Fujitsu Starts New Embedded CPU Family

New FR20 First in a Line of 32-bit RISC-like Microcontrollers

by James L. Turley

Bridging the gap between its traditional CISC microcontrollers and more expensive RISC processors, Fujitsu has announced the FR20, the first in a new line of 32-bit embedded processors promising RISC performance at microcontroller prices. Paralleling similar developments from Hitachi and NEC, Fujitsu created the FR family from scratch specifically for embedded applications.

Like the Hitachi SH7000 family and NEC V800 series processors *(see **070802.PDF**)*, the FR family combines a completely new 32-bit processor core with a smattering of on-chip I/O. And, like those competitors, the new processor architecture borrows from RISC principles to produce a small streamlined core with a compact instruction set. Devices in this market are aimed at high-volume applications, each with a different mix of performance, memory, and peripherals.

Although not architecturally compatible with either Fujitsu's low-end F^{2}MC family of 4-, 8-, and 16-bit MCUs or the SparcLite series of RISC microprocessors *(see **080804.PDF**)*, the FR architecture borrows techniques from both. At its core, the FR20 is a streamlined 32-bit engine with impressive integer performance. The chip adds peripheral functions like on-chip A/D and D/A converters that make it more suitable for embedded control than desktop computing.

The inaugural FR20 promises to be the first of a series, with follow-on parts to be geared toward audio/video processing, automotive control, and high-end printer controllers. Fujitsu is marketing the FR only in Japan at this time, with introduction in the U.S. and Europe no sooner than 1995.

Fujitsu Merges RISC and CISC

Fujitsu's approach is neither to extend a successful 8- or 16-bit architecture to 32 bits nor to water down an existing microprocessor, such as its own SparcLite. Rather, it has developed a new CPU architecture with a new instruction set to match. The programming model features an orthogonal 32-bit general-purpose register set. All registers are treated equally, with any one acting as either source, destination, or address pointer.

The 32-bit registers permit straightforward manipulation of standard operand sizes. Smaller 4-, 8-, and 16-bit operands are right-aligned to the least-significant byte of any register. The 32-bit registers also make linear memory addressing easier. There are only 16 registers rather than the customary 32, however, making them a limited commodity. Architecturally, the FR uses a linear 4G address space, although the initial FR20 implementation brings out only the low-order 24 bits.

Nearly all the FR's 168 instructions manage to fit in a single 16-bit word. The only exceptions are coprocessor operations and those that take large (20-bit or 32-bit) immediate operands. The compact instruction size was a key design goal for Fujitsu's engineers. Keeping instructions within two bytes reduces overall code size. The compact encoding size constrains all instructions to two operands, using the destination as a second source operand, like most CISC processors. Apart from reducing the number of registers needed for a given operation, two-operand instructions are simpler to encode than three-operand types, contributing to the compact instruction set and reducing CPU core size.

The instruction set itself, listed in Table 1, contains load/store, arithmetic, logical, and bit-manipulation operations. Variations of most instructions operate on the processor's Status Register, Stack Pointer Registers, and other special-purpose registers. Branch instructions can be made conditional on various flags in the Status Register.

Memory Addressing Aids Code Density

Another place where Fujitsu departs from RISC principles is the wide range of memory addressing modes. The basic load and store instructions have postincrement and predecrement memory addressing options, easing the task of dumping multiple registers to memory or storing a constant value. These options are accessed by using two special registers in the otherwise general-purpose register set, R13 and R15. When used for memory addressing, register R13 automatically increments itself and R15 decrements.

Another useful enhancement is the ability of logic instructions (AND, OR, EOR, etc.) to modify memory operands directly. These instructions bypass the internal register set completely, freeing the programmer from the usual tedium of loading an operand into a register only to store it back out. Unfortunately, the logical operations do not support the auto-increment/decrement addressing modes, so inverting a block of memory, for example, still requires some loop manipulation.

In addition to the usual load and store operations, the FR supports direct memory-to-memory moves. When used with registers R13 and R15, x86-style string moves are possible—a very un-RISC-like but useful feature. Although no faster than the in-and-out approach, memory-

to-memory transfers save a register and are commonly used in control applications.

