WHITE PAPER:

Motorola Version Three ColdFire® Processor

Advanced Processors for the Next Generation of Storage, Imaging, and Multimedia Products

October 1997

Table of Contents

Overview

First introduced in 1994, the Motorola ColdFire® Family has continued to be unique in its ability to provide compelling integrated solutions for today's embedded systems. Primarily designed to simplify system development and shorten time-to-market cycles, the ColdFire processor line is based on a memory-saving, variable-length (VL), 32-bit RISC instruction set that is optimized for efficient code density and cost-sensitive integrated systems. Now in its third generation, the advancements implemented in the newest version of the ColdFire processor family represent a significant leap forward in performance provided by better throughput and higher operational frequencies.

The Version Three (V3) ColdFire core delivers enhancements including a refined instruction prefetch pipeline, branch prediction capabilities, and higher frequencies of operation. These improvements allow the V3 core to provide up to 250% to 300% more performance than the Version 2 core, while increasing the operating system frequency by less than 50%, making it an attractive solution for new designs or upgrades to existing systems. In addition, the V3 core includes a high-speed multiply-accumulate unit, dedicated hardware divide, and enhanced debugging capabilities.

The V3 core also represents a well-defined performance migration path for past 68K and ColdFire microprocessor system designs. The ColdFire processors' foundation in Motorola's legendary 68000 (68K) architecture allows designers to leverage more than 18 years of tool development, code evolution, and engineering expertise. Because of its 68K roots, the ColdFire processor line provides the most logical solution for a vast number product designs that may be ready for the migration to the performance gains offered by a RISC architecture. In addition, V3 is 100% upward code compatible with V2 and offers a significant performance boost for V2 users who are ready to upgrade.

The V3 ColdFire core represents state-of-the art embedded microprocessor technology that is ideally suited for a new generation of advanced imaging, storage, and multimedia products such as high-resolution laser and inkjet printers, high-performance disk drives, and set-top boxes – applications that continually demand higher performance, yet are extremely cost sensitive.

Market Conditions

The Growing Demand for Performance

The market for 32-bit embedded processors is in a state of tremendous growth with enormous opportunities in imaging, storage, and multimedia – areas targeted by the third-generation ColdFire processors. The Internet and small office/home office are fueling the expansion of potent markets, and the use of embedded processors in these applications is expected to grow at a healthy pace well into the next millennium. System developers in these markets have stringent requirements, such as time-to-market pressures, cost issues, and assurance of a well-defined performance migration path, that influence their decision to utilize an embedded architecture. According to Micro Design Resources, Motorola 68K-based designs represented over 50 percent of the volume in the 32-bit embedded market in 1996, and many of these existing systems require the cost-effective performance upgrade that the ColdFire product line can provide. V3 products also provide an excellent upgrade path for hundreds of existing V2 designs, which include office equipment, robotics, industrial control, telecommunications, settop boxes, measurement equipment, and consumer products.

Code Compatibility

Another factor that is extremely important in choosing an architecture is code compatibility. Given the complexity of today's integrated embedded processors, reusing software code from previous efforts is extremely desirable as it often represents the most costly portion of system development. Motorola engineers have diligently maintained code compatibility across the entire ColdFire processor family. The V3 core is no exception. It is 100 percent upward-compatible with the V2 core and will be upward compatible with future versions of ColdFire cores as well.

In addition to compatibility with other family members, the ColdFire instruction set is based on a subset of the Motorola 68000 instruction set, which enables software code from existing Motorola 68K-based systems to be easily ported to the ColdFire architecture. This task has been made even easier by the creation of PortAsm/68K for the ColdFire product line, a source-level assembler porting tool (which Motorola provides free of charge) which translates 680x0 family assembly language to run on ColdFire processors. This represents enormous potential for the ColdFire architecture, given the huge installed

base of 68K designs. Many of these designs require additional performance, yet must maintain spartan development costs.

