

## Flash Forward

### The 68HC908GP20

## SILICON UPDATE

Tom Cantrell



Tom  
thinks the  
8-bit MCU  
market is  
getting

back on the growth  
track and flash MCUs  
are going to be a big  
part of it. Join him as  
he takes a look at a  
new player whose  
name you may already  
be familiar with.



At the Embedded Systems Conference West in 1998, with a bit of help from Tom Starnes of the market-research outfit, Dataquest, I presented a session entitled “8-Bits: Onward and Upward.”

Everyone knows I’ve got an 8-bit chip on my shoulder, so I asked Tom to provide independent confirmation of what I know in my gut. Despite the sizzle surrounding fancy chips, the 8-bit market is where it’s at when it comes to embedded.

In fact, even with the slowdown in ’98 (due to global market weakness and declining prices), the 8-bit MCU market is expected to get back on a growth track. Indeed, Starnes predicts that 8-bit MCU revenue will continue to exceed the combined total of 4-, 16-, and 32-bit (see Figure 1).

Don’t forget that, given the recent price cutting, a revenue perspective understates the 8-bit MCU’s popularity in terms of units. As Carl Sagan might have put it, we’re talking “billions and billions” of 8-bit MCUs per year.

I explained the key technology trends that should invigorate designers. These trends in-

clude things like speed (the 100-MHz MCU from Scenix), mixed signal (already starting to see regulators, hi-V I/O, temp sensors, etc.), intellectual property (the 8-bit MCU market is even bigger if you count all the cores that will be buried in ASICs), and so on.

But, the most apparent trend is the widespread move to flash MCUs. Consider...

- flash pioneer Atmel says flash share is doubling each year, far faster than the overall market
- Hitachi claims 30% of their H8 volume is already flash
- Microchip is increasing their lineup from 2 to 22 devices
- Philips is dealing themselves into the flash fray via a deal with Macronix

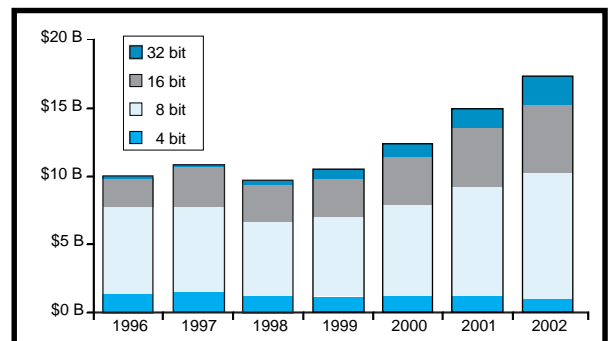


Figure 1—Although it’s been a bit soft of late, the 8-bit MCU market is expected to get back on track soon and remain dominant well into the future. (Source: Dataquest).

• Analog Devices kicks off their foray into MCUs with a flash '51

It looks plenty rosy for flash MCUs, but what's wrong with this picture? Let's ask numbers guru Tom Starnes for a hint. Hey, Tom, who's the unquestioned 8-bit MCU leader?

Why, Motorola, of course. Their chips (notably the 'HC05 family) far outsell even the most popular competitors like '51s and PICs.

Motorola may be number one, but not when it comes to flash MCUs. Sure, they've dabbled with specialized 68k (68F333) and 16-bit chips like the 'HC12 and 'HC16, but they haven't had anything to offer the mainstream 8-bit market in the way of flash MCUs.

Motorola may be late to the party, but judging by the 68HC908GP20, they're making up for their tardiness with a grand entrance.

### A WHOLE NEW 'HC05

As a matter of fact, the 'HC908 is the subject of the *Circuit Cellar Design99 Contest*, which is already under way. No doubt, those of you who've already entered are up to speed on this chip, so get back to work! The rest of you, read on.

Let's start with the big picture shown in Figure 2—'HC05 upward-compatible CPU, 20 KB of flash memory, and 512 bytes of SRAM. The peripherals include all the usual suspects: 33 I/O lines, 2 × 16-bit timer/counters, 8 × 8-bit ADC, UART, and SPI.

