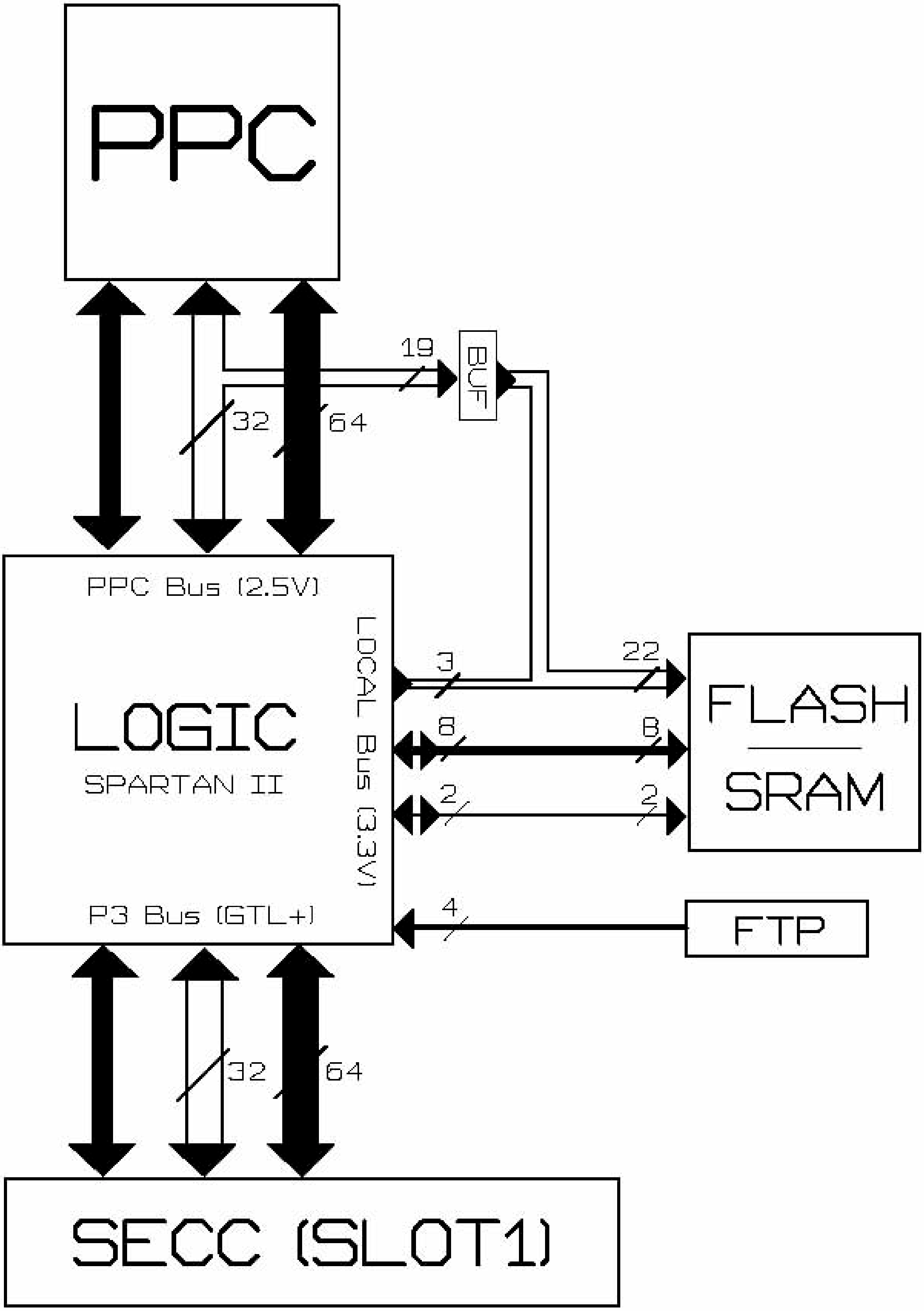
RIORED-J (RRJ) HARDWARE BOOK

© 2001 - 2002, Rodolphe Czuba

V. 3.0 – April 7th, 2002



RIORED-J

(c) SILICON FRUIT – April, 2001

MEMORY MAP

$FF00_0000	**FFFF_FFFF**	**16MB**	**FLASH Space** (Reset : $FFF00100)	
$FE00_0000	**FEFF_FFFF**	**16MB**	**SRAM Space**	
$FCF0_0000	FDFF_FFFF	17MB	Reserved	
$FCE0_0000	**FCEF_FFFF**	**1MB**	**LOCAL APIC Configuration Space**	Riored-J
$FC00_0000	FCDF_FFFF	14MB	Reserved	

$FBE0_0000	**FBFF_FFFF**	**2MB**	**X86 BIOS Aperture**	
$FBD0_0000	FBDF_FFFF	1MB	Not used	Mainboard
$FBC0_0000	**FBCF_FFFF**	**1MB**	**I/O APIC Configuration Space**	
$FB40_0000	FBBF_FFFF	8MB	Not used	
$FB30_0000	**FB3F_FFFF**	**1MB**	**SMI ACK**	
$FB20_0000	**FB2F_FFFF**	**1MB**	**INTR ACK**	
$FB10_0000	FB1F_FFFF	1MB	Not used	
$FB00_0000	**FB0F_FFFF**	**1MB**	**I/O Emulation Aperture**	
$MEMTOP+1	FAFF_FFFF		PCI Space	
$0010_0000	$MEMTOP	0.5-1.5GB	MAIN MEMORY	
$000E_0000	000F_FFFF	128KB	X86 BIOS 16-Bit Space (X86 Reset Vector : $000FFFF0)	
$000C_0000	000D_FFFF	128KB	EXPANSION Space (VIDEO BIOS = 32KB)	
$000A_0000	000B_FFFF	128KB	VIDEO BUFFER Space (mapped to PCI or AGP) / SMRAM	
$0000_0000	0009_FFFF	640KB	MEMORY SYSTEM (DOS)	

A - PC ARCHITECTURE :

Video Buffer Area ($A_0000 – B_FFFF)

The 128 KB graphics adapter memory region is normally mapped to a legacy video device on the hub interface/PCI (typically VGA controller). This area is not controlled by attribute bits and processor – initiated cycles in this region are forwarded to either the hub interface or the AGP or the Internal Graphics Device for termination.
This region is also the default region for SMM space (see below).

Expansion Area ($C_0000 – D_FFFF)

This 128 KB ISA Expansion region is divided into eight 16 KB segments. Each segment can be assigned one of four Read/Write states: read-only, write-only, read/write, or disabled. Typically, these blocks are mapped through Northbridge and are subtractively decoded to ISA space. Memory that is disabled is not remapped.
The 32 Kbyte Video BIOS is located between $000C0000 and $000C7FFF.

System BIOS Area ($E_0000 – F_FFFF) = 128KB

This area is a double 64 KB segment. These segments can be assigned read and write attributes**.**
It is by default (after reset) read/write disabled and cycles are forwarded to the SouthBridge.

High BIOS Area ($FFE0_0000 – FFFF_FFFF)

The top 2 MB of the extended memory region is reserved for system BIOS (High BIOS), extended BIOS for PCI devices, and the A20 alias of the system BIOS.
The processor begins execution from the High BIOS after reset.
The actual address space required for the BIOS is less than 2 MB but the minimum processor MTRR range for this region is 2 MB so that full 2 MB must be considered. The SouthBridges currently support a maximum of 1 MB in the High BIOS range.

BIOS Memory

SouthBridges support 1 Mbyte of BIOS memory space. This includes the normal 128-Kbyte space plus an additional 384 Kbyte (extended BIOS space) and 512 Kbyte of BIOS space (1M extended BIOS area).
The 128-Kbyte BIOS memory space is located at $000E_0000 – 000F_FFFF (top of 1 Mbyte) and is aliased at $FFFE_0000 (top of 4 Gbytes). This 128-Kbyte block is split into two 64-Kbyte blocks. Accesses to the top 64 Kbytes ($000F_0000 – 000F_FFFF) and its aliased region ($FFFF_0000 – FFFF_FFFF) are always forwarded to the ISA Bus and BIOSCS# is always generated. Accesses to the bottom 64 Kbytes ($000E_0000 – $000E_FFFF) are forwarded to the ISA Bus and BIOSCS# is only generated when this BIOS region is enabled (bit 6=1 in the XBCS Register). If this BIOS region is enabled, accesses to the aliased region at the top of 4 Gbytes ($FFFE_0000 – FFFE_FFFF) are also forwarded to ISA and BIOSCS# generated. If disabled, these accesses are not forwarded to ISA and BIOSCS# is not generated.

