# Motorola Redefines 68K Instruction Set

## ColdFire Strips Functions, Increases Speed for Embedded Control

by James L. Turley

The venerable 680x0 family underwent a transformation at the recent Microprocessor Forum as Motorola revealed the first details of a new derivative of the 68K architecture streamlined for embedded control. The new devices will not be available until mid-1995 but promise to deliver 68040 performance at 68000 prices.

Conceding the desktop general-purpose computing market to PowerPC and Pentium, Motorola is focusing 68K development on embedded applications. The new CPU architecture, dubbed ColdFire, will form the basis of a new family of embedded-control processors aimed between the 68300 family and the emerging embedded PowerPC products. In return for ColdFire's gains in performance, the new architecture sacrifices binary compatibility with members of the 68K product family.

| Instruction | Data Size | Description |
|---|---|---|
| Unsupported Instructions | | |
| ABCD | All | Add BCD Digits |
| SBCD | All | Subtract BCD Digits |
| NBCD | All | Negate BCD Digits |
| DIVS | All | Integer Divide, Signed |
| DIVU | All | Integer Divide, Unsigned |
| ROL | All | Rotate Left |
| ROR | All | Rotate Right |
| ROXL | All | Rotate Left with Extend |
| ROXR | All | Rotate Right with Extend |
| RTR | All | Return and Restore |
| RTD | All | Return and Deallocate |
| TAS | All | Test and Set |
| TRAPV | All | Trap on Overflow |
| MOVEP | All | Move Peripheral |
| EXG | All | Exchange Registers |
| CHK | All | Check Register Against Bounds |
| DBcc | All | Decrement and Branch |
| CMPM | All | Compare Memory Operands |
| Unsupported Data Formats | | |
| ADD | Byte, Word | Integer Add |
| SUB | Byte, Word | Integer Subtract |
| AND | Byte, Word | Logical AND |
| OR | Byte, Word | Logical OR |
| EOR | Byte, Word | Exclusive OR |
| NOT | Byte, Word | Logical NOT |
| LSL, LSR | Byte, Word | Logical Shift Left, Right |
| ASL, ASR | Byte, Word | Arithmetic Shift Left, Right |
| NEG, NEGX | Byte, Word | Negate, with Extend |
| CMP | Byte, Word | Compare |

Table 1. The ColdFire instruction set excludes divide, rotate, and BCD support but includes most other 68000 instructions. Byte-sized data operands are unsupported for nearly all instructions.

### Tiny CPU Mixes Old and New Features

Like other vendors producing new embedded CPUs, Motorola is walking the fine line between processor simplicity and capability. Simplifying the instruction set often means trading off silicon in the CPU for silicon in the ROMs, as code size increases.

In his presentation, Motorola's Joe Circello described how, in optimizing the 68K's variable-length instruction set, the design team removed some of the least-used and most silicon-hungry instructions from the base 68000 CPU core. What's left is the roughly 80% of instructions most embedded systems use anyway, according to Circello. Left out are rotate and integer divide instructions, those handling BCD data formats, and most support for byte and word operands. Several addressing modes were also removed. Misaligned operand support becomes a chip-dependent option.

Table 1 lists the instructions that are removed from the base 68000 instruction set for ColdFire. In contrast, Table 2 shows those extensions from later 68K generations that are included. Integer multiply is enhanced, while integer divide is eliminated entirely, a feature many new embedded processors also lack. Motorola will supply an integer divide software routine to replace the missing function. The company claims the emulated divide function runs faster than the hardware divider on a 68000 at the same frequency and adds support for 32-by-32 division as well.

Every feature of the new CPU is calculated to reduce the size, and therefore cost, of the basic die. As an example, it is more efficient (in terms of silicon expended, not necessarily in code density) to build a simple 32-bit adder plus the logic to sign-extend 8- or 16-bit quantities to 32 bits than it is to construct a more complex adder that can tolerate different operand sizes.

Sometimes the knife cut deep. The shift and rotate functions were evaluated and, based upon the assumption that either one could do the work of the other, the rotate logic was excised in favor of the more compact shifter.

| Instruction | Operand | Description |
|---|---|---|
| EXT | 8 | Sign Extend Operand to 32 Bits |
| MULS, MULU | 32 | Integer Multiply Signed, Unsigned |
| TRAPF | n/a | Trap |

Table 2. Some 68020 instructions have been added, including an improved 32 × 32 multiply, new TRAP instructions, and a now-vital 32-bit sign-extension instruction.

