MU9C4K64 Errata Sheet DRAFT

Errata Sheet for Revision A of the MU9C4K64 Routing CoProcessor

Device
MU9C4K64 Routing CoProcessor
Product Revision: A

Problem 1
When executing a "WRs [NFA]{MRn}" or "MOV [NFA],CR{MRn}" control state that writes to the last empty location of a device that is not the Lowest Priority device in a daisy chain (LPC = 1), the PA:AA bus outputs F:FFF (hex) instead of the NFA.

Scope
Problem 1 affects any application that:

- Writes the associated data into an external RAM using the PA:AA bus to drive the RAM address pins; AND
- Configures two or more MU9C4K64 devices into a daisy chain.

Applications that use a single MU9C4K64 are not affected (the LPC must be set to 0).

Fix
No silicon changes are planned for this problem.

Workaround
The following workarounds may be used to avoid Problem 1:

1. Use the MU9C8K64 instead of multiple MU9C4K64 devices. The MU9C8K64 does not exhibit this problem.

2. For applications that write to both segments of the memory, use the PA:AA bus values that are the result of the first WRs [NFA] cycle instead of the second, as shown in the following example:

   WRL [NFA] (Write the LOW segment to memory, do not set valid, use PA:AA)
   WRH [NFA] (Write the HIGH segment to memory, set valid, do not use PA:AA)

   Note that the segments can be used in any order. This method does not require any additional cycles.

3. For applications that write to one segment of the memory with the WRs [NFA] control state, use one of the following sequences:

   RD NFA (Read the Next Free Address)
   WRs aaa (Place the NFA on the AC bus and write the entry into the memory at that address)

   or

   RD NFA (Read the Next Free Address)
   WR AR (Write the Next Free Address into the Address register)
   WRs [AR] (Write the entry into the memory location pointed to by the Address register)

   The "WRs aaa" and "WRs [AR]" control states will place the address where the entry is written onto the PA:AA bus. Although these sequences take more cycles, the cycles may be separated from each other, allowing higher priority compares to be interleaved. Because of the relatively low incidence of table updates in most applications, there should be little or no impact on the overall system performance.

4. To learn new entries with the "MOV [NFA]" control state, the following sequence should be used:

   WRL CR (Write to one segment of the Comparand register)
   CMPWH DQ (Write DQ to the other segment of Comparand register and compare)

   if no match, then continue

   MOV NFA,CR (Move the Comparand register to the Next Free Address)
   CMP CR (Compare the contents of the Comparand register)

   The "CMP CR" control state will match the entry that was learned. The HPMA value that is placed on the PA:AA bus by this control state is the NFA to where the entry was learned. Note that the segments can be used in any order. This sequence is one compare cycle longer than the standard learn sequence.

Note: This document version has not completed MUSIC's internal approval process, therefore it should be rechecked with a released version or with factory personnel.

Problem 2

When executing a "NEXT" or a "RDs [HPM];NEXT" control state that is preceded by any other instruction that changes the PA:AA bus (including the "WRs [HPM]{MRn}" control state), the PA:AA bus will output F:FFF (hex) or the last accessed address instead of the address of the first matching location in each device that does not contain the Highest Priority match for that daisy chain.

Scope

Problem 2 affects any application that:

- Uses the PA:AA bus to drive the address pins of an external RAM for reads of associated data; AND
- Reads the associated data for multiple matches using the "NEXT" or "RDs [HPM];NEXT" control states; AND
- Configures two or more MU9C4K64 devices into a daisy chain.

Applications that use a single MU9C4K64 are not affected. Applications that do not allow multiple matches or that use only the Highest Priority match are not affected.

Fix

No silicon changes are planned for this problem.

Workaround

Any one of the following workaround alternatives will avoid Problem 2:

1. Use the MU9C8K64 instead of multiple MU9C4K64 devices. The MU9C8K64 does not exhibit this problem.
2. Use the "NEXT" or "RDs [HPM];NEXT" control states consecutively (for example, no other control states between them).
3. Between consecutive "NEXT" or "RDs [HPM];NEXT" control states, use only control states that do not change the PA:AA bus (for example, are listed as n/c in the Control State Overview table, such as NOP and register reads and writes).
4. If it is necessary to update the matching entry using the "WRs [HPM]" or "MOV [HPM],CR" control states, use the PA:AA bus values that are the result of the WRs or MOV cycle instead of the NEXT or RD cycle.

Additional Information

All releases of the MU9C4K64 RCP VHDL and Verilog models match the operation of the Rev. A silicon for Problem 1 and Problem 2. As a result, any version of these models may be used to test if an application is susceptible to the problems described above, and to test the effectiveness of any proposed workaround.

MUSIC Semiconductors' agent or distributor:

Worldwide Headquarters
MUSIC Semiconductors
5850 T.G. Lee Blvd, Suite 345
Orlando, FL 32822
USA
Tel: 407 850-1035
Fax: 407 850-1063

North American Sales
MUSIC Semiconductors
495 Union Ave., Suite 1B
Middlesex, NJ 08846
USA
Tel: 732 469-1886
Fax: 732 469-2397
USA Only: 800 933-1550 Tech Support
888 226-6874 Product Info

Asian Headquarters
MUSIC Semiconductors
110 Excellence Ave., SSEPZ 1
Carmelray Industrial Park
Canlubang, Laguna
Philippines 4028
Tel: +63 49 549-1480
Fax: +63 49 549-1024
Sales Tel/Fax: +632 723-6215

European Headquarters
MUSIC Semiconductors
P. O. Box 184
6470 ED Eygelshoven
The Netherlands
Tel: +31 43 455-2675
Fax: +31 43 455-1573

http://www.musicsemi.com email: info@musicsemi.com