MOTOROLA

Semiconductor Products Sector

This document can be
ordered by:
MCORE1/D

M•CORE1

Architectural Brief
M•CORE
microRISC Engine

The 32-bit M•CORE microRISC Engine represents a new line of Motorola microprocessor core products. The processor architecture has been designed for high-performance and cost-sensitive embedded control applications, with particular emphasis on reduced system power (energy) consumption, making it suitable for a number of battery-operated, portable and mobile products.

Total system power consumption is dictated by various components in addition to the processor core. In particular, memory power consumption (both on-chip and external) is expected to dominate overall power consumption of the core+memory subsystem. With this factor in mind, the Instruction Set Architecture (ISA) for M•CORE makes the trade-off of absolute performance capability versus total energy consumption in favor of reducing the overall energy consumption. This is accomplished, while maintaining an acceptably high level of performance at a given clock frequency.

By designing a streamlined execution engine, many of the same performance enhancements and implementation techniques used by desktop RISC designs are possible. By utilizing fixed-length instruction encoding and defining a strict load/store architecture, control complexity and overhead is minimized. The goal of minimizing the overhead memory system energy consumption is achieved by adopting a (relatively) short 16-bit instruction encoding. This choice significantly lowers the memory bandwidth needed to sustain a high rate of instruction execution. This careful selection of the instruction set for M•CORE, allows the code density and overall memory footprint (efficiency) of the M•CORE architecture to approach and surpasses the code density offered by traditional CISC architectures.

In addition to substantial cost and performance benefits, M•CORE also offers advantages in power consumption and power management. M•CORE minimizes power dissipation by using a fully-static design, dynamic clock management, and low-voltage operation. It automatically powers-down internal function blocks that are not needed on a clock-by-clock basis. Power conservation modes, which are invoked via three low-power mode instructions, provide for absolute lowest power consumption.

The primary features of the M•CORE processor include the following:

- 32-bit Load/Store RISC Architecture
- Fixed 16-bit Instruction Length
- 16 Entry 32-bit General Purpose Register File
- Efficient 4-Stage Execution Pipeline, Hidden from Application Software
- Single-Cycle Instruction Execution for most Instructions, Two cycles for Branches and Memory Access Instructions
- Support for Byte/Halfword/Word Memory Access
- Fast Interrupt Support with 16-Entry User-Controlled Alternate Register File
- Vectored and Autovectored Interrupt Support

This document contains information on a product under development. Motorola reserves the right to change or discontinue this product without notice.

SEMICONDUCTOR PRODUCT INFORMATION

- On-Chip Emulation Support
- Full Static Design for Minimizing Power Consumption
- Fully Supported by Industry-Leading Third-Party Development Tools

OVERVIEW

Figure 1 is a block diagram of the M•CORE processor. The following paragraphs provide an overview of the M•CORE processor.

