Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

FREESCALE'S FIRST FLEXIS MCUS

New 8- and 32-Bit Microcontrollers Offer Pin Compatibility

By Tom R. Halfhill {6/26/07-01}

Years ago, some crazy hot-rod mechanics crammed V8 engines into their classic Volkswagen Beetles. This hardware hack wasn't easy. The huge V8 transformed a cute Bug into a kludgy monstrosity. Freescale Semiconductor wants to bring a similar upgrade to embedded systems, only without the kludge quotient. At this week's Freescale Technology Forum in Orlando, Florida, Freescale is unveiling the first microcontroller family with pin-compatible 8- and 32-bit devices. These new Flexis-family MCUs for consumer and industrial applications will allow developers to pull an 8-bit chip out of a socket, replace it with a 32-bit part, update the firmware, reboot, and continue running the system as before—except with much more horsepower. Freescale estimates that a 32-bit Flexis MCU can deliver eight to ten times more throughput, depending on the application.

Microprocessor Report began covering Freescale's "Microcontroller Continuum" strategy last year. Our first article outlined the foundation of the strategy. (See *MPR 5/1/06-01*, "Freely Scaling From 8-Bits to 32.") Our second report described the new ColdFire V1 instruction-set architecture that forms the 32-bit half of the foundation. (See *MPR 8/28/06-02*, "Connecting the Continuum.") Now, Freescale has announced the first implementations, with enough information for an initial analysis.

Our conclusion: Flexis definitely isn't a kludge, though it does make some inevitable compromises. Overall, Flexis embodies a sensible strategy, both for Freescale and for embedded-system developers.

Wrestling ARM for 32-Bit Customers

It's easy to understand why Freescale is pursuing this difficult strategy. As more embedded applications outgrow the capabilities of 8-bit MCUs—because of their limited processing power, memory addressing, and I/O—there's a growing migration toward 32-bit MCUs. And the trend is to skip 16-bit MCUs altogether. If 32 bits is the logical endpoint, developers don't want to port their 8-bit designs more than once. Economics also plays a role: in modern fabrication processes, 32-bit MCUs can be as inexpensive to manufacture as 8-bit MCUs. The 8-bitters are usually fabricated in older, larger processes and are too pad-limited to benefit from further shrinkage.

ARM, the leading vendor of licensable 32-bit processor cores, quickly recognized the business opportunity of converting 8-bit users into new customers. In recent years, ARM has worked aggressively to promote its already popular architecture as the new standard for 32-bit MCUs. In 2004, ARM announced the Cortex-M3 core, designed specifically for this market. (See *MPR 11/29/04-01*, "ARM Debuts Logical V7.") The Cortex-M3 enabled startup Luminary Micro to introduce the first 32-bit MCUs priced as low as a dollar. (See *MPR 6/5/06-02*, "32 Bits for a Buck.") Other ARM licensees are busily expanding their MCU product lines, using ARM7-, ARM9-, and ARM11-family cores in addition to the Cortex-M3. (See *MPR 4/4/05-02*, "ARM-Based MCUs Flex Muscles.") In 2005, ARM acquired Keil, a leading vendor of 8-bit software-development tools.

Not surprisingly, Freescale wants a piece of the action, too. And unlike ARM, whose main business is licensing 32-bit processor cores, Freescale is a broad-based semiconductor