The V3 Architecture

The Pipelined Approach of the V3 ColdFire Core

The V2 core architecture, shown in Figure 1, consists of two independent and decoupled pipelines: the two-stage instruction fetch pipeline and the two-stage operand execution pipeline. The instruction fetch pipeline contains an instruction address generation (IAG) stage followed by the instruction fetch (or IC, instruction cycle) stage, while the operand execution pipeline includes the instruction decode (or DSOC, decode/select and operand cycle) and the execution stage (or AGEX, address generation and execute).

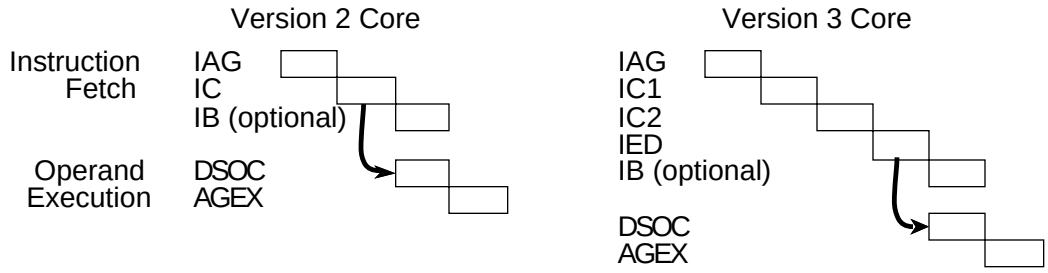

Figure 1: V2 and V3 core pipeline stages

An overriding goal of the V2 design was to attain the smallest possible core size. As a result, the V3 design leveraged two opportunities in the V2 design to greatly increase operating frequency. The first is associated with the processor's single-cycle internal bus, the "K bus." This 32-bit bus connects the core and debug module with any on-chip memories, including cache, RAM, and/or ROM, and delivers instructions and operands to the processor. The second opportunity is related to the placement of the instruction decode function within the processor's pipelines. With the advent of the V3 core, both of these opportunities have been leveraged.

The single-cycle K-bus memory access was addressed by dividing it into a two-stage pipelined process. This enabled higher frequency operation, because what used to take a single cycle now effectively is a pipelined access over two cycles.

The first K-bus access cycle, KC1, begins with a registered address local to the memory controller, accesses the memory array, and then registers the results in the memory

controller. This provides an a register-to-register bounded path within the memory controller and alleviates the timing pressure associated with the access times for the on-chip memories. During the second K-bus access cycle, KC2, the accessed data is selected from the appropriate source and routed back to the processor. This approach provides an entire clock cycle to source data from the appropriate memory controller. However, due to the pipelined nature of the K-bus, data is returned every cycle.

The second opportunity for frequency improvement involved the location of the instruction decode function. For the V3 design, a portion of the decode function was moved from the execution pipeline into a new stage in the instruction fetch pipeline, a concept that was borrowed from the design of the superscalar 32-bit MC68060 processor. This added stage to the instruction fetch pipeline, the instruction early decode stage (IED), is dedicated to performing this "early" instruction decode. The benefit of this approach is that time-critical instruction decode information is made available at the same time that the instruction is gated into the operand execution pipeline.

The instruction decode function in the V2 core is a fairly complex function that performs much like a hardware lookup table. The opcode provides the index into a hardware table that then produces a vector of control information. With the V2 core, the availability of the control vector occurs fairly late in the DSOC cycle.

In the V3 core design, by providing the early decode information at the same time the instruction register is loaded with the opcode and extension words, the processor can more efficiently control the sequencing of the operand pipeline at a much higher operating frequency.

The creation of the two-stage pipelined K-bus and the addition of the early decode structure represent the fundamental changes in the architecture of the V3 core. Figure 1 illustrates the pipeline structures for the V2 and V3 designs.