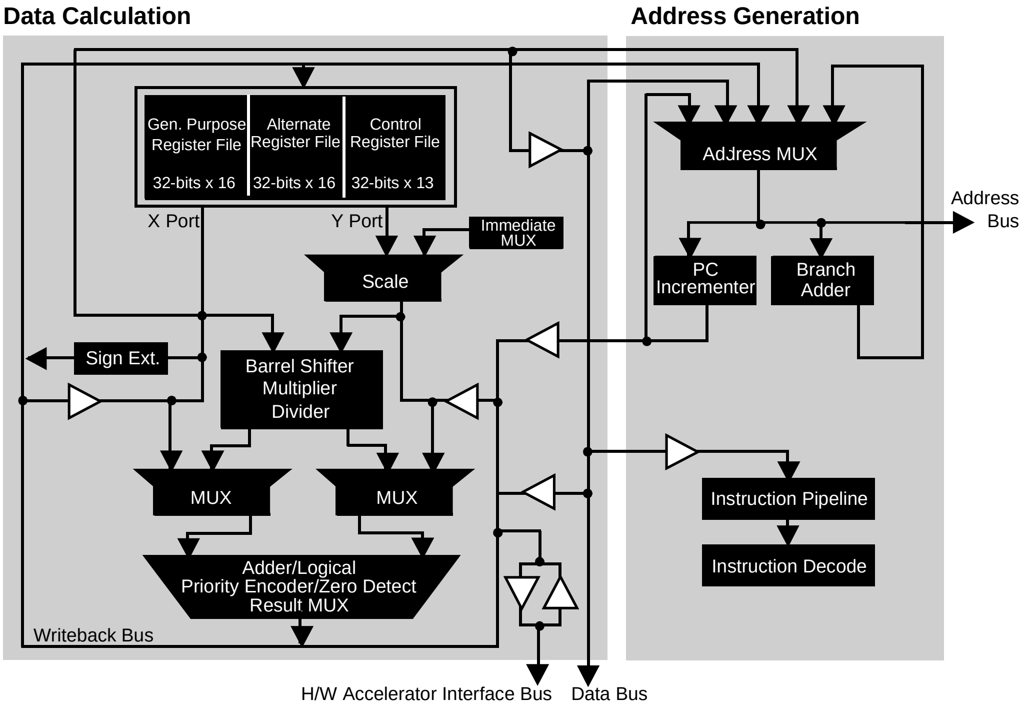
Data Calculation
Address Generation
Gen. Purpose Register File 32-bits x 16
Alternate Register File 32-bits x 16
Control Register File 32-bits x 13
X Port
Y Port
Immediate MUX
Scale
Sign Ext.
Barrel Shifter Multiplier Divider
MUX
MUX
Adder/Logical Priority Encoder/Zero Detect Result MUX
Writeback Bus
Address MUX
Address Bus
PC Incrementer
Branch Adder
Instruction Pipeline
Instruction Decode
H/W Accelerator Interface Bus
Data Bus

Figure 1. M•CORE Block Diagram

M•CORE PROCESSOR CORE

The M•CORE utilizes a four-stage pipeline for instruction execution. The Instruction Fetch, Instruction Decode/ Register File Read, Execute, and Register File Writeback stages operate in an overlapped fashion, allowing single clock instruction execution for most instructions.

Sixteen general purpose registers are provided for source operands and instruction results. Register R15 is used as the Link Register to hold the return address for subroutine calls and Register R0 is used as the current Stack Pointer by convention.

The execution unit consists of a 32-bit Arithmetic/Logic Unit (ALU), a 32-bit Barrel Shifter (Shifter), Find-First-One unit (FFO), Result Feed-Forward hardware, and miscellaneous support hardware for multiplication, division, and multiple register load and stores. Arithmetic and Logical operations are executed in a single cycle with the exception of multiply and divide. Multiply is implemented with a 2-bit per clock, overlapped-scan, modified Booth algorithm with early-out capability (for minimal die area /future implementations will be faster) to reduce execution time for operations with small multipliers. Divide is implemented with a 1-bit per clock early-in algorithm. A Find-First-One unit operates in a single clock cycle.

The Program Counter Unit has a PC incrementer and a dedicated Branch Address Adder to minimize delays during change of flow operations. Branch target addresses are calculated in parallel with branch instruction decode, with a single pipeline bubble for taken branches and jumps, resulting in an execution time of two clocks. Conditional Branches which are not taken execute in a single clock.

Memory load and store operations are provided for byte, halfword and word (32-bit) data with automatic zero extension of byte and halfword load data. These instructions can execute in two clock cycles. Load and store multiple register instructions allow low overhead context save and restore operations; these instructions can execute in (N+1) clock cycles, where N is the number of registers to transfer.

A single Condition/Code Carry (C) bit is provided for condition testing and for use in implementing arithmetic and logical operations greater than 32-bits. Typically, the C bit is set only by explicit test/comparison operations, not as a side-effect of normal instruction operation. Exceptions to this rule occur for specialized operations where it is desirable to combine condition setting with actual computation.

A 16-entry Alternate register file is provided to support low overhead interrupt exception processing, and both vectored and autovectored interrupts are supported by the CPU.