Feature	Freescale Flexis S08QE128	Freescale Flexis MCF51QE128
Architecture	HCS08	ColdFire-V1
Architecture Width	8 bits	32 bits
SRAM	4K, 5.8K, or 7.8K	4K or 8K
Flash Memory	64K, 96K, or 128K	64K or 128K
Serial Comm Interface	2 x SCI	2 x SCI
Serial Peripheral Interface	2 x SPI	2 x SPI
I^2C Interface	1 or 2	2 x I^2C
Keyboard	2 x KBI	2 x KBI
Interrupts	6 or 8 channels each	8 channels each
Gen. Purpose I/O	26–70 GPIO pins*	54 or 70 GPIO pins*
PWM	1 x 6 channels	1 x 6 channels
Timers (16-Bit)	2 x 3 channels	2 x 3 channels
Watchdog Timer	Yes (COP)	Yes (COP)
A-to-D Converter	12 bits, 10, 22, or 24 channels	12 bits, 22 or 24 channels
Analog Comparators	2 x ACMP	2 x ACMP
Internal Clock	RTC	RTC
Debug Interface	1-pin BDM	1-pin BDM
Core Frequency	50MHz	50MHz
Bus Frequency	25MHz	25MHz
IC Process	0.25 microns	0.25 microns
Voltage	1.8–3.6V	1.8–3.6V
Temp Range	–40° to +85°C	–40° to +85°C
Pin Counts	32, 44, 48, 64, or 80 pins	64 or 80 pins
Package Options	LQFP-32, QFP-44, QFN-48, LQFP-64, LQFP-80	LQFP-64, LQFP-80
Package Sizes	$7mm^2$, $10mm^2$, or $14mm^2$	$10mm^2$ or $14mm^2$
Power (Typical)	52.5mW @ 3.0V † 50MHz CPU / 25MHz bus	100.2mW @ 3.0V † 50MHz CPU / 25MHz bus
Dhrystone 2.1 Performance	n/a	0.94 Dmips/MHz (SRAM) 0.76 Dmips/MHz (flash)
Price (10kU)	$3.59 up	$3.80 up
Availability	Samples: Jul-07 Prod: 1Q08????	Samples: Jul-07 Prod: 1Q08????

Table 1. Feature summary of 8- and 32-bit Freescale Flexis microcontrollers. Although this table lists only two MCUs, they are the basis for about a dozen parts, counting all the variations of SRAM, flash memory, integrated peripherals, I/O ports, pin counts, and chip packages. Chips with identical packages and higher pin counts are pin compatible across the 8-bit S08 and 32-bit ColdFire architectures; these parts are highlighted in purple. *The number of GPIO ports doesn't include one input-only port and one output-only port. †Power-consumption ratings are preliminary "typical maximum" measurements, with the chips running at maximum speed and all peripheral clocks enabled; see Table 2 for more details. (n/a: data not available)

company with existing 8-, 16-, and 32-bit MCUs to defend. Freescale doesn't want to cannibalize those product lines, but neither does the Austin-based company want to lose existing customers to rival architectures. Therefore, Freescale's strategy is both defensive and offensive. Flexis should help Freescale hold onto 8-bit customers that might be tempted to defect to ARM, and Flexis may win new customers as well.

One complication of Freescale's strategy is that the company has two completely different 32-bit CPU architectures. One of them is ColdFire, an embedded derivative of the Motorola 68000 (68K), which dates to the late 1970s. The other is the Power Architecture (formerly PowerPC), freshly invigorated by the Power.org Consortium and renewed cooperation with IBM. Indeed, Freescale recently began invading ARM's territory by licensing 32-bit ColdFire and Power cores for the first time. (See *MPR 4/2/07-01*, "Freescale Licenses Power Cores.") Good arguments could be made for using either ColdFire or Power—or both architectures—to anchor the 32-bit end of Freescale's Microcontroller Continuum.

Pin Compatibility Is Real, But Limited

Freescale's Flexis family will debut with about a dozen 8- and 32-bit chips, counting all the variations of features and chip packages. These variations boil down to one basic 8-bit MCU (the MC9S08QE) and one basic 32-bit MCU (the MCF51QE). The 8-bit devices are based on Freescale's S08 architecture, which traces its lineage to the Motorola 6800 in the 1970s, long before Motorola's semiconductor division spun off and became Freescale in 2004.

Flexis 32-bit MCUs are based on ColdFire-V1, not the Power Architecture. It may seem puzzling that Freescale is renewing its devotion to ColdFire at a time when Power is a rising force. Freescale explains that ColdFire is still a popular 32-bit architecture for consumer and industrial applications—the same markets for which the initial Flexis MCUs are intended. Freescale currently offers more than 50 ColdFire products and continues to develop new ones. In addition, millions of embedded developers are familiar with the ColdFire/68K architectures, so they should feel immediately comfortable with the new ColdFire-based Flexis MCUs.