If you're unfamiliar with the Motorola architecture, let me explain. Like the '51 and PIC, the '05 is one of those oldies but goodies. As best I can recall, the first '05 chips were introduced about 20 years ago. So, it's no surprise that the basic architecture, comprising an accumulator, index

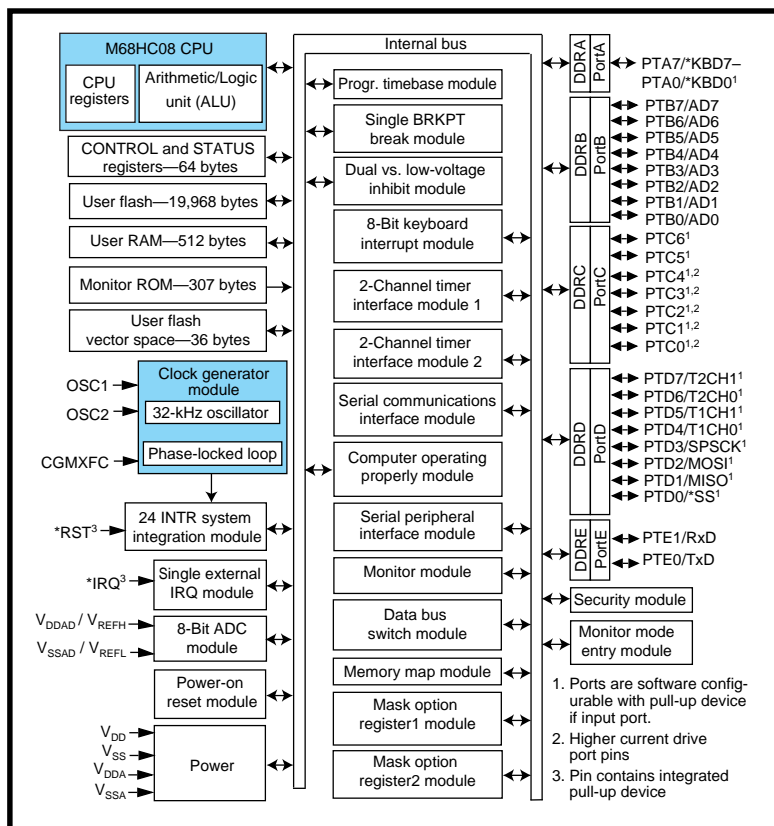


Figure 2—Although it's descended from the humble 68HC05, the 'HC908 delivers a lot more than you might expect from an entry-level micro, including a big chunk (20 KB) of flash memory.

register, and stack pointer, is about as simple as it gets.

The more recent '08 represents a midlife tweak of the '05 design. It's still simple, but with a bit more oomph, especially when it comes to toting the load of C.

Upgrades include boosting the index register from 8 to 16 bits as well as enabling all 16 bits of the stack pointer and program counter for full 64-KB addressing. There are also extra addressing modes, the most appreciated being stack relative for subroutine parameter passing.

And, there are a bunch of instructions including stalwarts such as decimal adjust accumulator (DAA) for BCD, looping (Z80-like decrement and branch), and multiply and divide (five and seven cycles, respectively).

The standard device features premium specs such as a -40° to +85°C temperature range, 3-5V ±10% V<sub>CC</sub> and up to 8-MHz throughput (at 5 V; 4 MHz at 3 V). Branches and instructions that access memory using the fancy addressing modes take 3-5 cycles, but some of the simple register ops

only take one cycle, legitimizing multi-MIPS performance claims.

Naturally, the 'HC908 is available in a 44-pin QFP for small-is-beautiful apps. But, Motorola also reminds us that DIP isn't dead with their 40-pin version for the through-hole crowd (especially welcome for prototyping, not to mention design contests).

### PECK O' PERIPHERALS

MCUs have come a long way in terms of the quantity and quality of built-in I/O—and the 'HC908 is no exception. Indeed, you don't have to get further than

traditionally mundane items like reset and the clock generator to see what I mean.

Besides external reset (i.e., yank \*RST low), the 'HC908 includes five additional ways to get the chip back on track including power on, watchdog timer, low-voltage inhibit, illegal opcode, and illegal address. Unlike lesser chips, there's no detective work needed to figure out what happened. The source of a reset is explicitly identified in a status register.

Motorola has a knack with PLLs, and they put it to use by supporting a 32-kHz watch-crystal operation. The frequency is boosted on-chip as high as 32 MHz, which is four times the internal bus clock.

Yes, PLLs have gotchas, especially when they're called on to increase frequency by a factor of a thousand. Lockup time, noise immunity, jitter, power consumption, and support circuits (see Figure 3) are all concerns.

