CTHG

CT60 + CTPCI HARDWARE GUIDE

Rev 8.0 – October 2000 to Jan 2012 - (c) Rodolphe Czuba

Please, try to not print on paper this file ! Using paper with computers is stupid and we need protect trees & oceans for CO2 absorbtion. Think it !

SUMMARY

FUNCTIONAL BLOCK DIAGRAM

ADDRESSES & REGISTERS

060 BUS SLOT

CONVENTIONS

EXPANSION BOARDS STACKING

PCI Device ID

Cables between main and PCI Slot board

CHIPS & REGISTERS EMULATION

THERMAL SENSOR

DIMM EEPROM

DIMM SDRAM for CT60

BURST with SDRAM

INTERRUPTS

CHIPSET ABE & SDR pinouts

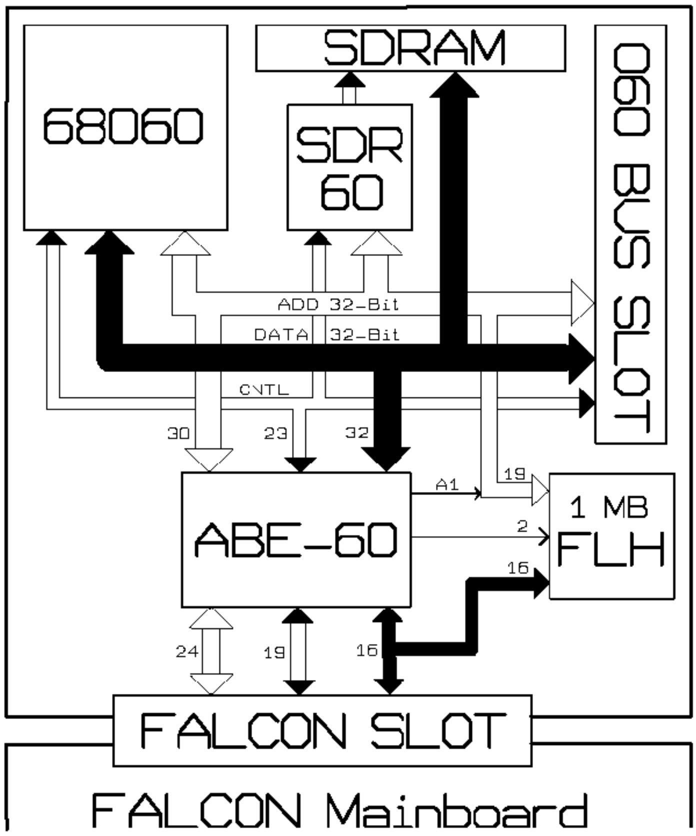
PLX9054 Bridge Overview

PLX9054 & PLX9080 (Milan) comparison

IDE PORT

PCI9054 Design Notes & PCI9054 Errata

CT60 FUNCTIONAL BLOCK DIAGRAM



CT60 FUNCTIONAL BLOCK DIAGRAM
(c) Rodolphe Czuba October, 2000

CONVENTIONS

Data Width	PCI 9054 & 060 Convention
1 byte (8 bits)	Byte
2 bytes (16 bits)	Word
4 bytes (32 bits)	Lword

For those of you who are a bit familiar with 64-bit processors like PPC or X86, don’t forget that the syntax for data size is not the same :

With 32-Bit processors :

- A **WORD** designates a **16-Bit entity**.
- **LONG-WORD** designates a **32-Bit entity**.

With 64-Bit processors :

- A **HALF-WORD** designates a **16-Bit entity**.
- A **WORD** designates a **32-Bit entity**.
- A **DOUBLE-WORD** designates a **64-Bit entity**.

PLX 9054 PCI bridge is set in MPC860 mode to connect with 68060. Thus the IBM PowerPC convention is used :
32-bits data is written D[0:31] with D0 for MSB and D31 for LSB.
Same for address bits and all control bits like TSIZ[0:1].

- PCI9054 Data D[0:31] bits are connected respectively to MC68060 D[31:0] data bits.
- PCI9054 Address A[0:31] bits are connected respectively to MC68060 A[31:0] address bits.
- Same for some control signals (PCI9054 TSIZ[0:1] with 060 SIZ[1:0] by example).

ADDRESSES & REGISTERS

68030 VIEW 24-Bit MAP

$xx000000	$xxDFFFFF	14 MB	ST-RAM
$xxE00000	**$xxEFFFFF**	**1 MB**	**TOS 4.0x ROM - BOOT**
$xxE00000	**$xxEFFFFF**	**1 MB**	**CT60 FLASH – CPU SPACE #3**
$xxF00000	$xxF00039	40 B	I/O IDE
$xxF00040	$xxF0FFFF	<64 KB	Unused
$xxF10000	$xxF9FFFF	576 KB	F030 BUS SLOT
$xxFA0000	$xxFBFFFF	128 KB	CARTRIDGE SLOT
$xxFC0000	$xxFEFFFF	192 KB	Unused
$xxFF0000	$xxFFFFFF	64 KB	I/O

CT60 32-Bit MAP

Falcon				
$00000000	$00DFFFFF	14 MB	ST-RAM	CACHE - NO BURST
$00E00000	**$00EFFFFF**	**1 MB**	**CT60/63 FLASH (for 24 bit add code)**	**CACHE - NO BURST**
$00F00000	$00F00039	40 B	IDE Port	NO CACHE - NO BURST
$00F00040	$00F0FFFF	<64 KB	*Reserved*	NO CACHE - NO BURST
$00F10000	$00F9FFFF	576 KB	F030 BUS Port	NO CACHE - NO BURST
$00FA0000	$00FBFFFF	128 KB	CARTRIDGE Port	NO CACHE - NO BURST
$00FC0000	$00FEFFFF	192 KB	*Reserved*	NO CACHE - NO BURST
$00FF0000	$00FFFFFF	64 KB	I/O	NO CACHE - NO BURST
SDRAM – Cache & Burst				
$01000000	**$04FFFFFF**	**64 MB**	**SDRAM (TT-RAM)**	
$01000000	**$08FFFFFF**	**128 MB**	**SDRAM (TT-RAM)**	
$01000000	**$10FFFFFF**	**256 MB**	**SDRAM (TT-RAM)**	
$01000000	**$20FFFFFF**	**512 MB**	**SDRAM (TT-RAM)**	
CT60 SLOT - Cache & Burst				
$21000000	$2FFFFFFF	240 MB	*Reserved*	
$30000000	**$3FFFFFFF**	**256 MB**	**SUPERVIDEL DDR SDRAM**	
$40000000	**$5FFFFFFF**	**512 MB**	**CTPCI : PCI MEM for 512 MB Space**	**NO CACHE**
$60000000	**$7FFFFFFF**	**512 MB**	**CTPCI : PCI MEM for 1GB Space**	**NO CACHE**
CT60 SLOT - No Cache/Burst				
$80000000	**$8000003F**	**256 B**	**CTPCI : PCI IO for 1GB Sp / EtherNAT**	**EtherNAT INT. Vect. $C4 & $C5**
$80000040	**$8FFFFFFF**	**255 MB**	**CTPCI : PCI IO for 1GB Space**	
$90000000	**$BFFFFFFF**	**768 MB**	**CTPCI : PCI IO for 1GB Space**	
$C0000000	**$DFFFFFFF**	**512 MB**	**CTPCI : PCI I/O for 512MB Space**	
$E0000000	**$E7FFFFFF**	**128 MB**	**CTPCI : CPLD Registers**	
$E8000000	**$EFFFFFFF**	**128 MB**	**CTPCI : PLX Registers**	
I/O – No Cache/Burst				
$F0000000	**$FBFFFFFF**	**192 MB**	**CT60/63 Registers**	
$FC000000	$FEEFFFFF	<48 MB	*Reserved for future CF & FLASH mem.*	
$FEF00000	**$FEF000FF**	**256B**	**CTPCI IDE Port (Secondary)**	
$FF000000	**$FFDFFFFF**	**14 MB**	FALCON 24-Bit ST-RAM SHADOW	
$FFE00000	**$FFEFFFFF**	**1MB**	**CT60/63 FLASH**	
$FFF00000	$FFF00039	57 B	IDE Port (Primary)	
$FFF00040	**$FFF00079**	**57 B**	**Reserved)**	
$FFF00080	$FFF000B9	57 B	Reserved for futur CF Reader	
$FFF000C0	$FFF000FF	85 B	*Reserved*	
$FFF00100	$FFF0FFFF	<63KB	*Reserved*	
$FFF10000	$FFF9FFFF	576 KB	*Reserved for future EXP. BUS Port*	
$FFFA0000	$FFFBFFFF	128 KB	FALCON 24-Bit CARTRIDGE SHADOW	
$FFFC0000	$FFFEFFFF	192 KB	*Reserved*	
$FFFF0000	$FFFFFFFF	64 KB	FALCON 24-Bit I/O SHADOW	

