Port Card Contoller (PCC) Hardware

Jim McLaughlin

jmclaugh@bnr.ca

Bell Northern Research

PCC / Port Card Hardware Interface (HWIO)

The HWIO is a FLEX device which interface the PCC's i960 microprocessor and the hardware blocks of the Port Card. The HWIO takes care of all control lines between itself and the hardware blocks, and is in turn controlled by the PCC(see Figure 1).

Because of pin constraints, the PCC and the Port Card hardware need to be interfaced over substantially fewer lines than would a full 32 bit address and data interface.

Many of the registers and memories in the Port Card hardware have limited access times because they are busy performing their real-time operations for most of a cell time. Because of this, reading from one of these locations will deliver data in between three and four cell periods. A direct fetch to the Port Card will take slightly less than 4 cell periods. A prefetch will get data in the same amount of time as the direct fetch, however the CPU will not be held during the period of time when the data is being retrieved.

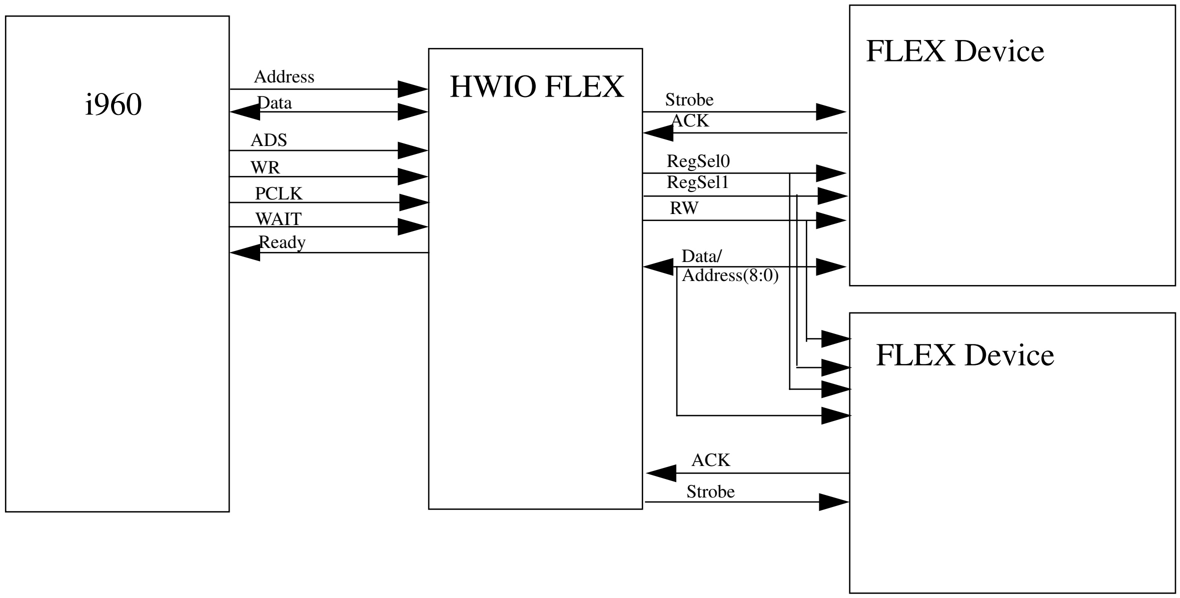
To help improve overall access time, a prefetch has been implemented. Because of the way the hardware blocks have been configured, there is a limit of one prefetch request per block at any given time.

The software must take great care when using the prefetch function. If the read following a prefetch is done before the data is ready, the CPU will be held until it is ready. If the read following a prefetch is to a different memory location, the CPU will be held until the first prefetech is completed, and will continue to hold the CPU until a second fetch for the new location has been completed. In this case, data might not be returned for up to eight cell periods.

The interface to each of the hardware blocks is achieved by 14 I/O pins, allowing prefetches to be kept independant. Following is a list of the 14 I/O pins and their functions:

- 9 bits for multiplexed address and data
- 1 bit read/write select
- 1 bit strobe
- 1 bit ACK
- 2 bits register select (1 for LSB/MSB select and 1 for address/data select)

Figure 1 PCC/ Port Card Hardware Interface



The Data Transfer Mechanism - PCC to HWIO

The interface to the PCC is handled via 18 bit data and 24 address lines. Accesses to the HWIO will appear to the processor as a plain 8/16/32 bit memory accesses, which sometimes have a number of wait states added (see Figure 2 and Figure 3).