Fujitsu enhanced the FR instruction set in other ways, such as adding procedural ENTER and LEAVE instructions, as well as multiple-register load and store instructions for a quick way to change machine state in a multitasking environment. An indivisible byte-exchange instruction permits reliable semaphore operations to external memory. Without such atomic operations, system designers requiring indivisible cycles are forced to develop their own hardware interlocks. These features help make FR object code denser than that of a true RISC processor.

The FR architecture also supports internal coprocessors, and certain opcodes have been reserved for coprocessor extensions. This leaves the door wide open for future enhancements, with new members of the FR family executing multiple instructions in parallel.

Modular Design Aids Integration

Fujitsu has evidently taken a page from Motorola's successful embedded controller plan and made the FR core modular enough to integrate with other standard cell designs in an ASIC. The FR processor core could be used as a megacell that the customer can use to develop very highly integrated embedded control devices. Fujitsu can also quickly develop derivative products simply by adding and subtracting peripheral functions directly on the same die as the FR core, much as Motorola has done with the 68300 family.

Figure 1 shows how the initial FR20 implementation includes an effective mix of on-chip peripheral and core functions. The separate 1K instruction and 1K data caches are large enough to be useful while keeping costs down. As it is, the caches eat up more silicon area than the CPU itself.

For I/O control, a two-channel D/A converter and an eight-channel successive-approximation A/D are included, along with an integral DRAM controller, chip-select logic, an eight-channel DMA controller, four UARTs, an up/down counter, timers, and a set of capture-and-compare registers for responding to external conditions. Twelve external interrupt inputs (plus a nonmaskable interrupt) round out the peripheral block.

The most interesting extension to the base FR architecture is the FR20's peripheral accelerator interface, or PAI. The PAI acts like an on-chip DMA controller, shuttling data between the FR20's peripheral sections and external memory. PAI transfers are initiated by two special instructions, LDRES (Load Resource) and STRES (Store Resource).

The FR20 has a fairly straightforward external bus interface. Nonmultiplexed address and data buses are managed by simple control signals. A single read strobe initiates input operations while four byte-enable lines indicate when output data is valid. Cycles are terminated with an active RDY line. Simple bus request/grant logic is also supplied.

Although the FR20 has a 32-bit external data bus, it can operate in a 16-bit mode. This is a good example of the FR20's cost-driven design philosophy. By keeping the instruction set narrow, a pair of standard byte-wide ROMs can feed the CPU at full speed. Even using the 32-bit bus and more (or wider) ROMs, designers reap the benefits of the compact instruction set in smaller overall code size.

Instruction	Description
Logic	
CMP	Compare
AND, ANDH, ANDB	Logical AND (Halfword, Byte)
OR, ORH, ORB	Logical OR (Halfword, Byte)
EOR, EORH, EORB	Logical Exclusive OR (Halfword, Byte)
BANDL, BANDH	Bitwise Logical AND (Low, High)
BORL, BORH	Bitwise Logical OR (Low, High)
BEORL, BEORH	Bitwise Logical Exclusive OR (Low, High)
BTSTL, BTSTH	Bit Test (Low, High)
LSL, LSL2	Logical Shift Left
LSR, LSR2	Logical Shift Right
ASR, ASR2	Arithmetic Shift Right
Arithmetic	
ADD, ADDC, ADDN	Add (with Carry, Negated)
SUB, SUBC, SUBN	Subtract (with Carry, Negated)
MUL, MULU, MULH	Arithmetic Multiply (Unsigned, Halfword)
DIV0S, DIV0U, DIV1	Arithmetic Divide (Signed, Unsigned)
DIV2, DIV3, DIV4S	
Load/Store	
LDI	Load Immediate Value (8, 20, 32-bit)
LD, LDUH, LDUB	Load Memory to Register (Unsigned Halfword, Byte)
LDM0, LDM1	Load Multiple Registers
ST, STH, STB	Store Register to Memory (Halfword, Byte)
STM0, STM1	Store Multiple Registers
MOV, DMOV,	Move Register/Memory to
DMOVH, DMOVB	Register/Memory
XCHB	Exchange Register with Memory
Flow Control	
JMP	Unconditional Jump
CALL	Call Subroutine
RET, RETI	Return from Subroutine
INT, INTE	Interrupt
Bcc	Branch on Condition
ENTER	Begin Procedure and Allocate Heap
LEAVE	Exit Procedure and Deallocate Heap
System Control	
ANDCCR	Logical AND to Condition Code Register
ORCCR	Logical OR to Condition Code Register
EXTSB, EXTUB,	Extend Operand (Signed, Unsigned,
EXTSH, EXTUH	Byte, Halfword)
COPOP, COPLD,	Coprocessor Operations
COPST, COPSV	
LDRES, STRES	Load, Store Resource
NOP	No Operation