Without discussing all the details of the 30 pages devoted to the clock generator in the datasheet, I'll just say that Motorola does a pretty good job

covering the bases. If you prefer, the PLL can be disabled in favor of a typical 4× crystal or external clock configuration.

One handy feature of watch-crystal mode is the built-in timebase feature, which generates an interrupt at one of eight programmable rates between 1 and 4096 Hz. Using these interrupts to wake the MCU from its low-power wait and stop modes can eliminate the need for an external real-time clock.

As usual, most of the 33 (31 in DIP) I/O lines wear multiple hats. Twenty-four lines feature bit-by-bit selectable pullups. All ports feature at least 10-mA drive, with five lines capable of 15 mA. Eight lines can be configured as so-called keyboard inputs, with edge- or level-interrupt capability.

The eight-channel multiplexed 8-bit ADC is a successive approximation type with conversion time on the order of 20 μs. A conversion-complete interrupt is an option, as are single or continuous conversions. Pins not required

for the ADC are available for general-purpose I/O use.

The two 16-bit timer/counters operate at up to the bus clock frequency of 125-ns resolution at 8 MHz and feature both input capture and output compare. For input capture, the counter latches when an active (programmable as high, low, or any) edge is detected on the input pin.

Output compare controls the output pin (programmable set, clear, or toggle) when the counter matches a preprogrammed value. The toggle-on-overflow mode is especially useful for PWM.

Those of you who've fiddled with output compare and PWM schemes have probably learned the hard way that glitches can occur because of asynchronous updates. For instance, setting a new compare value can generate runt or missing pulses depending on when the update occurs relative to the count.

The 'HC908 features a unique buffered mode that combines both channels in a ping-pong arrangement to get around the problem.

Unlike the stripped-down UARTs in some low-end chips, the 'HC908 SCI has all the trimmings. For one thing, there's a built-in data-rate generator that's good for standard rates up to 115 kbps, depending on the MCU clock rate.

In addition to all the usual formats, the 'HC908 offers the 9-bit data wakeup mode popular in multidrop applications. It features a full complement of error detection, including overrun, framing, and parity, as well as some sophisticated glitch suppression.

Finally, the four-wire (select, clock, data in, and data out) SPI-clocked serial port is versatile enough to adapt to various standards (including I<sup>2</sup>C), thanks to programmable master/slave mode, phase, and polarity.

Although it isn't Ethernet or Firewire, don't underestimate the SPI port's ability to move data

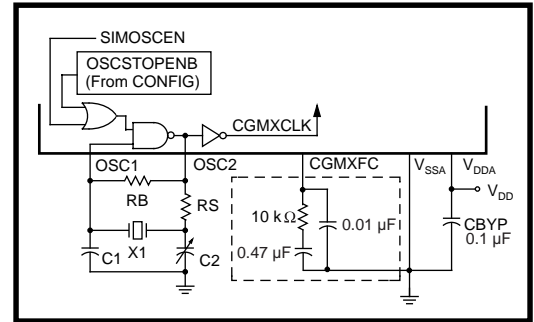


Figure 3—Operation from a 32-kHz watch crystal is a nice feature, but it does call for a few external discretes, including a loop filter that optimizes PLL stability and tracking.

around. Separate double-buffered receive and transmit registers can handle up to 4 or 8 Mbps (i.e., bus clock divided by two or one) in master and slave mode, respectively.

## COOL FLASH

Flash memory is the point of the 'HC908, so let's take a closer look. The 20-KB array is organized in 8-byte pages, with eight pages composing a row, which corresponds to the minimum program-and-erase granularity.

The eight-to-one ratio isn't a coincidence but a requirement. The flash technology dictates that a row should not be programmed more than eight times before it's erased.

Eliminating the need for a separate V<sub>pp</sub> programming voltage isn't easy, given the wide operating range (2.7–5.5 V). Nevertheless, with a built-in charge pump and voltage regulator, Motorola pulls it off.

Unlike other flash MCUs, the 'HC908 is completely self- or in-operation programmable. In a process akin to performing brain surgery on yourself, the chip executes software in one portion of the flash that programs the other. Lest things go awry, the 'HC908 incorporates a block-protect mechanism as malpractice insurance against a self-lobotomy.