Figure 2 Read With Internal and External Wait States Added

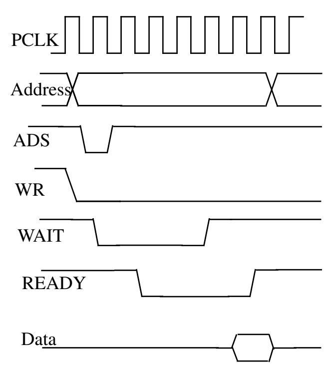
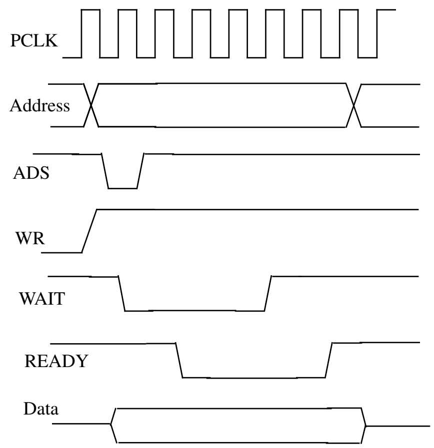


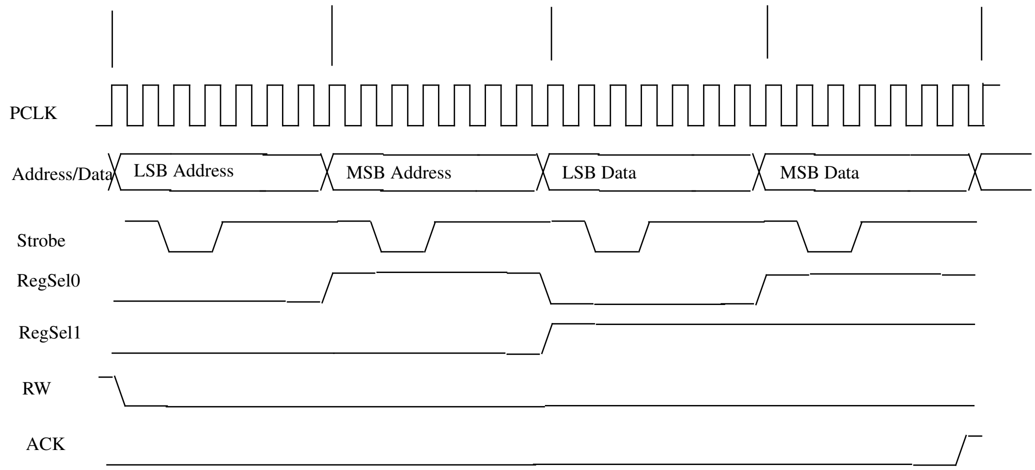
Figure 3 Write With Internal and External Wait States Added



The Data Transfer Mechanism -HWIO to H/W

When the PCC initiates a read or write operation, the HWIO has to transfer the request to the hardware. For a read operation, the HWIO will transfer the address (9 bits at a time) and then force the CPU to wait until the HWIO receives the ACK and read the data from the block. For a write operation, the HWIO must transfer both the address and the data (again 9 bits at a time) but does not force the CPU to wait for the block to ACK receipt of the write request (see Figure 4).

