# PCI 9080RDK-860

Design Note
July 6, 1998

## A. Product Status

The scope of this document encompasses the PCI 9080RDK-860 Rev 2.0

| Product status | Description | Production |
|---|---|---|
| PCI 9080RDK-860 | Production release | July 1998 |

## B. PCI 9080RDK-860 Design Note

### BDM Operation with tools that support the FRZ# Pin:

Design Issue:
The BDM port on the MPC860 is compatible with two different industry tools. The default connector on the 9080RDK-860 supports tools that use the FRZ# pin. However, before using the BDM port you must configure the BDM multiplexed pins correctly. The BDM pins are configured by bits 11 and 12 of the SIU Module Configuration Register (SIUMCR) (Page 12-15, section 12.4.1.1). Bits 11, and 12 must be programmed to b'11.

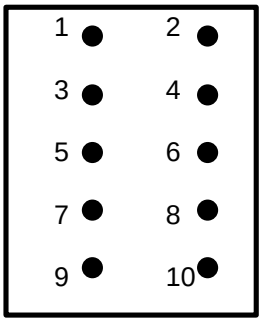
This register can be programmed by IOP software or programmed by the Hard Reset Configuration Word at reset. To program them at reset, MPC860 data lines D11, and D12 should be pulled high using a 1K resistor. Alternatively, the register can be programmed via IOP code.

It is recommended that data lines D11 and D12 be pulled high by connecting two 1K resistors between 5VCC and connector J2 pin 21 and pin 20 respectively. The resistors can be conveniently added to the solder-side of the board and a good source point for 5VCC is connector J1 pin 10. Pin 10 of connector J1 can be located by first locating pin 1. When looking at the solder side of the PCB with the gold fingers pointing downwards, pin 1 is the bottom left pin of connector J1. The pins alternate with pin 2 being directly above pin 1. Therefore, pin 10 is the 5$^{th}$ pin from the left in the top row of connector J1.

### BDM Operation with tools that support the VFLS[0:1] Pins:

The BDM port on the MPC860 is compatible with two different industry tools. The default connector on the 9080RDK-860 supports tools that use the FRZ# pin. To add support for tools that use the VFLS[0:1] pins you must add an additional connector to the prototype area. The new connector should have the pin-out as described below. Each pin below also contains a reference number to a device that the signal can be obtained from. For example, U19.1 refers to device U19 pin 1. Resistor pins are referenced as either T (top pad), B (bottom pad), or R (right pad) when viewing the board with the gold fingers pointing downward:

| | |
|---|---|
| Pin 1: VFLS0 (R59.T) | Pin 2: SRESET (R44.B) |
| Pin 3: GND (U19.1) | Pin 4: DSCK (R69.B) |
| Pin 5: GND (U19.1) | Pin 6: VFLS1 (R60.T) |
| Pin 7: HRESET (R43.B) | Pin 8: DSDI (R65.B) |
| Pin 9: 3.3 VCC (U19.2) | Pin 10: DSDO (R42.R) |




**New BDM Header (Standard 0.1" spacing)**

Before using the BDM port you must also configure the BDM multiplexed pins correctly. The BDM pins are configured by bits 9, 10, 11 and 12 of the SIU Module Configuration Register (SIUMCR) (Page 12-15, section 12.4.1.1). Bits 9, 10, 11, and 12 must be programmed to b'1111.

This register can be programmed by IOP software or programmed by the Hard Reset Configuration Word at reset. To program them at reset, MPC860 data lines D9, D10, D11, and D12 should be pulled high using a 1K resistor. Alternatively, the register can be programmed via IOP code.

It is recommended that data lines D9, D10, D11 and D12 be pulled high by connecting four 1K resistors between 5VCC and connector J2 pins 23, 22, 21, and 20, respectively. The resistors can be conveniently added to the solder-side of the board and a good source point for 5VCC is connector J1 pin 10. Pin 10 of connector J1 can be located by first locating pin 1. When looking at the solder-side of the PCB with the gold fingers pointing downwards, pin 1 is the bottom left pin of connector J1. The pins alternate with pin 2 being directly above pin 1. Therefore, pin 10 is the 5$^{th}$ pin from the left in the top row of connector J1.

## IDMA Operation:

Design Issue:
During IDMA transfers to and from the PCI bus, the PCI 9080, like any PCI interface device, may need to back-off the MPC860 by means of the RETRY~ pin. Back-off will occur:

a) to solve possible deadlock situations when another PCI master is attempting to access devices on the 9080RDK-860, while the MPC860 is performing a direct master operation, or
b) when the PCI bus is heavily loaded and the PCI 9080 needs to inform the MPC860 of a full FIFO during direct master write transactions.

