ColdFire Doubles Performance With v4

Changes to Core Design, Clock Speed Accelerate Motorola's Midrange CPU

by Jim Turley

Version 4 of the ColdFire architecture will double the performance of today's ColdFire v3 chips through a combination of relatively advanced design tricks and the less subtle effects of better process technology. Motorola continues to bring the performance of its ColdFire processor line up to par with that of competing architectures: the latest wrinkle in the ColdFire roadmap shows the chips reaching 200 Dhrystone MIPS in the middle of next year.

EMBEDDED PROCESSOR FORUM

At the Embedded Processor Forum earlier this month, Motorola chief architect Joe Circello described new branch-prediction and branch-folding features, a handful of new instructions, and design methods that allow Motorola to produce customer-specific ColdFire parts in less time than before.

More Process Than Processor

ColdFire is as much a process as it is a product for Motorola. In addition to being the successor to the 68000 dynasty, ColdFire is the only chip family in the company's arsenal that is entirely synthesized from start to finish. The ColdFire CPU cores, internal buses, caches, and peripherals are all synthesized from myriad designs carried out at Motorola's far-flung design centers. Even on-chip RAM and ROM are compiled. All logic uses edge-triggered D flip-flops for ease of verification. This fierce reliance on synthesis allows Motorola to create new ColdFire variations in a relatively short period of time, reacting to consumer demand more rapidly that it may otherwise have been able to.

This strategy starkly contrasts with that of the company's 68300-series chips, which were frequently made up of hard macrocells fitted together around a physical bus (the IMB). Although the 68300's design method allowed for customization, it lacked the flexibility of a synthesized design flow and the advantages in testing unfinished chips.

Mnemonic	Description	New to v4
MOVE3Q	Move 3-bit quick	●
MOVSX	Move with sign extend	●
MOVZX	Move with zero extend	●
MOVE.B	Move byte	
MOVE.W	Move word (16 bits)	
CMPI.B	Compare immediate byte	
CMPI.W	Compare immediate word	
Bcc (disp32)	Branch with 32-bit displacement	
BRA (disp32)	Branch with 32-bit displacement	
BSR (disp32)	Branch to subroutine	
SATS	Signed saturate	●
TSET	Test and set operand	

Table 1. Apart from its revised pipeline and new branch prediction, ColdFire v4 adds four new instructions and reinstates several 68020 instructions that were not present in ColdFire v3.

Version 4 Adds New Instructions

With version 4 of the core, Motorola is starting to turn ColdFire down a different path than the one well worn by the 68000. The architecture is now picking up new instructions that the 68K never had and rejiggering old instructions to work in new ways. The enhancements are meant to improve code density or improve performance with small operands, if not both.

The new ColdFire v4 instructions are listed in Table 1. The MOVSX (move with sign-extension) and MOVZX (move with zero-extension) instructions are entirely new, combining the features of the previous MOV and EXT instructions. Combining these operations into a single instruction saves code space and speeds the operations as well. All ColdFire v4 instructions execute in a single cycle, which is much faster (and more predictable) than the original 680x0 family from which they are descended. It's also faster than ColdFire v3, where reads from memory took three to four cycles.

The MOVE3Q (move 3-bit data quick) instruction is similar to the 68K ADDQ (add quick) and SUBQ (subtract quick) instructions, but it takes a 3-bit immediate operand from the tail of the instruction and stores it in a destination register or memory location.

The signed saturate (SATS) instruction, also new, adds the first taste of saturating arithmetic to the 68K since its inception eons ago. This instruction will saturate a signed 32-bit value in one of the data registers, depending on the condition codes.

Although ColdFire v4's arithmetic instructions do not support saturation intrinsically, the SATS instruction provides much the same effect without having to duplicate all of the existing arithmetic operations with new opcodes that support saturation. The separate instruction also allows programmers to forestall saturating intermediate results until desired.

The other seven "new" instructions in ColdFire v4 are actually 68020 instructions that have been reinstated. The three branch instructions with 32-bit displacements, for example, cure a longstanding problem with ColdFire that forced programmers to use short branches to long-displacement jumps in order to reach the other side of the address map. ColdFire was similar to Hitachi's SuperH architecture in this regard, although Hitachi has never adequately addressed the problem.

New MAC Adds Fractions

Motorola took a half-step toward adding floating-point capability to ColdFire with its new multiply-accumulate (MAC) unit. In addition to standard integer arithmetic, ColdFire v4 can now multiply and accumulate numbers in a signed "fixed-point fractional" format that extends the dynamic range and precision of the chip without the burden of full IEEE-754 floating-point math.