Table 1. The FR instruction set provides support for 8-,16-, and 32-bit operands for most instructions.

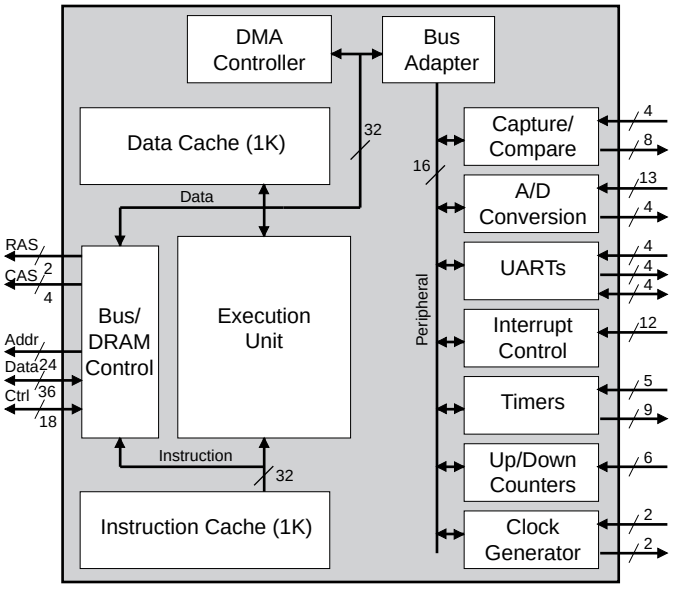

Figure 1. The FR20 mixes a completely new RISC-like CPU core with a mixture of general-purpose peripheral functions.

Caches Speed Execution Time

Significant chip area is devoted to the dual caches in the FR20. The five-stage instruction pipeline keeps the execution unit filled even during stalls or cache misses. Caches make deterministic execution timing problematic, though, and can confuse debugging. Either cache can be locked on a line-by-line basis so critical code or data is never pushed out. The two-way set-associative structure holds eight 16-bit words in each cache line.

Assuming the necessary code is already in cache, nearly all instructions execute in one clock cycle. Exceptions are logic operations on external memory operands, which take three clocks, and subroutine calls, which take two. Integer multiplication and division functions vary, with 16-bit multiplies taking three clock cycles; 32-by-32-bit multiplication takes five clocks. As in many CPUs, integer division is a stepped process, so total execution time depends on the data.

The overall die size of the FR20 is 70 mm^2. Of that, only 5.4 mm^2—less than 8%—is devoted to the CPU core, leaving the remaining space for peripheral functions, RAM cells, and I/O pads. The FR20 is fabricated in a 0.8-micron two-layer-metal process. From a nominal 5–V supply, the FR20 runs at 25 MHz, dissipating approximately 600 mW worst-case. Turning the voltage down decreases the maximum clock frequency to 16 MHz at 3.3 V. The design is not static, so clock-stopping is not an option, although the FR20 does have low-power sleep and stop modes.

	FR20	SH7604	V851	ARM700
Frequency (5V)	25 MHz	28.5 MHz	25 MHz	25 MHz
Dhrystone MIPS	28 MIPS	25 MIPS	29 MIPS	30 MIPS
Instruction Cache	1K	4K	n/a	8K
Data Cache	1K	unified	1K RAM	unified
Die Area	70 mm2	82 mm2	42 mm2	46 mm2
IC Process	0.8µ, 2M	0.8µ, 2M	0.8µ, 2M	0.8µ 2M
Voltage (Vdd)	3.3 V–5 V	3.3 V–5 V	2.0 V–5 V	3V–5 V
List Price	<$20	>$25	$19	$35
in Quantities of:	25,000	25,000	100,000	10,000

Table 2. The FR20 offers good performance in a small package, with a mix of I/O equivalent to the SH7032.