To summarize, the instruction fetch function in the V2 design is a two-stage pipeline: an instruction address generation (IAG) stage followed by an instruction cycle (IC) stage (which represents the single-cycle K-bus memory access). In contrast, the V3 fetch pipeline begins with the same instruction address generation stage, while the single IC stage is replaced with two stages reflecting the 2-cycle pipelined K-bus access, IC1 and IC2. Once the instruction is returned at the completion of the IC2 stage, another new stage — the instruction early decode (IED) — was added to perform the time-critical decode and exploit the speed advantage provided by having the opcode and its decode available as the

instruction enters the operand execution pipeline. At the conclusion of the IED stage, the instruction and early decode information is either loaded into the instruction buffer (IB) or routed directly into the first stage of the operand execution stages of the pipeline. The 2-stage fetch pipeline of the V2 design was expanded to a 4-stage pipeline in the V3 architecture, with both designs including the optional instruction buffer stage at the bottom of this hardware structure.

The operand execution pipeline is essentially unchanged between the V2 and V3 designs. While the 2-stage pipelined K-bus does add one cycle to the execution time of all instructions reading operands from memory, register-to-register instructions, as well as instructions which store to memory, retain their single-cycle execution times.

In general, the timing of the V3 design is more balanced across the pipeline stages than V2, which was one of the goals of the V3 development since it allows the new microarchitecture to be more scaleable to advanced process technologies and results in higher frequencies of operation. Because the primary design motivation for the V2 core was small die size, there was intentionally a minimization of pipelining. Because the initial V3 implementations are targeted for a manufacturing technology with 0.35 micron feature sizes, the advanced process supported the development goals of maximizing performance and frequency while maintaining a small core die size.

Reorganized Instruction Buffer

The IB in Figure 1 represents the instruction buffer stage. This stage provides the queuing mechanism that allows the instruction fetch and operand execution pipelines to operate independently. Instruction information can be loaded into the operand execution pipeline from the IB, or directly from the IED stage of the instruction fetch pipeline.

In the V2 core design, instructions are prefetched and typically written into the IB 32 bits at a time. The IB functions as a simple 3-entry, 32-bit wide, first-in-first-out (FIFO) buffer, and leaves the determination of instruction length and boundaries to the OEP.

In the V3 core design, the instruction buffer is reorganized as an eight-entry buffer, with each entry holding one complete machine instruction. Each entry in the V3 instruction buffer has four 16-bit components. The first two parts are required with every instruction and include a 16-bit operation word and 16 bits of early decode and branch acceleration prediction information. The two remaining 16-bit components are optional and are known as extension word one and extension word two. Because the ColdFire architecture

implements a variable instruction set, all combinations are possible (i.e., there can be zero, one, or two extension words).

By organizing the IB as machine instructions, the mechanism required to gate the next instruction from the IB into the OEP is greatly simplified. As the OEP completes an instruction, the IB simply outputs the entry from the next sequential location.

Change-of-Flow Acceleration

In general, as the number of pipeline stages increases in a given design, the processor's performance during change-of-flow operations (such as branches) decreases. Because the V2 design intentionally minimized size and pipeline depth, the performance issues associated with branch instructions were diminished compared to designs with longer pipelines.

Because the ColdFire core implements decoupled instruction fetch and operand execution pipelines, the increased depth of the IFP is visible on taken branch instructions because the current sequential prefetch stream must be discarded and a new fetch stream established at the target instruction address. The designers of the V3 core addressed this situation in a novel way. As it turns out, the early decode information provided in the V3 instruction fetch pipeline allows implementation of a unique branch acceleration technique.

While not used in the V3 design, a traditional method of accomplishing change-of-flow acceleration is known as branch caching. In this approach, a hardware lookup table is implemented providing an association between instruction addresses of branch instructions and their corresponding target addresses. Control information allows the branches to be "predicted" based on past execution history. Branch cache designs use various sizes and prediction algorithms, but all implementations require a sizable hardware investment to be effective.

Unlike a branch cache, the change-of-flow acceleration technique in the V3 design applies to all branches as they are prefetched, and it is not limited by the size of a branch cache or past execution history. Ultimately, this approach provides a substantial improvement in branch execution time, but at a much lower cost of implementation when compared to a branch cache.

One field provided by the early decode stage is the length of the instruction, which is one of the time-critical pieces of information needed as the instruction begins execution in the OEP. Once the instruction length is known, the instruction fetch pipeline can determine the location of instruction boundaries. In fact, this capability enables the reorganization of

the instruction buffer to a storage queue organized as machine instructions rather than a simple FIFO.