TOS :

From the 030, the FLASH chip is accessible (to program and read it) by the 030 ADDRESS SPACE #3.

From the 060, the TOS chip is NOT accessible.
The FLASH is seen at the TOS addresses when booting.
When programming the Flash in 060 mode, the ALTERNATE SPACE #3 must be used.

REGISTERS SUMMARY

SDR-60 chip

EE	EECL	$F0000000	I2C port for Clock setting & DIMM.
	EEDA	$F0800000	
TH	THCS	$F1000000	THermal sensor of the 060 (not implemented on CT63).
	THCK	$F1800000	
	THDA	$F1000000	
SDCNF		$F2000000	SDram CoNFiguration.
IDESWA		$F3000000	IDE SWAp (Swap the 2 IDE ports)

ABE-60 chip

SLP	$FA000000	Sleep = Turn OFF the ATX power supply.

CTPCI chip

INT PEND	$E0000000	Byte/word Read - No Write
ENABLE INT	$E0000001	Byte Read/Write
INT VECTOR	$E0000003	Byte Write - No Read
MISC.	$E0000023	Byte Write.

REGISTERS DETAIL

SDRAM EEPROM I2C Port :

EECL (EEprom serial CLock)

Write to $F0000000 → WRITE 0 to EECL line.
Write to $F0400000 → WRITE 1 to EECL line
Read to $F0000000 → READ from the EECL line on the D1 CPU data line.

EEDA (EEprom serial DAta)

Write to $F0800000 → WRITE 0 to EEDA line.
Write to $F0C00000 → WRITE 1 to EEDA line.
Read to $F0000000 → READ from the EEDA line on the D0 CPU data line.

060 THERMAL 3-wires Port (Not used on CT63) :

THCS (THermal Chip Select)

Write to $F1000000 → WRITE 0 to CS line.
Write to $F1400000 → WRITE 1 to CS line.

THCK (THermal ClocK)

Write to $F1800000 → WRITE 0 to CLK line.
Write to $F1C00000 → WRITE 1 to CLK line.

THDA (THermal DAta)

Read to $F1000000 → Read from the DO line on the D0 CPU data line.

SDRAM CONTROLLER

SDCNF (SDram CoNFiguration)

Write a long to $F2xx0000 with xx = [A23..A16]

Chip DensitY (EEPROM Byte #3 & #4)

A23 = cdy2
A22 = cdy1

			Byte#3	Byte#4
[cdy2,cdy2]	= 0,0	--> 8Mx8; 8x16	$0C	$09
	= 0,1	--> 16Mx8	$0C	$0A
	= 1,0	--> 16Mx16	$0D	$09
	= 1,1	--> 32Mx8; 32Mx16	$0D	$0A

NumbeR of DIMM Banks (EEPROM Byte #5)

A20 = nrb

[nrb]	= 0	--> 1 bank
	= 1	--> 2 banks

Module DensitY (EEPROM Byte #31 * EEPROM Byte #5)

A19 = mdy2
A18 = mdy1

[mdy2,mdy1]	= 0,0	--> 64MB
	= 0,1	--> 128MB
	= 1,0	--> 256MB
	= 1,1	--> 512MB

ReFresh RaTe (EEPROM Byte #12)

A16 = rfrt

[rfrt]	= 0	--> 15.360 uS
	1	--> 7.680 uS

IDE SWAP

IDESWA

Write to $F3800000 → SWAP IDE ports -> CTPCI is Primary IDE & F30 is secondary IDE.
Write to $F3000000 → Default/No SWAP -> F030 is Primary IDE & CTPCIis secondary IDE.

POWER CONTROL

SLP (SleeP)

Write to $FA800000 → Turn OFF the power supply.

INT PEND - Byte/word Read to $E0000000 - No Write

Data bits	Name Value	Inactive	Function
0	INTP	0	INT PLX
1	INTA	0	INT #A
2	INTB	0	INT #B
3	INTC	0	INT #C
4	INTD	0	INT #D
5	x	x	Not used – write 0 for future hardware
6	x	x	Not used – write 0 for future hardware
7	x	x	Not used – write 0 for future hardware

***Note :** Enable INT & INT PEND registers can be read by one WORD access at $E0000000*

ENABLE INT - Byte Read/Write to $E0000001

Data	Name	Default	Function
bits	Value		
0	ENBIP	0	ENaBle INT PLX
1	ENBIA	0	ENaBle INT #A
2	ENBIB	0	ENaBle INT #B
3	ENBIC	0	ENaBle INT #C
4	ENBID	0	ENaBle INT #D
5	x	x	Not used – write 0 for future hardware
6	x	x	Not used – write 0 for future hardware
7	x	x	Not used – write 0 for future hardware

Note : *Enable INT & INT PEND registers can be read by one WORD access at $E0000000*

MISC. - Byte Write to $E0000023.

Data	Name	Default	Function
bits	Value		
0	PCIRST	0	Set to 1 : Reset the PLX, the PCI slots & the PCI arbiter.
1	ITF	0	Set to 1 : Fast IDE timings for 75-100 Mhz clocks
2	PCIMAP	0	Set to 1 : PCI mapping = 2*512MB (default = 2*1GB)
3	x	x	Not used – write 0 for future hardware
4	x	x	Not used – write 0 for future hardware
5	x	x	Not used – write 0 for future hardware
6	x	x	Not used – write 0 for future hardware
7	x	x	Not used – write 0 for future hardware

Note : *This register is used by software to test the CTCPI presence.*
Write to $E0000020 : if bus error -> no CTPCI

CTPCI INT VECTOR - Byte Write to $E0000003 - No Read

Data	Name	Default	Function
bits	Value		
0	x	x	Not used – write 0 for future hardware
1	x	x	Not used – write 0 for future hardware
2	x	x	Not used – write 0 for future hardware
3	x	x	Not used – write 0 for future hardware
4	VE4	0	Int. Vector address bit#4
5	VE5	0	Int. Vector address bit#5
6	VE6	0	Int. Vector address bit#6
7	VE7	0	Int. Vector address bit#7

CTPCI board to PCI slots board cables

There are 2 ribbon cables between the CTPCI main board and the PCI connectors board.
The two cables are made with ATA 80 wires standart cables.

It is necessary using the system and master ends of the flat ATA cables. The middle (slave) connector of the flat cable cannot be used. So it is not possible to cut the flat cable to reduce the length using system and slave ends.

