ROB-in Rx / ROB-in Rx-P Programmer's Guide

G. J. Crone

February 15, 1999

DRAFT

Contents

1 Introduction **4**
- 1.1 Hardware description . . . 4
- 1.2 Principle of operation . . . 5

2 Initialisation and Reset **6**
- 2.1 Power up Initialisation of i960 Rx . . . 6
  - 2.1.1 Initialisation of the Memory Controller . . . 6
  - 2.1.2 Initialisation of the Address Translation Unit . . . 6
- 2.2 Reset state of local resources . . . 7
- 2.3 Software initialisation sequence . . . 7
  - 2.3.1 Setting FIFO thresholds . . . 8
- 2.4 Software Reset . . . 8

3 ROB-in Rx Memory map **9**

4 Register Definitions **10**
- 4.1 Free Page FIFO (FPF) . . . 10
- 4.2 Input FIFO Threshold Load Register (IFTLR) . . . 10
- 4.3 ROB-in Control Register (RCR) . . . 11
- 4.4 Free Page FIFO Threshold Load Register (FPFTLR) . . . 11
- 4.5 ROB-in Status Register (RSR) . . . 12
- 4.6 Used Page FIFO (UPF) . . . 13
- 4.7 Test Data Write Register (TDWR) . . . 14
- 4.8 Test Data Write With Error Register (TDWWER) . . . 14
- 4.9 Test Control Write Register (TCWR) . . . 14
- 4.10 Test Control Write With Error Register (TCWWER) . . . 14

5 PCI interface **15**
- 5.1 PCI to PCI bridge . . . 15
  - 5.1.1 Extended Bridge Control Register (EBCR) . . . 15
- 5.2 Address Translation Unit . . . 15
- 5.3 Supported Accesses . . . 16

6 S-LINK interface **17**
- 6.1 Reset . . . 17
- 6.2 UXOFF . . . 17
- 6.3 UTDO . . . 17

List of Figures

1 ROB-in Rx block diagram . . . 4
2 ROB-in Rx operation . . . 5
3 Initialisation flow . . . 7

DRAFT

List of Tables

1 i960 Rx Memory Bank Control Register 6
2 i960 Rx Memory Controller Wait States Registers 7
3 Memory map as seen by i960 Rx 9
4 ROB-in Control Register 11
5 ROB-in Status register 12
6 Used Page FIFO (UPF) 13
7 Memory map of ROB-in Rx hardware as seen from the PCI bus . 16

DRAFT

1 Introduction

1.1 Hardware description

The ROB-in Rx is a standard PCI card. The ROB-in Rx-P is a PMC version of the ROB-in Rx which differs (apart from the physical layout) only in the FIFO's used.

The ROB-in Rx consists of 6 main parts as shown in figure 1.

1. The i960 Rx (i960 RP or i960 RD) processor
2. The boot ROM (512Kx8 Flash RAM)
3. The program SRAM (256Kx32 Static RAM)
4. The fragment buffer SRAM (256Kx32 Static RAM)
5. The Free & Used Page FIFO's
6. The control logic (MACH 5 PLD)