The 384KB extended BIOS space resides at $FFF8_0000 – FFFD_FFFF. If this BIOS region is enabled (bit 7=1 in the XBCS Register), these accesses are forwarded to ISA and BIOSCS# generated.
The 1M extended BIOS space resides at $FFF0_0000 – FFF7_FFFF. If this BIOS region is enabled (bit 9=1 in the XBCS Register), these accesses are forwarded to ISA and BIOSCS# generated.
If disabled, these accesses are not forwarded to ISA and BIOSCS# not generated.

APIC Configuration Space ($FEC0_0000 – FECF_FFFF, $FEE0_0000 – FEEF_FFFF)

This range is reserved for APIC configuration space and IO APIC configuration space. The default Local APIC configuration space is $FEE0_0000 to $FEEF_FFFF.
On RRJ, if implemented, the space may be $F**FB**0_0000 to $F**FB**F_FFFF.
Processor accesses to the local APIC configuration space do not result in external bus activity since the local APIC configuration space is internal to the X86 processor. However, a MTRR must be programmed to make the local APIC range uncacheable (UC).
The I/O APIC(s) usually reside in the I/O Bridge portion (I/O Controller Hub) of the chipset or as a stand-alone component(s) on old PC.

I/O APIC units will be located beginning at the default address $FEC0_0000. The first I/O APIC will be located at $FEC0_0000. Each I/O APIC unit is located at $FEC0_x000 where *x* is I/O APIC unit number 0 through F(hex). This address range will be normally mapped via the hub interface to PCI.
Note : **The address range between the APIC configuration space and the High BIOS ($FED0_0000 to $FFDF_FFFF) is always mapped via the hub interface to PCI.**

System Management Mode (SMM) Memory Range

The Northbridge supports the use of main memory as System Management RAM (SMRAM) enabling the use of System Management Mode (SMM).
The Northbridge supports three SMM options :

- Compatible SMRAM (AB segment enabled)
- High Segment (HSEG)
- Top of Memory Segment (TSEG)

System Management RAM (SMRAM) space provides a memory area that is available for the SMI handler's code and data storage. This memory resource is normally hidden from the operating system so that the processor has immediate access to this memory space upon entry to SMM.
The GMCH provides three SMRAM options :

- **Below 1 MB** option that supports compatible SMI handlers.
- **Above 1 MB** option that allows new SMI handlers to execute with write-back cacheable SMRAM.
- **Optional larger write-back cacheable** T_SEG area of either 512 KB or 1MB in size above 1 MB that is reserved from the highest area in system DRAM memory.

The above 1 MB solutions require changes to compatible SMRAM handler's code to properly execute above 1 MB.

The HSEG and TSEG SMM transaction address spaces reside in this extended memory area.
HSEG ($000A_0000 – 000B_FFFF)
SMM-mode processor accesses to enabled HSEG are remapped to $000A_0000 – 000B_FFFF. Non-SMM-mode processor accesses to enabled HSEG are considered invalid are terminated immediately on the FSB. The exception to this is non-SMM-mode write-back cycles. They are remapped to SMM space to maintain cache coherency. AGP and hub interface originated cycles to enabled SMM space are not allowed. Physical DRAM behind the HSEG transaction address is not remapped and is not accessible.
TSEG (Top of Main Memory–TSEG)
TSEG can be up to 1 MB and is at the top of memory. SMM-mode processor accesses to enabled TSEG access the physical DRAM at the same address. Non-SMM-mode processor accesses to enabled TSEG are considered invalid and are terminated immediately on the FSB. The exception is non-SMM-mode write-back cycles. They are directed to the physical SMM space to maintain cache coherency. AGP and hub interface originated cycle to enabled SMM space are not allowed.

B - PowerPC ARCHITECTURE :

System X86 BIOS Area ($E_0000 – F_FFFF) = 128KB

This area can be accessed by the PPC by two ways :

- a direct access to $000E_0000 - 000F_FFFF
- a hardware translated access to $FBFE_0000 - FBFF_FFFF, that is converted to $FFFE_0000 - FFFF_FFFF.

High BIOS Area ($FFE0_0000 – FFFF_FFFF) = 2MB

This area can be accesses by the PPC at a translated addresses area : $FBE0_0000 – FBFF_FFFF.

The actual address space required for the X86 BIOS is less than 2 MB (typically it is 256KB !) but the minimum X86 processor MTRR range for this region is 2 MB so that full 2 MB must be considered. The SouthBridges currently support a maximum of 1 MB in the High BIOS range.

PPC FLASH/SRAM Space

The PowerPC Flash/Sram space is 16 MBytes and is located at $FF00_0000 – FFFF_FFFF.
The Reset Vector Address is $FFF0_0100.

RRJ uses a chip from AMD that integrates both a FLASH and a SRAM.
Different Flash/Sram configurations are possible :
Flash / Sram

- **2 / 0.5 MBytes with the Am41DL163D (FBGA-69)**
- 4 / 0.5 Mbytes with the Am41DL324D (FBGA-73)
- 4 / 1 Mbytes with the Am41DL328D (FBGA-73)
- 8 / 1 Mbytes with the Am41DL6408G (FBGA-73)

Am41DL163D, Am41DL324/328D and Am41DL6408G are pin compatible. RRJ provides the A21line for implementation of the Am29DL324D/328D.
The fourth configuration needs modifications of the PCB : a Single Buffer (AHC) on the board for the A22 line.

The standard RRJ Flash is 2 MB chip and this gives enough place for a complete new BIOS replacing the X86 BIOS and avoiding a X86 emulation of the mainboard PC BIOS. Sure this last option assumes that a certain model of mainboard is chosen.

2 / 0.5 MBytes Configuration with the Am41DL163D (FBGA-69) :
FLASH : $FFE0_0000 - FFFF_FFFF
SRAM : $FE00_0000 – FE07_FFFF

4 / 0.5 MBytes Configuration with the Am41DL324D (FBGA-73) :
FLASH : $FFC0_0000 - FFFF_FFFF
SRAM : $FE00_0000 – FE07_FFFF

4 / 1 MBytes Configuration with the Am41DL324D (FBGA-73) :
FLASH : $FFC0_0000 - FFFF_FFFF
SRAM : $FE00_0000 – FE1F_FFFF

8 / 1 MBytes Configuration with the Am41DL6408G (FBGA-73) :
FLASH : $FF80_0000 - FFFF_FFFF
SRAM : $FE00_0000 – FE1F_FFFF

PowerPC Reset Exception Vector

At Power-On, the PPC reads an instruction fetch with a single-beat load operation (no cache) at offset 00100. The IP bit in the MSR (Machine State Register) is set to 1 by default (MSR = 00000040) for a exception base address = $FFFx_xxxx. If the IP bit is cleared, the vectors are placed at $000x_xxxx ! It is what should be done during the initialization of the system…

BIT & BYTE ORDERING

A- Byte Ordering Overview

For **big-endian** data, the **MSB is stored at the lowest (or starting) address** and the LSB is stored at the highest (or ending) address. This is called big-endian because the big (most-significant) end of the scalar comes first in memory.

For **little-endian** byte ordering, the **LSB is stored at the lowest address** while the MSB is stored at the highest address. This is called little-endian because the little (least-significant) end of the scalar comes first in memory.

PPC processors use, by default, the big-endian mode. It is possible to configure a PPC to run in little-endian mode but this is not a ‘true’ little-endian : it is called PowerPC little-endian byte ordering (also referred to as munged little-endian because of the addresses change method).

X86 processors use the little-endian.

