Reed Electronics Group

MICROPROCESSOR REPORT

www.MPRonline.com

THE INSIDER'S GUIDE TO MICROPROCESSOR HARDWARE

FAULT TOLERANCE FOR CORTEX-M3

ARM Modifies MCU Core for Critical Embedded Systems

By Tom R. Halfhill {5/12/08-01}

ARM is enhancing its Cortex-M3 processor core with faster clock speeds, configurable debug logic, new power-saving features, and compatibility with third-party fault-tolerance technology. All the enhancements make the Cortex-M3 even more suitable for microcontrollers, but fault tolerance is especially important for automotive, medical, and military applications.

Cortex-M3 Release 2.0 was announced at last month's Embedded Systems Conference in San Jose. It's the first major modification since the 32-bit MCU core was unveiled in 2004. (See *MPR 11/29/04-01*, "ARM Debuts Logical V7.")

Designed to replace the aging but popular ARM7TDMI core in MCUs, the Cortex-M3 has been licensed to more than 20 companies. It's vital to ARM's strategy of capturing a larger share of the MCU market, especially in embedded systems needing more processing power than 8- and 16-bit MCUs can deliver. Two years ago, the Cortex-M3 enabled 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.")

Little Tweaks Add Up

Software compatibility is unchanged. The Cortex-M3 remains the only processor exclusively supporting the ARMv7-M instruction-set architecture, which consists entirely of Thumb and Thumb-2 16-bit instructions for greater code density. But ARM is responding to customer feedback by modifying the Cortex-M3 in several ways. To squeeze out a little more throughput, ARM straightened some critical paths in the core, so the maximum worst-case clock frequency is now 250MHz, assuming fabrication in a generic 90nm CMOS process with Artisan Advantage cell libraries. The original release of the Cortex-M3 reaches about 190MHz under similar conditions.

Most other enhancements in Release 2.0 target power consumption and design flexibility. For instance, some developers complained that the Cortex-M3's original debug/trace logic was too large for designs that must cram the core into very small chips. Heretofore, the only alternative was to omit the debug logic altogether, making verification more difficult. ARM has responded by making the debug features more configurable. Developers can choose one to four levels of data watchpoints and two to eight levels of hardware breakpoints, allowing finer trade-offs between debugging capabilities and gate counts.

A new wakeup interrupt controller (WIC) allows the Cortex-M3 to enter an ultralow-power standby mode that halts all clock signals to the processor. The WIC couples to the existing nested vectored interrupt controller (NVIC) and supports a configurable number of interrupts, at a cost of about 50 gates per interrupt. The WIC distinguishes between critical interrupts that need to wake up the processor and those that don't. Because the processor remains in deep-sleep mode without directly monitoring the interrupts, the WIC reduces dynamic power and static leakage. The Cortex-M3 retains its state during these slumbers, so wakeup is fast: about 12 clock cycles to enter the interrupt handler. And the WIC is an optional block—developers not wanting it can leave it out.

Another power saver is a new Artisan standard-cell library for implementing Cortex-M3 designs in a 0.18-micron ultralow-leakage fabrication process. ARM says this physical intellectual property (IP) can reduce static leakage

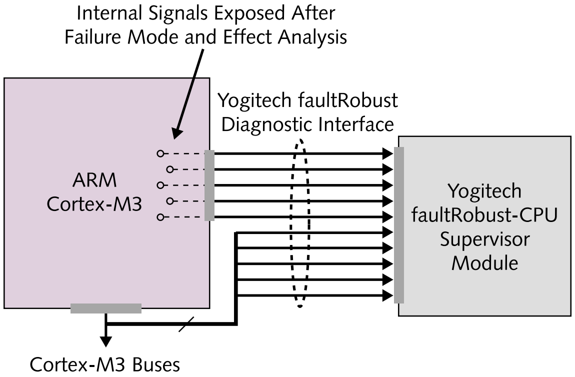

Figure 1. ARM's enhanced Cortex-M3 processor has an optional observation port called the faultRobust Diagnostic Interface (fRDI) that couples to Yogitech's fRCPU fault-supervisor module. ARM modified the Cortex-M3 to expose 130 internal signals to the fRCPU. Yogitech's faultRobust failure-mode and effect analysis (fRFMEA) determined which signals need monitoring for critical operation. Systems that don't need this safety feature can omit the module and observation port without affecting the size or performance of the Cortex-M3.

by a factor of 20× and shrink the die area by 20%, compared with conventional physical IP in a generic fabrication process. (Although 0.18 micron may not seem like state of the art, it's currently the sweet spot for MCUs.) The new Artisan library has configurable single-ported register files and a power-management kit that allows further tweaking by developers.