Cable #1 (connecteur J5)

Pin #	Signal Function	Pin #	Signal Function
1	INTA#	2	Ground
3	INTC#	4	INTD#
5	GNT1#	6	REQ1#
7	REQ2#	8	CLK3
9	CLK4	10	GNT4
11	AD30	12	AD29
13	AD26	14	AD25
15	AD23	16	AD22
17	AD19	18	AD18
19	Ground	20	Key
21	C/BE2#	22	Ground
23	IRDY#	24	Ground
25	DEVSEL#	26	Ground
27	LOCK#	28	PERR#
29	C/BE1#	30	Ground
31	AD14	32	AD13
33	AD10	34	NC
35	AD8	36	C/BE0#
37	AD5	38	AD4
39	AD1	40	Ground

Cable #2 (connecteur J6)

Pin #	Signal Function	Pin #	Signal Function
1	INTB#	2	Ground
3	RST#	4	CLK1
5	CLK2	6	GNT2#
7	GNT3#	8	REQ3#
9	REQ4#	10	AD31
11	AD28	12	AD27
13	AD24	14	C/BE3#
15	AD21	16	AD20
17	AD17	18	AD16
19	Ground	20	Key
21	FRAME#	22	Ground
23	TRDY#	24	Ground
25	STOP#	26	Ground
27	SERR#	28	PAR
29	AD15	30	Ground
31	AD12	32	AD11
33	AD9	34	NC
35	AD7	36	AD6
37	AD3	38	AD2
39	AD0	40	Ground

CT BOARDS STACKING

Here is the stacking of the daughter boards of the CT60/63 serie.

Top : EtherNAT
SuperVidel
CTPCI
Bottom : CT60/63
Falcon 030

PCI DEVICES ID

PCI Bridge	IDSEL = 20
PCI slot#1	IDSEL = 21
PCI slot#2	IDSEL = 22
PCI slot#3	IDSEL = 23
PCI slot#4	IDSEL = 24

060 BUS SLOT

A 060 bus Slot is present on the CT60 for some daughter cards : CTPCI, EtherNAT , SuperVidel.

The connector has 100 pins (2 connectors of 2x25 pins) and furnishes the following signals and power lines :

ADDRESS & DATA

A31-A0	**A**ddress Bus
D31-D0	**D**ata Bus

TRANFER CONTROL

/TS	**T**ransfer **S**tart
R/W	**R**ead **W**rite
/BS0, BS1/, /BS2, /BS3	**B**yte **S**elect (BS0 is for D31-D24 lane)
SIZ1, SIZ0	**SIZ**e
TT1, TT0, TM2, TM1, TM0	**T**ransfer **T**ype & **T**ransfer **M**odifier
/TA	**T**ransfer **A**cknowledge

ARBITRATION

/BR	**B**us **R**equest
/BG	**B**us **G**rant
/BB	**B**us **B**usy

INTERRUPTS

/TEA	**T**ransfer **E**rror **A**cknowledge
/RST	**R**e**S**e**T**
/I6	**I**nterrupt **6** : Sent by the daughter card to the CT60

MISC

/IDE	**IDE** decoding signal ($00F0xxxx & $FFF0xxxx).
/DTKCMB	**DTK** from **C**O**MB**EL - Reserved

CLOCK

CLK	**CL**oc**K** (CT60 clock : 64MHz or more)

POWER

-12V, +12V	Power supplies : used by some PCI cards and the fans (+12).
+3.3V, +5V	Power supplies : used by components and processors. Each pin can drive up to 6 Amperes.
GND (9)	**G**rou**ND** pins.

Pinout

Add Connector		**Data Connector**	
#A1	***#A2***	***#D1***	***#D2***
GND	GND	GND	GND
+5V	/BR	TM0	+12V
/TS	/BG	TM1	-12V
R/W	/BB	TM2	+3.3V
SIZ0	SIZ1	TT0	TT1
/TA	/TEA	/BS0	/BS1
/DTKC	CLK	/BS2	/BS3
/RST	GND	/I6	/IDE
A30	A31	D0	D1
-	-	-	-
-	-	-	-
A0	A1	D30	D31
GND	GND	GND	GND
#A49	***#A50***	***#D49***	***#D50***

CHIPS & REGISTERS EMULATION

The CT60 allows a **Falcon hardware emulation**.
With this Hardware Emulation, it is easy to implement a new chip replacing the old one of the Falcon motherboard and this at the same address(es) !

Examples :

- SUPER-VIDEL chip.
- SDRAM replacement of a part of the ST-RAM at the same addresses.
- ACIA for new PS/2 ports with a CPLD/FPGA.
- ACIA MIDI with a CPLD/FPGA.
- DSP56301 replacing 56001 at same addresses !
- FPGA emulating serial & parallel port of the Falcon (Zilog 85C30 and Yamaha).
- new SDMA for Audio.

Timing window
There is a time window from the start of the 060 access to the Falcon addresses ($00xxxxxx and $FFxxxxxx) up to the start (rising edge) of the 7th cycle of the CLK (bus and 060 clock). This method is only recommended for fast response new hardware on daughter board.

When the 060 inserts the address and TS to validate an access, a counter into ABE starts if the address is somewhere in the Falcon address space.
Until the end of the 6 th cycle, a card on the 060 slot bus of the CT60 can answer to terminate the access instead of a chip of the Falcon mb (with TA/ or TEA/ or both TA/ & TEA/ for a RETRY).
This termination of the access terminates and invalidates the Falcon access that was started.

At the begining of the 7th cycle the Falcon READ access continues and cannot be stopped. ABE drives data on the CT60 bus.
The time limit for the termination signal sampling is the end of 6 th cycle.

If you want **to use SDRAM on a daughter card** :
For 66 MHz SDRAM BURST READ you need 5-1-1-1 cycles.
The TA arrives the 5th cycle (first data) up to 8th (fourth data). This TA arrives before the end of the 6th cycle and the F30 access start is cancelled.
For 66MHz SDRAM BURST WRITE you need 3-1-1-1 cycles.
The TA arrives the 3rd cycle (first data) up to 6th (fourth data). This TA arrives before the end of the 6th cycle and the F30 access start is cancelled.

For **registers accesses on a daughter card**, you need 2 or 3 cycles.
If you want to write both to F030 mb AND your daughter card (an adress that is present on the two boards), don't send TA from the daughter card and the TA from mb will terminate the write access for you.

By example, this technic allows to write all VIDEL and SUPER VIDEL registers in the same time. The emulation is total! The only thing there is to do is to implement a bit in the daughter board to **switch ON/OFF the emulation**.

If the switch is ON :
- the daughter card address registers are at the same addresses than the F030 mb and :
- the TA must not be sent when writting these registers that are common to F030 and the daughter card.
- the TA must be sent before the 7th cycle when reading from register that is a common to F030 & daughter card.

If the switch is OFF :
- the daughter card address registers must be present at some specific addresses (not the same than the F030) and the TA is sent as usual byt the daughter card for all read & write accesses.

Example with $FFFF820E :
Switch is ON --> Write at $FFFF820E write to daughter card and F030 mb and this access is terminated by the TA from Falcon mb (ABE). The card don't send TA.
Switch is OFF --> Write at $FFFF820E write only to Falcon mb. You need to write to a 'new' address on the card to access the same register.

