MICROCONTROLLERS Jan 14, 2005

Instant Performance Enhancement with Flash µCs

The ultra-high-speed microcontrollers (UHSM) are 5V, 1-clock 8051s with integrated flash and SRAM. These UHSMs drop into existing 8051 applications and deliver an instant boost in speed. This application note discusses the simple steps that may be required to port an existing 8051application to the UHSM family. The benefits of doing so are described.

Introduction

The ultra-high-speed microcontrollers (UHSM) are 5V, 1-clock 8051s with integrated flash and SRAM. These UHSMs drop into existing 8051 applications and, with little or no effort, give an instant boost in speed. In most cases UHSMs are 100% compatible with original 8051s, so no hardware or code changes are usually required. Three UHSMs are currently available: the DS89C430, DS89C440, and DS89C450, each with 16K, 32K, or 64K of flash memory, respectively. This article describes simple steps that may be required to port an original 8051 application to an UHSM. The benefits of using an UHSM are presented.

Architecture

As mentioned above, the UHSM is a single-clock-cycle 8051, instruction compatible with the original 8051 which relied on a 12-clock cycle architecture. Reducing the number of clocks per instruction to one enables up to 12 times better performance than the original 8051 at equivalent clock frequencies. Optionally, this also provides the ability to run at lower clock frequencies and yield the same performance while reducing overall system power consumption. For greater performance and noise reduction, the UHSM also integrates a clock multiplier that allows either two or four times multiplication of the external crystal. The UHSM, for example, could be used in an existing 7.372MHz 8051 design and run clock quadrupled 29.49 MHz internally. Not only does the on-board clock rate of 29.49MHz improve performance, but by keeping this high frequency isolated internally on the microcontroller, external noise is also kept to a minimum. This greatly reduces EMI.

The UHSMs contain several additional features that make them an excellent choice for new designs:

- Dual datapointers with automatic increment/decrement and toggle select
- In-application programmable Flash
- 1K byte SRAM for MOVX
- Power Management Modes: Idle Mode, Stop Mode, Divide-by-1024 Mode
- Two serial ports

- Watchdog timer
- Power-fail reset and early warning power-fail interrupt

Compatibility

Code

The UHSM is 8051-instruction compatible, and in most cases no changes to the code are required. Code-based timing loops, however, must be rewritten and recalculated against the single-cycle instruction timings. To take further advantage of the performance enhancements of the UHSMs, a few other small code changes are required. One example is using the divide-by-4 option on the timers to allow higher baud rates. Another example is using the datapointer auto inc/dec options to speed up copy, clear, and compare operations.

The on-chip flash memory removes the requirement for external code memory, and the built-in SRAM enables applications to eliminate the requirement for external data SRAM.

Timer/Serial Port

The UHSM can run each timer in divide-by-12 mode (original 8051) off the external crystal, or run in divide-by-4 mode from the multiplied clock (1,2, or 4). This allows existing 8051 timer and serial code to run unmodified, and offers the option of higher baud rates if required by a new design. The CKCON register bits, TxM, manage the 12/4-clock selection.

Hardware

Because the UHSM is a 5V design, it will drop in 5V only systems with no change. External memory access is set to a default of 3 machine cycles (12 system clocks), but can be modified by the CKCON register through the use of stretch cycles to allow usage of slower data memories.

Performance

It is difficult to find a performance benchmark that any two people can agree is reasonable. In any case, speed of memory copies, CRC generation, and interrupt latency are probably interesting to most designers and are analyzed below. As a bonus, performance of the SHA-1 secure hash is thrown into the mix as a high-level C benchmark. SHA-1 is both memory and code intensive, and very relevant in the modern embedded application.

Competitive performance numbers in this section will be labeled "12-clock", "6-clock", or "1-clock," referring to the 8051 architecture behind each. The Philips P89C51RD2 and Atmel AT89C51RD2 are used for the 12- and 6-clock performance numbers, as they can run in either 12- or 6-clock mode and are 5V flash microcontrollers. For the 1-clock numbers, the DS89C440 is used. Note that the 6-clock microcontrollers are precisely twice as fast as the 12-clock microcontrollers. In the case of the UHSM, even though the clocks-per-machine cycle have been reduced to one, not all opcodes can be executed in a single cycle (e.g. DIV AB takes 10 cycles).