Fault Supervision for Critical Systems

All of ARM's enhancements are worthwhile, but the most interesting new feature of the Cortex-M3 is compatibility with a third-party fault supervisor. The supervisor is a synthesizable-IP module from Yogitech, a company based in Pisa, Italy (home of the world's most fault-tolerant tower). ARM has modified the Cortex-M3 to work with Yogitech's faultRobust-CPU (fRCPU) supervisor, which couples to the processor core over a special observation port. The fRCPU monitors numerous signals on this port to detect errors. Cortex-M3 systems not needing this safety feature can omit the module and the port. ARM says its modifications to the Cortex-M3 for fRCPU compatibility don't affect the processor's performance or size if the module and port are omitted.

Yogitech, founded in 2000, created its first fault-supervisor module in 2006, for the ARM968 processor. However, that module never reached market. The Cortex-M3—a much smaller processor core intended specifically for MCUs—is a better match for this technology. In 1Q07, Yogitech introduced its first faultRobust supervisors for on-chip memory subsystems and buses, including an fRBUS module for ARM's multilayer AHB. So far, Yogitech has sold three licenses for the memory module, one license for the bus module, and one license for the Cortex-M3 fRCPU module. (It's possible to build fault-tolerant systems without using all the faultRobust modules, if developers take different approaches to fault tolerance.) All of Yogitech's faultRobust licensees are in the development phase of their projects and prefer to remain anonymous at this time.

ARM's modifications to the Cortex-M3 expose 130 internal signals to the fRCPU supervisor. Yogitech determined which signals need monitoring by performing a failure-mode and effect analysis (FMEA) on the Cortex-M3. This type of analysis identifies the parts of a complex logic circuit that are relevant to safe operation. Yogitech refers to its proprietary FMEA technology as faultRobust-FMEA (fRFMEA), and Yogitech must perform this analysis on each different processor core the company wishes to support. Yogitech also needs cooperation from the processor vendor, because the processor requires internal modifications and the addition of the observation port. So far, the Cortex-M3 is the only processor core with a commercially available fRCPU supervisor. Figure 1 shows how the fRCPU tightly couples to the ARM Cortex-M3's optional observation port, officially called the faultRobust Diagnostic Interface (fRDI).

FaultRobust technology is an alternative to other methods of fault detection and tolerance, such as using two identical processor cores running in lockstep. The fRCPU fits on the same chip as the processor and shoulders much of the burden of error checking and correction. It works with single- or multicore CPU designs. By monitoring detailed fault information, the fRCPU allows the system to correct some run-time errors as well as detect them, so it improves both the reliability and availability of the system. Yogitech's failure-mode analysis allows the company to optimize the size of the fRCPU for the target processor, so the supervisor is as small as possible for the functions it performs. Yogitech estimates that a Cortex-M3 processor and fRCPU will require about half as many gates as an alternative design that replicates the Cortex-M3.

Specifically, Yogitech says that an fRCPU optimized for the Cortex-M3 requires about 34% as many gates as the processor does. In contrast, a dual-core redundant design with two Cortex-M3 processors running in lockstep requires 100% more gates for the second core plus 69% more gates for a dual-core comparator, a timeout watchdog, and the special layout precautions necessary to reduce the chances of common-cause failures. (Such a layout would typically have extra guard rings, power-supply separations, and so forth.) Therefore, an fRCPU-based design with a single Cortex-M3 core is only about 50% the size of the dual-core lockstep alternative. This difference saves significant silicon and power.