THERMAL SENSOR

The 68060 contains a Die Temperature Sensor with two external pins THERM0 & THERM1. The sensor is done with a temperature sensitive resistor which has a 780 ohms value at 25℃ and increases/decreases by steps of 2.8 ohms per ℃ unit. By example, a 060 core at 80℃ gives a resistance of 934 ohms between the two THERMx pins.
Equations : R60 = 780 + 2.8 x (TEMP – 25) or TEMP = (R60 - 710) / 2.8

The CT60 uses a small slow Analog/Digital converter (TI TLV0831) to obtain a 8-Bit value of the voltage between the THERMs pins.
The equation is : U60 = (3.34 x R60) / (1000 + R60) where R60 is the value of the core sensor resistor; 3.34 is the power supply and 1000 is the value of the resistor connected between the 3.34V and the positive THERM0/IN+ line.
TOLERANCES :

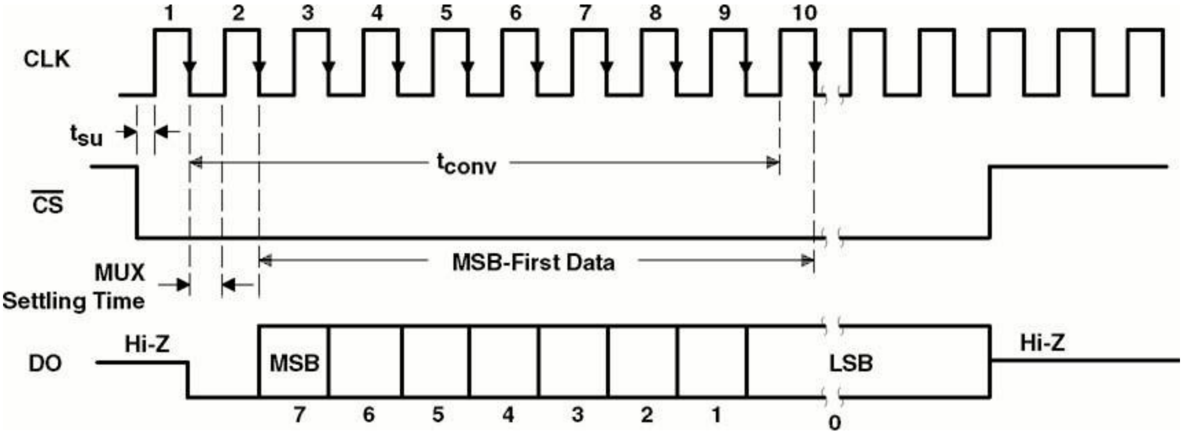
- Power supply : 3.3V +/- 4% → From 3.168V to 3.432V. Curently, it is 3.30 to 3.34.
- Resistor : 1K +/-1% → From 990 to 1010 Ohms. Curently, it is from 995 to 1005 ohms.

The AD converter uses a **REF voltage of 1.800 V**. With 0 to 0.007 V between the two pins of the AD converter, the digital result is 0. With 1.794 to 1.800 V, the result is 255. The value increases/decreases by **steps of 0.007 V**.
The equation is : Data = INT [U60/0.007].

At 0℃ :	Vin+ = 1.387 V	Data = 197	R60=710
At 25℃ :	Vin+ = 1.464 V	Data = 208	R60=780
At 50℃ :	Vin+ = 1.535 V	Data = 218	R60=850
At 100℃ :	Vin+ = 1.662 V	Data = 236	R60=990

ATTENTION : The varation of the data is not linear !!

The CPU must access the TLV831 by a basic bit-by-bit protocol. It is the software responsibility to respect the protocol & timings of the following chronogram, and assemble the bits.



f : Clock frequency	10 to 600 kHz (typical = 250)
tsu : Setup time, CS LOW before CLK goes HIGH	350 ns MIN
tpd : Propagation delay time :output data after CLK goes HIGH	500 ns MAX (typical = 200)
twh : Pulse duration, CS HIGH	220 ns MIN
tconv : Conversion Time (at 250kHz)	32 us

Three registers are present in the SDR60 chip.
The 060 CPU must drive THCS & THCK and read THDA by these registers.
The address $F1000000, $F1800000 & $F1000000 are used respectively for THCS, THCK & THDA.

THCS (Chip Select)

LONG WRITE at $F1000000	WRITE 0 to CS	Rising edge of CS (removed)
LONG WRITE at $F1400000	WRITE 1 to CS	Falling edge of CS (active)

THCK (Clock)

LONG WRITE at $F1800000	WRITE 0 to CLK	Falling edge of CLK
LONG WRITE at $F1C00000	WRITE 1 to CLK	Rising edge of CLK

THDA (Data Output)

LONG READ at $F1000000	READ from DO – Value is available on D0 of the CPU data bus.

For an example, see the example in the DIMM EEPROM chapter.

DIMM EEPROM

EEPROM DATA

The DIMM standard allows the loading of the manufacturer informations from a small 128 or 256 bytes EEPROM on the DIMM. Some of these informations are needed to configure the SDRAM controller of the CT60.

The following bytes are uses by the CT60 :

- **Bold** are used by the boot software to configure the SDRAM controller.
- Others are used only as user information in a SET UP menu.

Byte #2	Memory Type	FPM; EDO; NIBBLE; **SDRAM=$04**
Byte #3	**Number of Row Addresses**	**12=$0C; 13=$0D**
Byte #4	**Number of Column Addresses**	**8=$08; 9=$09; 10=0A**; 11=$0B
Byte #5	**Number of DIMM Banks**	**1=$01; 2=$02**
Byte #6 & 7	Module Data Width	**64=$4000;** 72; 80
Byte #8	Voltage Interface Level of this assembly	TTL; **LVTTL=$01;** HSTL; SSTL3; SSTL2
Byte #9	SDRAM Cycle Time (tCYC)	
Byte #10	SDRAM Access from Clock (tAC)	
Byte #11	SDRAM Configuration Type	**None=$00**; Parity; ECC
Byte #12	**Refresh Rate**	**15.625uS=$80; 7.81uS=$82**
Byte #17	Number of Banks on SDRAM Device	2; **4=$04**
Byte #27	Minimum ROW Precharge Time (tRP)	
Byte #28	Minimum ROW Active to Active Delay (tRRD)	
Byte #29	Minimum RAS to CAS Delay (tRCD)	
Byte #31	**Module Bank Density**	**32=$08; 64=$10; 128=$20; 256=$40; 512=$80**
Byte 64-71	Module Manufacturer's JEDEC ID Code	EX : $**A4000000 = IBM**
Byte 73-90	Module Part Number	
Byte 93-94	Module Manufacturing Date	
Byte 95-98	Module Serial Number	

Some features are initialized by the logic chip into the DIMM module when booting :
- **BURST Length** 1, 2, **4**, 8 , Page **4** is for **060, PPC, X86** processors
- **CAS Latency** **2**, 3, 4, … **2** is possible with **PC100** at 66 up to 80 MHz !

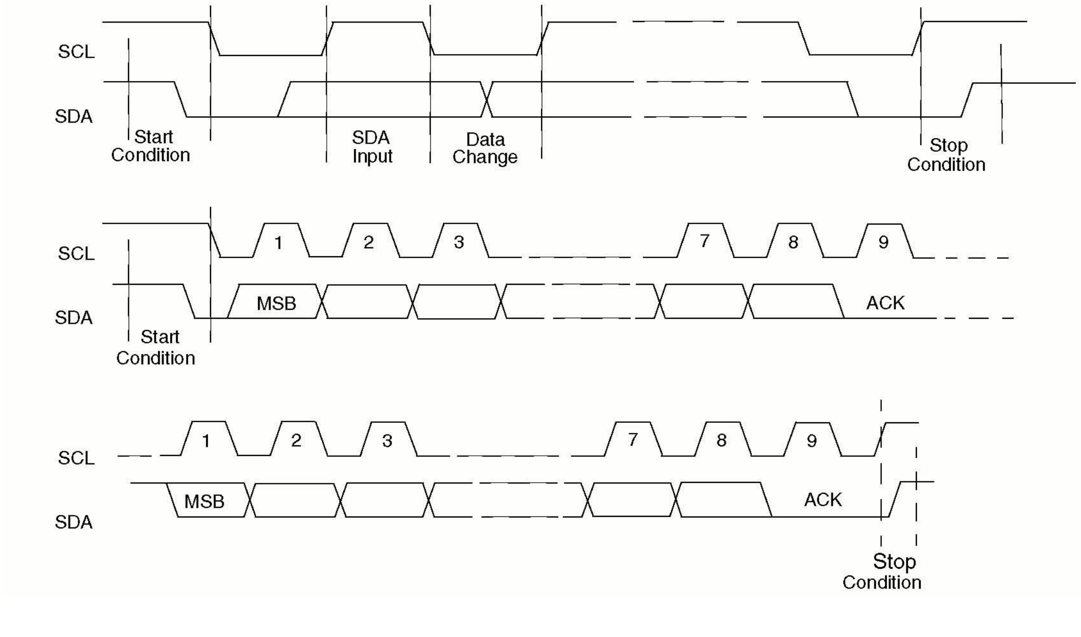
Remarks :

- Don't confuse SDRAM banks (2 or 4) with DIMM banks (1 or 2) !
- Bytes 128-255 are open for Customer Use and can be written – Not used with CT60.
- DIMM Density = Module Bank Density * Number of DIMM Banks (1 or 2).

