ROB-in RP Programmer's Guide

Alternative Logic Version

G. J. Crone

February 24, 1998

Contents

List of Figures

List of Tables

1 Introduction

1.1 Hardware description

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

1. The i960 RP 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 Used Page FIFO
6. The control logic (MACH 5 PLD)

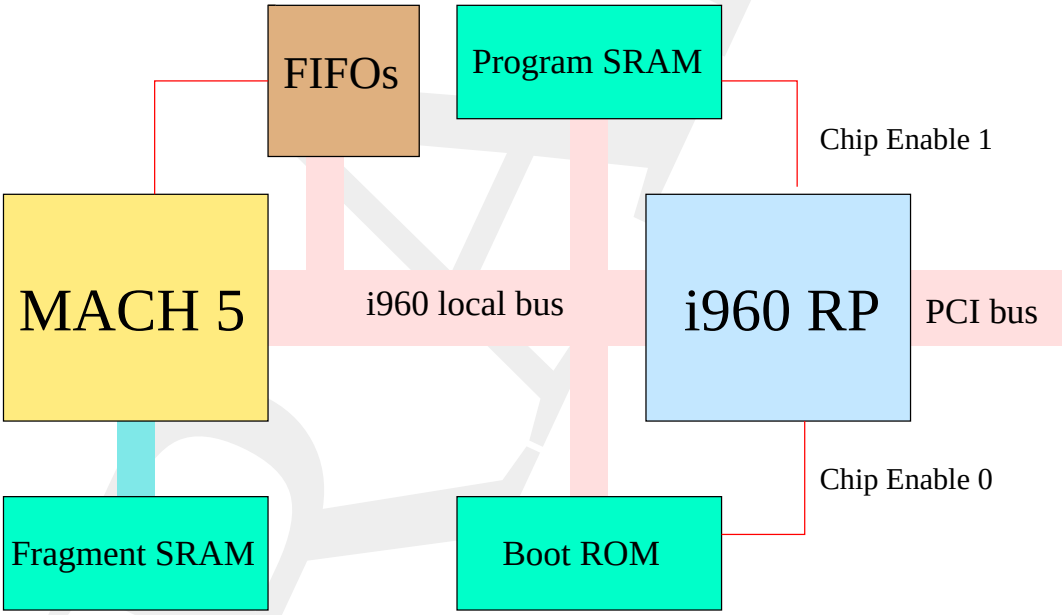


Figure 1: ROB-in RP block diagram

The boot ROM is connected to the i960 RP's memory bank 0. The program SRAM is connected to the i960 RP'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 4.1).

1.2 Principle of operation

The fragment memory is organised as linked lists of pages, using the last word of each page as the link to the next page. When the buffer is empty all the pages form a single linked list of free pages **(The links must be setup by software as part of the initialisation)**. The end of the list is marked by a special value in the link word (MSB set).