All that is true. But Power is popular, too, and it has more future potential than ColdFire does. Freescale isn't disclosing future plans, but *MPR* wouldn't be surprised if the Flexis family someday expands to include 32-bit MCUs based on Power cores as well as ColdFire cores. Such a move would be almost mandatory if Freescale introduces Flexis products for the automotive market. That market is vital for Freescale, and Power is central to Freescale's strategy in automotives.

Although pin compatibility is the major selling point of the Flexis family, it's not universal across the whole product line. The only pin-compatible 8- and 32-bit Flexis chips are those with identical packages and higher pin counts. For example, the first 32-bit Flexis MCUs will be available in low-profile quad flat packs (LQFP) with either 64 or 80 pins, depending on the number of general-purpose I/O (GPIO) ports they have. These chips are pin compatible with the 8-bit Flexis MCUs in identical LQFP-64 and LQFP-80 packages. The 8-bit Flexis MCUs with fewer than 64 pins don't have 32-bit counterparts. Freescale says it may extend pin compatibility to Flexis devices with smaller pin counts in the

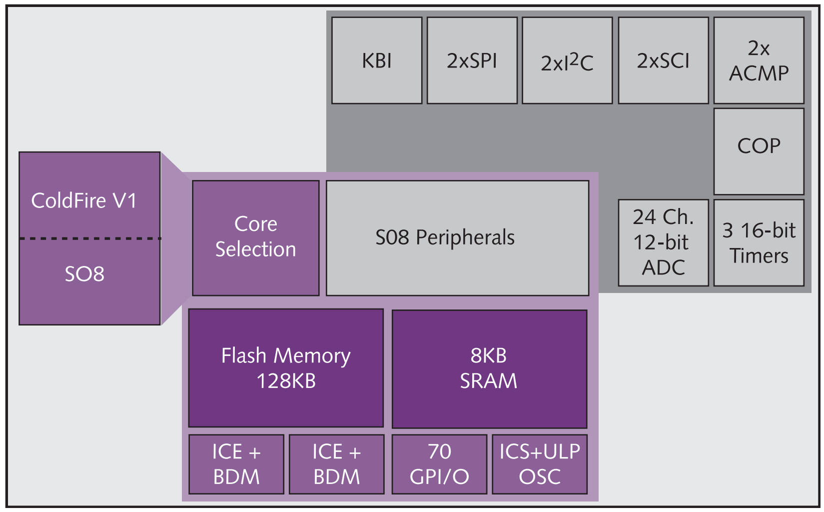

Figure 1. Flexis microcontroller block diagram. Essentially, the 8-bit and 32-bit MCUs are the same design with different processor cores. By keeping on-chip peripherals the same across both sides of the product line, Freescale can maintain a high degree of pin compatibility. Both architectures share the same peripheral drivers and one-bit debug port.

future, if customers want it. Table 1 summarizes the features of the initial Flexis devices, which are intended for consumer and industrial applications.

Even when 8- and 32-bit Flexis MCUs aren't pin compatible, they still share things in common. Their integrated peripherals are the same, and so are their peripheral drivers. Developers can reuse this software when upgrading a design. And all Flexis MCUs have a one-pin port for background debugging mode (BDM). Their on-chip real-time trace buffers are large enough to capture a few hundred instructions.

In addition, programmers can write software for all Flexis MCUs using the same development tools—although the compiled code will be different, of course. A new version of Freescale's CodeWarrior for Microcontrollers supports both CPU architectures. Freescale offers a free Special Edition, which limits the size of compiled code to 32KB on 8-bit MCUs and 64KB on 32-bit MCUs. For $999, developers can unlock this limitation. The $2,495 Standard Edition supports unlimited amounts of code and object-oriented programming in C++. The $4,995 Professional Edition adds code-coverage tools, profilers, a better debugger, and built-in support for real-time operating systems. GNU C and other third-party tools are coming, too. IAR Systems plans to ship a development environment for ColdFire Flexis devices in 3Q07 and for S08 Flexis devices in 1Q08.

