Optimal μCs and μC Website Information
by Dennis L Feucht

A comedian of former years, Allan Sherman, used to take classical-music tunes and write humorous lyrics to them. One of them has a line that goes:
"When you're driving in the desert and your car runs out of gas, Lotsa luck, pal. Lotsa luck."

To this I add the line,
"When you're searching for a website that has data, not pizzaz, Lotsa luck, pal. Lotsa luck!"

Some electronic component manufacturers have more to learn about providing technical data over the Internet. This applies in particular to microcontroller suppliers. Today, I got onto the Freescale website. In some ways it is simpler and better-organized than the sites of some of Freescale's competitors. I wanted to look at the instruction set of the category of parts that includes the MC56F8013. It is a 16-bit DSP/μC combination, in the category of the new Microchip dsPIC series. Lotsa luck.

The evolution of the microcontroller (μC) has been toward the DSP, with its Harvard architecture; the data and instruction memories are separated and given separate paths to the processor, thereby easing somewhat the *von Neumann bottleneck*: the single data path between memory and processor. Hardware multiplier and barrel shifter continue the evolution, followed by addressing units that behave like modulus N counters and can address a table in memory, rolling around to the other end of it. When RISC styling is included, more but never enough directly-addressable registers are added to the ALU and useful addressing modes that take more than a single CPU cycle to complete are discarded. This has the "advantage" of making the software engineer use more program memory to write several instructions to calculate the effective address before using it with one of those single-cycle instructions. I am told that compiler optimization is easier with RISC instruction sets though I do not hear how badly real-time, table-driven firmware is impaired by them.

The trend now seems to be to pull back from that extreme and (perhaps somewhat sheepishly but rightly) creep back toward the CISC addressing modes. The Microchip http://www.microchip.com dsPIC has. Some have never really left. The Freescale (formerly Motorola) μCs are in that category, growing out of the PDP-11-oriented architecture of the 6800. The crew who designed the 6800 microprocessor (μP) at Motorola - headed by Chuck Peddle and Will Mathys – went to MOS Technology near Philadelphia and designed a minimalist 6800 called the 6502, the μP that Steve Wozniak of Apple Computer designed into the Apple I and II products. (Commodore, maker of another early 6502-based microcomputer, acquired MOS Technology.) The 7000-transistor NMOS 6502 was offered for the initial price of $20 in unit quantities. Chuck Peddle told a small conference room of us at Tektronix that the 6800 (selling for around $300 each) would eventually come down to that price because it had a chip area no larger than $5 calculator chips in high volume production at that time, 1975. So, he must have reasoned, why not sell the 6502 at that price, blow away the competitors, and still make a profit? I marveled at the cost difference and wondered (in my mid-20s) what the world would become with $5 μPs. What design possibilities that would enable! And it did.

Bill Mensch http://www.wdc.com later developed the CMOS 6502, the 65C02, which is the current 6502 manifestation. The 6502 might still be the highest-volume μP ever, having been discovered by Mitsubishi and incorporated into multiple brands of VCRs and other consumer products. It has remarkably good addressing modes, and edged out the 6800 with indirect, indexed addressing. The instruction provides the base address of a table in memory and the index register is added to it to point to the indexed item in the table. In other μPs, a fixed number in the instruction was added to a base-address register instead – not nearly as useful.

The 6800 had two accumulator registers while the 6502 had only one. The 8080 and succeeding generations of µPs were more like RISC machines in that they had even more ALU registers so that dual-operand addressing of registers was possible. This avoids having to go to memory to execute an operation, but the tradeoff was a limited address space and a very clunky partitioning of it that was eventually done away with in the 80386. The Zilog Z80 appeared after the 8080, 6800, and 6502 and was a combination of their features, yet more like the 8080 register-oriented architecture and lacking that indirect, indexed addressing mode. Lotsa luck, pal.