Safety Standards Vary by Application

Yogitech designed the fRCPU and additional fault-supervisor modules to meet safety standards defined by the International Electrotechnical Commission (IEC). FaultRobust technology addresses an international standard known as the IEC61508 norm. This norm embodies the concept of

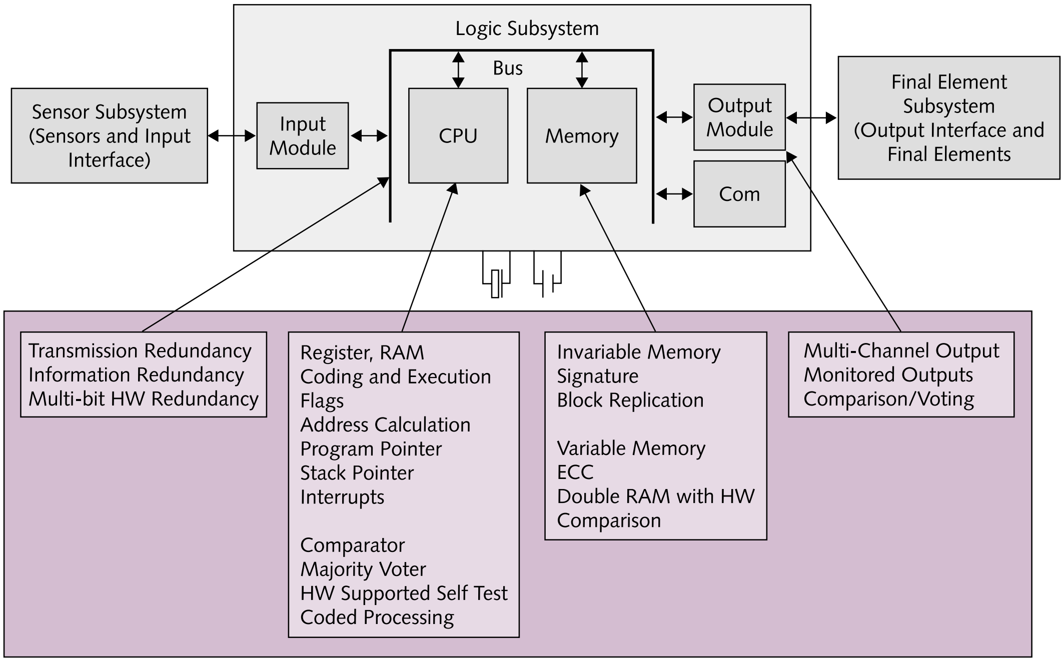

Figure 2. Diagnostics recommended for ICs rated HFT=0 by the IEC61508 norm. The chip must perform these diagnostics in a separate "channel" consisting of different hardware than the CPU and other subsystems under observation. The diagnostic channel may include a redundant processor core, but a smaller supervisor module like Yogitech's fRCPU is sufficient for some safety levels.

hardware fault tolerance (HFT), distinguishing between different levels of tolerance for different applications. Broadly speaking, if it's acceptable for a failed system to merely enter a safe mode (an idle state that cannot cause safety risks), then HFT=0. If a system must tolerate faults and remain available, then HFT>0. Higher HFT numbers define higher degrees of fault tolerance.

For HFT=0, the IEC61508 norm requires ICs to monitor the actions of the CPU and related subsystems in real time by using a diagnostic channel. The term "channel" is somewhat misleading, because it implies a simple conduit. Actually, the channel must perform numerous diagnostics using autonomous hardware blocks. Those blocks must have some degree of hardware diversity and isolation from the subsystems they observe to protect themselves from common-cause failures—such as short circuits, crosstalk errors, voltage fluctuations, and temperature gradients. The "mission channel" is the portion of the subsystem that performs the application functions. Figure 2 shows the various diagnostic techniques that the IEC61508 norm recommends for ICs.

To achieve HFT=0, the IEC61508 norm doesn't mandate ICs to provide full redundancy (e.g., the common method of using duplicate processor cores running in lockstep), but it does favor using diverse hardware in the diagnostic channel. Yogitech's fRCPU is an example of an optimized, tightly coupled diagnostic module that's smaller than a redundant processor core but sufficient to meet the requirements of some IEC61508 safety levels. Figure 3 illustrates these two approaches to achieving HFT=0.

In a sense, Yogitech's solution is also a "dual-core" design, because the fRCPU supervisor contains a processor core, too. However, the fRCPU's processor isn't a duplicate copy of the master CPU. It's a much simpler proprietary processor that can observe the master CPU's instruction stream but cannot drive the master CPU's bus. It's a supervisor, not a redundant slave.