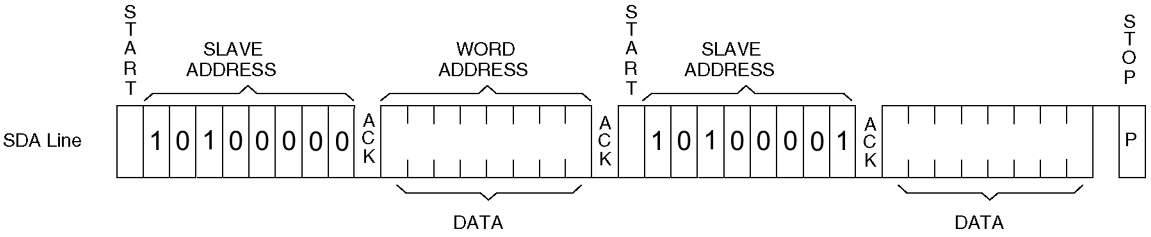
I2C 2-wire PROTOCOL

The EEPROM device conforms to the I2C 2-wire protocol. CT60 uses only the **random read operations** with the EEPROM.

During data input, the EEPROM samples the SDA signal on the rising edge of the clock (SCL). For correct device operation, the SDA signal must be stable during the clock low to high transition and data must change only when the clock (SCL) line is low.



RANDOM READ PROTOCOL & SOFTWARE



The slave address is 1010000. The eight bit is the R/W bit.

Random read operations allow the master to access any memory location in a random manner. Before issuing the slave address with the R/W bit set to one (Read), the master must first perform a dummy write operation. The master issues the start condition, slave address and then the word address it is to read. After the word address ACK, the master immediately re-issues the start condition and the slave address with the R/W bit set to one. This will be followed by an ACK from the slave and then by the eight bit word. The master will not ACK the transfer but will issue a stop and the slave stops transmission and goes into standby.
The device that controls the transfer is referred to as the master (SDR60 chip) and the device that receives the data (EEPROM) is referred to as the slave device. The master will always start a data transfer (SDA line) and will provide the serial clock (SCL line) for synchronization.

The 060 CPU must drive the SCL and SDA lines. These lines are connected to 2 pins of the logic chip.
The address $F00xxxxx is used for SCL and $F08xxxxx is used for SDA signal.

SCL (Clock)

LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL
LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL

SDA (Data)

LONG WRITE at $F0800000	WRITE 0 to SDA
LONG WRITE at $F0C00000	WRITE 1 to SDA
LONG READ at $F0800000	READ from SDA – Value is available on D0 of the CPU data bus.

EXAMPLE

If you want to read the Byte #3 from the EEPROM :

START condition

LONG WRITE at $F0800000	WRITE 0 to SDA	
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

SLAVE ADDRESS (Write at 1010000)
Write ‘1’

LONG WRITE at $F0C00000	WRITE 1 to SDA	
LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

Write ‘0’

LONG WRITE at $F0800000	WRITE 0 to SDA	
LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

…
Repeat for the values 10000 (the last 0 is for ‘write’)

ACK condition

LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG READ at $F0800000	READ from SDA	If =0, it’s an ACK
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

WORD ADDRESS DATA (# 3 in this example)
Write '0'

LONG WRITE at $F0800000	WRITE 0 to SDA	
LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

…
Repeat 5 times

Write '1'

LONG WRITE at $F0C00000	WRITE 1 to SDA	
LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

…
Repeat 1 time

ACK condition

LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG READ at $F0800000	READ from SDA	If =0, it's an ACK
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

START condition immediately after ACK

LONG WRITE at $F0C00000	*WRITE 1 to SDA*	
LONG WRITE at $F0400000	*WRITE 1 to SCL*	*Rising edge of SCL*
LONG WRITE at $F0800000	WRITE 0 to SDA	
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

SLAVE ADDRESS (Read at 1010000)
Write '1'

LONG WRITE at $F0C00000	WRITE 1 to SDA	
LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

Write '0'

LONG WRITE at $F0800000	WRITE 0 to SDA	
LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

…
Repeat same procedure for the values 10001 (the last 1 is for 'read')

ACK condition

LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG READ at $F0800000	READ from SDA	If =0, it's an ACK
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

READ WORD DATA
Bit#7

LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG READ at $F0800000	READ from SDA	DATA Bit#7
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

Bit#6

LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG READ at $F0800000	READ from SDA	DATA Bit#6
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

…
Repeat 6 times

Clock cycle (NO ACK)

LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG WRITE at $F0000000	WRITE 0 to SCL	Falling edge of SCL

STOP condition

LONG WRITE at $F0800000	WRITE 0 to SDA	
LONG WRITE at $F0400000	WRITE 1 to SCL	Rising edge of SCL
LONG WRITE at $F0C00000	WRITE 1 to SDA	

DIMM SDRAM for CT60

The CT60 supports the **PC100/133 standard SDRAM DIMMs**, but some obsolet/uneeded are not accepted.

GOOD SDRAM DIMMs for CT60
- **PC-100 & PC-133**.
- Must be **UNBUFFURED** type.
- Must be 64-bits (no parity or ECC = 72 / 80 bits).

Module Config.	**CHIPS / Side**	**SIDES** (1=single) (2=double)	**CHIPS Archit.**	**RAS Addr.**	**CAS Addr.**	**PAGE Lentgh** (4 banks)	**Refresh Rate** (uS)
64 MB	8	1	8Mx8b	12	9	8 KB	15.625
64 MB	4	1	8Mx16b	12	9	8 KB	15.625
128 MB	8	2	8Mx8b	12	9	8 KB	15.625
128 MB	4	2	8Mx16b	12	9	8 KB	15.625
128 MB	8	1	16Mx8b	12	10	**16 KB**	15.625
128 MB	4	1	16Mx16b	13	9	8 KB	7.8125
256 MB	8	2	16Mx8b	12	10	**16 KB**	15.625
256 MB	4	2	16Mx16b	13	9	8 KB	7.8125
256 MB	8	1	32Mx8b	13	10	**16 KB**	7.8125
512 MB	8	2	32Mx8b	13	10	**16 KB**	7.8125
512 MB	4	2	32Mx16b	13	10	**16 KB**	7.8125

NOT SUPPORTED SDRAM DIMMs
- **All DIMM with chips density < 64Mbits :**
  - **8MB, 16MB & 32MB** DIMMs.
  - **64MB** DIMMs **with 16 CHIPS and / or with chips on the 2 sides.**
  - All DIMMs with **2 logical banks chips** = obsolet (CT60 needs 4 logical banks chips).
- **REGISTERED** / **BUFFURED** DIMMs (generally for Work Stations & Servers, not PC).
- 512 MB DIMM with one physical bank (only 1 side populated).