Peripheral Compatibility Eases Transition

To assure skeptics that 8- and 32-bit MCUs can be truly pin compatible, Freescale staged a live demonstration for industry analysts in Austin last March. Freescale engineers showed a working development system with an 8-bit MCU. They shut down the system, yanked the S08 chip out of its socket, replaced it with a ColdFire device, flashed the firmware, and rebooted the system. Sure enough, it worked. The swap took only a few minutes.

This demo was impressive but immediately raised questions. Does pin compatibility limit the mix of integrated peripherals? How portable is the application software? How bad is code expansion after recompiling for 32 bits? What are the power-supply requirements? What about the effects on power consumption and thermals? Will the additional performance justify the higher power and cost? Now that Freescale is officially announcing the Flexis chips, we have some answers to these questions.

To maintain maximum compatibility with existing sockets and drivers, the initial 32-bit Flexis MCUs have the same integrated peripherals as the 8-bit Flexis MCUs, as the block diagram in Figure 1 shows. No more, no less. In fact, whenever possible, Freescale is reusing the same RTL for the peripherals and other modules. However, this decision imposes some restrictions on the 32-bit chips.

Today's 32-bit MCUs often sport Ethernet, USB, Bluetooth, LCD/video controllers, keypad controllers, control-area network (CAN) interfaces, and other luxuries uncommon in 8-bit MCUs. These features are absent from the initial Flexis devices. Usually, such features require more I/O pins. If future 32-bit Flexis MCUs add such features, they risk sacrificing pin compatibility with the 8-bit chips. This compromise is inevitable but will disappoint developers who crave a 32-bit upgrade for those features. Freescale's product-development roadmap does anticipate adding USB and CAN to both the S08 and ColdFire lines, which could allow Flexis devices to incorporate those interfaces while preserving pin compatibility. (Freescale currently sells ColdFire-based MCUs with USB, Ethernet, and CAN controllers outside the Flexis family, but they aren't pin compatible with any 8-bit MCUs.)

Memory is another reason for upgrading to 32 bits. Developers running short of memory in their 8-bit designs should welcome the extra breathing room of a 32-bit architecture. Although 8-bit Flexis MCUs can address up to 128KB of flash memory in 16KB banks, the 32-bit MCUs offer flat memory addressing for virtually any amount of SRAM or flash memory that will fit on the chips. Curiously, Freescale isn't fully exploiting this advantage yet. The initial 32-bit Flexis devices have no more than 128KB of flash—the same amount available in the 8-bit Flexis parts. (All the memory is internal. These MCUs have no external memory bus, which simplifies pin compatibility.) Freescale says future 32-bit Flexis chips may incorporate more than 128KB of flash.

More memory would be handy, because 8-bit code tends to expand when ported to 32 bits. Freescale estimates 30% to

Power Mode	Freescale Flexis S08QE128	Freescale Flexis MCF51QE128	Notes
Full Run Mode 50MHz / 25MHz	11.9mA (milliamps)	25.9mA (milliamps)	Full functionality and performance
Low-Power Run Mode 2MHz / 1MHz	647µA (microamps)	1,410µA (microamps)	Full functionality, less performance
Lower-Power Run Mode 32KHz / 16KHz	6.1µA (microamps)	9.5µA (microamps)	Full functionality, lowest performance
Stop 2 Wait Mode Lowest-Power Mode	300nA (nanoamps)	300nA (nanoamps)	Only SRAM, I/O, RTC (optionally) are powered; wakeup always requires reset (~6ms)
Stop 3 Wait Mode Power-Down Until Interrupt	450nA (nanoamps)	450nA (nanoamps)	CPU, bus clocks, most modules disabled; wakeup by interrupt is serviced without reset

Table 2. Flexis low-power modes. Freescale's preliminary datasheets list 15 power modes, some supporting multiple clock rates. This table summarizes the most important modes. In full run mode, Freescale measured preliminary "typical maximum" power at 3.0V and 25°C with the chips running at full speed, all peripherals enabled.

40% code expansion after recompiling a typical C program from the 8-bit S08 to the 32-bit ColdFire-V1. Of course, this estimate assumes the 8-bit software is written in C to begin with. Older code may be written in difficult-to-port assembly language. (Freescale says a customer survey found very little recent software written in assembly—mostly for time-critical interrupt-service routines.) If programmers move to C++, the overhead of object-oriented bureaucracy will probably bloat the code even more. However, there are instances when 32-bit code is more compact. Eight-bit programs that rely heavily on emulated math libraries can actually shrink when recompiled for 32 bits, because 32-bit processors have more function units capable of executing math-intensive code in hardware.