Figure 4 HWIO Write Cycle

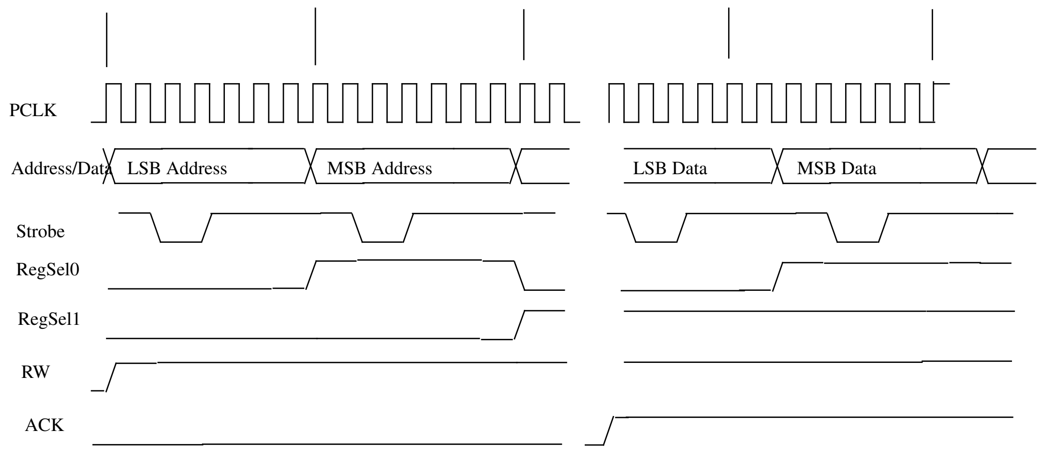


The Register Select lines are used to signal to the block which part of the address or data is being sent to or from HWIO. There are 2 bits of select lines to allow for 4 different transfers. Values 0-1 are used to signal the transfer of the address, and values 2-3 are used to signal the transfer of data:

Register Select = 0 -> address 0-8
Register Select = 1 -> address 9-17
Register Select = 2, -> data 0-8
Register Select = 3, -> data 9-17

There is one strobe line and one ACK line per FLEX device. The strobe selects the correct device and also strobes in/out the address and data during read/write operations. The ACK line is used by the FLEX device to signal the HWIO that the read/write request has been completed. In a read operation, once the block has finished getting the requested data, it will signal the interface hardware using the ACK line. The HWIO will then signal the block with the R/W signal and the Reg Sel lines for reading, and then strobes for both of the 9 bit blocks to be read (see Figure 5).

Figure 5 HWIO Read Cycle



The Prefetch Read Request

One prefetch request will be allowed at a time to each of the 3 blocks in the port card hardware (CM, DM, IPATH/ EPATH). There are two addresses for the same location in the Port Card hardware. One for the prefetch where the MSB of the address is set high, and the other is for the fetch, where the MSB of the address is set low. i.e. the MSB can be defined as the prefetch/fetch bit.

A read to the "shadowed" address of the prefetched address will initiate the prefetch request. e.g. to read data from memory location $8f368 using the prefetch read request, the PCC initiates a prefetch to address $0f3b8, then a read would be done to address $8f3b8.

During a prefetch request, the HWIO will return a flag in the "read" data (bit 0) to signal to the PCC if another prefetch to the same block is still in progress. When this bit is set, it means that the prefetch request was not allowed, as there will be no queuing support for prefetching in the hardware. If no prefetch is in progress, it will not set this bit in the read data, and start the prefetch process. *It will be necessary for the software to check this bit when utilizing the prefetch, or to ensure that two back to back prefetch requests do not occur.*

The interface hardware will initiate the process of shifting the address to the appropriate block, and then watch the ACK line to begin clocking the data back from the block when it is ready. It will internally set the prefetch in progress bit before shifting out the address and clear it when the last of the 9 bit blocks of data is read from the block.

When an actual read request comes from the PCC, the interface block will compare the fetch address to the prefetch address for the last access to that block. If it is the same address except for address bit 19, then it will transfer the prefetched data to the PCC. If the data is not ready but the fetch request is for the same address as the prefetch, the

HWIO will hold the PCC by employing wait states until the data is ready and will transfer it at that time. If the fetch request is for a different address than the prefetch, the data will be refetched. The HWIO will hold the PCC until the read is completed by the block and then transfer the data. *The real-time software needs to ensure that prefetch/fetch pairs are properly managed, otherwise the benefit of the prefetch will be lost. If this error happens a read operation may take as long as 8 cell periods.*

When a hardware block is in the middle of a prefetch request, a subsequent write request from the PCC will not be held until the prefetch has completed. Instead, the HWIO will accept the request, and invalidate the prefetched data when it arrives. The HWIO will then transfer the write data to the block.