REMARKS :
- **Don't confuse logical banks (2 or 4) with physical banks (1=Single Side or 2=Double Side) !**

PERFORMANCES :

The better system performances is obtained with 16 KB page lentgh DIMMs.

060 BURST with SDRAM

The CT60 bus clock = the 060 clock (060 in ‘Full Bus mode’).
The 060 uses **LINE BURST** to & from the system memory. SDRAM is well adapted for a such processor !
A BURST LINE is a length of **4 LONG-WORDs (16 Bytes)** that are transferred with only :

PAGE HIT (access to a logical SDRAM page already open) :
3,1,1,1 = 6 cycles for Burst Writes. Rate is 16 Bytes / 6 cycles = 178 MBytes/s (Each access in the same page).
5,1,1,1 = 8 cycles for Burst Reads. Rate is 16 Bytes / 8 cycles = 132 MBytes/s (Each access in the same page).

PAGE MISS (access to a new logical SDRAM page (must be precharged and open) :
7,1,1,1 = 10 cycles for Burst Writes. Rate is 16 Bytes / 10cycles = 107 MBytes/s (Each access in a new page).
9,1,1,1 = 12 cycles for Burst Reads. Rate is 16 Bytes / 12cycles = 89 MBytes/s (Each access in a new page).

The CT60 uses the 060 at the top of the possible performances with the mighty **COPYBACK** mode ! Instead of the WRITETROUGH mode like other TOS machines !

Copyback mode is active for all SDRAM memory area.
Copyback mode allows the 060 to write into the cache without writting into the SDRAM, what is so more performant ! The cache lines are pushed into SDRAM only when needed (060 needs place by example). With **two 8Kbytes caches**, it gives to coders the possibility to do some incredibly speedy routs residing at 100% into the caches !

The 060 uses BURST transfers with SDRAM in 99% of the cases. Here are the cases when the 060 don’t burst, this means, transfers Bytes, Words & Long-Words :

Byte, Word, and Long-Word READ Transfer Cycles from SDRAM

Accesses that are implicitly NONCACHABLE :

- Locked Read-Modify-Write accesses.
- Table Searches.

Accesses that are not allocate in the data cache on a read miss :

- Exception Vector Fetches.
- Exception stack Deallocation for an RTE Instruction.

Byte, Word, and Long-Word WRITE Transfer Cycles to SDRAM

Accesses that are implicitly NONCACHABLE :

- Locked Read-Modify-Write accesses.
- Table Searches.

Accesses that are not allocate in the data cache on a write miss :

- Exception stacking.

Cache Line pushes for lines containing a single dirty Long-word.

Write to WRITETHROUGH pages (ST-RAM !).

INTERRUPTS

CT60 adds a new interrupt for the 060 Bus Slot : **I6 (active low)**

I6 is the interrupt from the 060 BUS SLOT and is merged with the others from the Falcon. See table below for the priority position.

060 INTERRUPTS PRIORITY TABLE

NAME	LEVEL	ACTIVE	TYPE	SOURCE	PRIORITY
I6	**6**	**Low**	**Software**	**CTPCI**	**Highest**
I6	**6**	**Low**	**Software**	**SuperVidel**	
I6	**6**	**Low**	**Software**	**EtherNAT**	
INT6	6	Low	Software	F030 Bus Slot	
MFPINT	6	Low	Software	F030 MFP	
DSPREQ	6	Low	Software	F030 DSP	
INT5	5	Low	Software	F030 SCC	
VBL	4	Low	Auto	F030 VIDEL VSync	
INT3	3	High	Software	F030 Bus Slot	
HBL	2	Low	Auto	F030 VIDEL HSync	
INT1	1	High	Software	F030 Bus Slot	**Lowest**

INT1 & INT3 are NO MORE SUPPORTED with CT60 !
INT6 is also named MFPINT on atari documents because it is daisy chained with the MFP.

CTPCI INT

Note :
Because of a missing pull-up resistor on the CTPCI (very sorry for that !), the LINT interrupt is not used by software.
End of DMA transfers are checked by pulling method (060 checks the registers in the PLX)

Local Interrupt Output (LINT#)

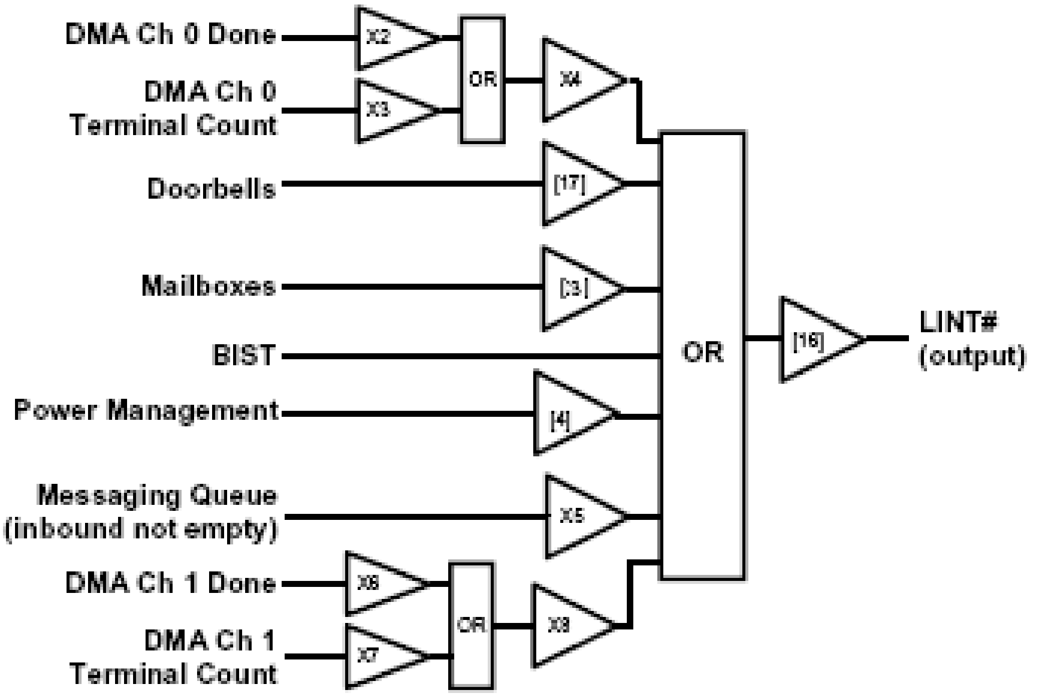
The PCI 9054 Local Interrupt (LINT#) output can be asserted by one of the following :

- PCI-to-Local Doorbell/Mailbox register access – **NOT possible on CTPCI**
- PCI BIST interrupt – **NOT possible on CTPCI**
- DMA Ch 0/Ch 1 Done interrupt
- DMA Ch 0/Ch 1 Terminal Count is reached
- DMA Abort Interrupt or Messaging Outbound Post Queue is not empty

The Local Interrupt Input Enable bit must be disabled (INTCSR[11]=0) when LINT# output is active.

LINT#, or individual sources of an interrupt, can be enabled or disabled with the PCI 9054 Interrupt Control/Status register (INTCSR). This register also provides interrupt status for each interrupt source.
The PCI 9054 Local interrupt is a level output. Interrupts can be cleared by disabling the Interrupt Enable bit of a source or by clearing the cause of an interrupt.

Note : PLX LINT# input mode is not supported on CTPCI.