Power Rises But Stays Reasonable

Any migration from 8 bits to 32 bits raises the specter of power and thermal problems. If someone crams a V8 engine into a VW, you can bet it won't be air-cooled. Fortunately, Freescale has taken steps to keep the power dissipation of 32-bit Flexis MCUs within reasonable boundaries. Indeed, by 32-bit standards, these chips are miserly.

As Table 1 shows, typical power consumption for an 8-bit Flexis MCU is about 52.5mW, which nearly doubles to 100.2mW for its 32-bit counterpart. Keep in mind that Freescale measured these numbers at 3.0V while running the chips at their top speed (50MHz CPU frequency, 25MHz bus frequency) with all peripherals enabled. All Flexis chips support a range of 1.8V to 3.6V, and they reach their maximum clock frequency at 2.1V. (Typical maximum power at that voltage wasn't available at press time.) The 2x power penalty for upgrading to 32 bits isn't excessive and shouldn't pose a problem, unless a system's power and thermal requirements are marginal already.

In some cases, there will be no penalty at all. Many embedded systems spend almost all their time idling in a low-power mode. They wake up briefly to perform a task, then return to their idle state. A 32-bit processor can perform the brief task more efficiently than an 8-bit processor, if the additional throughput outweighs the additional energy. Freescale estimates that a 32-bit Flexis MCU will deliver about 8x–10x more throughput than an 8-bit Flexis MCU while drawing about twice as much current. Therefore, a program matching this workload profile would run more efficiently on the 32-bit processor. Freescale is working on new development tools that will help engineers make these analyses.

Flexis MCUs have multiple low-power modes. Without sacrificing functionality, the CPU can drop its frequency to 2MHz, and the internal memory bus can drop to 1MHz. At those clock rates, the chip draws at least one order of magnitude less current than it does in full run mode at 50MHz/25MHz. Clocking the CPU even slower, to 32kHz—yes, that's kilohertz, not megahertz—still retains full functionality, but it reduces performance to a bare minimum. In this mode, current falls by two additional orders of magnitude.

Stop 2 mode is the lowest-power mode for Flexis MCUs. This mode puts the voltage regulator into a still-functional low-power state while trickling power to the SRAM, I/O pins, and (optionally) the real-time clock. The rest of the chip powers down. Wakeup from this mode always requires a chip reset, even if the wakeup source is an interrupt. Stop 3 mode maintains the voltage regulator in the same low-power state as Stop 2 mode, but the chip remains powered. The CPU, bus clocks, and most modules go to sleep, though some peripheral modules, and the real-time clock may remain active. An interrupt can awaken the chip from this mode without triggering a reset. Table 2 summarizes the Flexis power modes. Note that in Stop 2 and Stop 3 modes, the 32-bit chip draws little more or no more current than the 8-bit chip does.

To achieve such low power levels, Freescale uses extensive clock gating to shut off power to unused peripherals and other modules, even during full run modes. The internal voltage regulator, real-time clock, and crystal oscillator (which draws less than one microamp) eliminate the need for external sources, reducing overall system power, too. But with so much emphasis on low power, one may question why Freescale is manufacturing both the 8-bit and 32-bit Flexis chips in an ancient 0.25-micron CMOS process. Although it's common industry practice to manufacture MCUs in long-amortized process lines, a process shrink for at least the 32-bit chips could reduce their power consumption, bringing them more into line with the 8-bit parts. A process shrink would also make room on the 32-bit MCUs for more memory, which would exploit their superior memory addressing.

Freescale explains that power consumption in run mode is only one consideration for these chips. At smaller fabrication geometries, circuits may draw more current while idling in stop mode than they would if fabricated in a larger process. As mentioned above, it's common for some MCUs to spend most of their time idling until a task must be performed—depending on the application, of course. In those cases, drawing less current in stop mode can be more important than drawing less current in run mode. Freescale manufactures many other chips in state-of-the-art fabrication processes, so access to the latest technology isn't a problem for the company. Freescale says it deliberately chose a 0.25-micron process and optimized it for low stop currents because it delivers the right balance of low power, good performance, and low cost.