The second opportunity that resulted from the early decode information was the ability to add branch acceleration capabilities to the IFP. Because the instruction boundaries are known, the processor can scan for specific opcodes, namely, change-of-flow instructions, such as branch opcodes. Once one of these opcodes is detected, the target address is calculated and immediately gated into the instruction address generation (IAG) stage of the IFP. The net effect is that the detection and redirection of the target prefetch stream required for branches in the V3 core is moved from the OEP to the IFP.

The branch acceleration techniques in the third-generation ColdFire core are applied to change-of-flow instructions with program counter relative or absolute addressing modes. For conditional branches, a static prediction algorithm determines whether the branch should or should not be taken. Backwards branches (i.e., a jump in which the target address is at an earlier PC step in terms of numerical value), by default, are predicted as taken. In most cases, these backwards branches correspond to loop branches, which are usually taken.

Forward conditional branches are more difficult to predict. Based on analysis of large amounts of embedded application code, approximately half of all forward branches are taken, although actual behavior can depend highly on the specific application. Accordingly, the default prediction for forward branches is “not taken.” However, a new control bit in the condition code register, accessible either in supervisor or user mode, provides a mechanism to dynamically alter the prediction for forward conditional branches. When this new control bit is asserted, forward branches are predicted as taken.

Through the use of these branch acceleration mechanisms, the measured OEP performance of change-of-flow instructions in V3 approaches single-cycle execution.

Clock-Multiplied Core

With today's advanced process technologies, microprocessor cores are designed to run at increasingly higher frequencies. In the design of the V3 architecture, all components in the processor complex — the CPU, debug module, internal K-bus, and all attached memories — are currently designed to operate at a clock rate of up to 90 MHz using a 0.35 micron technology. However, it's not appropriate or efficient to run the rest of the microprocessor at the same speed.

The V3 core employs two separate clock domains for simplicity and power savings: a low-frequency clock domain drives the external bus as well as any on-chip peripherals and customer supplied logic (if present); a high-speed clock controls the processor complex. In addition, the V3 design supports any integer equal to two or greater for the clock multiplication ratio (up to the maximum frequency for the processor complex).