Nor did the atomic instructions TAS (test and set) and CAS (compare and swap) make the cut. In their place is a programmable bit in the CPU's control register. Setting the bit asserts an external signal, hopefully preventing an external arbiter from granting the bus away. Used properly, the system can lock any number of bus cycles until the bit is cleared.

## New CPU Not Code Compatible

The ColdFire CPU is compatible with 68000 and 68300 family devices—up to a point. The new core executes a subset of 68000 instructions plus a few 68020 and 68040 operations. The result is neither a subset nor a superset of those chips, and thus it is not binary compatible with any existing 68K processor.

As an artifact of its cost-conscious design, the chip does not completely decode its opcode space. Missing instructions and addressing modes leave "holes" in the opcode map, and ColdFire does not bother to trap the freshly amputated instructions. Using earlier-generation 680x0 ROMs in a ColdFire-based system is virtually impossible, as instructions dropped from the ColdFire instruction set produce undefined results. Trapping unimplemented instructions for software emulation is not possible.

Motorola has not made any irreversible architectural decisions in this regard, so individual ColdFire devices could be more robust about handling unimplemented instructions. Optimistically, leaving the opcode map open permits simple enhancements to the instruction set in future devices.

CLARENCE TOWERS

**Motorola's Joe Circello describes the tradeoffs in the ColdFire CPU core design.**

Developing new code for ColdFire or simply porting existing code requires a new ColdFire-compatible compiler or assembler that emits only the supported instructions and addressing modes. Most software development tools already support the different 680x0 family devices through simple compiler switches. The new architecture could be implemented as another option.

The size of ColdFire object code relative to 68000 code is application dependent and, at this point, speculative. Simulations indicate that ColdFire code should be about 15% smaller than 68000 code by virtue of its new instructions, or 5% larger than 68040 code. The majority of the increase is due to reduced support for 8- and 16-bit operands, which must now be extended to 32 bits with an additional instruction. Other contributors are the fewer number of read/modify/write operations and instructions that are simply missing.

The fault model has been greatly simplified compared with the 68040. Even compared with the 68000, fault handling has been reduced to its basic elements. All stack operations are long-word (32-bit) aligned. Exceptions force exactly two long words onto the stack. The 32-bit program counter is followed by a 16-bit status register and a 16-bit format word describing the cause of the exception. Previously reserved bits in the format word supply information formerly found in larger 68K stack frames. External bus errors cannot be easily restarted in this model because the failing address is not on the stack.

The exception vector table retains a 680x0 layout with 256 vectors of four bytes each. The table is more sparsely populated than for a true 680x0, because many vectors are never used. Floating-point exceptions, for example, are never taken. The location of the table is programmable on 1M (12-bit) granularity boundaries.

All register-to-register operations execute in one clock cycle. Disregarding memory access time, stores also take one clock cycle, loads and memory-to-memory moves take two, and logic operations on memory operands take three clocks to complete. Both forward and backward branches take from one to three clocks to process, depending on whether the branch is taken or not.

## Core Trades Size for Speed

Based on a 0.5-micron three-layer-metal CMOS process, the ColdFire core is a minuscule 4.4 mm$^2$. This size places it among the smallest of any announced 32-bit CPU core, about even with VLSI's new 0.5-micron ARM6 at about 4 mm$^2$ *(see **0813MSB.PDF**)*, and less than Hitachi's SH70xx 8-mm$^2$ core size. Significantly, ColdFire is also about one-third the size of the company's own 0.8-micron CPU32+ core used in the 683xx series.

All the streamlining paid off. Motorola estimates that adding the full 68000 instruction set and fault model would have increased ColdFire's size by a factor of 1.9. That is to say, the design team realized a nearly 50% reduction in area purely through the optimization process. Circello credits Motorola's CAD tools, used to develop the 68040 and 68060, for the impressive feat.

Interestingly, some implementations of the 68000 instruction set are not so large. Using the same process geometry, a 68EC000 core is only slightly larger than ColdFire, at 4.6 mm$^2$. This dichotomy highlights one of the inherent tradeoffs of RISC design. Although the ageing 68000 has a microcoded CPU that is hardly optimized by today's standards, it would still be smaller than an equivalent ColdFire. On the other hand, ColdFire executes nearly all instruction in one clock cycle, compared with the 68000's four.

First silicon of the device has not yet been built.