The programming operation isn't for the faint of heart. As Figure 4 shows, it's a rather convoluted process with specific sequencing and timing that must be followed exactly. The algorithm relies on a margin-read scheme to program the flash memory just enough to gather the 10-year data-retention spec, without overstressing and compromising the 100-cycle/row

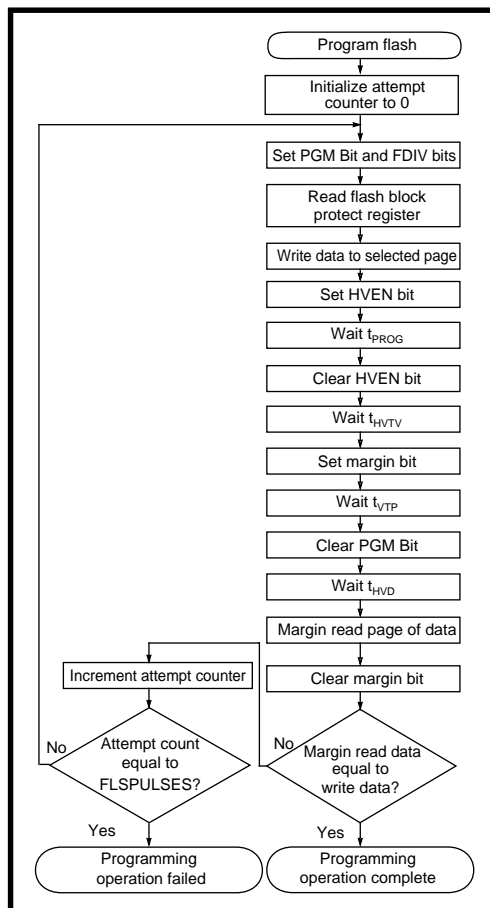


Figure 4—The 'HC908 programming algorithm is finely tuned to match the requirements of the flash circuitry. Sloppy coding or trying to roll your own algorithm is definitely a no-no.

write endurance spec.

The concept of self-programming raises the question of how to get a factory-fresh blank chip programmed in the first place. The answer is a tiny (307 bytes) built-in ROM monitor with just enough smarts to get past the who's-on-first dilemma.

The monitor includes only a few commands—read and write memory (i.e., SRAM), READ the stack pointer, and GO. On this minimal foundation, arbitrarily complex functions, such as programming the flash, can be built.

The monitor is entered in one of two conditions. First, if the chip is new or recently erased (i.e., the RESET vector is blank). However, normal operation of an already-programmed chip can be bypassed in favor of the monitor by applying a high voltage ( $V_{DD} + 2.5\text{ V}$ ) to the \*IRQ pin.

The monitor incorporates a security feature to discourage prying eyes. On entry, the host must send a sequence of eight bytes that match those stored at \$FFF6–FFFF.

If you care about keeping your code secret, don't leave these locations blank. If the security check fails, attempts to read or execute from flash are blocked. But, the monitor permits the flash to be erased, so at least the absentminded won't have to throw the chip away.

## HERE COMES DE BUG

Besides flash-memory programming, the monitor provides the hooks for a low-cost debugging scheme, which Motorola exploits in their

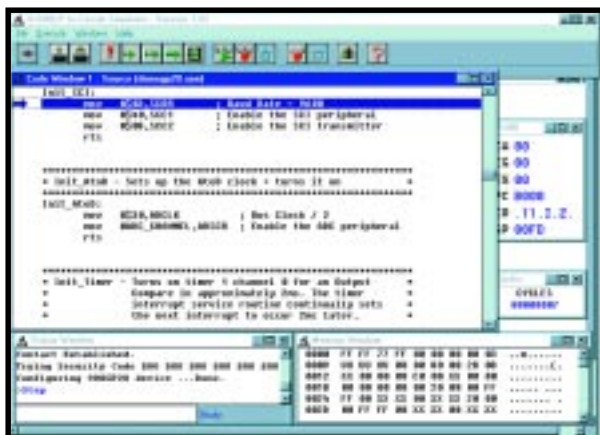


Photo 2—The P&E Micro software included in the kit (and free on the web) features an assembler and simulator, as well as tools that work specifically with the Motorola hardware, like an in-circuit simulator, debugger, and programmer.