The PCI bus and some other CPU busses (i.e. 68K familly) use a bit format where the most-significant bit (msb) for data is D31, while the PPC processors data bus uses a bit format where the msb is D0 (DH0). Thus, PCI data bit AD31 or 68K data bit D31 equates to the processor’s data bits DH0 and DL0, while PCI data bit AD0 and 68K data bit D0 equates to the PPC processor’s data bits DH31 and DL31.

Big-Endian Mode

The following example demonstrates the operation of a system in big-endian mode.
Starting with a program that does the following:
store string (‘hello world!’) at 0x000
store pointer (0xFEDCBA98) at 0x010
store halfword (0d1234) at 0x00E
store byte (0x55) at 0x00D

If the data is stored into local memory, it appears as shown :

DH0 (MSb)			**DH31**	**DL0**			**DL31**
‘H’	’E’	“L’	‘L’	‘O’	‘ ‘	‘W’	‘O’
$00	$01	$02	$03	$04	$05	$06	$07
‘R’	‘L’	‘D’	‘!’		55	12	34
$08	$09	$0A	$0B	$0C	$0D	$0E	$0F
FE	DC	BA	98				
$10	$11	$12	$13	$14	$15	$16	$17

Little-Endian Mode

The following example demonstrates the operation of a system in little-endian mode.
Starting with the same program :

D63 (MSb)							**D0**
‘O’	‘W’	‘ ’	‘O’	‘L’	‘L‘	‘E’	‘H’
$07	$06	$05	$04	$03	$02	$01	**$00**
34	12	55		’!’	’D’	’L’	’R’
$0F	$0E	$0D	$0C	$0B	$0A	$09	$08
				98	BA	DC	FE
$17	$16	$15	$14	$13	$12	$11	$10

B- Byte-Ordering Mechanisms

PPC		X86
DH0-DH7	←→	D7-D0
DH8-DH15	←→	D15-D8
DH16-DH23	←→	D23-D16
DH24-DH31	←→	D31-D24
DL0-DL7	←→	D39-D32
DL8-DL15	←→	D47-D40
DL16-DL23	←→	D55-D48
DL24-DL31	←→	D63-D56
A[0:28]	←→	A[31:3]

PPC A[29:31] are generated by the logic.

LINE TRANSFERS CONVERSION MECHANISM

A- Burst order

A line transfer reads or writes a cache line, the unit of caching on the PowerPC and P6 family processor system bus. This is 32 bytes aligned on a 32-byte boundary. While a line is always aligned on a 32-byte boundary, a line transfer need not begin on that boundary.
A line is transferred in four eight-byte chunks, each of which can be identified by two address bits. The chunk size is 64-bits.

Unfortunately, there is a big difference between PPC and P6 processors : the burst order.
PPC processors use a Wrap addressing when performing a cache line transfer and X86 processors use Toggle mode addressing. The following two tables show the differences for the requested addresses $08 & $18.
Note there is no difference at $00 and $10 and for Writes (cache write-back) because writes are always at $00 for both processors.

PPC

For a line transfer, A[0:28]# carry the upper 29 bits of a 32-bit physical address. Address bits A[27:28]# determine the transfer order, called burst order. The following table specifies the transfer order used for a 32-byte line, based on address bits A[27:28]# specified in the transaction's Address Phase.

A[4:3]# (binary)	**Requested** Address (hex)	**1st** Address Transferred (hex)	**2nd** Address Transferred (hex)	**3rd** Address Transferred (hex)	**4th** Address Transferred (hex)
00	00	0	8	10	18
01	08	8	**10**	18	**00**
10	10	10	18	0	8
11	18	18	**00**	8	**10**

X86

For a line transfer, A[31:3]# carry the upper 29 bits of a 32-bit physical address. Address bits A[4:3]# determine the transfer order, called burst order. The following table specifies the transfer order used for a 32-byte line, based on address bits A[4:3]# specified in the transaction's Request Phase.

A[27:28]# (binary)	**Requested** Address (hex)	**1st** Address Transferred (hex)	**2nd** Address Transferred (hex)	**3rd** Address Transferred (hex)	**4th** Address Transferred (hex)
00	00	0	8	10	18
01	08	8	**00**	18	**10**
10	10	10	18	0	8
11	18	18	**10**	8	**00**

B- Conversion mechanism

The two tables show that the second and the fourth double-word must be swapped during the burst.
For that a buffer must be installed into the logic.
Consider the four double-words are designated, in the transfer order, D1, D2, D3, D4 and the burst is started at address $ 08 or 18. Note the little<->big endian converter is active too.

Cycle 1 : D1 is transfered through the logic bridge from the X86 to the PPC data bus.
Cycle 2 : D2 is stored into the logic chip. PPC receives one Wait State (WS) from the logic.
Cycle 3 : D3 is stored into the logic chip. PPC receives one Wait State (WS) from the logic.
Cycle 4 : D4 is transfered through the logic bridge from the X86 to the PPC data bus.
Cycle 5 : D3 is sent to the PPC.
Cycle 6 : D2 is sent to the PPC.

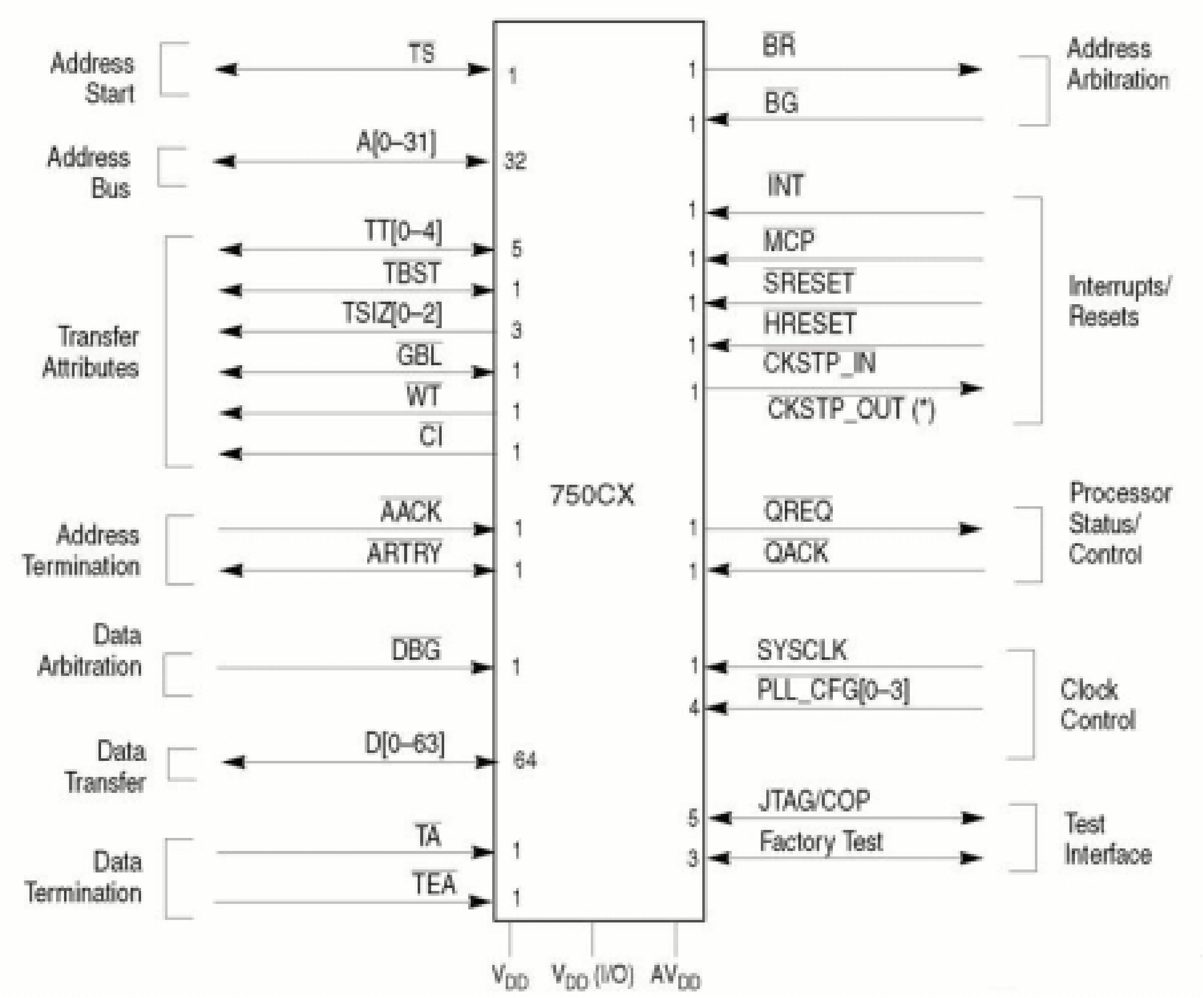
Result : PPC receives the line in the following order : D1, D4, D3, D2.
D4 and D2 are well swapped !