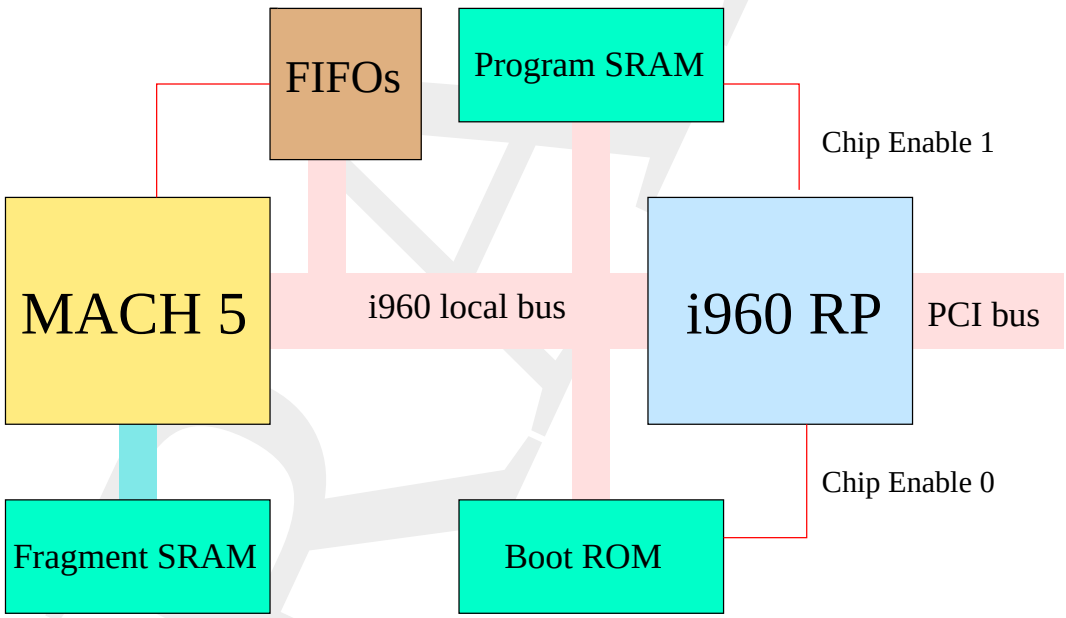


Figure 1: ROB-in Rx block diagram

The boot ROM is connected to the i960 Rx's memory bank 0. The program SRAM is connected to the i960 Rx's memory bank 1. The MACH 5 is responsible for decoding addresses for the fragment buffer and FIFO's.

The board is designed to accept data from S-LINK. It has a socket for an S-LINK destination card and provides access to several S-LINK signals through the control register.

The board also has 6 LEDs, 4 of which are user programmable via the ROB-in Control Register (see section 4.3).

DRAFT

1.2 Principle of operation

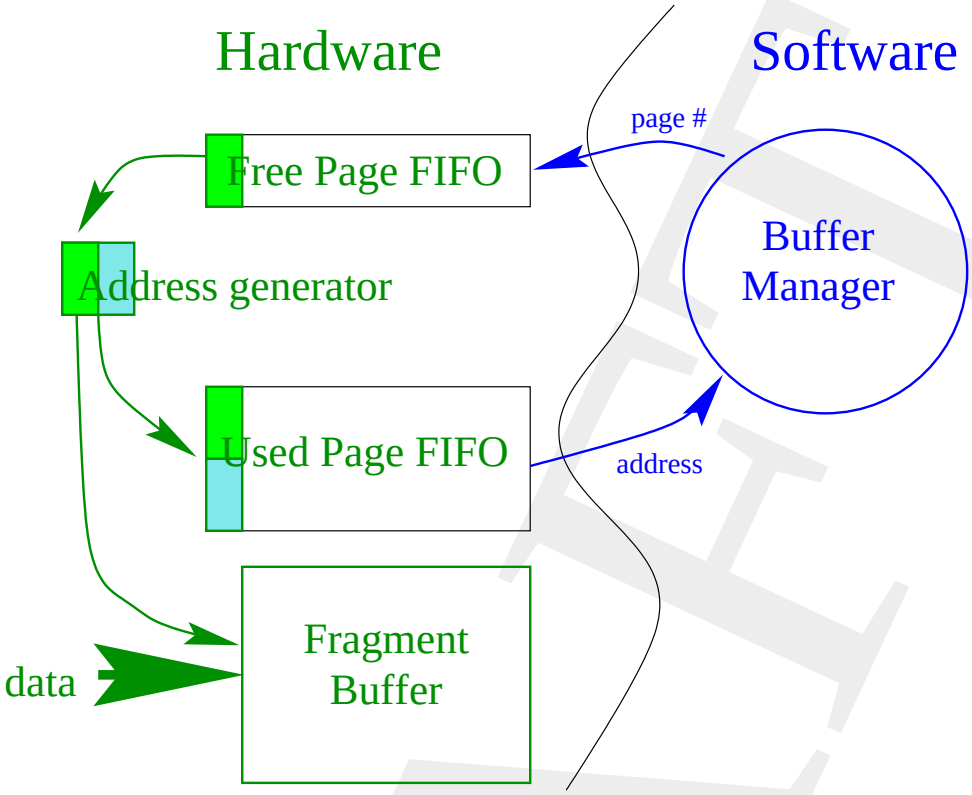


Figure 2: ROB-in Rx operation

Data arrive at the front end input and are routed to the fragment buffer under control of the MACH. The fragment memory is organised as 1024 pages of 256 32bit words (1024 bytes). The MACH determines which page of the buffer to write to by loading the upper bits of its address generator from the Free Page FIFO. After writing the last word of an event fragment, or the last word of a page (whichever comes first), the address bits are written into the Used Page FIFO along with some status information. It is the responsibility of the buffer management software, whether running on the on board i960 Rx or an external processor, to keep the Free Page FIFO supplied with page numbers of free pages as shown in figure 2.