POWER MANAGEMENT FEATURES

The M•CORE processor was designed for industry-leading power efficiency. The instruction set and the machine were designed to effectively access internal and external memory which is where a major portion of the chip and system’s power consumption can be attributed. Since the M•CORE is a 16-bit instruction machine, it can efficiently interface to external memory through a 16-bit interface. M•CORE 16-bit instruction mapping gives the user a compact memory image as well, which minimizes the number of accesses to both internal and external memory.

Both static and dynamic power-enhancing features are included in the architecture and implementation. In terms of the M•CORE architecture, three instructions were added to allow the system designer an avenue to optimize his design with various power saving modes. As defined by M•CORE, these instructions are WAIT, DOZE, and STOP. The functionality of these modes are not dictated by the core but are configured by an external power management module. The M•CORE processor core provides output signals associated with the execution of each of these instructions that may be monitored by external logic to control operation of the core as well as the rest of the system.

The first implementation of M•CORE is optimized for power consumption. The M•CORE has a compact die area of 2.2 mm sq. in 0.36 (L effective) micron CMOS technology. The logic has been minimized, as well as the routing capacitance. Gated clocks played a major role in minimizing unnecessary or spurious bus transitions in the data path portion of the design.

CODE DENSITY

The M•CORE engine minimizes the overhead memory system by using (relatively) short 16-bit instruction encoding. This choice significantly lowers the memory bandwidth needed to sustain a high rate of instruction execution. The careful selection of instruction for M•CORE, allows for a compact data structure and a small overall memory footprint for the M•CORE architecture. M•CORE supports 8, 16 and 32-bit data, but is highly optimized for use out of 16-bit off-chip memory. This allows a design based around M•CORE to use less expensive and smaller memories, decreasing overall system cost, while also using less memory on-chip for more integrated solutions.

HARDWARE ACCELERATOR INTERFACE (HAI)

The M•CORE engine provides support for task acceleration by external hardware blocks which are optimized for specific application-related operations. Data is transferred between the core and an Accelerator Block by one or more of several mechanisms as appropriate for a particular implementation. These external hardware blocks may be as simple as a block for performing a population count, or a more complicated function such as a DSP acceleration block capable of high speed multiply/accumulate operation or data encryption.

DEBUG INTERFACE

The M•CORE architecture supports on-chip emulation (OnCE™) circuitry through the JTAG interface, which provides a means of interacting with the M•CORE processor core and any peripherals. This allows a user to examine registers, memory, or on-chip peripherals facilitating hardware/software development on the M•CORE processor core. Internal Status and Control registers are accessible via the serial scan chain during OnCE™. To achieve this, special circuits and dedicated pins for the M•CORE processor core are provided to avoid sacrificing any user-accessible on-chip resources.

TARGET APPLICATIONS

The M•CORE architecture represents a new level of performance for embedded applications -- an innovative, ultra-low power microRISC core designed to power a new generation of portable and mobile applications. Smaller in size and offering full 32-bit performance out of 16-bit memory, M•CORE delivers a high-performance, low-power solution ideally suited for battery-powered and wireless applications, including digital phones, pagers, electronic personal assistants, and for automotive safety products like ABS and Airbags, where system cost and high temperature rages are key variables. Combining performance and cost efficiencies in a compact, low-power design, M•CORE is the natural solution for any application where battery life and system costs are as important as MIPS.

DEVELOPMENT TOOLS AND EVALUATION SYSTEMS

A solid selection of third-party development tools is available for the M•CORE processor, including highly optimized compilers, instruction-set simulators, debuggers, target monitors, software/hardware co-verification tools, real-time operating systems and evaluation boards. The M•CORE architecture is also supported by an Application Binary Interface (ABI) Standard, that defines all the interfaces required by a compiler writer to create a toolchain for M•CORE. This standard allows interoperability between other ABI compliant M•CORE tools. To provide additional support and protection, M•CORE development tools are fully tested to ensure “customer-ready” validation before being released.

M•CORE SIGNAL GROUPS

The M•CORE signal groups are referenced in Figure 2.