Performances : 6 cycles instead of 4, but only for 2 transfer cases among the 4, what gives a total time of 20 cycles instead of 16 (for 4 lines, each at a different address : two not are swapped and two are swapped.
The performances are decreased by **25% for reads, what may not be significant for the user !**

SYSTEM BUS

A- PPC750cx signals

‘#’ signifies the signal is active at the low level



Functional Groupings	Name	I/O	Notes
Address Arbitration	BR#	O	NOT USED by RRJ1 – USED by RRJ2
	BG#	I	
Address Bus	A[0:31]	IO	
Address Transfer Attributes	TS#	IO	
	TT[0:4]	IO	
	TBST#	IO	NOT USED by RRJ1 – Pull-up – USED by RRJ2
	TSIZ[0:2]	O	
	GBL#	IO	NOT USED by RRJ1 – Pull-up – USED by RRJ2
	WT#	O	
	CI#	O	NOT USED
Address Termination	AACK#	I	
	ARTRY#	IO	
Data Arbitration	DBG#	I	
Data Transfer	DH[0:31]	IO	
	DL[0:31]	IO	
Data Termination	TA#	I	
	TEA#	I	Only used by RRJ watchdog
Interrupts & Reset	INT#	I	Receives the INTR via the logic.
	MCP#	I	SMI# connection
	CKSTP_IN#	I	NOT USED – Pull-up
	CKSTP_OUT#	O	NOT USED
	SRESET#	I	Connected to HRESET
	HRESET#	I	
Processor Status & Control	QREQ#	O	NOT USED
	QACK#	I	NOT USED – Pull-up
Clock Control	SYSCLK	I	100 to 133 MHz
	PLL_CFG[0:3]	I	OPTION : Core CLK Multiplier Configuration

Power pins

Avdd	PLL supply
Vdd (40)	CORE supply
Ovdd (24)	IO supply
Gnd (53)	Ground

Pins with pull-up resistor (10K)

BVSEL	I	Bus Voltage : 0=1.8V / **1=2.5V**
DBWO#	I	Not used / L2_TSTCLK in test mode
L1_TSTCLK	I	Factory Test
LSSD_MODE#	I	Factory Test

Test Interface (JTAG/COP)

TRST#	I
TMS	I
TCLK	I
TDI	I
TDO	O

B- PPC750cx Transfer Type encodings

750CX Bus Master Transaction	Transaction Source	TT [0-4]	60x Bus Specification Command
Single-beat WRITE	Caching-inhibited or write-through store	**0** 0 0 1 0	Write-with-flush
Burst WRITE (nonGBL/)	Cast-out, or snoop copyback	**0** 0 1 1 0	Write-with-kill
Single-beat READ	Caching-inhibited load or instruction fetch	**0** 1 0 1 0	Read
Burst READ	Load miss, store miss, or instruction fetch	**0** 1 1 1 0	Read-with-intent-to-modify
Atomic Single-beat WRITE	stwcx	1 0 0 1 0	Write-with-flush-atomic
N/A	N/A	1 0 1 1 0	Reserved
Atomic Single-beat READ	Iwarx (caching-inhibited load)	1 1 0 1 0	Read-atomic
Atomic Burst READ	Iwarx (load miss)	1 1 1 1 0	Read-with-intent-to-modify-atomic

TT0 = 0 → normal access
= 1 → atomic access

TT1 = Read/Write : 1=R / 0=W

TT2 = 0 → single
= 1 → burst
TT3,TT4 = [1,0]

ECIWX & ECOWX optional intructions are not supported by RR-J.

C- SLOT1 signals

Request

A[31:3]#	IO	GTL+	
ADS#	IO	GTL+	
REQ[4:0]#	IO	GTL+	

Response

RS[2:0]#	I	GTL+	
TRDY#	I	GTL+	Writes only

Data Response

D[63:0]#	IO	GTL+	
DRDY#	IO	GTL+	
DBSY#	IO	GTL+	

Snoop (option)

HIT#	IO	GTL+	OPTION
HITM#	IO	GTL+	OPTION

Arbitration

BPRI#	I	GTL+	Priority Agent Bus Request
LOCK#	IO	GTL+	For ATOMIC cycles
BNR#	IO	GTL+	Block Next Request

Interrupts & Reset

RESET#	I	GTL+	
LINT0/INTR	I	CMOS 2.5V	Local APIC INTerrupt 0 / INTeRrupt – Default mode : APIC
SMI#	I	CMOS 2.5V	NOT present on PPC750cx : connected directly to MCP#

Option

LINT1/NMI	I	CMOS 2.5V	Local APIC INTerrupt 1 / Non Maskable Interrupt – Default mode : APIC

Status & Control

SLP#	I	CMOS 2.5V	NOT USED : SleeP (Power Management)
STPCLK#	I	CMOS 2.5V	NOT USED : StoP CLocK (Power Management)

Clock control

BCLK	I	CMOS 2.5V	Bus CLocK (100/133)
BSEL[1:0]	IO	3.3V	OPTION : Bus Frequency Select : 0:1 = 100 MHz / 1:1 = 133 MHz

Pins with pull-up resistor (10K)

FERR#	O	CMOS 2.5V	Fpu ERRor
THERMTRIP	O	CMOS 2.5V	THERMal TRIP : CPU is stopped !
TESTHI1	O	CMOS 2.5V	

Pins with pull-down (1K)

SLOTOCC#	O		Presence Detect
VID[3:0]	O		Voltage ID : 0101 = 1.8V
VID 4	O		Voltage ID : always 0

Power pins

VCC_CORE	I		Core power source
VCC_L2 / 3.3V	I		3.3V source for Riored-J Logic & PPC IO (2.5V), FLASH & DOC
VTT	I		GTL+ Terminators source : 1.5V
GND			