The buffer management software can build an index of which data are stored in the used pages based on the information returned in the Used Page FIFO and in the Fragment Buffer itself. For the input logic to correctly allocate a new page for each event fragment data must be framed with the Beginning of Block and End of Block control words as described in [2]. No other words are treated as special by the hardware. The buffer management software must look in the fragment buffer to find event identification information.

NB: The ROB-in Rx provides no interrupts for the Used Page FIFO. The buffer management software must poll for new pages.

DRAFT

2 Initialisation and Reset

2.1 Power up Initialisation of i960 Rx

In order to use the ROB-in Rx the i960 Rx processor must be allowed to run the initialisation code in the boot ROM. It should be jumpered to initialise in mode 3. This allows the i960 Rx processor to set up the PCI configuration registers before the host's BIOS queries them.

The initialisation code in the boot ROM is responsible for:

- Setting the address, size and wait states for the program SRAM in the i960 Rx's memory controller.
- Setting the Address translation value and size in the Address Translation Unit (ATU).

The memory layout shown in the rest of this document assumes that the standard initialisation code has been executed!

2.1.1 Initialisation of the Memory Controller

The i960 Rx's Memory Bank Control Register (MBCR) is set up to enable a 2 Mbyte memory bank 1 for the program SRAM as shown in table 1.

Bit	Value	Description
31:24	0	— reserved —
23:20	0101	Memory bank 1 size – 2Mbytes
19	0	— reserved —
18	1	Memory Bank 1 Extended MWE3:0# Bit
17	1	Memory Bank 1 Write Enable bit
16	1	Memory Bank 1 enable bit
15:08	0	— reserved —
07:04	1000	Memory bank 0 size – 16Mbytes
03	0	— reserved —
02	1	Memory Bank 0 Extended MWE3:0# Bit
01	1	Memory Bank 0 Write Enable bit
00	1	Memory Bank 0 enable bit

Table 1: i960 Rx Memory Bank Control Register

The wait states for the SRAM and boot ROM are set in the Memory Bank Read Wait States Registers (MBRWS0:1) and the Memory Bank Write Wait State Registers (MBWWS0:1) as shown in table 2.

2.1.2 Initialisation of the Address Translation Unit

The Primary Inbound ATU Limit Register (PIALR) is set to a value which indicates the size to be 4 Mbytes.

DRAFT

Bit	Bank 0 Value (ROM)		Bank 1 Value (SRAM)		Description
	Read	Write	Read	Write	
31:19	0	0	0	0	— reserved —
18:16	011	011	001	001	Address to first data wait states
15:11	0	0	0	0	— reserved —
10:08	100	000	001	001	Data to data wait states
07:03	0	0	0	0	— reserved —
02:00	010	010	000	000	Additional recovery wait states

Table 2: i960 Rx Memory Controller Wait States Registers

The Primary Inbound ATU Translate Value Register (PIATVR) is set to the base address of the program memory.

It seems from experience, although it is not documented in intel's manual [1], that **the PIALR must be set some time before the PIATVR**.

2.2 Reset state of local resources

At power up XOFF is asserted on the S-link interface, the Free Page, the Used Page and the input FIFOs are held reset and all the user programmable LEDs are off.

The ROB-in Rx-P comes up with the test inputs enabled to allow programming of the Input FIFO Thresholds (see 2.3).

2.3 Software initialisation sequence

To avoid loss of data the S-LINK UXOFF signal should remain asserted until the FIFOs have been initialised as show in figure 3.

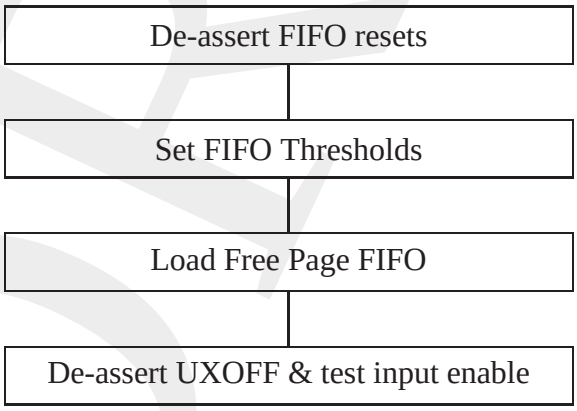


Figure 3: Initialisation flow

DRAFT

2.3.1 Setting FIFO thresholds

With the FIFO's used on the ROB-in Rx-P the **first 2 writes** to the FIFO's **always** set the thresholds, **not just writes to the FPFTLR or IFTLR**! Therefore the FIFO thresholds should always be set before loading the FPF ore enabling external inputs.