Making Safety-Level Trade-Offs

ARM's Cortex-M3 and Yogitech's fRCPU have been certified to meet Safety Integrity Level 3 (SIL3) of the IEC61508 norm. Table 1 shows how different safety levels correlate with Safe Failure Fractions (SFF) and the HFT. Basically, a higher SIL number indicates higher availability for an embedded subsystem, expressed by the SFF percentage. For automotive-safety systems, SIL2 is considered the minimum level, suitable for antilock-brake controllers. SIL3 is required

	Hardware Fault Tolerance		
Safe Failure Fraction	**0**	**1**	**2**
< 60%	n/a	SIL1	SIL2
60% < 90%	SIL1	SIL2	SIL3
90% < 99%	SIL2	SIL3	SIL4
≥ **99%**	SIL3	SIL4	SIL4

Table 1. The IEC has defined these standards for fault tolerance in embedded subsystems. The higher the Safety Integrity Level (SIL), the higher the subsystem's availability. In this table, purple type highlights the SIL ratings considered suitable for automotive-safety systems. (Data source: Yogitech)

Price & Availability

Final RTL for ARM's Cortex-M3 Release 2.0 processor core is scheduled for availability in 2Q08. Final RTL for Yogitech's faultRobust memory- and bus-supervisor modules is available now. The faultRobust supervisor module for the Cortex-M3 (fRCPU ARMcm3 Release A1.1) is scheduled for availability in 3Q08. Neither ARM nor Yogitech publicly discloses licensing fees. For more information, visit:

- *www.arm.com/products/CPUs/ARM_Cortex-M3.html*
- *www.fr.yogitech.com*

for automotive-stability controllers and drive-by-wire networks. SIL4 is an even higher level that requires redundant chips, but it usually doesn't apply to automotive systems.

The IEC defines the SFF as the ratio of the average rate of "safe failures" plus detected "dangerous failures" of the subsystem to the total average failure rate of the subsystem. A "safe failure" is one that doesn't cause the subsystem to enter a hazardous state that might lead to a safety malfunction. A "dangerous failure" is the opposite. (Keep your insurance paid up.)

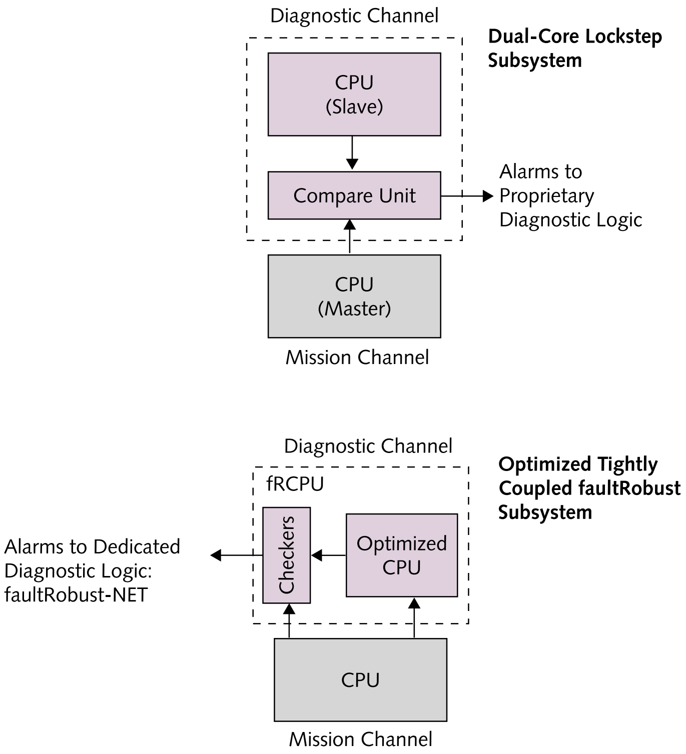
Diagnostic Channel
Dual-Core Lockstep Subsystem
CPU (Slave)
Compare Unit
Alarms to Proprietary Diagnostic Logic
CPU (Master)
Mission Channel
Diagnostic Channel
fRCPU
Optimized Tightly Coupled faultRobust Subsystem
Alarms to Dedicated Diagnostic Logic: faultRobust-NET
Checkers
Optimized CPU
CPU
Mission Channel

Figure 3. A common way to design HFT=0 systems is to use redundant processor cores running in lockstep (top). However, this method doubles the die area and power consumption of the CPU subsystem, and it requires extensive layout modifications and interconnection precautions to avoid common-cause failures. A smaller diagnostic module, tightly coupled to the CPU, can provide enough diversity and safety while saving silicon and power (bottom). The cost savings become even more significant in multicore designs.

Yogitech's fRCPU supervisor has been certified as SIL3-compliant by Tüv Süd, an international engineering-services organization that performs safety testing and IEC certification.

