MOTOROLA
SEMICONDUCTOR
ENGINEERING BULLETIN

Order this document
by EB165/D

EB165

Hardware Implications of *xmem* as a *st* followed by a *ld*

The MC88110 supports an exchange memory (**xmem**) instruction that is a combination of a load and store instruction. The **xmem** instruction is normally a read access followed by a write access (as it was implemented in the MC88100). However, the **xmem** instruction can also function as a write access followed by a read access if the **xmem** bit is set in the data MMU/cache control register (DCTL). The MC88110 drives the LK signal during an **xmem** operation to indicate that the access is part of an atomic data access sequence. TheLK signal is only asserted during **xmem** transactions.

Generally, the **xmem** instruction is used to perform operations on semaphores. A location in memory is reserved for the semaphore flag. This flag is used to indicate whether a resource (a block of memory, a printer, etc.) is currently being used by another processor. When a processor needs to access the resource, it performs an **xmem** transaction, transferring the old value of the flag into a register and transferring a "set" value into the location of the flag. The processor must then test the value of the flag in the register. If the flag was already set, the processor must wait before accessing that resource.

Figure 1 illustrates a multiprocessing system comprised of two MC88110s with local buses, each connected to a system bus with system memory and system resources. For this system, the two different forms of **xmem** are evaluated.

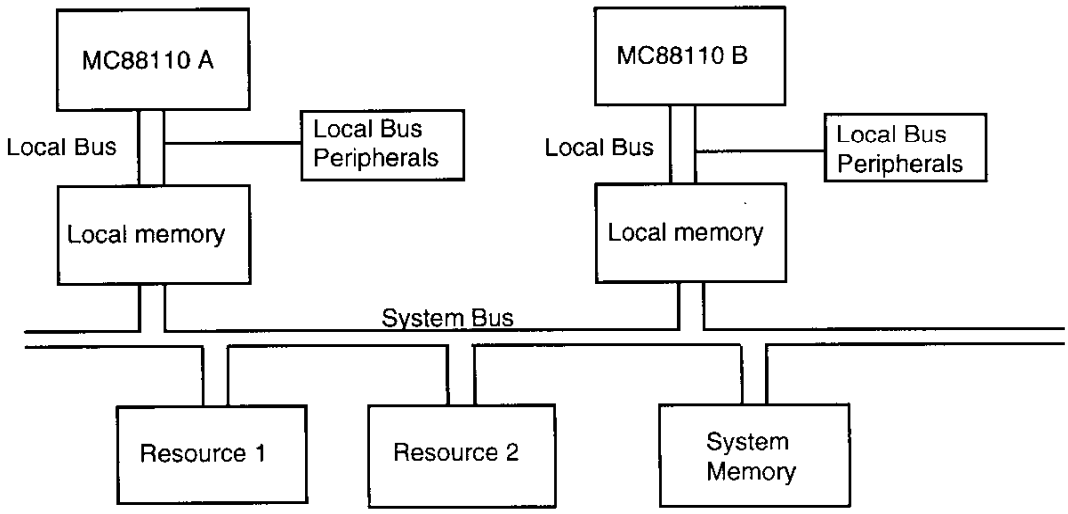

Figure 1. Multiprocessing System

Read-Before-Write xmem

While performing a read-before-write **xmem**, if the system arbiter negates the grant to the initiating processor in between the read and write operations, the value of the semaphore could be changed by another processor between the two accesses. Then, when the first processor completes the write portion of the **xmem**, the value stored in its register would be incorrect. Therefore, with a read-before-write **xmem**, the processor initiating the operation must remain on the system bus until the entire transaction is complete. This can cause a performance degradation by reducing system bus bandwidth.

Write-Before-Read xmem

Using the write-before-read **xmem** option can improve system performance over the read-before-write option by reducing the total time a processor remains master of the system bus in order to access a semaphore. This is accomplished with extra hardware in the system memory controller. This added hardware is explained in the next section. When the initiating processor begins an **xmem** transaction, the memory system latches the address and the write data. Then, the memory system can terminate the processor's bus mastership by negating its grant. The memory system will then perform the **xmem** transaction locally and store the read data in a latch for the processor to access later. In this way, the system bus is not tied up while the memory access is performed. Another device may perform a transaction across the system bus while the memory controller is performing the **xmem** transfer.

The write-before-read option also allows for a performance gain on the MC88110's local bus. After the memory controller has latched the MC88110's address and data and negated the grant to the MC88110, another bus master can access peripherals across the local bus. Note that the MC88110's performance is not mproved because the MC88110 does not run any other transactions across the bus until the **xmem** operation is complete.

EXTRA HARDWARE REQUIRED FOR THE WRITE-BEFORE-READ OPTION

Although the write-before-read option improves system performance, it also requires some extra hardware in the memory subsystem. Figure 2 shows one possible way to implement this extra control with discrete logic. The block labeled DRAM control logic is the control required for the DRAM even without this added option. The latches on the address bus are octal transparent latches. The 74F652s on the data bus are octal, non-inverting transceiver/register. They can transmit data directly from an input bus or from an internal register. Data on the A or B side of the 74F652 is clocked into its internal registers when the appropriate clock pin goes high. The block labeled "Extra Control Logic" is a logic array that uses the LK and R/W signals as inputs and generates the control signals for the 74F652 and the rR/W signal for the DRAM control logic.

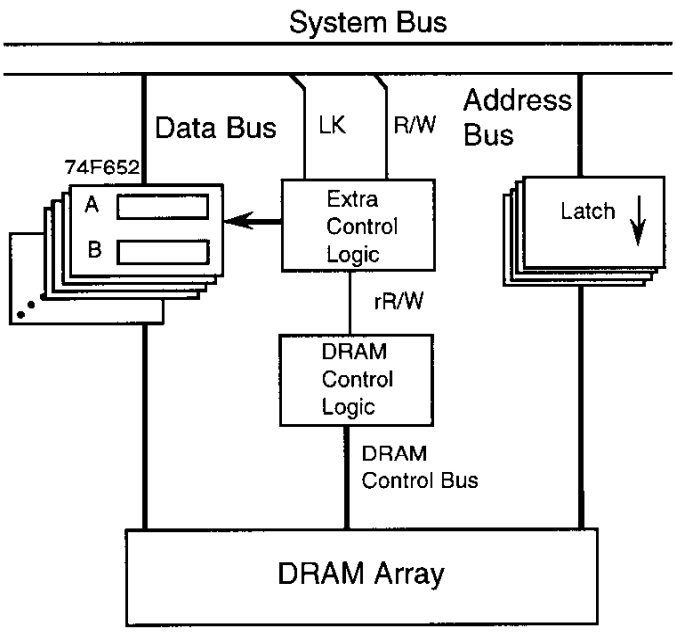

Figure 2. Extra Hardware Required for a Write-Before-Read xmem

The following table shows a list of the different input combinations to the control logic block and the resulting actions taken by the system in response to these inputs. When the LK signal is negated (that is, not an **xmem** transaction) the control logic generates the appropriate signals to the 74F652 to make it seem transparent on the bus. Therefore, if LK is negated and the R/W signal indicates a read, the data on the B side of the bus is transferred to the A side of the bus. For a write, the information on the A side of the bus is transferred to the B side of the bus.

Inputs	Actions
No LK and Read	Transfer B data bus to A data bus, rR/W = Read
No LK and Write	Transfer A data bus to B data bus, rR/W = Write
LK and Write	(1) Latch address, latch data into register B (2) Terminate MC88110's tenure. (3) rR/W = Read, Store B data bus in A latch (4) R/W = Write, Transfer B latch to B data bus
LK and Read	Transfer A latch to A data bus (no DRAM access)

Table 1. Inputs and Actions of the Control Logic Block

When LK is asserted with a write signal from the processor, an **xmem** transaction has started. The address for the **xmem** and the corresponding write data are latched so that the transaction can be terminated and the MC88110 will no longer be tying up the system bus. The data is latched into register B in the 74F652. The rR/W signal sent to the DRAM control logic indicates a read. Once the data is read from memory, it is stored into register A in the transceiver/register. The rR/W signal is then changed to indicate a write. The data previously latched into register B is transferred to the B side of the data bus and written into memory at the location of the semaphore.

Although the initiating processor is no longer the bus master, it still asserts its bus request signal for the second portion of the **xmem** transfer. When it again receives a grant, it will assert the LK signal and indicate

a read. Since the read data has already been latched into the 74F652, the DRAM does not need to be enabled. The extra control logic just transfers the data from the A latch onto the A bus and the **xmem** transaction is complete.

One problem can arise from this configuration. It is possible that, while the initiating processor is requesting the bus, another processor may assume ownership of it and try to read from the semaphore location. If the second processor is allowed to read from memory at this location, the value it reads will be incorrect. Therefore, the extra control logic must monitor all transactions to memory. If a read of the semaphore address is attempted while the control logic is in the middle of a semaphore, that read must be retried until the proper value has been loaded into the semaphore location.

Literature Distribution Centers:

USA: Motorola Literature Distribution: P.O. Box 20912; Phoenix, Arizona 85036.
EUROPE: Motorola Ltd.; European Literature Center; 88 Tanners Drive Blakelands, Milton Keynes, MK14 5BP, England.
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.