NB: It is recommended to set the FIFO thresholds for both Input and Free Page FIFO's in this manner even on the ROB-in Rx as future versions may use the same FIFO's as used in the ROB-in Rx-P.

2.4 Software Reset

There is no board level reset available from software but each component other than the i960 Rx can be reset via the control register (see section 4.3). The i960 Rx can be reset by writing to the Extended Bridge Control Register EBCR in the PCI to PCI bridge (see section 5.1).

DRAFT

3 ROB-in Rx Memory map

The complete memory map of the ROB-in Rx processor is shown in table 3.

0000 0000	Internal Data RAM
0000 0400	Reserved
0000 1000	Peripheral Memory Mapped Registers
0000 2000	ATU Outbound Direct Addressing Window
8000 0000	ATU Outbound Translation Windows
9002 0000	Not Decoded
A000 0000	Program Memory
A010 0000	Reflection of Program Memory
A020 0000	Free Page FIFO (FPF)
A021 0000	Input FIFO Threshold Load Register (IFTLR)
A022 0000	ROB-in Control Register (RCR)
A024 0000	Free Page FIFO Threshold Load Register (FPFTLR)
A026 0000	ROB-in Status Register (RSR)
A028 0000	Not Decoded
A02A 0000	Used Page FIFO (UPF)
A02C 0000	Test Data Write Register (TDWR)
A02D 0000	Test Data Write With Error Register (TDWWER)
A02E 0000	Test Control Write Register (TCWR)
A02F 0000	Test Control Write With Error Register (TCWWER)
A030 0000	Fragment Memory
A03F FFFF	Not Decoded
FEF8 0000	Flash ROM
FEFF FF2F	Initialization Boot Record (IBR)
FEFF FF60	Reserved
FF00 0000	i960 Core Processor Memory-Mapped Register Space
FFFF FFFF	

Table 3: Memory map as seen by i960 Rx

The address lines for the registers are only partially decoded by the MACH logic so each register is reflected many times.

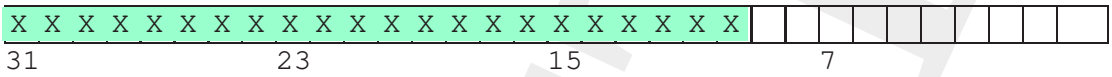
DRAFT

4 Register Definitions

All registers are accessible through the PCI bus as well as from the on board i960 Rx processor. All register accesses are **32bit only**.

4.1 Free Page FIFO (FPF)

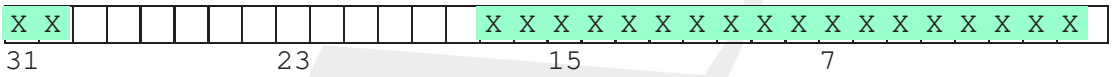
Offset: 0x00200000
Access: Write only



The Free Page FIFO is a 1024 word deep FIFO from which the input logic sets the upper 10 bits of its address generator at the start of writing each page. This FIFO should be loaded with the **page number** of each free page in the lower 10 bits.

4.2 Input FIFO Threshold Load Register (IFTLR)

Offset: 0x00210000
Access: Write only



The IFTLR allows the Programmable Almost Empty (PAE) / Full (PAF) thresholds of the input FIFO to be set. See section 4.4 for a description of threshold operation.

On the ROB-in Rx-P the input FIFO is a single 36 bit wide FIFO (IDT23641). The PAF signal is used to control UXOFF see section 6.2.

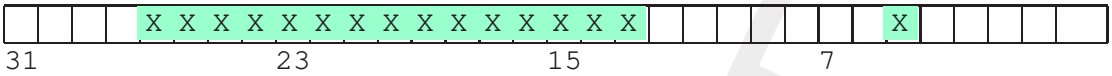
On the ROB-in Rx the input FIFO is actually constructed from 2 18 bit FIFOs and the programmable threshold status signals are routed to separate destinations. The PAE / PAF thresholds are not used to control UXOFF (this is derived from the half full status which is not programmable see section 6.2). The 12 bit threshold values written to bits 29:18 of this register control the PAF which is fed back into the input FIFO itself accumulated over a page and routed to the UPF where it can be read on bit 18. The other PAE / PAF signals are not currently used.