Pinout

SIGNAL	PIN	PIN	SIGNAL
NAME	(L)	(R)	NAME
VCC_VTT	**A001**	**B001**	**GND**
GND	**A002**	B002	Reserved
VCC_VTT	**A003**	B003	SMI#
Reserved	A004	B004	Reserved
Reserved	A005	**B005**	**VCC_VTT**
GND	**A006**	B006	*STPCLK#*
FERR#	A007	B007	Reserved
Reserved	A008	B008	*SLP#*
Reserved	A009	**B009**	**VCC_VTT**
GND	**A010**	B010	Reserved
Reserved	A011	B011	Reserved
Reserved	A012	B012	Reserved
TESTHI1	A013	**B013**	**VCC_CORE**
BSEL1	A014	B014	Reserved
THERMTRIP#	A015	B015	Reserved
Reserved	A016	B016	LINT1/NMI
LINT0/INT	A017	**B017**	**VCC_CORE**
GND	**A018**	B018	PICCLK
PICD0	A019	B019	Reserved
Reserved	A020	B020	Reserved
Reserved	A021	B021	BSEL0
GND	**A022**	B022	PICD1
Reserved	A023	B023	Reserved
Reserved	A024	B024	Reserved
Reserved	A025	**B025**	**VCC_CORE**
GND	**A026**	B026	Reserved
Reserved	A027	B027	Reserved
Reserved	A028	B028	Reserved
Reserved	A029	**B029**	**VCC_CORE**
GND	**A030**	B030	D62#
Reserved	A031	B031	D58#
D61#	A032	B032	D63#
D55#	A033	**B033**	**VCC_CORE**
GND	**A034**	B034	D56#
D60#	A035	B035	D50#
D53#	A036	B036	D54#
D57#	A037	**B037**	**VCC_CORE**
GND	**A038**	B038	D59#
D46#	A039	B039	D48#
D49#	A040	B040	D52#
D51#	A041	**B041**	**GND**
GND	**A042**	B042	D41#
D42#	A043	B043	D47#
D45#	A044	B044	D44#
D39#	A045	**B045**	**VCC_CORE**
GND	**A046**	B046	D36#
Reserved	A047	B047	D40#
D43#	A048	B048	D34#
D37#	A049	**B049**	**VCC_CORE**
GND	**A050**	B050	D38#
D33#	A051	B051	D32#
D35#	A052	B052	D28#
D31#	A053	**B053**	**VCC_CORE**
GND	**A054**	B054	D29#
D30#	A055	B055	D26#
D27#	A056	B056	D25#
D24#	A057	**B057**	**VCC_CORE**
GND	**A058**	B058	D22#
D23#	A059	B059	D19#
D21#	A060	B060	D18#
D16#	A061	**B061**	**GND**
GND	A062	B062	D20#
D13#	A063	B063	D17#
D11#	A064	B064	D15#
D10#	A065	**B065**	**VCC_CORE**
GND	**A066**	B066	D12#
D14#	A067	B067	D7#
D9#	A068	B068	D6#
D8#	A069	**B069**	**VCC_CORE**
GND	**A070**	B070	D4#
D5#	A071	B071	D2#
D3#	A072	B072	D0#
D1#	A073	**B073**	**VCC_CORE**
GND	**A074**	B074	RESET#
BCLK	A075	B075	Reserved
Reserved	A076	B076	Reserved
Reserved	A077	**B077**	**VCC_CORE**
GND	**A078**	B078	Reserved
Reserved	A079	B079	Reserved
Reserved	A080	B080	A29#
A30#	A081	**B081**	**GND**
GND	**A082**	B082	A26#
A31#	A083	B083	A24#
A27#	A084	B084	A28#
A22#	A085	**B085**	**VCC_CORE**
GND	**A086**	B086	A20#
A23#	A087	B087	A21#
Reserved	A088	B088	A25#
A19#	A089	**B089**	**VCC_CORE**
GND	**A090**	B090	A15#
A18#	A091	B091	A17#
A16#	A092	B092	A11#
A13#	A093	**B093**	**VCC_CORE**
GND	**A094**	B094	A12#
A14#	A095	B095	A8#
A10#	A096	B096	A7#
A5#	A097	**B097**	**VCC_CORE**
GND	**A098**	B098	A3#
A9#	A099	B099	A6#
A4#	A100	**B100**	**GND**
BNR#	A101	B101	SLOTOCC#
GND	**A102**	B102	REQ0#
BPRI#	A103	B103	REQ1#
TRDY#	A104	B104	REQ4#
DEFER#	A105	**B105**	**VCC_CORE**
GND	**A106**	B106	LOCK#
REQ2#	A107	B107	DRDY#
REQ3#	A108	B108	RS0#
HITM#	A109	**B109**	**VCC_5V**
GND	**A110**	B110	HIT#
DBSY#	A111	B111	RS2#
RS1#	A112	B112	Reserved
Reserved	A113	**B113**	**VCC_L2 / 3.3V**
GND	**A114**	B114	Reserved
ADS#	A115	B115	Reserved
Reserved	A116	B116	Reserved
Reserved	A117	**B117**	**VCC_L2 / 3.3V**
GND	**A118**	B118	Reserved
VID2	A119	B119	VID3
VID1	A020	B120	VID0
VID4	A121	**B121**	**VCC_L2 / 3.3V**

D- SOCKET 370

Request

A[31:3]#	IO	GTL+	
ADS#	IO	GTL+	
REQ[4:0]#	IO	GTL+	

Response

RS[2:0]#	I	GTL+	
TRDY#	I	GTL+	Writes only

Data Response

D[63:0]#	IO	GTL+	
DRDY#	IO	GTL+	
DBSY#	IO	GTL+	

Snoop (option)

HIT#	IO	GTL+	OPTION
HITM#	IO	GTL+	OPTION

Arbitration

BPRI#	I	GTL+	Priority Agent Bus Request
LOCK#	IO	GTL+	For ATOMIC cycles
BNR#	IO	GTL+	Block Next Request

Interrupts & Reset

RESET#	I	GTL+	
LINT0/INTR	I	CMOS 2.5V	Local APIC INTerrupt 0 / INTeRrupt – Default mode : APIC
SMI#	I	CMOS 2.5V	NOT present on PPC750cx : connected directly to MCP#

Option

LINT1/NMI	I	CMOS 2.5V	Local APIC INTerrupt 1 / Non Maskable Interrupt – Default mode : APIC

Status & Control

SLP#	I	CMOS 2.5V	NOT USED : SleeP (Power Management)
STPCLK#	I	CMOS 2.5V	NOT USED : StoP CLocK (Power Management)

Clock control

BCLK	I	CMOS 2.5V	Bus CLocK (100/133)
BSEL[1:0]	IO	3.3V	OPTION : Bus Frequency Select : 0:1 = 100 MHz / 1:1 = 133 MHz

Pins with pull-up resistor (10K)

FERR#	O	CMOS 2.5V	Fpu ERRor
THERMTRIP	O	CMOS 2.5V	THERMal TRIP : CPU is stopped !

Pins with pull-down (1K)

CPUPRES#	O		Presence Detect
VID[3:0]	O		Voltage ID : 0101 = 1.8V
VcoreDET	O		Indicates the type of core (2.0V or other).

Power pins

VCC_CORE	I		Core Power
VCC_2.5	I		2.5V Source (not used by P3/used by Celeron : may be not present ?!)
VCC_CMOS	O		CMOS signals Termination Voltage : connected to VCC2.5
Vref	I		GTL+ Vref source = 2/3 of VTT
VTT	I		GTL+ Terminators source : 1.5V

There are no 3.3V and 5V Supplies !

Power and Ground Pins

The operating voltage of the Pentium III processor for the PGA370 socket is the same for the core and the L2 cache; VCC CORE . There are four pins defined on the package for voltage identification (VID). These pins specify the voltage required by the processor core. These have been added to cleanly support voltage specification variations on current and future processors.

For clean on-chip power and voltage reference distribution, the Pentium III processors in the FC-PGA package have 75 VCC CORE , 8 V REF , 15 VTT, and 77 VSS (ground) inputs. VCC CORE inputs supply the processor core, including the on-die L2 cache. VTT inputs (1.5V) are used to provide an AGTL+ termination voltage to the processor, and the V REF inputs are used as the AGTL+ reference voltage for the processor. Note that not all VTT inputs must be connected to the VTT supply.

Three additional power related pins exist on a processors utilizing the PGA370 socket. They are VCC 1.5 , VCC 2.5 and VCC CMOS .

The VCC CMOS pin provides the CMOS voltage for the pull-up resistors required on the system platform. A 2.5V source must be provided to the VCC 2.5 pin and a 1.5V source must be provided to the VCC 1.5 pin. The source for VCC 1.5 must be the same as the one supplying VTT. The processor routes the compatible CMOS voltage source (1.5V or 2.5V) through the package and out to the VCC CMOS output pin. Processors based on 0.25 micron technology (e.g., the Intel Celeron processor) utilize 2.5V CMOS buffers. Processors based on 0.18 micron technology (e.g., the Pentium III processor for the PGA370 socket) utilize 1.5V CMOS buffers. The signal VCORE DET can be used by hardware on the motherboard to detect which CMOS voltage the processor requires.

A VCORE DET connected to VSS within the processor indicates a 1.5V requirement on VCC CMOS .