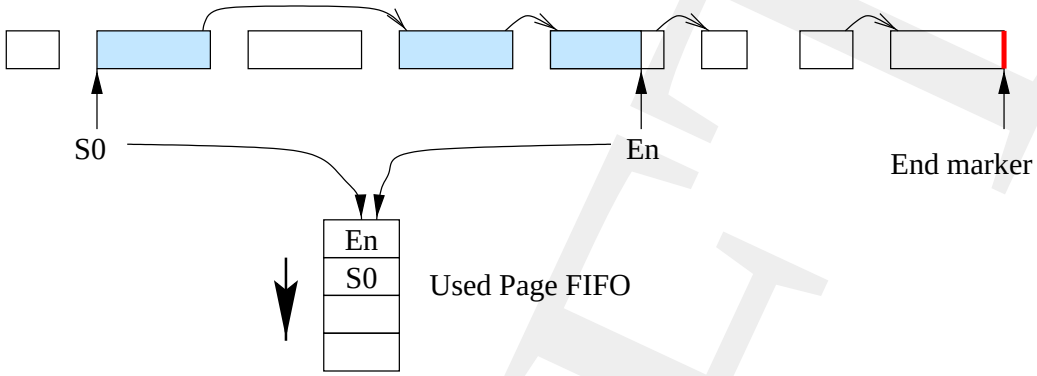


Figure 2: ROB-in RP fragment input

Data arriving at the front end input are clocked into the current page in the fragment buffer until either the last word of the page or the last word of the fragment is reached. At this point the last word of the page is read and loaded into the address generator to become the new current page. After writing the last word of an event fragment the page number of the first page and the address of the last word used in the last page of the event are written into the Used Page FIFO[1] along with some status information as shown on 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.

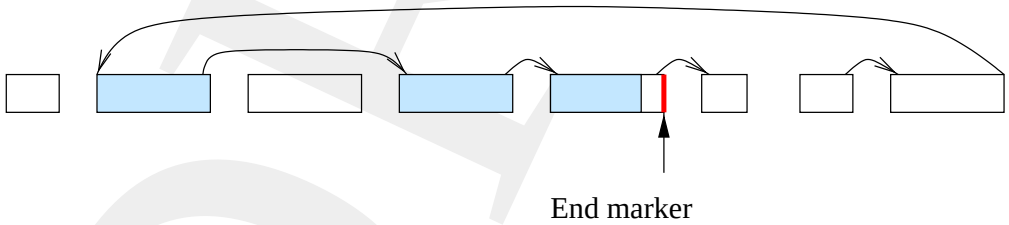


Figure 3: ROB-in RP fragment freeing

When the buffer manager has finished with an event fragment it must attach it

[1]In the current version of the hardware thee first and last page values are written at 2 consecutive entries in the FIFO. In a future version thet will be written as different bit fields in a single entry.

to the end of the free page linked list and mark the new end of the free page list (any links within the fragment are still valid) as shown in figure 3.

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

2 Initialisation and Reset

2.1 Power up Initialisation of i960 RP

In order to use the ROB-in RP the i960 RP 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 RP 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 RP'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 RP'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 RP 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.

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 RP 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.

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 4.

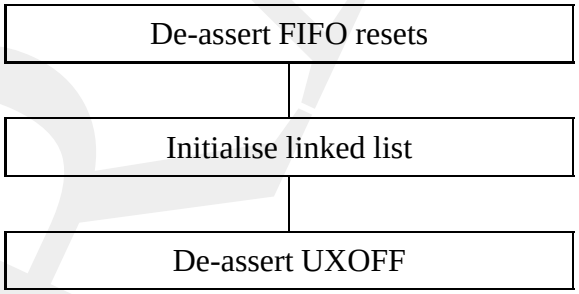


Figure 4: Initialisation flow

2.4 Software Reset

There is no board level reset available from software but each component other than the i960 RP can be reset via the control register (see section 4.1). The i960 RP can be reset by writing to the Extended Bridge Control Register EBCR in the PCI to PCI bridge (see section 5.1).

3 ROB-in RP Memory map

The complete memory map of the ROB-in RP 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	Not Decoded
A022 0000	Control Register
A024 0000	Not Decoded
A026 0000	Status Register
A028 0000	Not Decoded
A02A 0000	Used Page FIFO
A02C 0000	Not Decoded
A030 0000	Fragment Memory
A03F FFFF	Not Decoded
FEF8 0000	Flash ROM
FEFF FF2F	Initialization Boot Record (IBR)
FEFF FF60	Reserved
FF00 0000 FFFF FFFF	i960 Core Processor Memory-Mapped Register Space

Table 3: Memory map as seen by i960 RP

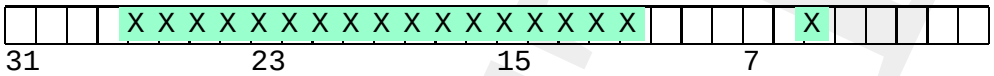
The address lines for the registers are only partially decoded by the MACH logic so each register is reflected many times.