NB: The test input enable bit must be set in the RCR to access this register.

DRAFT

4.3 ROB-in Control Register (RCR)

Offset: 0x00220000
Access: Write only



The structure of the RCR is shown in table 4. Bits 11:2 are all related to the S-LINK interface (see section 6 and [3] for details).

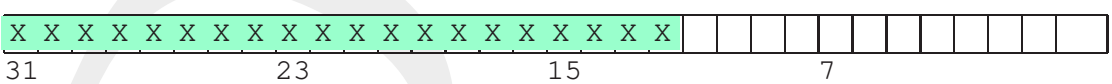
Bit	Default	Description
31	0 (off)	User LED 3 red
30	0 (off)	User LED 2 yellow
29	0 (off)	User LED 1 yellow
28	0 (off)	User LED 0 yellow
27:13	0	— reserved, write as 0 —
12	0	Test Inputs Select – Setting this bit allows the i960 Rxto write to the input of the input FIFO via the registers IFTLR, TDWR, TDWWER, TCWR and TCWWER
11:08	0000	S-LINK return lines URL(3:0)
07:06	00 (32bit)	S-LINK data width UDW(1:0)
05	0	— reserved, write as 0 —
04	0	S-LINK test data select (UTDO)
03	1	S-LINK Transmit Off (UXOFF) – This bit is not connected directly to the S-LINK UXOFF but is ORed with status lines from the FIFOs (see section 6.2)
02	1	S-LINK Reset (URESET)
01	1	Reset Input FIFO
00	1	Reset Used & Free FIFOs

Table 4: ROB-in Control Register

NB: Since the RCR is write only it is recomended that the user code keep track of its contents.

4.4 Free Page FIFO Threshold Load Register (FPFTLR)

Offset: 0x00240000
Access: Write only



The Free Page FIFO Threshold Load register allows the Programmable Almost Empty (PAE) / Full (PAF) thresholds of the Free Page FIFO to be set.

On the ROB-in Rx-P the FIFO is an ∫dt23641. On the ROB-in Rx the FIFO is an ∫dt72225LB (see [4] for a detailed description).

PRELI

ROB-in Rx ONLY

The PAE signal is used to generate UXOFF when there are too few pages available in the FPF (see section 6.2).

This register provides direct access to the $\int$dt FIFO input with the LOAD and WRITE ENABLE lines asserted. Each write to this register sets one of the threshold registers in turn. The first value written will be loaded into the Empty Offset register (the Programmable Almost Empty threshold), the next to the Full Offset register, the next to the Empty Offset again.

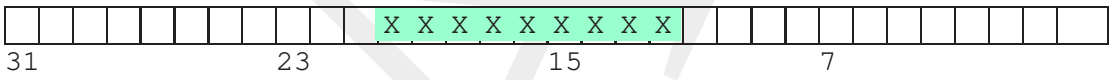
Since the PAF signal is not used only the Empty Offset register needs to be loaded but **to avoid confusion it is recommended that both be programmed.**

The first word written to an empty FIFO is automatically presented at the FIFO's output port so the PAE signal is asserted when there is one more page available than the selected threshold. For example setting a threshold of 0 causes PAE (and hence UXOFF) to be asserted when there is still 1 page available.

The default value for the Empty Offset register is 0x7f (127) which would cause UXOFF to be asserted whenever there are less than 128 pages available in the Free Page FIFO (see section 6.2).

4.5 ROB-in Status Register (RSR)

Offset: 0x00260000
Access: Read only



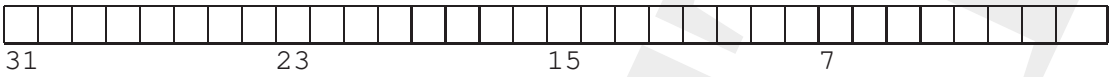
Bit	Description
31	Used Page FIFO (UPF) is empty
30	Combined error flag
29	Free Page FIFO (FPF) is empty
28	Input data had a Begin of Block control word without a preceding End of Block control word
27	Input data had an End of Block control word without a preceding Begin of Block control word
26	Input data arrived without a preceding Begin of Block control word
25	S-LINK Link Down (LDOWN)
24	Input FIFO half full
23	Input Data available
22	Input FIFO empty
21	X-OFF status
20:12	Unassigned
11:00	Availability counter

Table 5: ROB-in Status register

The ROB-in Status Register reflects the current state of the ROB-in Rx. Bits 31:24 are also presented along side the UPF (see section 4.6) allowing status information to be read at the same time as the FIFO. The structure of the RSR is shown in table 5.

4.6 Used Page FIFO (UPF)

Offset: 0x002A0000
Access: Read only



Bit	type	Description
31	L	Used Page FIFO empty
30	L	Combined error flag
29	L	Free Page FIFO empty
28	L	Input data had a Begin of Block control word without a preceding End of Block control word
27	L	Input data had an End of Block control word without a preceding Begin of Block control word
26	L	Input data arrived without a preceding Begin of Block control word
25	L	S-LINK Link Down (LDOWN)
24	L	Input FIFO half full
23	F	S-LINK Link Data Error (LDERR)
22	F	Last page of event
21	F	Data arrived without a preceding Begin of Block control word
20	F	End of Block control word without Begin of Block control word
19	F	Begin of Block control word without End of Block control word
18	F	Input FIFO asserted almost full
17:08	F	Page number of used page
07:00	F	Number of last word written within page

Table 6: Used Page FIFO (UPF)

The Used Page FIFO contains the addresses of the last word written to each used page along with the status of the ROB-in Rx at the time the address was loaded into the FIFO.

Since the input logic always clocks in 32bit words its address generator increments per word not byte! In order to use the address bits from the UPF to access the data in the fragment buffer they must be shifted 2 places left.

Not all of the bits in this register are actually taken from the FIFO Some of the bits are current status information. In particular the Most significant bit is the

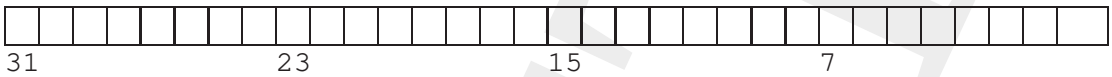
DRAFT

current empty / not empty state of the FIFO itself. This basically flags whether or not the lower bits are valid. The structure of the Used Page FIFO is shown in table 6, the FIFOed bits are flagged with ‘F’ and the “live” bits are flagged with ‘L’.

NB: Bits 18–23 are the accumulated status for the whole page.

4.7 Test Data Write Register (TDWR)

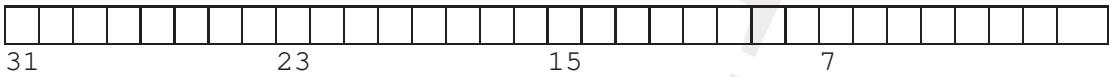
Offset: 0x002C0000
Access: Write only



Data written to this register are presented at the input to the input FIFO just as if they had been transferred accross the S-LINK. **In order to use this facility the test data enable bit must be set in the RCR.**

4.8 Test Data Write With Error Register (TDWWER)

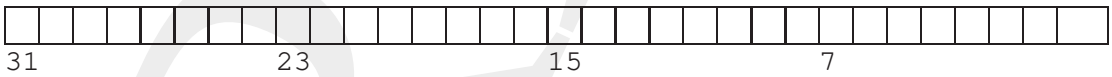
Offset: 0x002D0000
Access: Write only



This register functions just as the TDWR (section 4.7) except that the bit normally controlled by S-LINK LDERR is asserted at the time that the data is clocked into the input FIFO.

4.9 Test Control Write Register (TCWR)

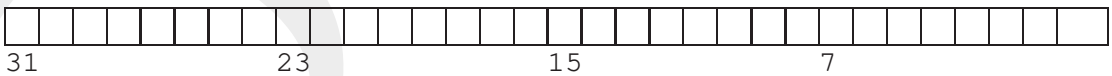
Offset: 0x002E0000
Access: Write only



This register functions just as the TDWR (section 4.7) except that the bit normally controlled by S-LINK control word bit is asserted at the time that the data is clocked into the input FIFO.

4.10 Test Control Write With Error Register (TCWWER)

Offset: 0x002F0000
Access: Write only



This register functions just as the TCWR (section 4.9) except that the bit normally controlled by S-LINK LDERR is asserted at the time that the data is clocked into the input FIFO.

DRAFT

5 PCI interface

The i960 Rx processor embedded in the ROB-in Rx shows up on the PCI bus as two devices, the Address Translation Unit (ATU) and the PCI to PCI bridge. Since all resources of the ROB-in Rx, other than the boot ROM, occupy a contiguous 4Mbyte address space the ATU can be configured by the boot ROM to map this area on to the PCI bus.

5.1 PCI to PCI bridge

Vendor ID 0x8086, Device ID 0x0960

There are no devices on the i960 Rx's secondary PCI bus so the PCI to PCI bridge is not used. However, there is one register in the PCI to PCI bridge, the Extended Bridge Control Register (EBCR), which can be written to cause a reset of the i960 Rx processor core.

5.1.1 Extended Bridge Control Register (EBCR)

Offset: PCI Configuration Address Offset: 40H
Access: Read/Write

Since the secondary PCI bus is not used in the ROB-in Rx the only bit of interest is the local bus reset bit (bit 5) which resets all devices on the i960 Rx's local bus including the processor core. This bit is automatically cleared after operation.

An example of how to reset the i960 Rx from the host processor is shown in the code fragment below.

```
status = pcibios_find_device (PCI_VENDOR_ID_INTEL,
                              0x0960, pci_index,
                              &pci_bus, &pci_device_fn) ;
if (status == 0)
{
   pcibios_write_config_word (pci_bus, pci_device_fn,
                              0x40, 0x20);
}
```

5.2 Address Translation Unit

Vendor ID 0x8086, Device ID 0x1960

The ATU Device ID and Vendor ID fields are left at their default values by the initialisation code in the boot ROM. This means that the ROB-in Rx shows up as Device ID 0x1960 with Vendor ID 0x8086 (int$_e$l).

The memory map of the ROB-in Rx as seen from the PCI bus is shown in table 7. The first 4096 bytes of the program memory are hidden by the Messaging Unit registers but the whole of the SRAM is visible if you use the reflection at base address + 0010 0000.

DRAFT

base address + 0000 0000	Messaging Unit
base address + 0000 1000	Program Memory
base address + 0010 0000	Reflection of Program Memory
base address + 0020 0000	Free Page FIFO (FPF)
base address + 0021 0000	Input FIFO Threshold Load Register (IFTLR)
base address + 0022 0000	ROB-in Control Register (RCR)
base address + 0024 0000	Free Page FIFO Threshold Load Register (FPFTLR)
base address + 0026 0000	ROB-in Status Register (RSR)
base address + 0028 0000	Not Decoded
base address + 002A 0000	Used Page FIFO (UPF)
base address + A02C 0000	Test Data Write Register (TDWR)
base address + A02D 0000	Test Data Write With Error Register (TDWWER)
base address + A02E 0000	Test Control Write Register (TCWR)
base address + A02F 0000	Test Control Write With Error Register (TCWWER)
base address + 0030 0000	Fragment Memory
base address + 003F FFFF	

Table 7: Memory map of ROB-in Rx hardware as seen from the PCI bus

5.3 Supported Accesses

The program SRAM supports 32, 16 and 8 bit accesses. **The Fragment memory supports 32 bit accesses only.**

The ROB-in Rx address space is marked as prefetchable in the PCI configuration registers.

DRAFT

6 S-LINK interface

6.1 Reset

The S-LINK reset protocol is not handled by the control logic. After asserting URESET in the RCR (see section 4.3), the software must wait for the LDOWN signal to be de-asserted before releasing URESET. The LDOWN signal can be seen in the RSR (see section 4.5).

6.2 UXOFF

The S-LINK UXOFF signal is generated by an OR of three signals:

- The UXOFF bit in the ROB-in Control Register (see section 4.3).
- The Programmable Almost Empty flag of the Free Page FIFO (see section 4.4).
- The Half Full (ROB-in Rx) or PAF (ROB-in Rx-P) flag of the input FIFO.

The current state of the S-LINK UXOFF signal is available through the RSR (see section 4.5).

6.3 UTDO

DRAFT

References

[1] i960 Rx I/O Microprocessor Developer's Manual
Intel Corporation.

[2] ATLAS Read-Out Link Data Format, Robert McLaren and Owen Boyle

[3] The S-LINK Interface Specification, Robert McLaren Owen Boyle and Erik van der Bij

[4] IDT 1995 Specialised Memories, FIFOs & Modules Data Book. Integrated Device Technology, Inc.

DRAFT