Fermilab

MVME2600-to-NNSLD Command Protocol

Rev C PCB, 10 September 1998, Paul Kasley

Communication between the PPC host and the NNSLD PMC module occurs through the eight 32-bit mailbox registers on the PMC. Four mailboxes are available for passing messages from the PPC to the NNSLD and four service the reverse path.

NNSLD Register Addresses in PPC Memory

The NNSLD registers appear in PCI I/O space on the PPC. The mapping of PCI I/O space into PPC memory is determined by the mapping registers in the Raven chip and by Base Address Register 0 in the NNSLD's PCI configuration space at offset $10-$13.

With S5933 BAR0=$01000181 (PPC bug command "CSAW 0 0 10 10 01000181"), PPCbug's default mapping places the NNSLD at $8100 0180 through $8100 01BF.

NNSLD NAME	NNSLD ADRS	PPC ADRS	PPC NAME	COMMENT
IBX1L	0000	8100 0180	IBX1L	PPC > ADSP (inbound)
IBX1H	0001	8100 0182	IBX1H	"
IBX2L	0002	8100 0184	IBX2L	"
IBX2H	0003	8100 0186	IBX2H	"
IBX3L	0004	8100 0188	IBX3L	"
IBX3H	0005	8100 018A	IBX3H	"
IBX4L	0006	8100 018C	IBX4L	"
IBX4H	0007	8100 018E	IBX4H	"
OBX1L	0008	8100 0190	OBX1L	ADSP > PPC (outbound)
OBX1H	0009	8100 0192	OBX1H	"
OBX2L	000A	8100 0194	OBX2L	"
OBX2H	000B	8100 0196	OBX2H	"
OBX3L	000C	8100 0198	OBX3L	"
OBX3H	000D	8100 019A	OBX3H	"
OBX4L	000E	8100 019C	OBX4L	"
OBX4H	000F	8100 019F	OBX4H	"
AFIFO_L	0010	8100 01A0	AFIFO_L	PPC hangs if FIFO is empty
AFIFO_H	0011	8100 01A2	AFIFO_H	"
MWAR_L	0012	*not visible*		
MWAR_H	0013	*not visible*		
APTA_L	0014	*not visible*		
APTA_H	0015	*not visible*		
APTD_L	0016	*not visible*		
APTD_H	0017	*not visible*		
MRAR_L	0018	*not visible*		
MRAR_H	0019	*not visible*		
AMBEF_IN	001A	8100 01B4	AMBEF_IN	Mailbox flags for ADSP-inbound
AMBEF_OUT	001B	8100 01B6	AMBEF_OUT	Mailbox flags for ADSP-outbound
AINT_L	001C	*not visible*		

AINT_H	001D	*not visible*		
AGCSTS_L	001E	*not visible*		
AGCSTS_H	001F	*not visible*		
MWTC_L	002C	*not visible*		
MWTC_H	002D	*not visible*		
MRTC_L	002E	*not visible*		
MRTC_H	002F	*not visible*		
not visible		8100 01A4	MWAR	
not visible		8100 01A8	MWTC	
not visible		8100 01AC	MRAR	
not visible		8100 01B0	MRTC	
not visible		8100 01B8	INTCSR	
not visible		8100 01BC	MCSR	

Note: The high byte of OBX4H can be used for direct hardware access (see "AMCC S5933 PCI Controller Data Book", section 9.1.3). When this feature is not used, OBX4H bits 8-15 will contain garbage and should be ignored.

PPC vs NNSLD Byte Ordering

Byte order as viewed from the PPC is the reverse of that on the PMC. A longword ordered as "ABCD" on the PMC appears as "DCBA" in the PPC memory.

Words written from the NNSLD as AB order are seen at the PPC as BA order. Long-words written from the NNSLD as H-AB, L-CD are seen at the PPC as DCBA. For example, 0009=8765 & 0008=4321 show as PPC 81000190=21436587. NNSLD MCSR=000000E6 shows at PPC 810001BC=E6000000.

Command Protocol

The NNSLD produces a single response for each command. The NNSLD will not accept a subsequent command until after the current command has been fully executed and the response has been read by the PPC host.

1. PPC writes optional data to inbound mailboxes 1 through 3.

    Inbound mailbox flags corresponding to the mailbox bytes written are set.

2. PPC writes command code to inbound mailbox 4-low.

    When it sees the flags for IBX4L get set, the NNSLD copies the command and data from the inbound mailboxes and begins processing the command.

3. PPC polls outbound mailbox 4-high flag in the mailbox flag register for completion or waits for an interrupt to occur on OBX4H flag set.

    When finished, the NNSLD copies its response to outbound mailboxes 1 through 3. It then copies the command to OBX4L and writes 0x0000 to OBX4H to set the corresponding bit in the flag register.

4. PPC reads outbound mailboxes 1 through 4.

> *The NNSLD now monitors the outbound mailbox hardware flags and waits until it sees all of them get cleared. The NNSLD now knows that the host has retrieved the results of the command. The NNSLD is now ready to accept another command.*

5. The PPC checks that OBX4H is $0000 before it initiates a new command.

<u>RD-CAMAC Command Example</u>

1) Write data to inbound mailbox

```
PPC1-Bug>md 81000180
81000180  00000000 00000000 00000000 00000000  ................
81000190  00000000 00000000 00000000 000000F8  ................
PPC1-Bug>mw 81000180 02200000
Effective address: 81000180
Effective data   : 02200000
PPC1-Bug>mw 81000184 f0f00400
Effective address: 81000184
Effective data   : F0F00400
PPC1-Bug>mw 81000188 55050000
Effective address: 81000188
Effective data   : 55050000
```

2) Write command to IBX4L

```
PPC1-Bug>mw 8100018c 07000000
Effective address: 8100018C
Effective data   : 07000000
```

3) Check for completion and read response in outbound mailboxes

```
PPC1-Bug>md 81000180
81000180  02200000 F0F00400 55050000 07000000  . ......U.......
81000190  D3030000 0004ACEC 00000000 070000F8  ................
PPC1-Bug>
```

4) Write data to inbound mailboxes in preparation for the next command

```
PPC1-Bug>MW 81000180 ABCDEF12
Effective address: 81000180
Effective data   : ABCDEF12
```