Memcopy

Table 1 gives a clock breakdown of a standard 8051 copy loop using two datapointers. The UHSM is 9 times faster than a 12-clock 8051, and 4.5 times faster than a 6-clock 8051.

Table 1. Memcopy Routine Showing UHSM Clock Cycle Advantage

Code	12-Clock	6-Clock	1-Clock
MOVX A,@DPTR	24	12	2
INC DPTR	24	12	1
INC DPS	12	6	3
MOVX @DPTR,A	24	12	2
INC DPTR	24	12	1
INC DPS	12	6	3
DJNZ R0, LOOP	24	12	4
Total Clock Cycles	144	72	16

Table 2 shows the results of an optimized memcopy implementation using the Auto-Increment and Auto-Toggle features of the UHSM. The UHSM with optimized memcopy routine is 18 times faster than a 12-clock 8051, and 9 times faster than a 6-clock 8051.

Table 2. Optimized Memcopy Routine Clock Cycles.

Code	1-Clock
MOVX A,@DPTR	2
MOVX @DPTR,A	2
DJNZ R0, LOOP	4
Total Clock Cycles	8

CRC16

Use of a CRC is important in many embedded applications and is used to verify data integrity. The CRC16 example in Table 5 of Appendix 1 in the *Book of DS19xx iButton Standards*[1] is an optimized implementation. When running against a 12-clock 8051, the UHSM is over 12 times faster; against a 6-clock 8051, it is over 6 times faster.

Interrupt Latency

Interrupt latency can be described in two ways: time delay until an interrupt is serviced, and time to fully service the interrupt vector.

Because interrupt vectoring can only occur between instructions, the longest opcode plus calling time is the worst-case latency. On the 8051s evaluated for this article, DIV AB is the longest instruction, so the worst case "latency until vector" will be DIV AB plus the implicit vectoring LCALL. The 8051 core inserts the LCALL instruction to force execution to change to the interrupt vector routine. Using this example, the UHSM is 5.5 times faster than a 12-clock 8051, and 2.7 times faster than a 6-clock 8051. See **Table 3**.

We will use a simple interrupt service routine to compare the UHSM against the original 8051s for "latency until return." The execution time is the amount of time it takes between the first interrupt vector instruction until the RETI is complete. In this example, the UHSM is 7.2 times faster than a 12-clock 8051, and 3.6 times faster than a 6-clock 8051. See **Table 4**.

Table 3. Worst-Case Latency of Interrupt Vector

Code	12-Clock	6-Clock	1-Clock
DIV AB	48	24	10
Implied LCALL	24	12	3
Total Clock Cycles	72	36	13

Table 4: Example Interrupt Service Routine

Code	12-Clock	6-Clock	1-Clock
CPL P1.1	12	6	2
RETI	24	12	3
Total Clock Cycles	36	18	5

C Example: SHA-1 Secure Hash

The use of security functions is prevalent in embedded systems, and the hash SHA-1 is a security function widely used today. The secure hash algorithm is easily coded in C. For this example, the Keil C compiler version 7.5 is used to build the implementation across each of the microcontrollers being compared. For all platforms, compiler options were selected to use dual datapointers, internal memory, Level 8 optimizations, and optimize for speed. All micros were run at 11.0592MHz. The UHSM is 11 times faster than the 12-clock 8051, and 5.5 times faster than the 6-clock 8051. **Table 5** lists the results of one block SHA-1 runs.

Table 5. SHA1 Results Showing UHSM Advantage

Code	12-Clock	6-Clock	1-Clock
SHA-1 Single Block (hashes/second)	3.19	6.41	35.59

Conclusion

Using an UHSM such as the DS89C4X0 series allows the designer to drop-in a replacement for an existing 8051 design, freshen an older design, or create a new design that an original 8051 would not be able to achieve. The UHSM allows flexibility of software and hardware, as it does not require a change of tools, source code, or hardware environment. Advanced features may be used as needed, and an immense boost in speed is achieved even without use of these features. The UHSM is the easiest upgrade path for 8051 microcontroller-based systems, and should be considered for new applications requiring processing power that standard 8051s cannot provide.

[1] The Book of DS19xx iButton Standards.