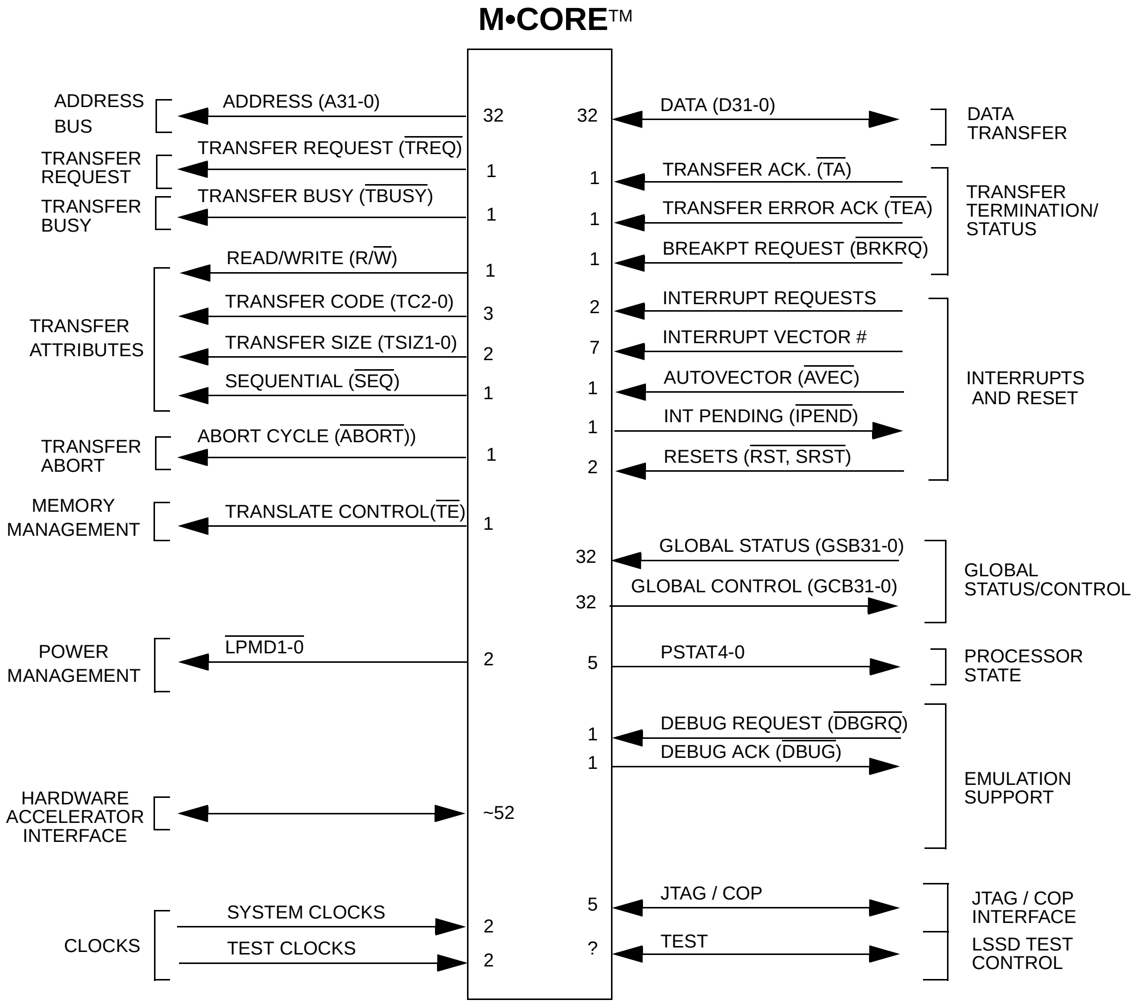

Figure 2. M•CORE Signal Groups

MORE INFORMATION

For more information on the M•CORE architecture please contact your local sales office or *M•CORE Market Development at (512) 342-6974*

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters can and do vary in different applications. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

Literature Distribution Centers:
USA/EUROPE: Motorola Literature Distribution; P.O. Box 20912, Arizona 85036.
JAPAN: Nippon Motorola Ltd.; 4-32-1, Nishi-Gotanda, Shinagawa-ku, Tokyo 141 Japan.
ASIA-PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Center, No. 2 Dai King Street, Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong.