



MOTOROLA INC.

Microprocessor Products Division
M68000 Applications Engineering
3501 Ed Bluestein Blvd.
Austin, Texas 78721

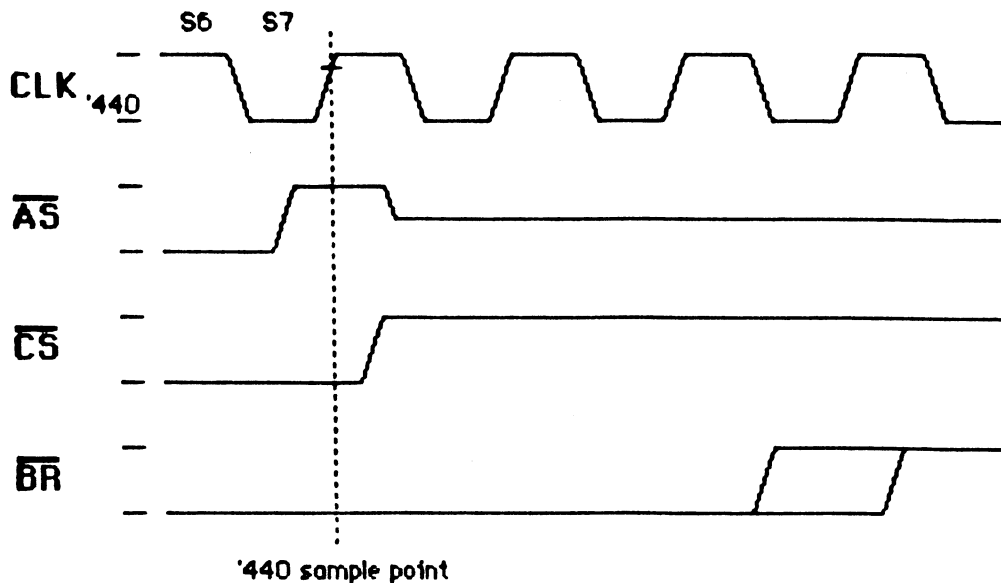
**MC68440 RS8 Mask
Errata Sheet
November, 1984**

The following is an explanation of three problems that exist on the current mask set (RS8) version of the MC68440 Dual Direct Memory Access Controller (DDMA). This mask set is available under the part number SC87876.

Problem #1

If the DDMA is being accessed in the MPU mode (either during a chip select or an interrupt acknowledge), is simultaneously requesting and has been granted the bus, and the negation of \overline{CS} or \overline{TACK} is delayed from the negation of \overline{AS} , then the channel that is causing the bus request may be aborted erroneously with an address error exception flagged in the corresponding Channel Status and Error Registers.

The signal timing that will cause this error is shown in the diagram below. The problem occurs due to the fact that the DDMA will attempt to take control of the bus as soon as it detects \overline{BG} asserted and \overline{AS} negated internally. Since \overline{CS} missed the setup time to the rising edge of the clock on which \overline{AS} negated was sampled, when the DDMA starts to service the channel that was requesting bus ownership, \overline{CS} will still be asserted internally. The DDMA reacts to this situation as though it had chip selected itself, which is defined as an Address Error for the channel being serviced.



There are several solutions to this problem that utilize the same technique; to prevent the DDMA from assuming bus mastership while \overline{CS} or \overline{IACK} is internally asserted. Some possible fixes are listed below.

- 1) Guarantee that the negation of both \overline{AS} and $\overline{CS}/\overline{IACK}$ meet the input setup time to the same rising edge of the DDMA clock; thus, when the negation of \overline{AS} and the assertion of \overline{BG} is detected internally, the internal $\overline{CS}/\overline{IACK}$ will also be negated.
- 2) Negate \overline{BG} to the DDMA while \overline{CS} or \overline{IACK} is asserted; thus forcing the DDMA to synchronize the assertion of \overline{BG} and the negation of $\overline{CS}/\overline{IACK}$ at the same time. This assumes that there will be some delay from the negation of \overline{CS} to the assertion of \overline{BG} so that the internal \overline{CS} will be negated by the time the internal \overline{BG} is asserted.
- 3) Assert the Halt bus exception code to the DDMA during \overline{CS} or \overline{IACK} cycles. This will force the DDMA to synchronize and debounce the \overline{BEC} pins before it takes control of the bus. The same effect will occur if a \overline{BEC} encoding of 5 or 6 (the undefined codes) is asserted instead of the Halt. Thus, in systems using an 'LS148 to generate the \overline{BEC} encodings, the unused 5 and 6 inputs to the 'LS148 can be tied to \overline{CS} and \overline{IACK} , respectively, for a quick and easy fix.
- 4) If an address error occurs for a channel when the MAR or DAR contains a valid address for the operation programmed into the channel, simply

clear the channel error register and restart the channel to continue the operation. This may not be feasible, since a request from a device may have been lost.