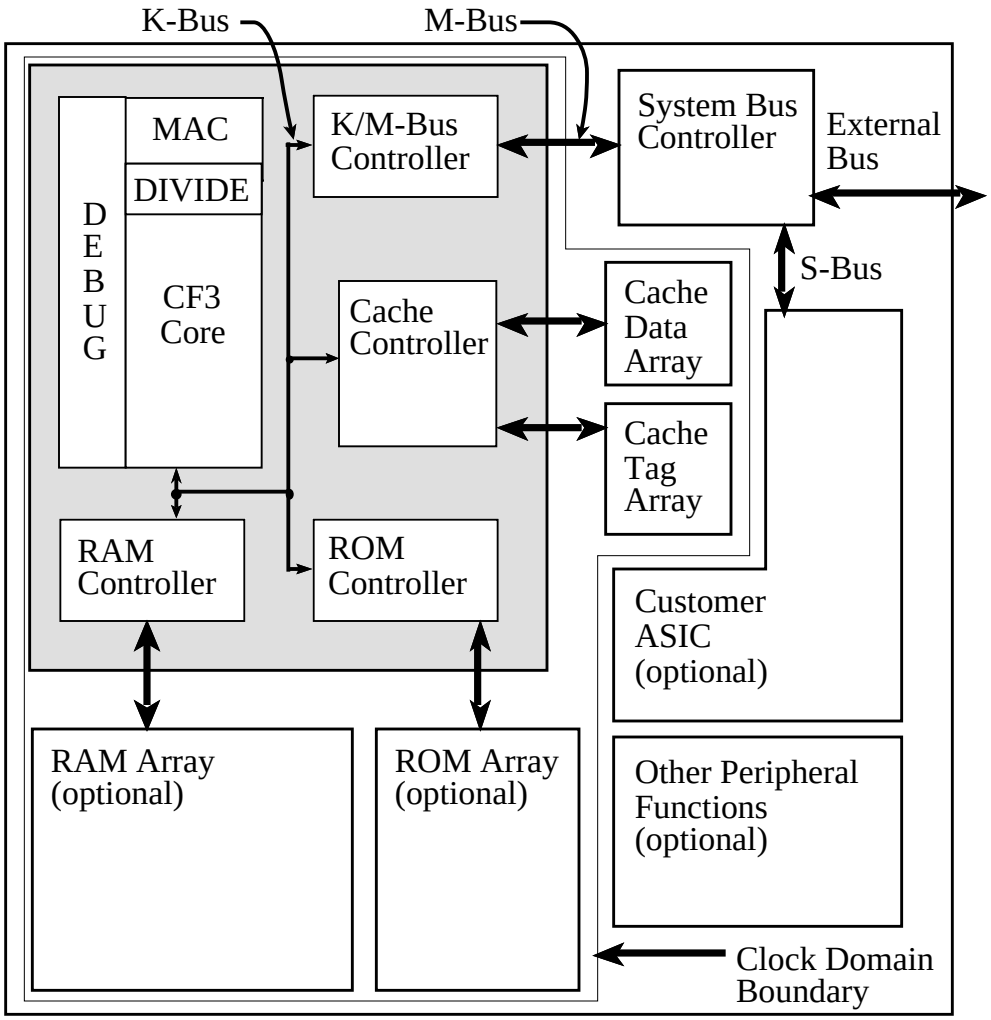

Figure 2: Functional block diagram of a representative embedded microprocessor implementation using the V3 ColdFire core.

The incorporation of multiple clock domains reveals a unique strength of the ColdFire architecture. In microprocessor architectures without multiple internal buses, it is

very difficult to set up multiple clock domains. Because of its hierarchical bus structure, establishing different clock domains in ColdFire processors was relatively easy. The clock boundary can be very well defined, as it consists of a registered boundary at all inputs and outputs from the processor complex. In terms of a functional ColdFire system diagram, this boundary is located at the master bus (M-bus) level (see Figure 2).

This architecture simplifies the design task in terms of timing, drive capabilities, and test for standard part configurations or in cases where the customer integrates a proprietary ASIC around a ColdFire core. The registered boundary isolates the processor complex and provides for a simplified implementation of clock distribution trees, an important consideration in high-frequency designs. In addition, running a large portion of the chip at a lower speed directly translates into a reduction in power dissipation.

Multiple clocks also solve a number of other problems. The portion of the circuit running at a lower speed is much simpler to design, whether it's a customer ASIC or standard Motorola peripherals. This also simplifies the reuse of existing designs for peripheral modules as well as the migration of any existing custom logic from other ColdFire cores.

Performance Impact

The development work done on the V3 core will have positive effects for some time, since many of the architectural changes will make it much easier for Motorola engineers to reach the performance milestones established for the ColdFire architecture’s migration path.

In order to evaluate the performance of a given processor versus any other processor, three fundamental factors must be considered: the effective processing rate in clock cycles per instruction; the CPU's operating frequency; and the number of instructions required to complete a given task. If two processors use the same instruction set (as in a comparison of the V3 vs. V2 cores), the number of instructions required is eliminated as a factor since they both execute the same code.

The V3 design added stages to the instruction fetch pipeline, so the effective cycles-per-instruction for the V3 core is slightly higher than V2. This factor is offset by the much greater increase in maximum operating frequency that the V3 core can achieve. Figure 3 shows 18 benchmarks, most of which represent actual application code for a wide variety of systems. Using typical V2 and V3 device configurations and the same 70 ns bursting

DRAM memory system, V3 provides 2 to 3 times the performance of V2, with the geometric mean at 2.5x.