Motorola's fractional format is similar in concept to the Q-format used in TI's fixed-point DSP chips, Siemens' TriCore (see MPR 11/17/97, p. 13), and some microcontrollers. Some of Motorola's 56K DSP and 68HC16 chips also use the fractional format. Outside of Motorola's own devices, however, there is no hardware support for this data format.

ColdFire's datapath is made more complex by the addition of fractional data, but it's not nearly as complex as a full FPU would be. As with Q-format numbers, Motorola's fractional values can be added and subtracted using normal integer instructions.

Branch Prediction Speeds Pipeline

Some of the credit for ColdFire v4's performance improvement must go to its branch handling. The v4 core implements a branch-caching and branch-folding mechanism that has not been seen on a 68K chip since the 68060, another Joe Circello design (see MPR 11/18/92, p. 14).

The new design uses a combination of a branch target cache (BTC) and a branch history table (BHT) to avoid as many bubbles in its execution pipeline as possible. The BHT is a 128-entry hash table; each entry holds the two-bit prediction state. Index entries into the table are hashed from their physical addresses. The BTC contains the address of a branch instruction, the address of its target, and some additional control fields.

ColdFire's BHT uses the familiar two-bit prediction system, where each branch is tagged as strongly or weakly taken or not taken. Every correct prediction reinforces the prediction, while an incorrect prediction weakens it. It takes two consecutive mispredictions to reverse the direction (taken or not) of the prediction.

If ColdFire v4 fetches a branch instruction that "hits" in the BTC, its target address is immediately routed to the address-generation unit, and the fetch stream is redirected, as Figure 1 shows. Assuming the prediction was correct, this results in a zero-latency branch. Even the nominal latency of the branch instruction itself is removed from the pipeline (a technique called branch folding), an ideal result.

If the branch is not found in the BTC, ColdFire v4 hashes its address to index into the 128-entry BHT. This process takes one cycle. If the branch is mispredicted, the penalty is eight cycles.

Subroutine returns also gain a new feature. ColdFire v4 maintains a four-entry LIFO (hardware stack) that records the address of the instruction following the most recent subroutine call (JSR or BSR) instruction. On the surface, this would seem to duplicate the role of the software stack, and it does. Because the LIFO is smaller and kept on chip, however, its contents can be fetched much more quickly than an address on the stack, potentially cutting several cycles off of subroutine-return latency.

The only drawback to this scheme is reality. In actual programs, assembly-language programmers often modify the return address on the stack to redirect program flow. Without consistency checks between the stack and the on-chip LIFO, ColdFire v4 would run a very good chance of crashing most programs. Thus, the LIFO must be checked against the contents of the stack on every RTS (return from subroutine) instruction. The comparison is done at the same time that the (predicted) return address is routed from the LIFO to the address-generation unit. If the predicted address is correct, RTS latency is only two clock cycles. If the stack has been altered and the predicted return address is now stale, latency is 10 cycles.

Price & Availability

The first ColdFire v4 chip is due to begin sampling in 2Q99 at 150 MHz. The chip, which has not been named, is expected to be priced in the $20–$30 range.

For more information, contact Motorola (Austin) at 512.895.2134 or visit *www.motorola.com/ColdFire.*

Some Instructions Execute Early

Circello detailed some other changes to the ColdFire core during his presentation at the Forum. ColdFire v4 has separate instruction and data buses (Harvard architecture), a first for ColdFire. The dual buses are mandatory now that ColdFire can execute one instruction every clock cycle.

ColdFire v4 is also more aggressive about executing and retiring instructions early. Previous ColdFire chips (in fact, most microprocessors) execute instructions only in the

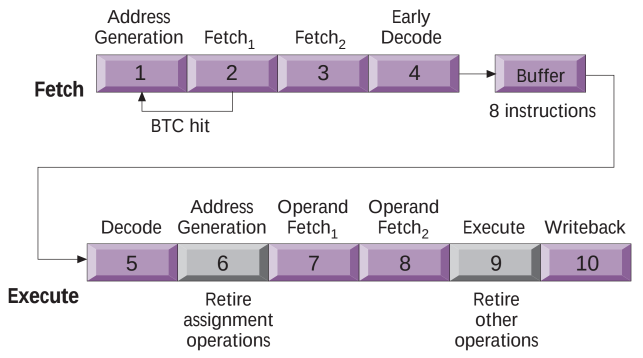

Figure 1. ColdFire v4 breaks its pipeline into fetch and execute portions, separate by an eight-instruction buffer. Many operations can be executed early, in the operand-address-generation stage, leaving the full execute stage for more complex operations.

execute stage of their pipeline. ColdFire v4 has the ability to execute instructions two stages earlier in the pipe, reducing latency and freeing resources for other instructions. By taking advantage of logic in the operand-address-generation (OAG) stage, ColdFire v4 retires many instructions before they ever reach the execute stage.