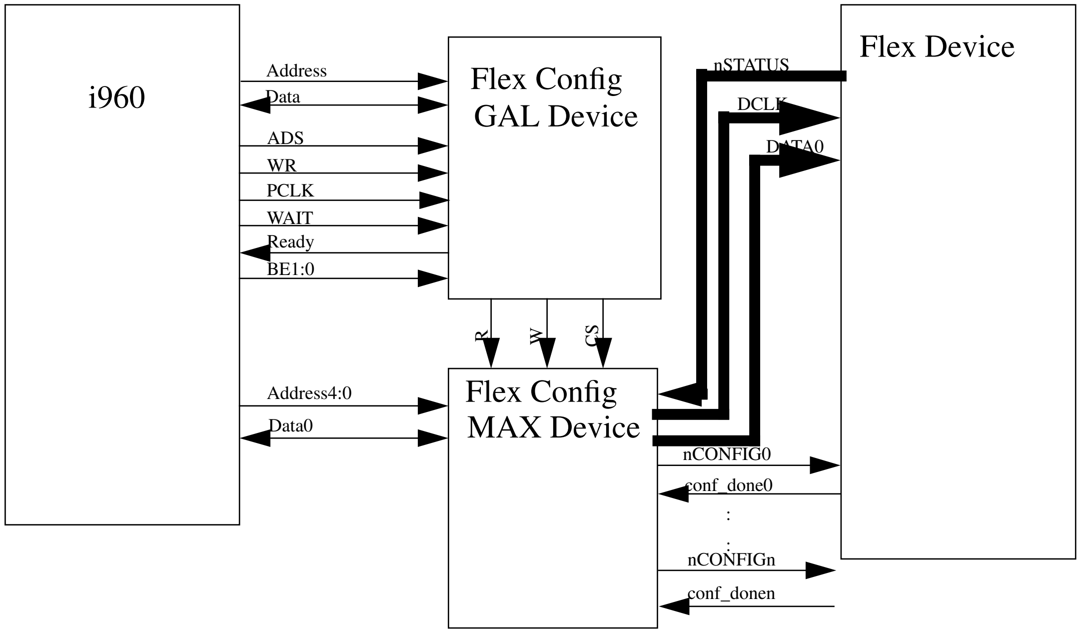
Timeouts

When a read request arrives at the HWIO and there is already a request in progress, the processor is held until the request has finished. Timeouts are needed to prevent the CPU from waiting forever for a bad memory cycle to finish. A programmable timer will watch the number of wait states added to a memory cycle. If it takes longer that the preprogrammed number of cell times (0-31),then the HWIO will release the PCC and then it will toggle an interrupt line. This kind of memory problem is a catastrophic one and the processor will have to deal with it appropriately.

PCC Configuration Of FLEX Devices

Since the FLEX devices are static RAM based, ie. volatile, upon power up, the FLEX devices must be configured. Most of the CPU bus control circuitry is contained in the FLEX devices, but there will be a small amount of non-volatile programmable logic to allow configuration of the FLEX devices. To be able to program or reprogram individual FLEX devices at any time, a more complex interface to the devices is required(see Figure 6).

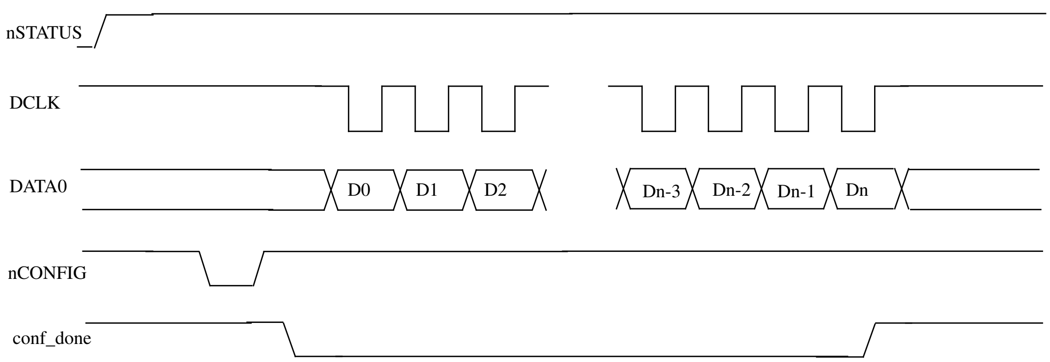
Figure 6 **FLEX Config Block Diagram**



There are 5 dedicated configuration pins on each FLEX device. They are as follows::
nStatus : common signal to all devices - active low signalling download error
DCLK : common signal to all devices - the strobe for Data0
Data0 : common signal to all devices - the data bit being loaded during download
nCONFIG : dedicated signal per device - active low signal to start reconfiguration
conf_done : dedicated signal per device - bit is set by FLEX after successful download

The configuration pins need to be controlled as shown in Figure 7.