Gate-level simulations of a 50-MHz device, with an on-chip 4K unified cache, produce 44 Dhrystone MIPS. For comparison, a 25-MHz 68040 delivers 25 Dhrystone MIPS, or about half the performance at half the clock speed. Not all ColdFire implementations will include the cache; the modular design allows for RAM arrays, cache, ROM, or no on-chip memory at all.

ColdFire-based integrated processors will not share the IMB (inter-module bus) that distinguishes most 683xx devices. Carrying the overhead of any standard bus, even an internal one, was deemed too expensive. Instead, the company is preparing a set of fully synthesizable peripheral modules specifically for the new family. Indeed, the CPU core itself is fully synthesized. Leveraging the commercial and in-house synthesis tools first used on the 68060 allows Motorola to quickly and efficiently produce derivative products to meet customer demand.

Properly done, synthesis tools can create a chip design that is independent of process technology, voltage, and frequency. Derivative products can be developed at a higher level of abstraction than the physical cut-and-paste technique of the 683xx series. Currently, the module development is in its early stages. The CPU core will not be available as part of the company's FlexCore ASIC-development program until 4Q95.

## RISC Principles in the Service of CISC

The fact that ColdFire runs as fast as a 68040 at similar clock speeds—with a die nearly one half the size— calls into question the usefulness of 15 years of "enhancements" made to the architecture. Over time, the additions to the 68020, 68030, 68040, and 68060 have apparently made little impact on the intrinsic performance of the processor. Admittedly, most new instructions were aimed at improving the performance of desktop applications, which are not measured in simple Dhrystone MIPS. Performance improvements have been due mostly to smaller geometries, IC process improvements, instruction cycle reductions, and the addition of on-chip caches.

Additional instructions beyond the basic core set are utilized only by assembly language programmers who can hand-tune their applications to the peculiarities of a new instruction set. Compiler users, on the other hand, have no such control. Unless the compiler itself emits the new opcodes, arcane instructions remain unused. The BFFFO (bit-field find first one) instruction, first found on the 68020, is a good example. Compiler writers are themselves under pressure to produce timely releases, so exotic instructions remain largely ignored. Very little is gained for the addition of so much silicon.

The impressive achievements of the ColdFire design highlight a paradox in recent embedded processor announcements. How can ColdFire, based on a 15-year-old CISC architecture, achieve a smaller core size, faster clock rate, and better performance than other RISC cores ostensibly designed at the outset for scalability?

Part of the answer lies in ColdFire's relatively small register set, inherited from the 680x0. A new RISC processor would never be designed with only eight data registers. Gone, too, are provisions for register windows and other fancy performance features. ColdFire supports only 24-bit addressing and big-endian byte ordering, and it has no hooks for memory management. Finally, Motorola has shown that it has excellent layout tools with which to tweak the design.

## ColdFire Poised to Supplant 680x0

The development of ColdFire is a bold, proactive move on Motorola's part. It has taken the oldest and most widely used embedded architecture and turned it into something competitive with the latest embedded RISC CPUs. Preserving some measure of 68K compatibility allows designers to reuse their experience with the architecture, if not their actual binaries.

When ColdFire devices reach general availability next year, they will be positioned between the 68300 series and the MPC500 embedded PowerPC family. This is a good fit in Motorola's product line; developers can trade off degrees of 680x0 compatibility for performance and growth potential.

The presence of ColdFire casts a shadow across the upper end of the 683xx line. Upgrading that family with 68040- or '060-derived cores may not make sense anymore. ColdFire provides more performance, at a lower cost, than a high-end 683xx device would.

At the top end, the 68060 appears to be the end of the line. Apart from minor voltage and frequency variations, it could well be the last implementation of the 68K instruction set the company develops. With the low-end 683xx bottled up and the high-end 68060 running out of steam, ColdFire is now the heir apparent to Motorola's entire 32-bit CISC realm.

The 68K architecture has served the computer industry long and well. That it lasted this long, and was the genesis for ColdFire, is a testament to its flexibility. Motorola can only hope that ColdFire will be as successful. ♦

## For More Information

General sampling of the first ColdFire-derived device is scheduled to begin 2Q95. Additional devices in the family will be announced beginning in the latter half of 1995. Coincident with the first chip from Motorola will be the announcement of the first commercial product based on a ColdFire processor.

For more information contact Tom Spohrer at Motorola, 512.891.2917 or e-mail *toms@oakhill.sps.mot.com.*