This problem will be fixed in the next mask set by forcing the DDMA to wait until \overline{AS} , \overline{CS} and \overline{TACK} are all negated before it assumes bus mastership. If any of the above fixes is implemented by external hardware, future mask versions of the DDMA will operate properly in the system.

Problem #2

The second problem is similar to the first in that it also causes an incorrect address error. Two configurations may cause this error.

Channel configuration A:	16-bit memory port size 8-bit device port size 16-bit operand size Dual address transfer Device to memory
Channel configuration B:	16-bit memory port size 16-bit device port size 8-bit operand size Dual address transfer Auto-request (maximum or limited) Odd initial MTC value

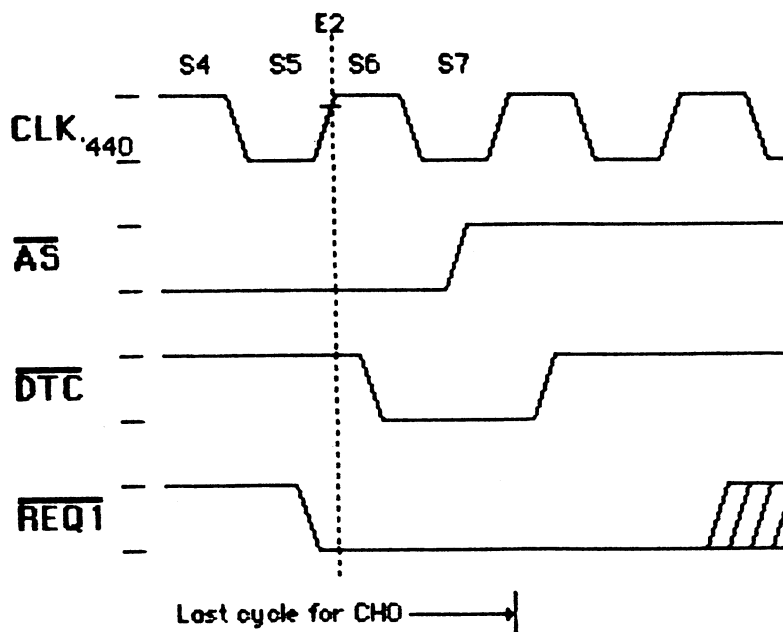
When a channel configured as shown above (subsequently referred to as the primary channel) is terminated, it is possible that it will be left in an illegal state. The illegal state will not cause an error on the primary channel since it is generated during channel termination; but it may cause an error on the other channel (subsequently referred to as the secondary channel) if a particular set of circumstances occurs.

For configuration A, the DDMA will always read two operands (bytes) from the device before writing them as a word to memory. However, if the first byte read from the device is accompanied by the assertion of \overline{DONE} by the device, the DDMA will immediately write the byte to memory and terminate the operation. Also, the MAR for the channel will be left at an odd address, which causes the channel to detect an address error that

will be ignored due to the channel termination. If an attempt were made to restart the channel without modifying the MAR, then an address error would be signalled for the channel and the operation would be aborted.

For configuration B, the MAR and DAR must both be initialized with even values, since the DDMA will perform transfers of two operands (bytes) at a time to reduce bus utilization. However, if the initial MTC value is odd, the DDMA will perform a single byte read followed by a single byte write during the last transfer. As in the configuration A case, this will leave the MAR (and the DAR) at an odd address so that an address error will be detected internally and ignored. If an attempt were made to restart the channel without reinitializing the MAR and DAR, then an address error would occur for that channel.

In addition to the occurrence of the conditions described above for the primary channel, if the secondary channel of the DDMA is configured for external request generation and has a "late" request, in either the burst or cycle steal mode, at a precise time, then improper operation will occur. In order to cause this error, a request for the secondary channel must not be detected until E2 (refer to page 4-27, figure 4-21 of the MC68440 Advanced Information Data Sheet, ADI-1002) of the last bus cycle executed to service the primary channel as shown below.



Now for a description of the actual failure mode. If either of the above sets of circumstances occurs, the secondary channel will be terminated incorrectly with an address error exception signalled in the CSR and CER. This is due to a speed path within the DDMA that causes the address error that is detected by the primary channel, but ignored, to be taken as an address error for the secondary channel. There are several possible fixes for this problem, as listed below.

- 1) Do not allow late requests on the secondary channel during the last bus cycle to service the primary channel. This is not feasible in asynchronous systems, but it points out the fact that this error will occur only if a late request occurs during the last cycle for the primary channel.
- 2) Do not use odd transfer counts, either in the device for configuration A, or in the MTC for configuration B. Again, this is not always feasible, particularly for data communications devices using asynchronous protocols.
- 3) When the secondary channel is terminated with an address error, simply clear the error and attempt to restart it. If the address error does not recur immediately, then it can be assumed that the primary channel conditions caused the address error. This method may not be feasible, since the request from the device for the secondary channel may have been lost.

This problem will be fixed in the next mask set.

Problem #3

The V_{IH} specification for the CLK pin is 2.4 Volts rather than 2.0 Volts. The next mask set will meet the 2.0 Volt specification.