4 Register Definitions

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

4.1 ROB-in Control Register (RCR)

Offset: 0x00220000
Access: Write only



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

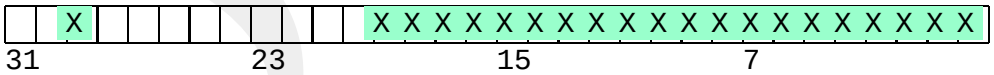
Bit	Default	Description
31	0 (off)	User LED 3 red
30	0 (off)	User LED 2 yellow
29	0 (off)	User LED 1 yellow
28	0 (off)	User LED 0 yellow
27:12	X	— unassigned —
11:08	0000	S-LINK return lines URL(3:0)
07:06	00 (32bit)	S-LINK data width UDW(1:0)
05	X	— unassigned —
04	0	S-LINK test data select (UTDO)
03	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 Page FIFO and free page pointer

Table 4: Control Register

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

4.2 ROB-in Status Register (RSR)

Offset: 0x00260000
Access: Read only



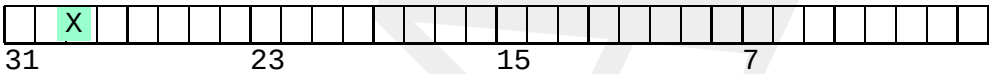
The Status Register reflects the current state of the ROB-in RP. Bits 31:24 are also presented along side the Used Page FIFO (see section 4.3) allowing status information to be read at the same time as the FIFO. The structure of the Status Register is shown in table 5.

Bit	Description
31	Used Page FIFO is empty
30	S-LINK UXOFF asserted (see section 6.2)
29	Unassigned
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
21:00	Unassigned

Table 5: Status register

4.3 Used Page FIFO (UPF)

Offset: 0x002A0000
Access: Read only



The Used Page FIFO contains the addresses of the last word written to each used page along with the status of the ROB-in RP 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 *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.

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

Bit	Description
31	Used Page FIFO empty
30	S-LINK UXOFF (see section 6.2)
29	Unassigned
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	Fifoed S-LINK Link Data Error (LDERR)
22	Last page of event
21	Data arrived without a preceding Begin of Block control word
20	End of Block control word without Begin of Block control word
19	Begin of Block control word without End of Block control word
18	Input FIFO asserted almost full
17:08	Page number of used page
07:00	Number of last word written within page – Only valid for end of event entries

Table 6: Used Page FIFO

5 PCI interface

The i960 RP processor embedded in the ROB-in RP 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 RP, 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 RP'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 RP 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 RP the only bit of interest is the local bus reset bit (bit 5) which resets all devices on the i960 RP's local bus including the processor core. This bit is automatically cleared after operation.

An example of how to reset the i960 RP 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 RP shows up as Device ID 0x1960 with Vendor ID 0x8086 (intel).

The memory map of the ROB-in RP 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.

base address + 0000 0000	Messaging Unit
base address + 0000 1000	Program Memory
base address + 0010 0000	Reflection of Program Memory
base address + 0020 0000	Not Decoded
base address + 0022 0000	Control Register
base address + 0024 0000	Not Decoded
base address + 0026 0000	Status Register
base address + 0028 0000	Not Decoded
base address + 002A 0000	Used Page FIFO
base address + 002C 0000	Not Decoded
base address + 0030 0000	Fragment Memory
base address + 003F FFFF	

Table 7: Memory map of ROB-in RP 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 RP address space is marked as prefetchable in the PCI configuration registers.

6 S-LINK interface

6.1 Reset

The S-LINK reset protocol is handled by the control logic. There are no requirements made of the software.

6.2 UXOFF

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

- The UXOFF bit in the ROB-in Control Register (see section 4.1).
- The Half Full flag of the input FIFO.

The current state of the S-LINK UXOFF signal is available through the ROB-in Status Register (see section 4.2).

6.3 UTDO

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 fifo manual?? idt 72225