X2 = Channel 0 Done Interrupt Enable bit (DMAMODE0[10]).
X3 = Channel 0 Interrupt after Terminal Count bit (DMADPR0[2]).
X4 = Local DMA Channel 0 Interrupt Enable bit (INTCSR[18]) and DMA Channel 0 Interrupt Select bit (DMAMODE0[17]).
X5 = Inbound Post Queue Interrupt Not Empty and Inbound Post Queue Interrupt Mask bits (QSR[5:4]).
X7 = Channel 1Interrupt after Terminal Count bit (DMADPR1[2]).
X6 = Channel 1 Done Interrupt Enable bit (DMAMODE1[10]).
X8 = Local DMA Channel 1 Interrupt Enable bit (INTCSR[19]) and DMA Channel 1 Interrupt Select bit (DMAMODE1[17]).

The numbers 3, 4, 16 & 17 represent bit numbers in the INTCSR register :

- Power management is not supported on CTPCI (bit #3)
- Bit #4 should be set to 1 if Mailboxes are used.
- **Default value of bit #16 = 1 (LINT# enable).**
- Bit #17 should be set to 1 if Doorbells are used.

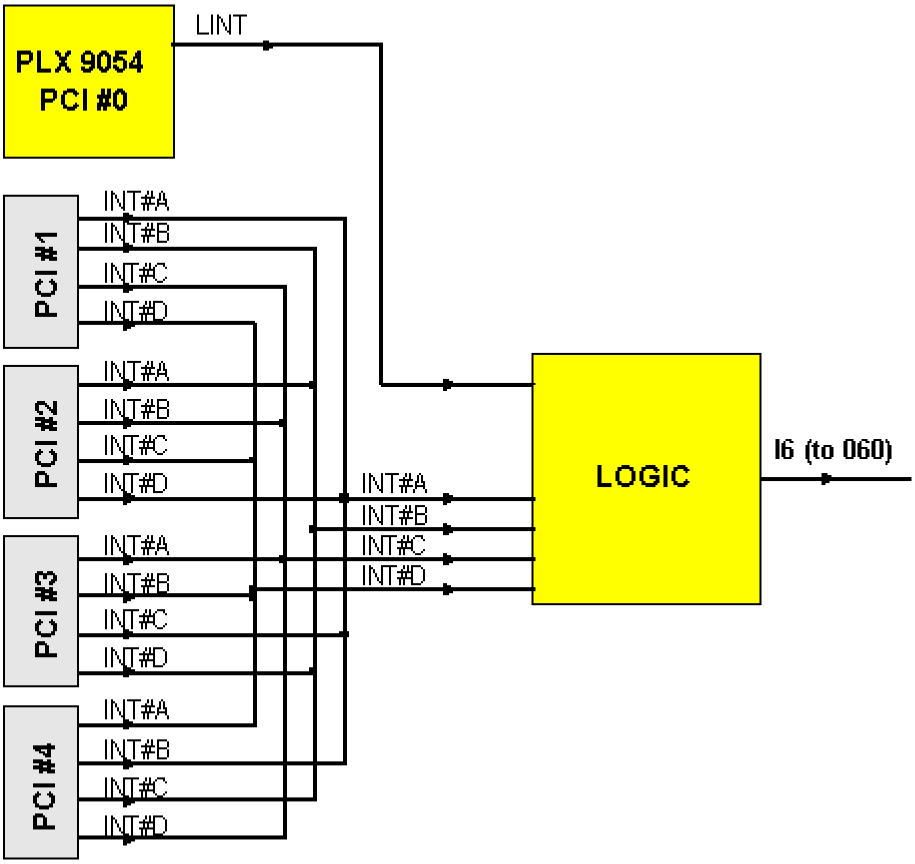
A DMA channel can assert a Local Bus interrupt when done (transfer complete) or after a transfer is complete for the current descriptor in Scatter/Gather DMA mode.
The Local processor can read the DMA Channel 0 Interrupt Active bits to determine whether a DMA Channel 0 (INTCSR[21]) or DMA Channel 1 (INTCSR[22]) interrupt is pending.

The Channel Done bit(s) (DMACSR0[4] and/or DMACSR1[4]) can be used to determine whether an interrupt is one of the following :

- DMA Done interrupt
- Transfer complete for current descriptor interrupt

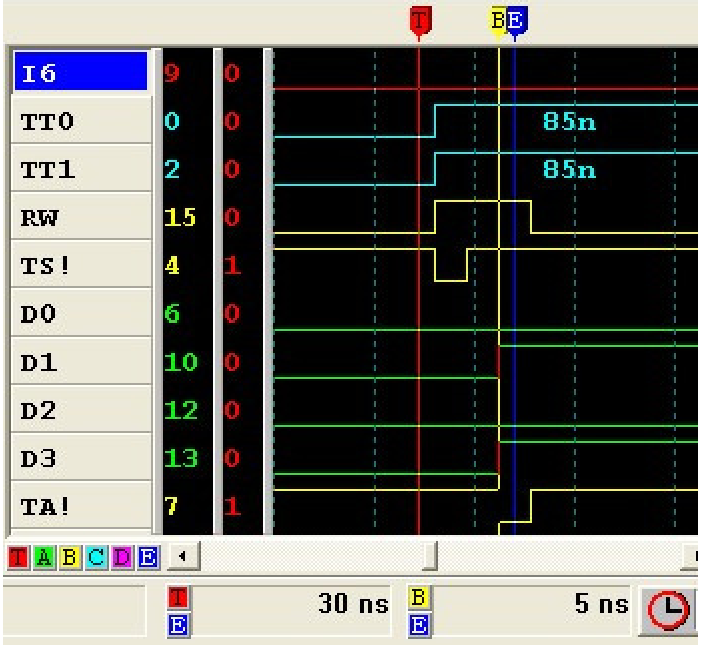
The Done Interrupt Enable bit(s) (DMAMODE0[10] and/or DMAMODE1[10]) enable a Done interrupt.
In Scatter/Gather DMA mode, a bit in the Next Descriptor Pointer register of the channel (loaded from Local memory) specifies whether to assert an interrupt at the end of the transfer for the current descriptor.
A DMA Channel interrupt is cleared by the Channel Clear Interrupt bit(s) (DMACSR0[3]=1 and/or DMACSR1[3]=1).

CTPCI INT



INT (level 6)	VECTOR
LINT	$x9
INT#A	$xA
INT#B	$xB
INT#C	$xC
INT#D	$xD

INT acknowledge cycle Timings



T-E = 3 cycles **B-E** = Vector Data set up time

CHIPSET ABE & SDR PIN-OUT

SDR-60

1 VCC
2 rstf
3 PGND
4 ta
5 PGND
6 PGND
7 PGND
8 VCC
9 PGND
10 a10
11 a11
12 a12
13 a13
14 a14
15 a15
16 a2
17 a3
18 GND
19 a4
20 a5
21 a6
22 a7
23 a8
24 a9
25 dm3
26 dm1
27 dm2
28 dm0
29 GND
30 BOOT_IDE output
31 cs3
32 clk
33 PGND
34 PGND
35 cs2
36 GND
37 VCC
38 PGND
39 ideled
40 d2
41 d1
42 VCC
43 d0
44 PGND
45 PGND
46 ma12
47 GND
48 ma11
49 ba1
50 ba0
51 ma10
52 ma9
53 ma8
54 ma7
55 VCC
56 ma6
57 ma5
58 ma4
59 ma3
60 ma2
61 ma1
62 GND
63 TDI
64 ma0
65 TMS
66 PGND
67 TCK
68 PGND
69 PGND
70 PGND
71 ras
72 GND