E- P6 BUS PROTOCOL

The modern North Bridges (NB) are optimized to support the Pentium II or Pentium III processor with the bus clock frequencies of 100 MHz or 133 MHz.
Many NB has an 8-deep In-Order Queue to support up to eight outstanding pipelined address requests on the host bus.
Host-initiated I/O cycles are positively decoded to AGP/PCI or NB configuration space and subtractively decoded to HUB interface (to South Bridge).
Host initiated memory cycles are positively decoded to AGP/PCI or DRAM and are again subtractively decoded to hub interface.
AGP semantic memory accesses initiated from AGP to DRAM are not snooped on the host bus.
Memory accesses initiated from AGP using PCI semantics and from either hub interfaces to DRAM will be snooped on the host bus.
All transactions are processed in the order that they are received on the host bus.

1- Transactions

Transaction	**REQa[4:0]#**	**REQb[4:0]#**	**Action**
Interrupt Acknowledge	01000	00x00	Interrupt acknowledge cycles are forwarded to the hub interface. A single byte of data is returned on HD[7:0]#.
Special	01000	00x01	See separate table in Special Cycles section.
I/O Read	10000	00xLEN#	I/O read cycles are forwarded to the hub interface A or AGP. I/O cycles to the NB configuration space are not forwarded to AGP or the hub interface.
I/O Write	10001	00xLEN#	I/O write cycles are forwarded to the hub interface A or AGP. I/O cycles to the NB configuration space are not forwarded to AGP or the hub interface.
Memory Read & Invalidate	00010	00xLEN#	Host-initiated memory read and invalidate cycles are forwarded to DRAM. The NB initiates an MRI (<=8 bytes) cycle to snoop a hub interface or AGP initiated write cycle to DRAM.
Memory Code Read	00100	00xLEN#	Memory code read cycles are forwarded to DRAM, hub interface or AGP.
Memory Data Read	00110	00xLEN#	Host-Initiated memory read cycles are forwarded to DRAM, the hub interface or AGP. The NB initiates a memory read cycle to snoop a hub interface or AGP initiated read cycle to DRAM.
Memory Write (no retry)	00101	00xLEN#	This memory write is a writeback cycle and cannot be retried. The NB forwards the write to DRAM.
Memory Write (can be retried)	00111	00xLEN#	The standard memory write cycle is forwarded to DRAM, hub interface or AGP.

NOTES:
1. LEN# = data transfer length as follows :
    00 <= 8 bytes (BE[7:0]# specify granularity).
    10 = 32 bytes (BE[7:0]# all active).

Interrupt Acknowledge Cycles

A processor agent issues an Interrupt Acknowledge cycle in response to an interrupt from an 8259-compatible interrupt controller. The Interrupt Acknowledge cycle is similar to a partial read transaction, except that the address bus does not contain a valid address. Interrupt Acknowledge cycles are always directed to the hub interface.

Upon recognizing the interrupt request, the P6 family processor issues a single Interrupt Acknowledge (INTA) bus transaction. INTR must remain active until the INTA bus transaction to guarantee its recognition.

Partial Reads

Partial Read transactions include I/O reads and memory read operations of less than or equal to eight bytes (four consecutive bytes for I/O) within an aligned 8 byte span. The byte enable signals, BE[7:0]#, select which bytes in the span to read.

Cache Line Reads

A read of a full cache line (as indicated by the LEN[1:0]=10 during request phase b) requires 32 bytes of data to be transferred. This translates into four data transfer cycles for a given request. Since the NB is the only response agent in the system, it is always selected as the target and will determine whether the address is directed to DRAM, the hub interface or AGP and provide the corresponding command and control to complete the transaction.

Partial Writes

Partial Write transactions include I/O (maximum of four bytes and memory write operations of eight bytes or less within an aligned 8-byte span. The byte enable signals (BE#[7:0]) select which bytes in the span to write. I/O writes crossing a 4-byte boundary are broken into two separate transactions by the host.

Cache Line Writes

A write of a full cache line requires 32 bytes of data to be transferred, which translates into four data transfers for a given request.

Locked Cycles

The NB support resource locking due to the assertion of the LOCK# line on the host bus as follows:

•Host<->DRAM Locked Cycles

The NB supports host to DRAM locked cycles. The P6 bus protocol ensures that the host bus will execute any other transactions until the locked cycle is complete. The NB arbiter may grant another hub interface or AGP device; however, any cycles to DRAM requiring cache coherency will be blocked.

•Host<-> I/O Controller Hub Locked Cycles

Any host to the ICH locked transaction will initiate a locked sequence to the hub interface. The P6 bus implements a bus lock mechanism that ensures that no change of bus ownership can occur from the time one agent has established a locked transaction (i.e., the initial read cycle of a locked transaction has completed) until the locked transaction is completed.
Note that for host transactions to hub interface, a “LOCK” special cycle is issued to establish the lock prior to the initial read and a “UNLOCK” special cycle is issued to the hub interface after the host lock transaction is completed.

Any concurrent cycle that requires snooping on the host bus is not processed while a LOCK transaction is occurring on the host bus.

Locked cycles from the hub interface to DRAM are not supported.

•Host<->AGP Locked Cycles

The AGP interface does not support locked operations; therefore, both host locked and non-locked transactions destined to AGP are propagated in the same manner.
On some NB, Host-AGP lock cycle result in a un-predictable system behavior.

Cache Coherency Cycles : Only for the snooping version of RRJ

1- The MCH generates an implicit writeback response during host bus read and write transactions when a processor asserts HITM# during the snoop phase. The host-initiated write case has two data transfers; the requesting agents data followed by the snooping agents writeback data.

2- The MCH performs a **memory read and invalidate cycle of length 0** (MRI[0]) on the host bus when a hub interface or AGP FRAME# snoopable DRAM write cycle occurs.

3- The MCH performs a **memory read cycle with length = 0** (MR[0]) on the host bus when a hub interface or AGP FRAME# snoopable DRAM read cycle occurs.

Memory Read and Invalidate (length > 0)

A Memory Read and Invalidate (MRI) transaction is functionally equivalent to a cache line read. The purpose of having this special transaction is to support write allocation (write miss case) of cache lines in the processors. When a processor issues a MRI, the cache line is read as in a normal cache line read operation; however, all other caching agents must invalidate this line if they have it in a shared or exclusive state. If a caching agent has this line in the Modified State, it must be written back to memory and invalidated and it is . The NB captures the write-back data.
It is illegal for a bus agent to assert HIT# on this transaction.

Memory Read and Invalidate (length = 0)

A Memory Read and Invalidate transaction of length zero (MRI0) does not have an associated Data Response. Executing the transaction informs other agents in the system that the agent issuing this request requires exclusive ownership of a cache line that maybe in the Shared State (write hit to a shared line).
Agents with this cache line invalidate the line. If this line is in the modified state, an implicit write-back cycle is generated and the NB captures the data.

The NB generates MRI(0) transactions for the hub interface and AGP memory write cycles to DRAM.

Memory Read (length = 0)

A Memory Read of length zero, MR(0) does not have an associated Data Response. This transaction is used by the NB to snoop for the hub interface to DRAM and AGP FRAME# snoopable DRAM read accesses.

The NB performs single MR(0) cycles for the hub interface reads less than or equal to 32 bytes and for AGP FRAME# master standard read or read line directed to DRAM.

2- Special cycles

A Special Cycle is defined when REQa[4:0] = 01000 and REQb[4:0]= xx001.
The first address phase Aa[35:3]# is undefined and can be driven to any value.
The second address phase, Ab[15:8]# (BE[7:0]#) defines the type of Special Cycle issued by the processor.

A special Cycle is “posted” into the MCH and the FSB transaction is terminated immediately after the cycle has been broadcast. It does not wait for the cycle to propagate or terminate on the hub interface interface.

The following table specifies the cycle type and definition as well as the action taken by the northbridges, that are supported by the RRJ :

BE[7:0]#	Cycle Type	Action
0000 0111	**SMI Acknowledge**	This transaction is first issued when an agent enters the System Management Mode (SMM). Ab[7]# is also set at this entry point. All subsequent transactions from the host with Ab[7]# set are treated by the NB as accesses to the SMM space. No corresponding cycle is propagated to the hub interface. To exit the System Management Mode the host issues another one of these cycles with the Ab[7]# bit deasserted. The SMM space access is closed by the MCH at this point.