In addition to the fRCPU supervisor, Yogitech sells other faultRobust IP modules that monitor the memory subsystem, on-chip peripherals, and on-chip buses. A private on-chip diagnostic network called faultRobust-NET (fRNET) ties all the faultRobust modules together, collecting and filtering any error messages generated. All faultRobust modules are certified by Tüv Süd for SIL3 compliance and are synthesizable-IP blocks that developers can drop into their chip designs. The fRCPU supervisor is the most complex and important faultRobust module. Figure 4 is a block diagram of the fRCPU.

The fRCPU checks for transient and permanent faults by comparing data from the Cortex-M3 with predicted values. To predict these values, the fRCPU's own processor core must be able to execute many of the same instructions and have duplicate copies of many of the same registers as the Cortex-M3. So, in effect, Yogitech has created an ARM-compatible processor. However, the fRCPU processor is much simpler than the Cortex-M3 and supports only the subset of ARM instructions and registers that Yogitech's failure-mode analysis has deemed critical. Unlike a redundant dual-core design, the fRCPU cannot substitute if the Cortex-M3 fails altogether. But because the fRCPU is based on completely different RTL—with its own execution logic, pipeline design, and clock timing—it's more immune to common-cause failures than a redundant processor would be.

A scoreboard in the fRCPU keeps track of the Cortex-M3's output and compares it with data from the same operations performed at different times. This function checks for data consistency. To detect I/O errors, the fRCPU checks signals and events on the observation port for protocol consistency. A timeout counter (somewhat like a watchdog timer) guards against software problems that might disable the master processor. In addition, the timeout counter can detect problems with the fRCPU itself, providing a self-check mechanism.

Yogitech says the fRCPU can detect faults in the master processor even more quickly than a redundant processor running in lockstep can. Typically, a redundant processor won't notice a fault until erroneous output appears on the conventional I/O bus. In contrast, the fRCPU constantly monitors the CPU's internal signals over its dedicated observation port, so faults are detectable immediately.

Detecting Errors and Controlling Failures

If a monitored value veers from the expected value, the fRCPU generates an error message. The message includes detailed fault information, such as the specific operation or register bank involved in the fault. Using this information, the system can control the failure in a safe manner. Of course, it's possible for the fRCPU to suffer a fault as well, and its fault could mask a CPU fault. To prevent this calamity, the fRCPU has self-checking circuits that assure the integrity of its diagnostic functions. All these features enable the fRCPU to meet the SIL3 rating for automotive-safety systems.

Consider the example of an antilock-brake controller in an automobile. It tries to keep the wheels from completely locking up and sending the car into an uncontrollable skid. To stop the car in a straighter path, the brake controller must rapidly apply or release the brakes on each wheel of the car in response to sensor input from the wheels. Each wheel must brake independently. When a wheel locks up, a sensor signals the brake controller to release the brake on that wheel. When a wheel rotates beyond a defined threshold, the sensor signals the controller to reapply the brake on that wheel. The result is the rapidly throbbing brake pedal but straight-line panic stop that is familiar to drivers of these cars.

Suppose a value in a CPU register wrongly changes. The anomaly may have been caused by a transitory hardware fault in the processor or even a soft error triggered in the circuit by cosmic rays from outer space. If the anomaly goes undetected, the brake controller's software might not apply the brakes when needed or might apply the brakes when not needed. Either way, the situation could be dangerous.

Yogitech's fRCPU, monitoring the processor in real time, would notice the anomaly and raise an error-specific alarm. Responding to this detailed error message, the system can automatically force the brake actuator into a safe state and either reset or interrupt the processor. Even if the error condition persists, the system can maintain a safe state. The block diagram in Figure 5 illustrates how such a system might look.

Alternative Fault-Tolerant Designs

HFT=0 is considered sufficient for most automotive systems, but other critical systems may require higher hardware fault tolerance (HFT>0). An example is an aircraft-control system. Achieving HFT>0 requires multiple channels—fully independent redundant processor cores or multiple redundant chips. Naturally, such systems are more expensive and consume more power. Higher HFT levels are also required for systems that simply cannot tolerate the halt of a critical function, such as drive-by-wire control in an automobile. Soft errors are becoming a larger problem with chips manufactured in deep-submicron fabrication processes, and those errors could cause an unacceptable loss of critical system availability.