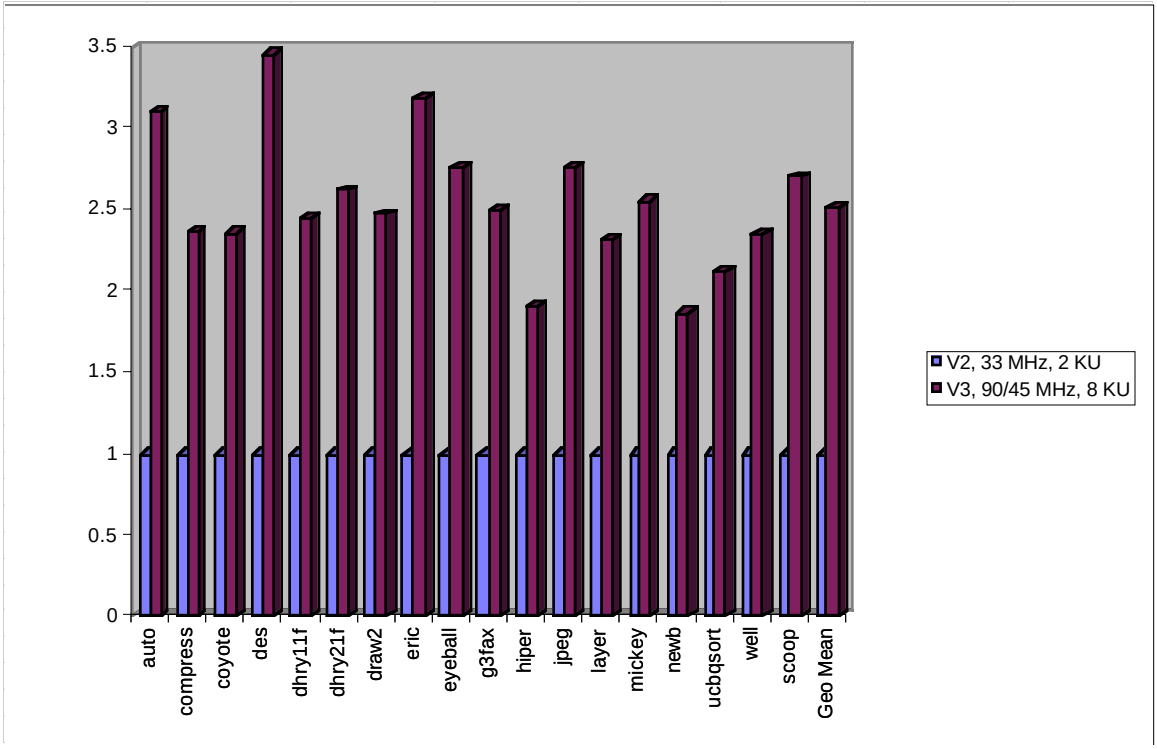

Figure 3: Benchmarks of customer-provided code demonstrate a typical 2.5x performance boost for V3 over V2.

Other Version 3 Enhancements

MAC and Hardware Divide

To this point, the analysis of modifications from the V2 to the V3 core has focused on the instruction fetch pipelines. The operand execution pipeline, the other major part of the CPU, calculates and fetches operands from memory or registers and then executes instructions. The operand execution pipeline remains virtually identical between V2 and V3 designs, except for the addition of two optional execute engines in the execution stage. These optional execute engines, a multiply-accumulate unit and a hardware divider, can be included in V2 designs as required.

The standard execution stage includes an ALU for operand address calculations, plus basic arithmetic and logical instruction execution. The other standard execute engine is a barrel shifter to perform shift operations and other miscellaneous instructions.

The optional execution engines consist of a high-speed multiply-accumulate (MAC) unit and a hardware divider. All four execute engines function as three-terminal devices, receiving two input operands and generating an output result.

The MAC and divider are specialized units that execute specific instruction types. The MAC instruction set is designed for high-speed digital signal processing (DSP) applications, while the hardware divide carries out integer divide and remainder instructions.

The MAC unit provides high-speed arithmetic computation via a multiply-accumulate function. The basic function consists of two multiply values, producing a result, which is then either added or subtracted from an accumulator. This operation is appropriate for a wide range of signal processing applications, such as digital signal filtering, JPEG and MPEG compression, and DCT transformations. For example, applications such as disk drives and printers often require FIR filtering for servo motor control.