3- Host Responses

RS[2:0]#	**Description**	**North Bridge support**
000	**Idle**	
001	**Retry**	This response is generated if an access is to a resource that cannot be accessed by the processor at this time and the logic must avoid deadlock. HUB directed reads and writes, DRAM locked reads, AGP reads and writes can be retried.
101	**No Data Response (Write)**	This is for transactions where the data has already been transferred or for transactions where no data is transferred. Writes and zero length reads receive this response.
110	Implicit Writeback	This response is given for those transactions where the initial transactions snoop hits on a modified cache line.
111	**Normal (Read)**	This response is for transactions where data accompanies the response phase. Reads receive this response.

Cache Coherency Cycles : Only for the snooping version of RRJ

Implicit Writeback

The NB generates an **implicit writeback response** during host bus read and write transactions when a processor asserts HITM# during the snoop phase. The host-initiated **write case** has two data transfers; the requesting agents write data followed by the snooping agents writeback data.

F- FEATURES NOT SUPPORTED

- **DEFERRED REPLY**
The P6 transaction ‘Deferred Reply’ is not supported because PPC doesn’t support it.

- **SNOOPING**
The PPC on RRJ don’t snoop the data flow between :

- PCI and MEMORY.
- USB and MEMORY.
- AGP and MEMORY (PCI initiated transfers) !

It is possible to solve this minor problem with carefully programming of the drivers !
Snooping may be implemented in the next version of RRJ.
Note that in the case of several PPC on RRJ, the snooping between the PPC is effective, what is the essential !

- LOCAL APIC & I/O APIC
The LOCAL APIC emulation is not implemented.
It is not possible to read the INT vector number by the APIC serial bus and the classical INT ACK cycle (via the PCI bus + North bridge) must be used. Be carefull because many motherboards are BIOS initialized in the APIC mode because all modern X86 CPU have this feature. The APIC protocol allows 24 INT instead of the 16 classical IRQ. If implemented, this may avoid to share several devices on the same IRQ !

- ECIWX & ECOWX optional PPC intructions are not supported.

LOCAL BUS

The **Local Bus** must not be confused with the **FSB (Front Side Bus)** that is the bus of the PPC.

The Local Bus is constitued by a 8-Bit DATA Bus, a 23-Bit ADDress bus and 5 control signals.

This bus is used by two devices :

- the FLASH/SRAM chip from AMD (2MB/512KB expandable up to 8MB/1MB).
- the FPGA that is used with the FTP to load the Flash from a Host PC computer.
  See the next chapter for further information.

The 2MB Flash needs 21 Address bits and a 8MB needs 23 address bits (LA0-LA22).

Two control signals are sent to the FLASH/SRAM chip for read & write (LOE# & LWE#). Flash & Sram Chip Selects are done from the PA6 add line and a single inverter gate (AHCT).

The RRJ hardware is optimized for a 90 ns access time FLASH/SRAM chip and a bus at 100 MHz.
This gives the following data rates :

WRITE FLASH (PROGRAM) : 9 cycles + 1 idle cycle between each FLH access (PPC feature).
To program, there are 3 command bytes followed by the data byte.
→ 1 byte / (10 x 4 cycles x 100) = **2.5 MBytes/s.**
READ FLASH (1 Byte wide) : 13 cycles + 1 idle cycle between each FLH access (PPC feature).
→ 1 byte / 14 cycles x 100 = **7.14 MBytes/s.**
READ FLASH & SRAM (8 Bytes wide) : (13 cycles x 8) + 1 idle cycle between each 8 bytes FLH access (PPC feature).
→ 8 byte / (13 x 8 cycles + 1) x 100 = **7.61 MBytes/s.**
WRITE SRAM (8 Bytes wide) : (9 cycles x 8) + 1 idle cycle between each 8 bytes SRAM access (PPC feature).
→ 8 byte / (9 x 8 cycles + 1) x 100 = **10.96 MBytes/s.**

FTP : Flash Transfert Port

RioRed-J is furnished to developers with a special port to allow the transfert of software (BIOS) into the on-board flash. It is a necessity when debugging a new BIOS by example. For prodcution, the flash could be programmed before the soldering.

This port uses 3 wires protocol from a standart PARALLEL port of the PC. Those three signals are direct from the signals DATA 0, STROBE# and SELECT_IN# . The transfer rate is from 100 to 1000 KBytes/s, depending of the software and the hardware at each side. The ECP // mode is recommended.
The FTP port uses a standart HE13 male 2x13 pin connector that has to be prolongated by a standard (DB25 connector + 26 wires ribbon cable) backpanel metal bracket.

In the following explanations, the 'HOST' term describes the PC (Personal Computer) you use to send the code in the Riored-J Flash.

The link between the HOST and RRJ must be done by a standart parallel cable or a cable that uses the minimal connections as described below.

For the standart port, the signal names are doubled : the first is the SPP (Standart Parallel Port) name and the second is the ECP (Extended Capabilities Parallel Port).

A- SPP/ECP standart connector (PC backpanel Female D-SUB25)

SPP	ECP	IO	SPP	ECP	IO
1 Strobe#	HostCLK#	Out	14 Autofeed#	HostACK	Out
2 Data 0	Data 0	I/O	15 Fault#	PeriphREQ	In
3 Data 1	Data 1	I/O	16 Init#	ReversREQ#	Out
4 Data 2	Data 2	I/O	17 Select_In#	1284 Active	Out
5 Data 3	Data 3	I/O	18 Gnd		
6 Data 4	Data 4	I/O	19 Gnd		
7 Data 5	Data 5	I/O	20 Gnd		
8 Data 6	Data 6	I/O	21 Gnd		
9 Data 7	Data 7	I/O	22 Gnd		
10 Ack#	PeriphCLK#	In	23 Gnd		
11 Busy	PeriphACK	In	24 Gnd		
12 PError	ACKReverse#	In	25 Gnd		
13 Select	X-flag	In			

B- RRJ FTP connector

Female DB25 (on a metal bracket)

1	nc	Out	14	nc
2	Data 0	I/O	15	nc
3	nc	I/O	16	nc
4	nc	I/O	17	nc
5	nc	I/O	18	Gnd
6	nc	I/O	19	Gnd
7	nc	I/O	20	Gnd
8	nc	I/O	21	Gnd
9	nc	I/O	22	Gnd
10	HostCLK	In	23	Gnd
11	nc	In	24	Gnd
12	nc	In	25	Gnd
13	FTP-On#	In		

On-Board HE13 Male 2x13 pins

1	nc	Out	14	nc
2	Data 0	I/O	15	nc
3	nc	I/O	16	nc
4	nc	I/O	17	nc
5	nc	I/O	18	Gnd
6	nc	I/O	19	Gnd
7	nc	I/O	20	Gnd
8	nc	I/O	21	Gnd
9	nc	I/O	22	Gnd
10	HostCLK#	In	23	Gnd
11	nc	In	24	Gnd
12	nc	In	25	Gnd
13	FTP-On#	In	26	nc

C- LINK CABLE

You have to use a ECP cable or to construct yourself your cable by connecting the following pins of two Male DB25 connectors :

Host		RRJ		
1	→	10	HostClock#	
2	→	2	Data 0	
17	→	13	FTP-On#	
18	-----	18	GND	
19	-----	19	GND	
20	-----	20	GND	To ensure good transferts,
21	-----	21	GND	it is important to connect at least 6 GND wires (two per signal) !
22	-----	22	GND	Use of a Ground Shielded cable is recommended.
23	-----	23	GND	The lengh of the cable should be reasonable : 2 meters.
24	-----	24	GND	
25	-----	25	GND	

Warning : a such minimal cable is not symetrical ! The left side must be connected to the HOST and the right side must be connected to RRJ. To avoid this detail you should add the connections 10 to 1 and 13 to 17 (left to right order).