To achieve HFT>0, the IEC61508 norm requires each channel to have dedicated diagnostics. One channel cannot do double duty by performing diagnostics on the other. Yogitech's faultRobust technology can meet the needs of these fault-tolerant systems by using two processor cores and two fRCPU supervisors. Figure 6 shows an example configuration. The twin modules can provide detailed diagnostics covering many possible failure scenarios. This configuration can rapidly switch from one channel to another or reallocate tasks between channels, thereby maintaining system availability.

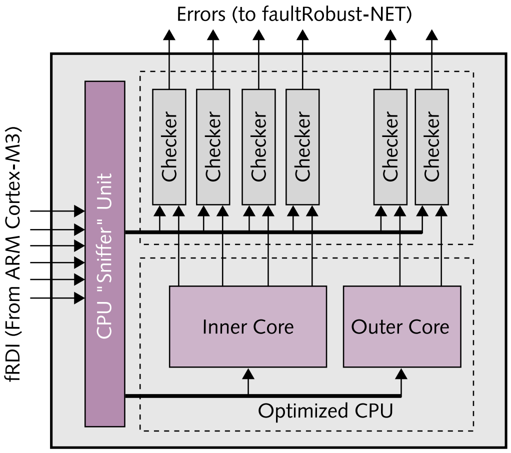

Figure 4. Yogitech faultRobust-CPU (fRCPU) block diagram. The fRCPU contains its own CPU core, but it's highly optimized for monitoring the primary CPU for faults and is much smaller than a duplication of the primary CPU. The fRCPU tightly couples to the primary CPU over the faultRobust Diagnostic Interface (fRDI). This observation port exposes critical internal signals from the CPU. At top, the fRCPU connects to a private on-chip network (fRNET) that links to other Yogitech faultRobust supervisors, such as the modules for on-chip peripherals, memory, and buses.

Even a dual-channel system isn't perfect. A famous example of fault *intolerance* is the European Space Agency's Ariane-5 disaster of 1996. Less than a minute after launch, the rocket's onboard guidance computer tried to convert a 64-bit value into a 16-bit value, but the software botched the conversion. The computer misinterpreted the resulting error message as navigational input and ordered the rocket engines to make an

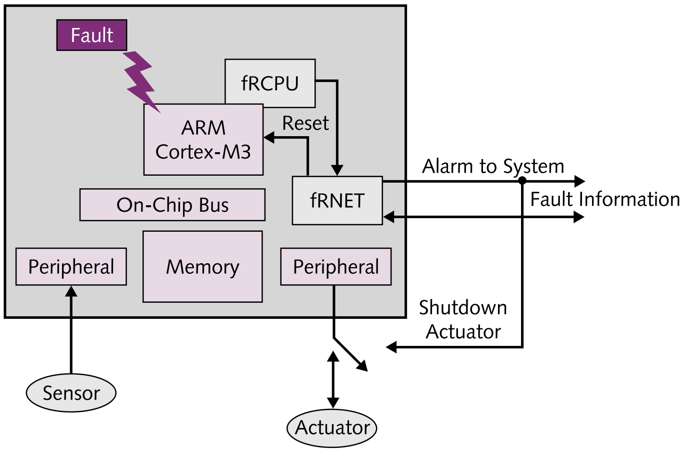

Figure 5. When Yogitech's fRCPU supervisor detects a CPU fault, it generates an error message. The fRNET module sends a global error signal that can switch the actuator into a safe state and immediately reset the CPU. After reset, the CPU can read the diagnostic information stored in the fRCPU to determine what happened. If the fault was transient, the CPU can switch the actuator back to its normal operating mode, preserving system availability.

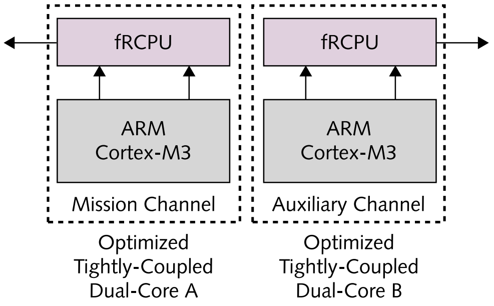
fRCPU
fRCPU
ARM Cortex-M3
ARM Cortex-M3
Mission Channel
Auxiliary Channel
Optimized Tightly-Coupled Dual-Core A
Optimized Tightly-Coupled Dual-Core B

Figure 6. For higher degrees of fault tolerance (HFT>0), two processor cores and two of Yogitech's fRCPU supervisors can form a dual-channel system. This approach is required for more-critical systems, such as drive-by-wire control in an automobile.

impossible course correction. The rocket—fortunately unmanned—went wild and self-destructed. The $370 million payload of four U.S. scientific satellites was lost.

The Ariane-5 actually had a redundant guidance controller that was supposed to handle such problems. Unfortunately, it ran the same software as the primary controller, so it suffered from the same bug. Although Yogitech's fRCPU supervisor can handle some software errors, it's designed primarily to detect hardware faults. One solution can't solve all problems. Designing a truly fault-tolerant system demands rigorous attention to both hardware and software development.

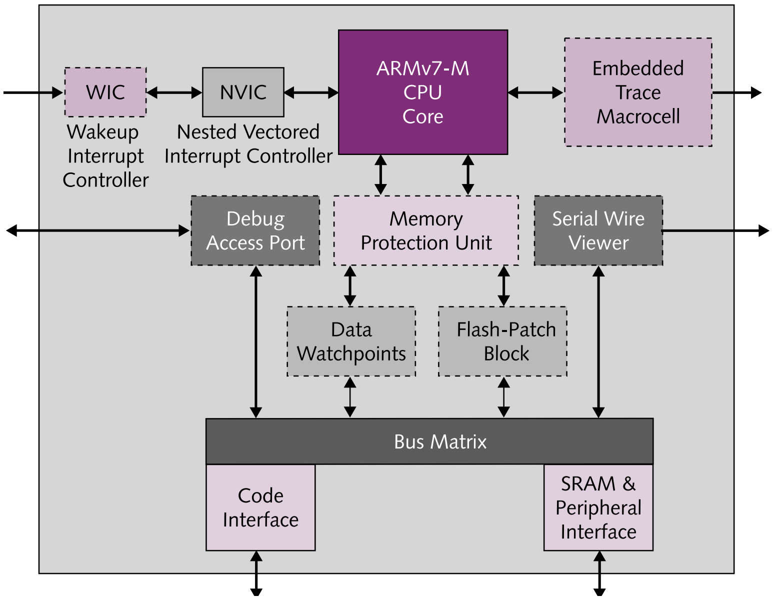

Figure 7. Dotted lines indicate configurable elements of the ARM Cortex-M3 Release 2.0 processor core. A significant reason for the small size of this 32-bit processor is its unique implementation of the ARMv7-M instruction-set architecture, which supports only the Thumb and Thumb-2 subsets of 16-bit instructions. The Cortex-M3 cannot execute traditional 32-bit ARM instructions.

Subtle But Welcome Improvements

Figure 7 is a block diagram of the Cortex-M3 Release 2.0 processor core, with configurable features indicated by dotted lines. When configured with 16 interrupts, a debug access port, all hardware breakpoints, the instrumentation-trace macrocell, memory-protection unit (MPU), and embedded-trace macrocell (ETM), the Cortex-M3 has a little more than 80,000 gates. Size-conscious developers can save gates by omitting some of these features. The MPU has 17,000 gates; the ETM has 7,500 gates; and the full complement of breakpoints requires about 4,000 gates.

Although the Cortex-M3 isn't as configurable as processor cores from ARC International, MIPS Technologies, and Tensilica, it's reasonably flexible for a processor that lacks a customizable instruction-set architecture. Notably, a minimal configuration of the Cortex-M3—even with its full bus matrix and a minimum complement of interrupts—is about 20% smaller than an ARM7TDMI-S, long a staple of 32-bit MCUs.

The Cortex-M3's enhancements are subtle but appropriate for ARM's customers. Luminary Micro and STMicroelectronics are already producing Cortex-M3 MCUs. A Norwegian startup named Energy Micro is reportedly developing even lower-power MCUs based on the core. And on May 8, Zilog announced that it has licensed the Cortex-M3 to expand its line of ARM-based 32-bit MCUs.

As the Cortex-M3 continues gaining traction in the marketplace, compatibility with Yogitech's faultRobust technology will extend the processor's reach into automotives, medical devices, and some military applications. Curiously, ARM hasn't announced similar compatibility for the Cortex-R4F, a more powerful processor core that ARM introduced for the automotive market in 2006. (See *MPR 10/30/06-01*, "ARM Thumbs a Ride.") We won't be surprised if that upgrade comes later. ◇