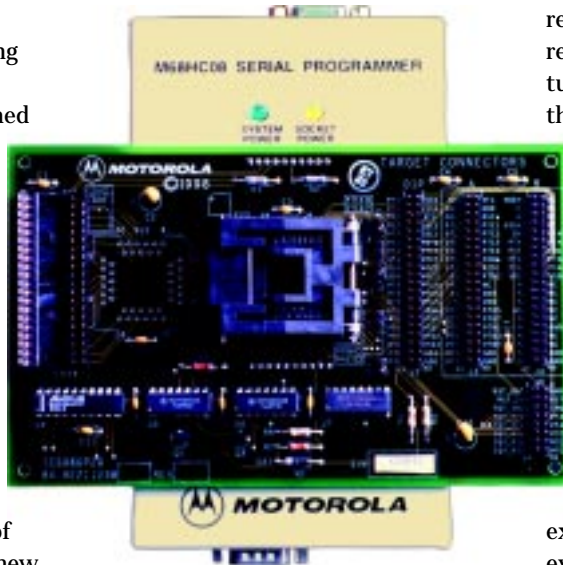


Photo 1—The \$295 development kit includes the hardware, software, and documentation needed to program and debug assembly-language apps for the 'HC908.

evaluation kit (see Photo 1).

At \$295, the kit may seem pricey, but it's more upscale than most. Its sturdy metal enclosure base unit hosts the 'HC908 personality card and connects to a PC serial port. The QFP MCU on the personality card is socketed so the 100-cycle endurance limit isn't a showstopper for us crash-and-burn types.

A ribbon cable with a 40-pin DIP on the end targets your own app hardware as well as the requisite wall-wart supply, which includes mains adapters that offer worldwide compatibility.

Also, there are printed versions of the documentation, and the package includes a nice development suite from P&E Microcomputer Systems. It features an IDE (see Photo 2) wrapped around various components including an assembler, programmer, a soft simulator, an in-circuit simulator, and an in-circuit debugger.

The soft simulator

represents the traditional approach, relying only on the PC to host a virtual chip. It's a good way to check out the chip and try some programming without spending the bucks (it's free via the web).

The in-circuit debugger does a fairly good job of controlling and monitoring execution of an application-programmed chip. The process is aided by the fact that the 'HC908 incorporates a hardware breakpoint so the flash doesn't wear out by jamming in software breaks. There's no real-time trace, but the system does execute at full speed between debug events.

In-circuit simulation combines the total visibility of a soft simulator with real-world I/O. Instruction execution is completely pantomimed by the PC, but operations that talk to pins are passed over to the real chip. It's much slower than real time and, because of the to and fro, even slower than purely soft simulation. But, it does enable you to perform basic reality checks of your circuit design and wiring.

Although the documentation says the P&E software needs little more than Windows 3.x or 95/98 and 640 KB of RAM, it didn't pass my clunker (33-MHz '386) test. The software-only stuff (assembling, soft simulating, etc.) worked, but communication with the hardware was flaky. No problems on my 300-MHz Pentium II, though, so the threshold of pain must be somewhere in between.

## FLASH TO THE FUTURE

Motorola didn't kick off the flash-memory trend, but their entry into the market will add fuel to the fire, especially since the chip is priced at \$4.95. That's enough to get any designer's creative juices flowing, as I expect we'll see when Design99 entries start rolling in.

Hey, Elizabeth, any chance we forgot to put in the no-employees boilerplate in our contest rules? ☹

***You don't need to pay for a development kit for Design99. [www.circuitcellar.com](http://www.circuitcellar.com) features free development software and tools, and well as information on third-party sources.***

*Editor's note: Sorry, Tom, no luck. But, I've heard that some people are having problems obtaining an 'HC908. Now they can check our web site for updated information on distributors.*

*Tom Cantrell has been working on chip, board, and systems design and marketing in Silicon Valley for more than ten years. You may reach him by e-mail at [tom.cantrell@circuitcellar.com](mailto:tom.cantrell@circuitcellar.com), by telephone at (510) 657-0264, or by fax at (510) 657-5441.*

## SOURCES

### **68HC908GP20**

Motorola

(512) 328-2268

Fax: (512) 891-4465

[www.mot-sps.com/sps/general/chips-nav.html](http://www.mot-sps.com/sps/general/chips-nav.html)

### **Development suite**

P&E Microcomputer Systems, Inc.

(617) 353-9206

Fax: (617) 353-9205

[www.pemicro.com](http://www.pemicro.com)