The PCI Record

EPICS Collaboration Meeting

- 22-24 May 2002 at BESSY, Germany

David Thompson

Thompsondh@ornl.gov

Outline

- Need for PCI bus support and hardware at SNS
- PCI fundamentals
- pciRecord design
- Results
- Directions

SNS needed PCI support in EPICS

- PPC MVME2100 used at SNS
- R3.14 IOC core runs on PC based platforms supported by several OS
- PCI bus may supplant VME in some applications
  - Good platform for FPGA, with FPGA used in SNS for MPS, low level RF, timing and RTDL
  - BNL is using quad DSP on PMC board
  - SNS's Machine Protection System (MPS) uses an FPGA on a PMC board

PCI Bus Basics

- 16/32/64 bits sync at 33/66 MHz.
  - Address & data muxed on same lines
  - Signaling for bus management, interrupts, different modes
- Supports multiple masters
- Faster than VME, older PC-ISA busses

  Can transfer data at 33 M words/sec in burst mode
- Supports auto-configuration

PCI Bus Basics – Config space

- Used by OS to allocate PCI I/O and memory spaces
- Based on geographic addressing
  PCI standard completely specifies registers in config space
- Unique Vendor Ids allow drivers to bind to hardware
- Registers in config space include: ID Register – 32 bits, version number, comand and status registers, Interrupt control, and base address registers

PCI Bus Basics – Addressing

- OS calls are provided by vxWorks provide access to config space
- 6 Memory Base Address Registers support up to 6 memory or I/O windows
- Memory maps PCI I/O, memory, and non-cacheable memory mapped devices in CPU's memory space
  - » Standards based:

PCI standards

IBM's CHRP

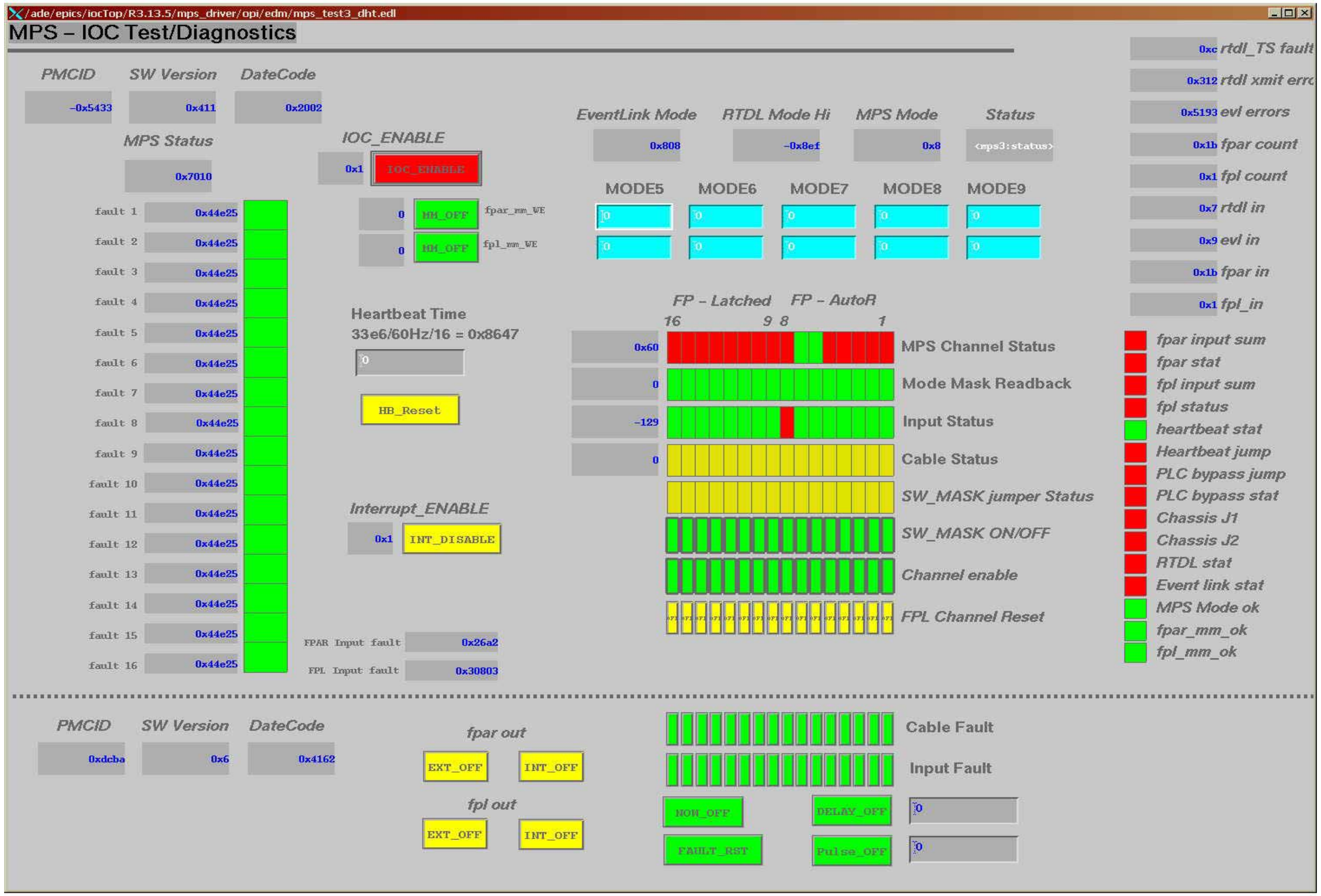
Accessing PCI

- It’s just like other memory mapped I/O?
- Not quite, PCI is little-endian byte order
  - » 0x00000001 is 0x01,0x00,0x00,0x00 or 0x0001,0x0000
  - » NOT 0x00,0x00,0x00,0x01
- Some PCI-I/O bus bridges support byte swapping
- Some PCI-I/O bus bridges do address remaping
  - » If access is char or short the bottom address bits are changed to make the byte order in memory act like big-endian memory

pciRecord - Goals

- Quick tool for debugging and testing MPS board prior to developing a driver
- Used to drive registers in MPS hardware test fixture
- Provide foundation for other PCI hardware device support

MPS Test Screen with pciRecord in use



/ade/epics/iocTop/R3.13.5/mps_driver/opi/edm/mps_test3_dht.edl

MPS – IOC Test/Diagnostics

PMCID	SW Version	DateCode
-0x5433	0x411	0x2002

MPS Status: 0x7010

IOC_ENABLE: 0x1

fpar_mm_WE: 0

fpl_mm_WE: 0

fault 1	0x44e25
fault 2	0x44e25
fault 3	0x44e25
fault 4	0x44e25
fault 5	0x44e25
fault 6	0x44e25
fault 7	0x44e25
fault 8	0x44e25
fault 9	0x44e25
fault 10	0x44e25
fault 11	0x44e25
fault 12	0x44e25
fault 13	0x44e25
fault 14	0x44e25
fault 15	0x44e25
fault 16	0x44e25

Heartbeat Time
33e6/60Hz/16 = 0x8647

0

Interrupt_ENABLE: 0x1

FPAR Input fault: 0x26a2

FPL Input fault: 0x30803

EventLink Mode	RTDL Mode Hi	MPS Mode	Status
0x808	-0x8ef	0x8	<mps3:status>

MODE5	MODE6	MODE7	MODE8	MODE9
0	0	0	0	0
0	0	0	0	0

FP – Latched FP – AutoR
16 9 8 1

0x60	MPS Channel Status
0	Mode Mask Readback
-129	Input Status
0	Cable Status
	SW_MASK jumper Status
	SW_MASK ON/OFF
	Channel enable
	FPL Channel Reset

0xc	rtdl_TS fault
0x312	rtdl xmit erro
0x5193	evl errors
0x1b	fpar count
0x1	fpl count
0x7	rtdl in
0x9	evl in
0x1b	fpar in
0x1	fpl_in

🟥	fpar input sum
🟥	fpar stat
🟥	fpl input sum
🟥	fpl status
🟩	heartbeat stat
🟥	Heartbeat jump
🟥	PLC bypass jump
🟥	PLC bypass stat
🟥	Chassis J1
🟥	Chassis J2
🟥	RTDL stat
🟥	Event link stat
🟩	MPS Mode ok
🟩	fpar_mm_ok
🟩	fpl_mm_ok

PMCID	SW Version	DateCode
0xdcba	0x6	0x4162

Cable Fault

Input Fault

0

0

PCI Record fields – Specifications

- VID: The vendor ID of the device
- ID: The Vendor’s device ID
- NUM: Which of possibly more than one device
- BSEL: Which Base Address Register to map (0-5)
- OFFS: The memory location relative to the BAR base address
- BEND: Endian-ness of data located in the BAR window
- BSIZ: 8/16/32 bit reads or writes

PCI Record fields – I/O

- VAL: The latest value scanned by the record, accessed according to byte order, size, and offset inputs to the record.
- WRI: When put to this field will perform a write to the memory location specified by the record

PCI Record fields Informational

- BBAR: The contents of the selected BAR
- BEA: The address, in memory space, that the BAR maps
- BTYP: The type of memory the bar maps; I/O, Memory, Non-cachable
- BWND: The size of the memory window mapped by the selected bar

PCI Record fields Informational – Config space

- CSTA: The status register, in config space
- CMDR: The command register

- ABUS: The PCI bus number in config space
- ADEV: The PCI device address in config space

Confessions of a newbe

- First pass not 4 char field names
  - Four character limit is mostly gone in R3.13.5 version of IOC-CORE but will not be officially removed until 3.15
- Did not use DBLINKs on first pass
- Did not use “Inst IO” to specify location of data
- Thanks goes to Coles Sibley, Ernest Williams, and the “Application Developer's Guide”

Issues

- Is the BAR to address translation a vxWorks kernel function or should it be an EPICS library function
- Is record support needed for ai,ao,bi,bo,mbbi, etc?
  - This record support already exists in the MPS driver