Figure 7 Passive Serial Configuration Timing Diagram (MAX interface to FLEX)



Software For Programming FLEX

The software will have full control over the 3 write lines (DCLK, DATA0, nCONFIG) and the other 2 read lines (nSTATUS, conf_done) of the FLEX. To configure a device, the software toggles the nCONFIG line, monitor the nSTATUS line, write a bit to the Flex and toggle the DCLK line. To access these configuration lines, the software will write to the locations in the following table. For a detailed description for programming the FLEX refer to the "FLEX Configuration Design Document" [1]

Configuration Pins	Memory Address
DCLK1, DCLK0	1000 0080H
DATA1, DATA0	1000 0084H
conf_done1, conf_done0	1000 0088H
nstatus1, nstatus0	1000 008CH
nconfig8, nconfig0	1000 00A0H
nconfig9, nconfig1	1000 00A4H
nconfig10, nconfig2	1000 00A8H
nconfig11, nconfig3	1000 00ACH
nconfig12, nconfig4	1000 00B0H
nconfig13, nconfig5	1000 00B4H
nconfig14, nconfig6	1000 00B8H
nconfig7	1000 00BCH

PCC HWIO Loopback Testing

The HWIO block will have loopback testing capability via a register in the HWIO block to enable the loopback mode.

A write cycle will need to be follwed by 4 reads to the following specific registers to verify that the address and data are correct.

There may also be an external loopback mode to the Port Card Hardware Blocks, which will be transparent to the HWIO block.

Loopback Function	Memory Address
HWIO Lpback Mode	1000 0040H
Not Used	1000 0044H
Not Used	1000 0048H
Not Used	1000 004CH
Address Lpbk Read LSB	1000 0050H
Address Lpbk Read MSB	1000 0054H
Data Lpbk Read LSB	1000 0058H
Data Lpbk Read MSB	1000 005CH

PCC / MCP Interrrupts

The level of interrupt from the PCC which is allowed by the MCP is to be programmable by the MCP. It will write to a 3 bit register the level of the lowest priority interrupt that it is willing to accept at the present time. This register is located on the Port Card. The level of allowed interrupt for a specific PCC will not necessarily be the same as all the other PCCs in the system.

To generate an interrupt, the PCC will write three bits into registers located in the Port Card to set the PCC to MCP IRQ level, then toggle another bit in the register. The output of the register will go to the IRQ pin on the backplane and utimately to the MCP. As well, the PCC will be able to read the 3 bit "mask" that was written by the MCP. The output of the PCC's register will be 3 lines holding the priority of the interrupt requested. This value will be compared in the IRQCTRL hardware to the value of "mask" written by the MCP. If it is lower priority, then the interrrupt line to the MCP is not affected. If it is higher or equal priority, then the single interrupt line to the MCP is toggled.

S/W on the PCC and MCP

For a detailed description of the MCP to PCC and PCC to MCP communication, refer to the "Inter-processor Cummunication Interface"[2] document.