When IDMA is used, it is possible that the RETRY~ assertion can cause an internal transfer error acknowledge (TEA~) from the MPC860's memory controller to its PowerPC core.

Solution1:
According to the Motorola MPC860 User's Manual (Page 13-45, section 13.5.10.1): "If a burst access is acknowledged on its first beat with a normal TA~, but with the BI~ signal asserted, the following single beat transfers initiated by the MPC860 to complete the 16 byte transfers recognizes the RETRY~ signal assertion as a transfer error acknowledge"[1]. This situation exists with IDMA, due to its burst cycle being split into single beat transfers by the BI~ signal. The RETRY~ assertion results in the cycle aborting, and "Any data that was previously read from the source into the internal storage is lost"[2] (Page 16-97, section 16.11.3.14.2). It is recommended

[1] _____ (Motorola, Inc., 1996), Section 13.5.10.1 (page 13-45).
[2] Ibid., Section 16.11.3.14.2 (page 16.97).

that designers use the PCI 9080 on-chip DMA controllers instead of the MPC860 IDMA controller for transfers to and from the PCI bus. This achieves higher performance, and completely solves the Design Issue described above.

Solution2:
If designers still need to use the MPC860 IDMA controller, they should limit the amount of PCI traffic to the PCI 9080 chip (i.e. Direct Slave transfers). IDMA transfers will function correctly if deadlock situations can be avoided (Direct Master and Direct Slave at the same time).

**Note:** there are no design issues with using IDMA for local to local transfers.

## Bursting Operation:

The following section is relevant for designers who wish to operate the PCI 9080RDK-860 with bursting capability. The PCI 9080 should be programmed to burst 4 mode and prefetch 4 mode, please refer to the PCI 9080 databook section 2.2.3.2. The following issues are due to the MPC860 not able to burst 1, 2 or 3 long words at a time.

### Direct Slave Write with Bursting Enabled:

Design Issue:
The MPC860 burst size is always fixed to 16 bytes long (consult MPC860 User's Manual MPC860UM/AD Page 13-33 for more info). The PCI 9080 is accessed as a 32-bit port and hence the bursts must be 4 beats in length. If the PCI 9080 cannot access the PCI bus to complete the data transfer it will burst whatever is left in the Direct Slave FIFO. This may result in bursts of 2 or 3 beats, which the MPC860 cannot accept.

Solution 1:
In systems where the burst rate on the PCI bus is controllable, limit bursts to multiples of 4 long words up to 32 in total.

Solution 2:
In systems where the burst rate cannot be controlled, Direct Slave bursting must be disabled. (single cycle mode should be used, refer to PCI 9080 databook section 2.2.3.2.)

### Direct Slave Read with Bursting Enabled:

Design Issue:
The MPC860 burst size is always fixed to 16 bytes long (consult MPC860 User's Manual MPC860UM/AD Page 13-33 for more info). The PCI 9080 is accessed as a 32-bit port and hence the bursts must be 4 beats in length. Depending on PCI bus availability (traffic level), the PCI9080 may burst read whatever free space is left in the Direct Slave Read FIFO. This may result in bursts of 2 or 3 beats on the local bus.

Solution 1:
Enable Direct Slave prefetch and set the limit to 4. This will guarantee all bursts arrive as 4 beats.
Solution2:
Perform PCI burst read of 4 long words at a time.

**Direct Master with Bursting Enabled:**

Design Issue:
The MPC860 general-purpose chip-select machines (GPCM) do not support bursting (consult MPC860 User's Manual MPC860UM/AD Page 15-73). The MPC860 accesses PCI 9080 Registers, Direct Master Memory, and Direct Master IO by using the MPC860 GPCMs. This results in these features being non-burstable.

Solution:
Use the PCI 9080 DMA controller to transfer data from the local bus to PCI bus or from PCI to local bus to obtain burst performance.

**DMA Bursting:**

Design Issue:
The MPC860 burst size is always fixed to 16 bytes long (consult MPC860 User's Manual MPC860UM/AD Page 13-33 for more info). The PCI 9080 is accessed as a 32-bit port and hence the bursts must be 4 beats in length. If the PCI 9080 cannot access the PCI bus to complete the data transfer it will burst whatever is left in the DMA FIFO. This may result in bursts of 2 or 3 beats.

Solution:
Enable the PCI 9080 Local Bus Latency Timer with a value of 0xC. This will cause the PCI 9080 to release the local bus during the DMA transfer. As a result, the FIFOs have time to empty and the PCI9080 never bursts in less than 4 beats, unless the DMA transfer size was not a multiple of 4 long words. The DMA threshold register should also be set to 0x2 to force the PCI 9080 DMA controller to wait until enough data is collected in the DMA FIFO before writing to the local bus (MPC860 bus).