An analysis of a representative segment of code used in a disk storage application revealed a small number of multiply instructions (about 11 percent). As a baseline, the V2 without a MAC unit produced a relative performance of one. Using C-language macros, the source level code was recoded and recompiled to generate MAC instructions. The resulting code, when run on a V2 integrated with a MAC unit, increased performance by approximately 50 percent, despite the fact that the multiply instructions accounted for only a small percentage of the dynamic execution stream. In addition, hand optimization of the assembly code (a common practice used to optimize DSP code) raised the performance level approximately 70 percent above the baseline. Because the core is synthesizable, optional units such as the MAC unit can easily be added to any ColdFire core.

The hardware divide can also provide significant performance improvements in certain types of applications. For instance, in imaging applications, this enhancement can improve overall application performance by up to 10 or 15 percent, depending on the page image.

Enhanced Illegal Opcode Handling

In the V3 core design, the entire 16-bit opcode space is fully decoded. This is very useful for adapting existing application code from 68K designs as any attempt to execute a nonlegal ColdFire opcode generates the illegal instruction exception. This enables designers to emulate previous instruction sets and automatically flag any unsupported instructions.

Debug Module, Rev. B Enhancements

While customers and developers alike resoundingly declared the ColdFire debug module as a great leap forward in processor debug capability, they have provided a few suggestions to improve it even further. The Rev. B debug module boasts notable enhancements based on input from both customers and emulator developers. The Rev. B debug module is backward compatible. In addition, the Rev. B debug module will work with emulators designed for the Rev A. debug module, and can be incorporated into V2 or V3 ColdFire products.

The Rev. B debug module includes a new serial background debug mode (BDM) command that allows the processor to dump out the current value of the program counter (PC) anytime, unobtrusively. This new command provides a convenient mechanism to allow emulators to monitor executed PC values to develop performance analysis tools. In the original design, the three internal hardware breakpoint resources could be configured for single- or double-level triggers by ANDing conditions together. In the Rev. B design, the breakpoint resources can also be ORed together to specify trigger events.

In response to emulator developers who voiced a need to perform concurrent BDM activity while the internal breakpoint resources were active, Motorola added breakpoint registers that were not shared between the breakpoint functions and the BDM functions.

Physical Implementation

As with all ColdFire cores, the V3 core is 100 percent synthesizable. This is one of the most significant and unique features of the ColdFire architecture for a number of reasons. First, it means that with minimal effort any ColdFire device can quickly be redeployed in a new process technology. Function-level "parameterization" is used for all ColdFire core components, including the MAC, hardware divider, debug module, and all processor-local memories. This architectural philosophy allows a myriad of specific

configurations to be easily created based on the application requirements. Additionally, all K-bus memory controllers are designed to support a range of array sizes. These two powerful concepts allow the system designer to create a ColdFire processor core customized for their application by providing configurability to system components that were previously defined only by the MPU vendor. In fact, 18 different ColdFire core configurations have been deployed in 10 different process technologies (5 outside of Motorola) in only 12 months. Synthesizability also extends the useful life of previous design efforts by allowing both customers and Motorola to reuse existing ColdFire peripherals or modules in future designs. The architecture's fully static operation and small chip size (approx. 3 square mm for a 0.35 micron CMOS process) minimize overall power requirements, and make the ColdFire architecture ideal for embedded system designs.

Deployment

The V3 core has already been licensed and deployed to selected major customers and Motorola business units. The first 53xx ColdFire standard product will be available in early 1998.

The V3 core performance exceeds 70 MIPS in a 90/45 MHz configuration supported by a memory subsystem using 70 ns bursting DRAMs. A V3 core in this configuration, with its standard 8K unified cache, will provide 2.5 to 3.0 times the performance of a V2 ColdFire device with a 2K unified cache at 33 MHz with the same external memory (see Figure 3). Combined with its low cost (future highly integrated 53xx devices will be priced at less than $20 each in volume), highly embeddable architecture, and mature development environment, the V3 ColdFire core is an ideal embedded processing solution for the next generation of mass storage, imaging, and multimedia products.