Some instructions stop in the OAG stage, some stop in the execute stage, and some can be executed in either place. Predictably, address-generation instructions (such as LEA #$DC00, A0) always complete in the OAG stage, but so do many assignment instructions. Instructions that need both address generation and an arithmetic or logic operation (such as ADD [A0], D4) must wait for the execute stage.

Last, ColdFire v4 is smart about collapsing commonly used constructs into a single operation. If two instructions will execute in different stages and have no dependencies, they will execute together in a single cycle. This "instruction folding" is ColdFire's first move toward superscalar dispatch.

New Debug Features Work Independently

As part of the cleanup work involved in any microprocessor design effort, Motorola also enhanced ColdFire's debug support, creating "revision C" of the debug logic. (Motorola's ColdFire cores and debug logic are independent entities that can be mixed in any combination.) The new debug circuitry adds three more instruction breakpoint registers, two address breakpoint registers, and a second set of data and mask registers. As before, emulators access these features through a three-pin serial port.

Perhaps most significant, ColdFire's debug handler can now be interrupted, unlike previous revisions of the debug unit. This improvement was made at the request of real-time developers, who previously had to watch in horror as their disk drive, print head, or other motion-controlled device flailed away while the processor handled a debug event, blithely ignoring other interrupts.

Motorola architect Joe Circello describes the changes made to the ColdFire v4 core design.

Better Performance, Better Compatibility

Motorola is expecting the first ColdFire v4 parts to hit 150 MHz when they appear in 2Q99. That clock rate is nearly twice as fast as the current ColdFire record holder, the 90-MHz MCF5307. The faster clock rate is due partly to Motorola's anticipated use of 0.25-micron process technology and partly to unclogging some bottlenecks in the ColdFire v3 pipeline. At 150 MHz, the company expects to deliver 200 Dhrystone 2.1 MIPS.

On the software front, Motorola has dealt with a nagging issue for all ColdFire chips: namely, their frustratingly incomplete compatibility with older 68K parts. The ColdFire architecture and programmer's model obviously borrow heavily from the 68K, but the differences are just numerous enough to prevent 68K binary code from running on ColdFire parts. For developers without access to the original source code, this was a tantalizingly awkward situation.

Two years ago, Motorola finally awakened to the possibility that 68K programmers might want to port their code to ColdFire, and it provided an assembly-level source translator developed by MicroAPL, a British software concern.

This year, Motorola has again tapped MicroAPL for a binary-emulation library. The emulation library allows any ColdFire v3 or (eventually) ColdFire v4 chip to execute the entire 680x0 instruction set, *sans* floating-point code. Unimplemented 68K instructions are trapped and emulated in library routines, with the same results and side effects as the original processor. The library even allows programmers to specify which 680x0 chip they wish to emulate; features are added and subtracted as required. In its minimum (68EC000) configuration, the library requires 50K of object code.

ColdFire Surges Up Through the Middle

ColdFire has turned from a curiosity to a mainstream part of Motorola's overall product strategy. The chips have overtaken their 68K progenitors in performance, as promised. By the middle of 1999, the fastest ColdFire chip will crank out 2× the performance of even the fastest 68060 processor.

The one feature ColdFire does abdicate to the 68K is floating point. For high-end motion-control systems, such as in robotics, double-precision floating-point support is mandatory, and ColdFire still does not provide this feature. It is the FPU alone that keeps the 68K family alive. Were Motorola to add an FPU to its ColdFire processors, the lucrative 68K market would soon collapse.

The addition of the "fixed-point fractional" number format to ColdFire v4 is a shrewd move to split market hairs. Printer makers and customers in low-end motion control (such as disk drives) want the added precision of fractional data, but they don't want the added cost of a full IEEE-754 FPU. Robotics makers, on the other hand, can't sacrifice precision or compatibility. ColdFire v4 allows Motorola to serve one group without alienating the other.

As the 68K wanes, ColdFire will step in to take its place. In many cases, it already has. Motorola will continue to push PowerPC at the high end and M•Core at the low end, but in future years, ColdFire will serve the vast mainstream of Motorola's customer base. The company's emphasis on development tools, quick product spins for special applications, and fast time to market are all indicative of a new business strategy that relies less on processor architecture and more on easing customers' design tasks. M