typ.	Max.	Min.	Typ.	Max.
A			1.60			0.063
A1	0.05		0.15	0.002		0.006
A2	1.35	1.40	1.45	0.053	0.055	0.057
B	0.22	0.32	0.38	0.010	0.012	0.014
C	0.09		0.20	0.004		0.008
D		16.00			0.630	
D1		14.00			0.551	
D3		12.35			0.486	
e		0.80			0.0314	
E		16.00			0.630	
E1		14.00			0.551	
E3		12.35			0.486	
L	0.45	0.60	0.75	0.020	0.024	0.030
L1		1.00			0.039	
K	0^o (Min.), 7^o (Max.)					

TQFP80.TBL