MCP to PCC Interrups

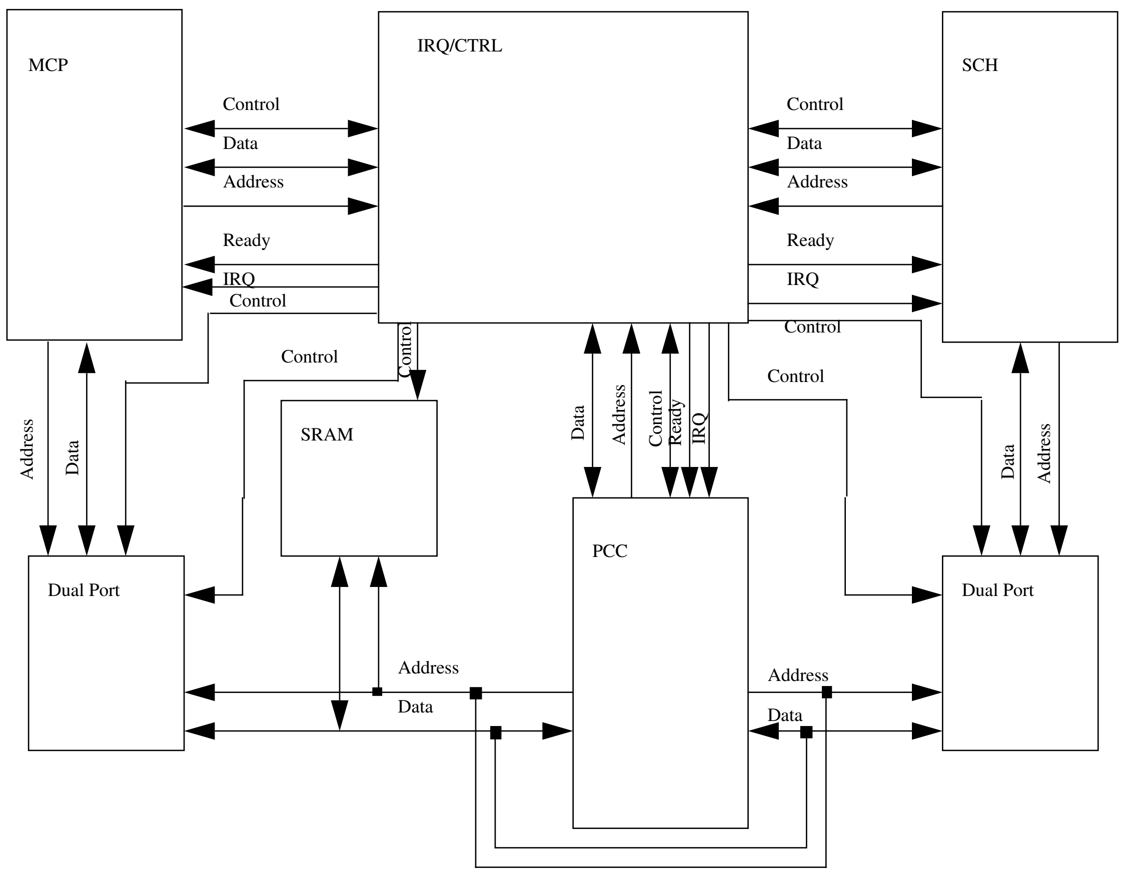
The MCP will interrupt the PCC via a register located on the PCC and is also mapped into the MCP's memory address space. To generate an interrupt to the PCC, the MCP will toggle a specific bit in the register. The output of the register will go to the IRQ input of the PCC. The PCC will treat the interrupt as a high priority one and will not be able to mask it off.

Port Card Memory

The port card memories consist of memory accessable by the MCP, PCC and the SCH. Between the MCP and PCC, and the PCC and the SCH there are 8Kx16 dual port SRAMs. The PCC also has 512K bytes of SRAM memory (see Figure 8).

The control circuitry for the PCC/MCP dual port will be contained in GAL devices. The PCC/SCH dual port control will also be contained in GAL devices.

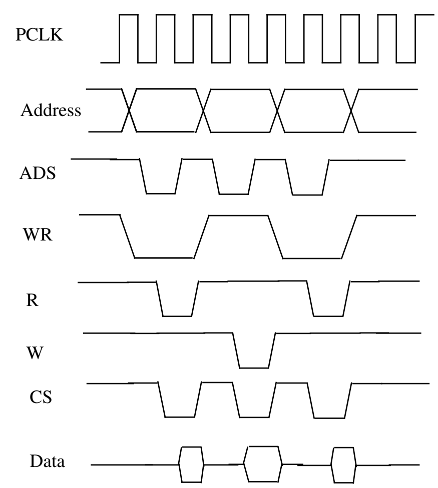
Figure 8 Port Card Memory Interconnection



Accesses to the dual port and the SRAM will be 16 bit wide. The software will take care of arbitration for the dual port memories.