Price & Availability

Freescale plans to deliver early samples of the MC9S08QE128 and MCF51QE128 microcontrollers in LQFP-80 and LQFP-64 packages in July, with manufacturing samples following in November. Volume production is planned by the end of this year. Prices for the 8-bit MCUs in 10,000-unit quantities start at $3.09 for parts with 64KB of flash memory in an LQFP-64 package. Prices for the 32-bit MCUs in the same quantities start at $3.59 for parts with 64KB of flash in an LQFP-64 package. An evaluation board compatible with 8- and 32-bit devices costs $325. A demonstration board with 8- and 32-bit daughtercards, a serial port, and a USB debug port is $99.

Freescale's cross-compatible software-development environment, CodeWarrior for Microcontrollers, is available in three editions. The free Special Edition limits the size of compiled code to 32KB for S08-based Flexis MCUs and 64KB for ColdFire-based Flexis MCUs; $995 unlocks the compiler and permits unlimited code size. The $2,495 Standard Edition supports C++. The $4,995 Professional Edition adds code-coverage tools, profilers, a better debugger, and support for real-time operating systems.

For more information about Flexis, visit:

- *www.freescale.com/files/pr/flexis.html*

Analyzing the Facets of Flexis

There are two ways to evaluate Flexis MCUs. One way is to disregard their 8/32-bit compatibility and merely compare them singly to other 8- and 32-bit MCUs. Another way is to acknowledge pin compatibility as their distinguishing feature, and compare them with other alternatives for upgrading an 8-bit design to 32 bits. MPR leans toward the latter approach.

If 8/32-bit compatibility doesn't matter, it's possible to find cheaper 8-bit MCUs with similar features, or comparably priced 32-bit MCUs with better features. S08-based Flexis chips start at $3.09 (with 64KB flash), which is a little pricey by 8-bit standards, although the Flexis devices are among the most well-endowed examples. ColdFire-based Flexis chips start at $3.59 (with 64KB flash), which is competitive for a 32-bit MCU. But many other 32-bit MCUs have more-advanced integration features, such as on-chip controllers for Ethernet, CAN, USB, LCDs, keypads, and wireless. Flexis MCUs tend to have more-traditional integration features, such as UARTs, SPIs, GPIOs, timers, and analog-to-digital converters.

These differences reflect Freescale's decision to transplant existing S08 peripherals into both the 8- and 32-bit Flexis chips. Normally, 32-bit MCUs boast newer I/O standards and features. As mentioned above, Freescale's roadmap anticipates adding more of those features to Flexis, which will make the new family more competitive for the latest embedded applications. Until then, developers who don't care about 8/32-bit compatibility can probably find better deals—even from Freescale, which has other families of 32-bit MCUs with those features already.

Compatibility is what Flexis is really all about. Developers that are using Freescale's S08 chips should be very interested in the opportunity for a 32-bit upgrade without redesigning the whole system. The savings in reengineering costs may be worth much more than saving a few cents on each MCU. Of course, this analysis depends on numerous factors, such as the suitability of the hardware design, the portability of the software, and the projected sales volume of the product.

Even for developers not already using the S08, Flexis offers a path to a "future-proof" design—or at least, a design as future-proof as anyone can realistically expect these days. Developers can make an 8-bit system that's ready for a 32-bit upgrade at any point. Another attractive possibility is a dual 8/32-bit design that supports multiple products with different features at different prices. The latter approach is similar to what semiconductor vendors do with the multiple bond-out options on chips, except that Flexis can raise it to the system level.

All told, Flexis makes good business sense for Freescale and good technical sense for some developers—especially those already wedded to Freescale. It's never been easier to upgrade an 8-bit design to 32 bits. Of course, some compromises are inevitable. If the Flexis family evolves to incorporate better feature integration, more memory, Power cores, and perhaps the advantages of newer fabrication technology, Freescale's Microcontroller Continuum will grow even stronger. ◇