Nowadays, the prominent RISC µCs have many registers and weak memory addressing capabilities. The idea is to bring into registers the local variables of a routine, then compute away without leaving the CPU. For many real-time routines, however, this simply does not work. Too many processes are happening concurrently and too many variables, tables, and other data structures must be accessed. The CISC addressing modes, requiring two or three memory cycles to complete, are often faster because the µP concurrently calculates the address while the next instruction is fetched from memory. This instruction fetch-execution pipelining was first incorporated into the 6502, which has an average of two memory cycles per instruction. This is not quite as good as the RISC single-cycle ideal but when the addressing power is included, I would rather opt for a 6502-like machine over a PIC or AVR. For this reason, I am somewhat disheartened that the 6502 has fallen to a commercial state of relative obscurity while an old General Instruments µP that did not succeed in the market and was shelved has metamorphosed into the Microchip PIC and has gained prominence. That gives me hope that the 6502 might re-emerge. Renesas, http://www.renesas.com a merger of Mitsubishi and Hitachi µC lines, might make this possible if the I/O capability – especially the PWM generators – and speed are enhanced, along with global engineering support. VHDL for 6502 cores for FPGA implementation are freely available on the Web. What is needed now is the powerful I/O to make the "core" µP into a fully competitive µC. I would like to see Atmel take their attractive µC I/O, replace the RISC AVR core with a 6502, and offer it to CISC-lovers.

In passing, I am also perplexed (though with some explanation for it) that µPs with stack-based architecture have not been better received. A stack gives you an indefinitely large register base. If the top two or three items on the stack are what most routines use, then these items can be kept in ALU registers while the continuation of the stack, handled automatically by stack-controller hardware, pushes and pops the stack to and from stack memory as required. The register-based RISC and DSP Harvard-architecture advantages are achieved without the severity of the register limitation. The commercially-defunct Harris RTX-2000 stack machine has a separate routine return stack. Both stacks, with separate memory and data paths, achieve more than the two-path Harvard architecture advantage. There is some rumbling that the RTX might reappear, and in a low-cost process at a decent price. With the µC I/O offerings of Atmel or Microchip added to an RTX-2000 with hardware multiply-accumulate capability and a barrel shifter, the world of real-time embedded computing could become a much better place. I would like to see Atmel or Microchip and the holder of the Harris RTX-2000 get together on this. It would put Harris (or its RTX successor) back in the forefront of microcontrolling and would give Atmel or Microchip a competitive edge. The RTX is not the only stack-based architecture with working silicon. Forth inventor Chuck Moore also has a multi-100-MHz stack machine. I can think of at least three or four other tested stack machines. They are, like the 6502, minimalist designs with a small "core" of transistors, yet highly functional and lacking abundant I/O.

Now back to the mundane side of the story. Do not plan for too little time in your schedule for the task of obtaining technical data on µC parts from the Web. I did. I supposed that with a few mouse clicks I would be looking at not only the Freescale µC (or digital signal controller, DSC) of choice but also its instruction set. Some of the technical data is in the hardware "datasheet," but in vain did I seek the instruction set. After over an hour of frustrating search of the website, I shot off an email to Freescale:

“In some ways your website is simpler and better organized than some of your competitors, but in one basic respect it is absolutely HORRID. Try to find the instruction set of the MC56F8013, for instance. I cannot find it. Consequently, I cannot design a Freescale processor into a multi-thousand-unit-per-year product I am designing because I DO NOT KNOW WHAT IT IS! I would think that something as BASIC to uCs as the set of op-codes would be given prominence. You people REALLY need to fix this, and soon!”

I do not mean to single out Freescale. They can be excused in part because their 16-bit-wide 56F8000 series has about five times as much functional capability in an IC as the competition at the same price. I have had frustrating times on the websites of their competitors too. These sites share one undesirable characteristic in common. The advertising and promotion of the company is emphasized too much over the provision of the required technical product information needed by designers. Would that application engineers at these companies manage the websites! Meanwhile, lotsa luck.

And who writes the data sheets? Take a look at the Atmel http://www.atmel.com AVR AT90PWM2/3 data sheet. The part itself is worth a good look for power control applications. But who writes the specifications? It is written in ether Chinglish or Bharatish, but not English, and is repeatedly annoying to try to read through. (At least a global replace of “thanks to” with “because of” would help. We thank people, not μCs or their parts, for benefits!) Bad English in technical literature is a detriment, especially to those for whom English is a second language, and in the case of the AT90PWM2/3 detracts from the merits of a fine μC for power electronics applications. With annoyance you can read it, but lotsa luck, pal. Lotsa luck.