Using the MIPS32® M4K® Processor Core SRAM Interface in Microcontroller Applications

October 2007

MIPS Technologies, Inc.
1225 Charleston Road
Mountain View, CA 94043
(650) 567-5000

Introduction

There are several features of the MIPS32® M4K® core that make it well suited for use in the microcontroller application space. This paper will address one of those features: the SRAM interface, which is a standard feature of the MIPS32 M4K core.

The M4K core does not internally support instruction cache (I-cache) or data cache (D-cache) as standard features because of the size and cost restraints in typical microcontrollers. The microcontroller environment must fit the maximum number of general purpose IO in the smallest package possible.

What is needed for microcontroller applications is an interface that will allow for tight coupling between the processor core and the memory system with a minimum of interface logic. The MIPS32 M4K core SRAM interface is an excellent solution.

M4K Core SRAM Interface: Basic Description

The M4K core SRAM interface is the general-purpose high-speed memory interface to the M4K core. It provides a low latency interface into both the instruction memory and data memory paths. Both single-cycle and multi-cycle memory access is supported.

It must be noted that the SRAM interface does not connect to external memory devices directly but must make use of an external memory controller. The external memory controller, which is usually a section of the general system controller supplied by the chip designer, must make use of the Fixed Mapping Table (FMT) and the SRAM interface to provide the complete memory control logic. The design of this memory controller is simplified since these buses only deal with instruction and data accesses. No other system traffic such as interrupt requests appear on this interface.

A complete description of the entire SRAM interface is beyond the scope of this paper; such a description can be found in the paper, “Working with Dual SRAM for MIPS32® M4K™ Cores Application Note,” available through MIPS Technologies. A description of the basic features of the interface follows.

Dual Mode Operation

The native configuration of the SRAM interface is called Dual Mode. In this mode, the instruction and data paths are isolated from each other. There are independent data read and write buses for data (D-SRAM) along with D-SRAM interface control signals and an independent instruction side (I-SRAM) interface along with its complementary I-SRAM control signals.

The I-SRAM interface has the capability to redirect signal inputs, allowing D-SRAM read cycles to be redirected to the I-side if required. This allows implementation of the modified Harvard architecture, which

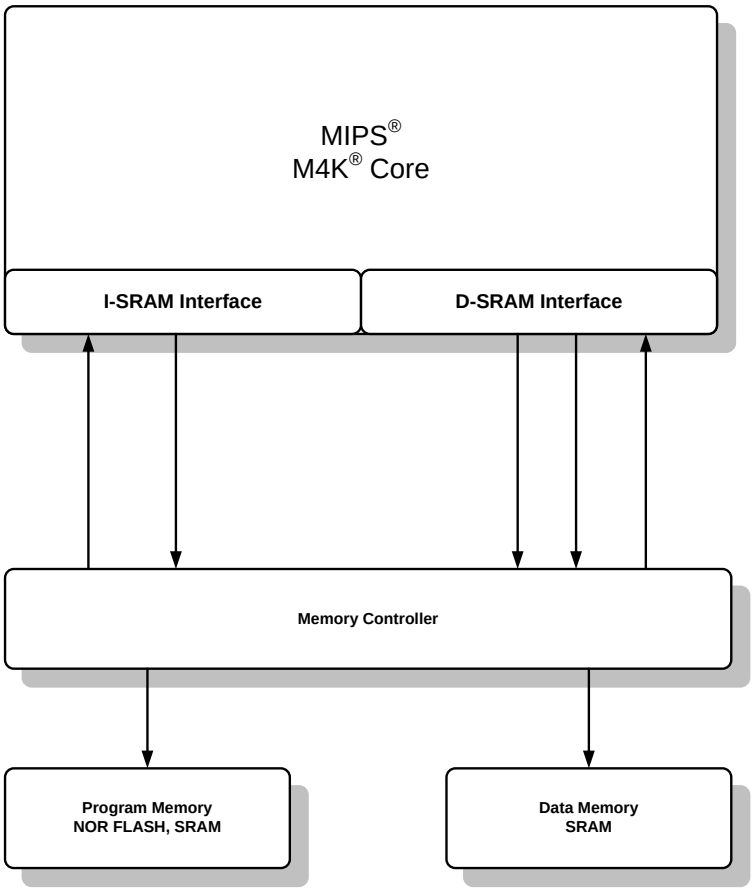
is a normal feature of microcontroller-based systems, allowing non volatile data to be stored in program memory.

Dual Mode allows simultaneous transactions to occur on the I-SRAM and D-SRAM interfaces, removing any stalls that might appear on a common bus interface from slowing program execution. In this mode, the core achieves its nominal performance of 1.5 DMIPS/MHz.

In Dual Mode, there are a total of 209 signals that are exposed by the SRAM interface that must be connected to the external memory controller block. The D-side interface uses just slightly more than half of these signals since it provides both data write and data read paths.

The Dual Mode structure is shown below in Figure 1.0.

Figure 1.0
M4K Processor Core
Dual Mode SRAM Interface



Unified Mode

Unified Mode is a configuration option for a unified interface in which the I-SRAM and D-SRAM signals are merged together to save on the total number of signals that must be routed from the core. In Unified Mode, the D-SRAM interface (apart from the data write bus) is completely disabled and all data read cycles are automatically redirected to use the I-SRAM read bus.

In this mode, the ability to fetch instructions and read data at the same time is not available and any data transaction will take priority over an instruction fetch. Since the program instructions and data share the same bus into the MMU, there is a performance degradation of up to 40% to deal with the arbitration that must happen between the two data paths.