Price & Availability

The FR20 is the first FR-architecture device to be released, but it will initially be available only in Japan. Quantity pricing has not been disclosed.

For more information, contact M. Somasundaram at Fujitsu Microelectronics at 408.456.1075. In Japan, contact Makoto Awaga at (81)44.754.3411, or e-mail *awaga@gmd.ed.fujitsu.co.jp*.

The current design uses a fairly conservative fabrication process for a company with Fujitsu's resources. This is a typical strategy for new devices in this price/performance range. According to the MPR Cost Model *(see **081203.PDF**)*, the estimated manufacturing cost for the FR20 is $14. The company plans two process shrinks: the first to Fujitsu's 0.5-micron CS-50 process *(see **080504.PDF**)*, with an attendant speed increase to 50 MHz; the second down to 0.35-micron, with 80-MHz operation. Scheduled sampling dates for these versions are not available.

More Bang for the Buck?

The optimized CPU, dual caches, and internal five-stage pipeline keep the FR20 running at a brisk pace. Fujitsu claims a peak execution rate of 28.4 Dhrystone MIPS at 25 MHz. This speed places it at roughly the same performance level as the more expensive 68040 at the same frequency. The two processors, however, are clearly aimed at different applications.

A more appropriate comparison might be with NEC's V851 *(see **081303.PDF**)*, Hitachi's SH7604, or the ARM700. All of these are similar, highly integrated embedded processors, which, like the FR20, profess to be based upon RISC principles. As Table 2 shows, all are roughly the same size and price, and all offer a mix of I/O on the chip. Below the surface, there are several architectural differences among these competitors.

Like the FR20, the 7604 makes do with 16 registers. Doubtless this was done to shave an addressing bit from the already compact 16-bit instruction sets. But where the FR20 has more than 75 instructions, the 7604 has only 56. Both ARM and V8xx have a larger, 32-register set and use longer, 32-bit opcodes (although the V8xx mixes in some 16-bit instructions as well).

Another worthy competitor, Motorola's 68300 fam-

ily, also has 16 registers (though only 8 are useful for data manipulation) with a variable-length instruction set that frequently tacks extension words and operands onto its 16-bit instructions. Fujitsu has done a good job squeezing useful functions into a tight code space.

Even the internal feature mix between the FR20 and the 7604 is roughly equal. Both offer on-chip cache, although Hitachi merges the 7604's into a unified 4K. Both include DMA engines, DRAM controllers, interrupt controllers, and timers. The 7604 drops the A/D functions of earlier 703x devices. For its part, NEC chose to emphasize smaller size, and therefore cost, by leaving most I/O functions off the V851. Unlike other V8xx chips, the V851 does not offer on-chip floating-point support.

Although Fujitsu touts the FR20 as a "high performance RISC processor," there are many deviations from the true RISC path made in the name of practical usefulness. In the end, it may be an academic argument. The FR20 provides a competitive mix of functions and performance, particularly emphasizing compact code size. Many architectural decisions have to be made in developing any new processor family; for the FR20, Fujitsu has chosen to let all the decisions fall on the side of lower cost. RISC architectural issues aside, the FR20 has the makings of a strong competitor in the increasingly crowded embedded processor arena.

It takes more than strong silicon to make a winner these days. Any new processor starts off handicapped by lack of development tools, installed software, and customer familiarity. With so many low-cost CPUs to choose from, why bet on an unknown newcomer? Motorola's 68300 family has the advantage of a 15-year legacy of software and development tools. ARM has an installed base of Newton users. Even if price were not an issue, many developers might opt for the tried-and-true over the uncertain benefits of a new product family. For the time being, engineers outside of Japan are spared those decisions, as Fujitsu is not yet marketing the FR family outside of its home country.

With the PDA market facing an uncertain future, and next-generation Nintendo and Sega processors locked up, many major processor vendors are left with vast development efforts to amortize and little observable market share to show for it. Consumer electronics provides many opportunities; perhaps embedded processors like the FR20 will carve out a niche in embedded applications yet to come. ♦