10	→	1	HostClock#
13	→	17	FTP-On#

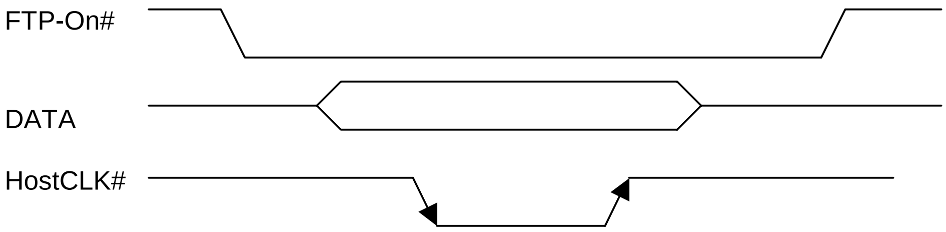
D- FTP PROTOCOL & CYCLES

The FTP protocol uses a subset of the ECP protocol.
Like with ECP, the HostCLK is used, but the PeriphACK (what should be named RRJ-ACK) is not used to acknowledge the transfert.

The FTP on-board logic contains a 22 bits address counter (A21-A0) needed for the 2MB Flash.

The order to send the 22 bits is MSB (Most Significant Bit) first.

Figure of the FTP Forward Cycle (Write) – 1 byte



1- FTP is activated. (FTP-On#) for a transfer.
2- Data is placed on data line D0 by Host.
3- Host indicates valid data by asserting HostCLK# low following by high (pulse = 1us).
4- Data is removed of data lines by Host.
5- FTP is desactivated at the end of the transfer.

LOGIC INTERFACE

FPGA logic elements

FLASH access :

- FLASH read for boot : 8, 16, 24, 32 or 64 bits access.
- FLASH read & write for programing : 8 bits access.

SRAM access : 8, 16, 24, 32 or 64 bits access.
DOC access : 8 bits access for read & write.

WATCHDOG : Generates TEA.

ARBITER : between PPC & NB.

DATA CONVERSIONS :

- BURST Linear (PPC) ←→ BURST Interleaved (X86).
- Big-endian ← → Little-endian (X86).

TRANSFERS Request & Responses Protocols conversion.

IO needs on FPGA

	(2.5)	**(3.3)**	**(GTL+)**	
	PPC	**LBUS**	**X86**	**TOTAL**
DATA interface	64	8	64	**136**
ADD interface	32	3	29	**64**
CNTL interface	17	2	16	**35**
FTP interface		**4**		**4**
TOTAL	**113**	**17**	**109**	**239**

Discret logic

- 74ALVCH16827 - 20-Bit Buffer : Local ADD (LA21-LA3).
- 74AHC1G04 – Single gate inverter : INTR (2.5V) → INT/ (2.5V) conversion.
- 74AHC1G04 – Single gate inverter : PADD inversion for FLASH/SRAM selection.

FPGA Solutions

SINGLE CHIP with SPARTAN II : XC2S150-5 FG456C (260 IO)

1K COST : **33$ -** Space : 23x23mm - 30 to 33 IO per blocks with 3 Vref per block

2.5V Pins : need is 113
→ 3 blocks of 33 IO = 99 IO
→ 1 block of 30 IO + 3 Vref : Rest 16 for GTL+

3.3V Pins : need is 17
→ 1 block of (29 IO + 3 Vref) : Rest 12 for GTL+

GTL+ Pins : need is 109
→ 3 blocks of (29 IO + 3 Vref) = 87 IO
87 + 16 + 12 = **115**

5 Blocks use their Vref pins for GTL+ → 15 IO are removed → 245 IO - FREE Pins : 6

SINGLE CHIP with VIRTEX 150 : XCV150-4 FG456C (260 IO)
1K COST : **85$ -** Space : 23x23mm

CHIPSETS

Bold	= compatible with RRJ
S	= SDRAM
R	= RAMBUS
D	= DDR SDRAM
EXT	= AGP with Add-on card & no AGP On-chip.
INT	= AGP On-chip.

Northbridge	**Chipset**	**FSB**	**RAM**	**AGP**	**Bus**	**SMP**	**Socket & processor types**
INTEL							
82443BX	**440BX**	66/100	66/100S	2x EXT	GTL+	2	?? : P2
82443GX	**440GX**	100	100 S	2x EXT	A/GTL+	2	?? : P2 & P2 XEON
82443MX100	with Southbridge !	66/100	66/100S	NO	GTL+	1	?? : P2 & Mobile Celeron
82810 GMCH	**810**	66/100	100 S	2x INT	AGTL+	1	Socket 370 : Celeron
82810E GMCH	**810E**	66/100/133	100 S	2x INT	AGTL+	1	Socket 370 & Slot1 : P2/P3/Celeron
82815 GMCH	**815**	66/100/133	100/133S	4x INT	GTL+	1	Socket 370 : P3 & Celeron
82815E GMCH	**815E**	66/100/133	100/133S	4x I/E	GTL+	?	Socket 370 : P3 & Celeron
82815EP MCH	**815EP**	66/100/133	100/133S	4x EXT	GTL+	1	Socket 370 : P3 & Celeron
82815EM GMCH2	**815EM**	100	100 S	4x I/E	GTL+	1	Socket 370 : P3 & Mobile Celeron
82820 MCH	**820**	100/133	400 R	4x EXT	AGTL+	2	Socket 370 & SC242 : P2/P3
82840 MCH	**840**	100/133	400 R	4x EXT	AGTL+	?	Slot1 : P3 & P3 XEON
82850 MCH	850	**A:200 / D:400**	400 R	4x EXT	AGTL+	?	**Socket 423 : Pentium 4**
VIA							
VT8501	Apollo MVP4	66/100	100 S	2x INT	**3.3V**	1	Socket7/Super7:Pentium/K6/K6-2/Cyrix6x86
VT82C691	**Apollo Pro**	66/100	100 S	2x AGP	GTL+	1	Slot1 : P2 & Socket 8 : Pentium Pro
VT82C693	**Apollo Pro Plus**	66/100	100 S	2x AGP	GTL+	1	Slot1 : P2 & Socket 370 : Celeron
VT8601	**Apollo Pro Media**	66/100/133	100/133S	2x INT	GTL+	1	Slot1 : P2/P3 & Socket 370 : Celeron
VT82C693A	**Apollo Pro 133**	66/100/133	100/133S	2x AGP	GTL+	1	Slot1 : P2/P3 & Socket 370 : Celeron
VT82C694X	**Apollo Pro 133A**	66/100/133	100/133S	4x AGP	GTL+	1	Slot1 : P2/P3 & Socket 370 : Celeron
ALI							
??	Aladdin V						
SIS							
SIS530	??						

PARTS & PRICES

IBM

PPC750CXE 550 MHz	**$275.00**	**1K**
	$192.50	10K
	$181.50	50K

XILINX VIRTEX

XCV50-4 FG256C	**$ 55.40 / 20**	**Insight US**	**Available**
	$ 35,87 / 1K	**Insight US**	**5 to 6 weeks**
XCV150-4 FG456C	**$ 85.00 / 1K**	**Memec FR**	
	$ 73.00 / 50K	**Memec FR**	
XCV150-5 FG456C	**$102.00 / 1K**	**Memec FR**	
	$ 87.00 / 50K	**Memec FR**	
XCV150-6 FG456C	**$122.00 / 1K**	**Memec FR**	
	$104.00 / 50K	**Memec FR**	
XCV200-4 FG456C	$112.43 / 1K	Insight US	5 weeks
XCV300-4 FG456C	$164,00 / 1K	Insight US	5 to 6 weeks

XILINX SPARTAN II

XC2S50-5 FG256C	**$ 16.00 / 1K**
	$ 12.00 / 50K
XC2S50-6 FG256C	**$ 18.00 / 1K**
	$ 14.00 / 50K
XC2S150-5 FG456C	**$ 33.00 / 1K**
	$ 25.00 / 50K
XC2S150-6 FG456C	**$ 38.00 / 1K**
	$ 29.00 / 50K

ALTERA MAX7000B

EPM7256B FC256-5	$108	/?
EPM7256B FC256-7	$46	/?
EPM7512B FC256-5	??	/ ?
EPM7512B FC256-7	$43	/ 1K
EPM7512B FC256-7	$48	/ Small Qty

LITTERATURE

Advanced PC Architecture (Pentium PRO & Pentium II)

Authors : Buchanan & Wilson
Editor : Addison Wesley

Architecture des machines Pentium

Authors : Dan Anderson & Tom Shanley
Editor : Mindshare