In this mode, the average performance of the core is approximately 1.2 DMIPS/MHz. However, in this mode the total number of active signals exposed by the SRAM interface core is 122, a savings of 87 signals from Dual Mode. Reducing the number of active signal used by the interface makes this a more cost-effective approach in very cost-constrained designs where the total die area is more important than absolute performance.

The SRAM interface Unified Mode is shown in Figure 2.0.

Transaction Abort

The SRAM interface provides the ability to abort an instruction transaction anywhere in the M4K core's 5-stage pipeline. This allows immediate response from the external system controller to external events such as interrupt requests or requests through the EJTAG debug interface. Quick response to external interrupt events is crucial when dealing with the highly deterministic nature of typical microcontroller applications. Many processor cores only acknowledge these events after the instruction pipeline has been fully cleared.

Narrow Device Support

The ability to interface to narrow memory devices via specific byte-lane steering interface signals allows the M4K core to assemble a 32-bit word from successive byte reads, which are then assembled internally within the SRAM interface. This eliminates the need for the external memory controller to provide these external latches or assemble the entire 32-bit word externally.

This function is usually used in combination with the Bus Lock and Bus Stall in interfacing peripheral devices such as analog to digital converters. These types of mixed signal devices generally do not provide 32-bit wide interfaces.

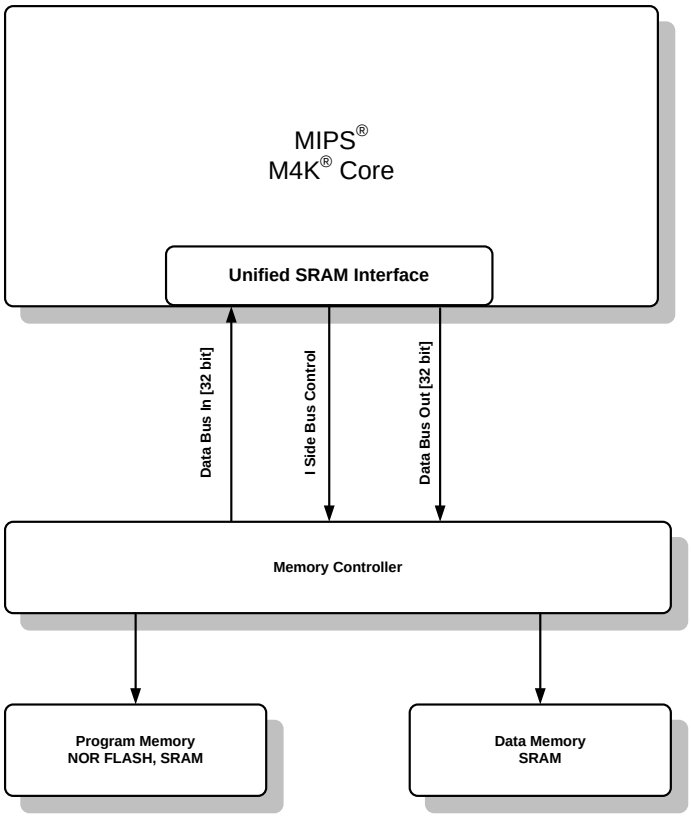
Bus Lock and Bus Stall

The SRAM interface provides input control signals to lock the data bus from further CPU write transactions until the external memory controller de-asserts this signal. This lets the atomic bus operation complete without fear of multi-threaded memory cycle contamination. The SRAM interface also provides control signals that allow the bus to be stalled, so designers can connect slower memory and peripheral devices to the system. These devices may include slower nonvolatile RAM and mixed-signal devices that require additional wait states before being able to present requested data onto the bus.

Cache Coherency Signaling

While the M4K core does not contain any cache memory, it provides the ability to indicate whether or not the current memory address present on the address bus can be cached or not. An external memory controller can make use of these status signals to implement an L2 cache structure.

Figure 2.0
M4K Processor Core
Unified SRAM Interface



M4K SRAM Interface – Application in the Microcontroller Space

With all of these features, designers can recognize multiple benefits from leveraging the M4K SRAM interface in the microcontroller system environment.

First, the tightly coupled interface means that most transactions are completed within a single clock cycle. The exception is when a designer implements bus stalls to allow for slower memory devices to complete the cycle. Control logic connected to this interface must only deal with instruction and data transactions, which makes the entire logic design less complicated. The bus may also be locked to allow for atomic transactions to complete without interference from queued cycles.

Fixed Mapping Table (FMT) also reduces the amount of logic and decoding required in the external memory controller. Apart from the memory-mapped peripheral and absolute size of the memory devices available to the microcontroller, the active regions are within defined boundaries.

In Dual Mode, the instruction fetch path and data read/write path are isolated from each other. These independent data paths allow the memory controller logic to optimize the pathways for the type and size of memory devices.

The ability to abort a transaction anywhere in the pipeline allows for quick response and precise breakpoint control when used in a debug environment.

Basic L2 cache may be implemented using native signaling from the M4K core, simplifying the L2 cache controller since no decoding need be done to determine if the active transaction is within a cached region.

The MIPS32 M4K core SRAM interface provides a high speed, easy to use and highly-configurable memory interface into the M4K core. It contains no additional overhead in protocol or signals to deal with anything but instruction and data memory, enabling chip designers to extract maximum performance out of the M4K core with a minimum of external logic. Ultimately, this capability translates into a higher performing microcontroller system at minimum cost—a critical combination for 32-bit microcontrollers as they increasingly replace lower performing 8-bit microcontrollers.