PCC accesses to the memories will be 0 wait state to the SRAM, and 1 wait state to the dual port.. SCH accesses to the dual port will be 1 wait state. MCP access to the dual port will be 3-5 wait states to allow for bus syncronization between two different clock speeds(see Figure 9).

Figure 9 PCC to SRAM and Flex Configuration Timing (0 Wait State)

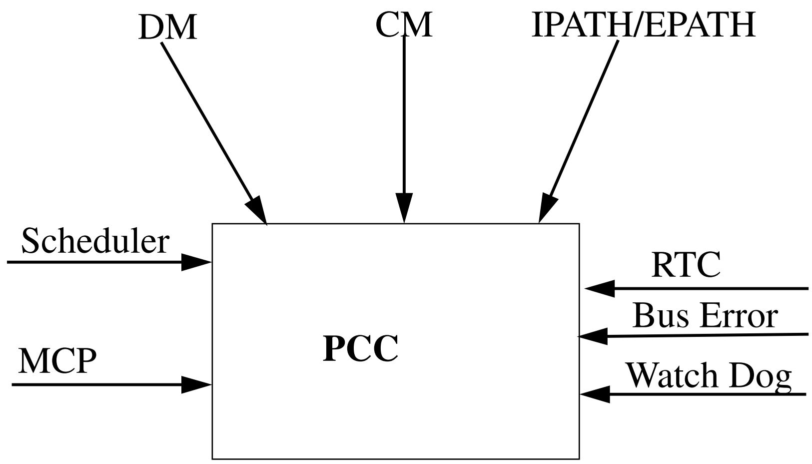


PCC Interrupt Sources

The PCC can be interrupted by a number of sources. It has the control to mask all of the sources except for the NMI. The sources are as follows:

NMI from Watchdog Timer - There is a timer which counts to 1 second and must be written to by the PCC in order to verify processor sanity. If the processor doesn't write to the Watchdog timer within 1 second, an NMI will be generated to the PCC. If the PCC still doesn't write to the Watchdog timer within another second, the PCC will then be reset.

Figure 10 PCC Interrupt Sources



The PCC may be interrupted by the MCP. The PCC does not mask interrupts from the MCP therefore there is no mask bit.

The PCC may be interrupted by the SCH. The PCC masks all interrupts from the SCH by setting the SCH_PCC_IRQ_MASK bit, and eables interrupts bys resetting the bit.

There is a Real Time Counter (RTC) for the PCC which genereates an interrupt to the PCC every 17 mS. The PCC masks RTC interrupts by setting the PCC_RTC_IRQ_-MASK bit, and enables interrupts by resetting the bit.

When the PCC is accessing the memory of the Port Card hardware through the HWIO any bus errors will generate an interrupt. The PCC masks the interrupt by setting the BUS_ERROR_IRQ_MASK bit, and enables it by resetting the bit.

Each of the Port Card hardware blocks contains error counters. When any of these counters overflows, the block generates an interrupt. All of these interrupts are multiplexed to a single interrupt on the PCC. The software will read a memory location which will contain one bit for each source to find out which block originated the interrupt. The PCC masks this interrupt by setting the HW_IRQ_ALL_MASK bit, and enables it by resetting the bit.

ITM and ETM Control

For both the ITM and ETM, there are 3 control signals which need to be set by the PCC to set Trace mode, Source mode, Continuous mode and counter clear. Trace mode is used to capture cells into the Trace memory, Source mode is used to insert cells into the card from the Trace memory. Continuous mode determines whether the hardware should Source/Trace 2K cells or Source/Trace cells continuously. Counter clear ensures that the counter is reset before starting Trace mode.

Both Trace and Source modes must be turned off when the PCC is accessing the memory, and each of the control bits has restrictions concerning how they are accessed by the software.

Trace: is restricted to “once through” or Non-Continuous mode.
- cells are stored without being frame aligned at the start of memory
- counter must be reset with the Clear bit before starting Trace

Source: may be used in Non-Continuous or Continuous mode
- cells must be loaded by software and ensure that they are frame aligned
- counter must be reset with the Clear bit before starting Source

Continuous: may only be changed when both Trace and Source modes are off
- has no function when memory being accessed from PCC

Clear: a bit which must be toggled to reset the counter