73 VCC
74 cs1
75 cs0
76 cas
77 we
78 a31
79 a30
80 a29
81 a28
82 a27
83 a26
84 VCC
85 a25
86 a24
87 a23
88 a22
89 GND
90 GND
91 a21
92 a20
93 a19
94 a18
95 a17
96 a16
97 tbi
98 rst60
99 GND
100 TEST SDR
101 TEST SDR
102 tci
103 PGND
104 PGND
105 PGND
106 IDE
107 PGND
108 GND
109 VCC
110 PGND
111 ipl2f
112 ipl1
113 i6
114 GND
115 ipl2
116 bs1
117 ts
118 tt1
119 PGND
120 PGND
121 ipl0
122 TDO
123 GND
124 siz1
125 rsto
126 eeda
127 VCC
128 thcs
129 rw
130 siz0
131 Reserved
132 PGND
133 PGND
134 eecl
135 bs0
136 bs2
137 bs3
138 thck
139 ipl0f
140 ipl1f
141 VCC
142 thdi
143 rst
144 GND

ABE-60

1 VCC
2 avec
3 a13
4 ct60
5 a14
6 a15
7 ta
8 VCC
9 bs0
10 flhoe
11 flhwe
12 dtkcmb
13 halt
14 a2
15 a3
16 tt0
17 tm0
18 GND
19 bg1
20 tm1
21 bg2
22 ts
23 bg0
24 tt1
25 exp/   - TEST ABE
26 exp2/ - TEST ABE
27 tm2
28 as
29 GND
30 BOOT_IDE input
31 bs1
32 clk
33 bs2
34 i6
35 dtk
36 GND
37 VCC
38 clkf
39 a2f
40 fc2
41 slp
42 VCC
43 bs3
44 a3f
45 bb
46 fc1
47 GND
48 a16
49 a17
50 a18
51 a19
52 a20
53 a21
54 a22
55 VCC
56 a23
57 a24
58 a25
59 a26
60 a27
61 a28
62 GND
63 TDI
64 a29
65 TMS
66 a30
67 TCK
68 a31
69 d31
70 d30
71 d29
72 GND

73 VCC
74 d28
75 d27
76 d26
77 d25
78 d24
79 d23
80 d22
81 d21
82 d20
83 d19
84 VCC
85 d18
86 d17
87 d16
88 d15
89 GND
90 GND
91 d14
92 d13
93 d12
94 d11
95 d10
96 d9
97 d8
98 d7
99 GND
100 d6
101 d5
102 d4
103 d3
104 d2
105 d1
106 br2
107 d0
108 GND
109 VCC
110 fd15
111 fd14
112 fd13
113 fd12
114 GND
115 fd11
116 fd10
117 fd9
118 fd8
119 fd7
120 fd6
121 fd5
122 TDO
123 GND
124 fd4
125 fd3
126 fd2
127 VCC
128 fd1
129 fd0
130 bg30
131 uds
132 lds
133 a1f
134 berr
135 br0
136 rw
137 fc0
138 br1
139 abdir
140 bgk
141 VCC
142 tea
143 rst
144 GND

PLX 9054 BRIDGE OVERVIEW

The PCI 9054, a 32-bit 33-MHz PCI Bus Master I/O Accelerator, is the most advanced general-purpose bus Master device available. It offers a robust PCI Specification v2.2 implementation enabling Burst transfers up to 132 MB/second. The PCI 9054 incorporates the industry leading PLX Data Pipe Architecture™ technology, including DMA engines, programmable PCI Initiator and Target Data-Transfer modes, and PCI messaging functions

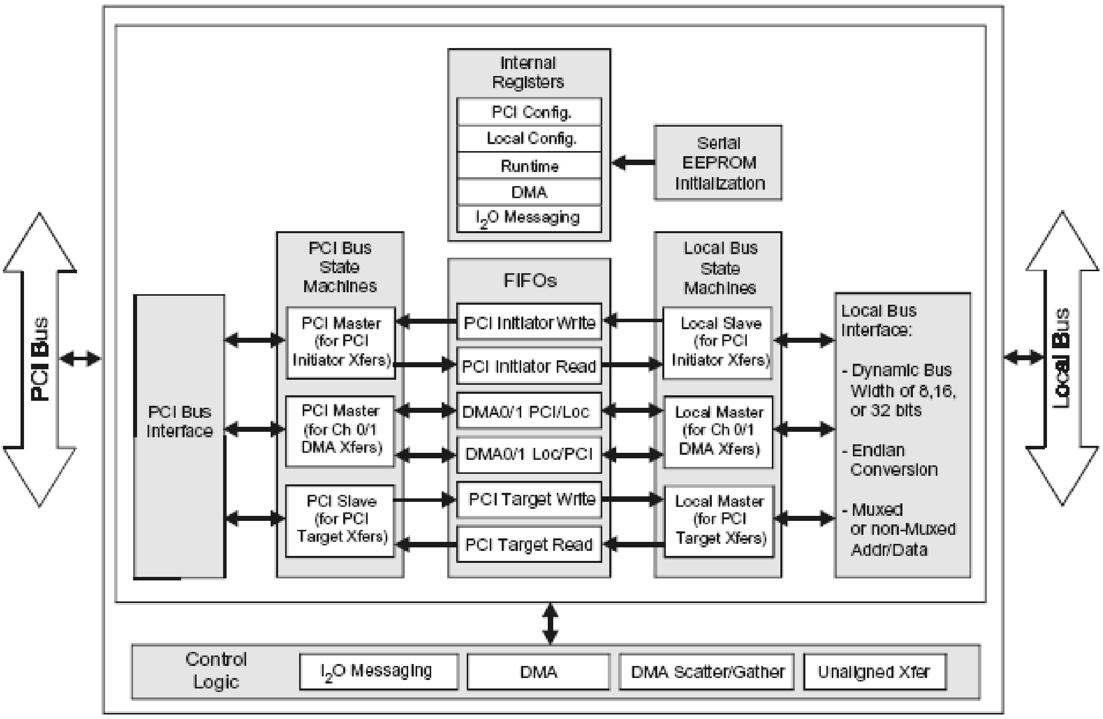


Figure 1-2. PCI 9054 Internal Block Diagram

.

All registers implemented in the PCI 9080 (Milan Bridge) are implemented in the PCI 9054. The PCI 9054 includes many new bit definitions and several new registers.

The Local processor programs the PCI 9054 registers, then sets the Local Init Status bit (LMISC[2]=done).
Note: *Some systems may hang if PCI Target reads and writes take too long (during initialization, the PCI Host also performs PCI Target accesses). The value of the PCI Target Retry Delay Clocks (LBRD0[31:28]) may resolve this.*

Features of bridge on the CTPCI

• PCI Specification version 2.2 (v2.2) compliant 32-bit, 33-MHz Bus Master Interface Controller with PCI Power Management features for adapters and embedded systems
• PCI v2.2 Vital Product Data (VPD) configuration support
• PCI ↔Local Data transfers up to 132 MB/s
• 3.3V, 5V tolerant PCI and Local signaling supports Universal PCI Adapter designs, 3.3V core, lowpower CMOS in 176-pin PQFP
• Industrial Temp Range operation
• Subsystem ID and Subsystem Vendor ID.

Contains Subsystem ID and Subsystem Vendor ID in the PCI Configuration Register Space in addition to System and Vendor IDs. The PCI 9054 also contains a permanent Vendor ID (10B5h) and Device ID (9054h).

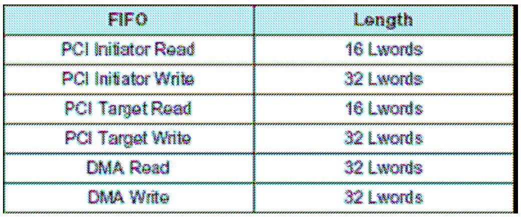
• PCI Dual Address Cycle (DAC) support
• PCI Hot Plug and CompactPCI Hot Swap compliant
• I2O™ v1.5-Ready Messaging Unit
• Two independent DMA channels for Local Bus memory to and from PCI Host Bus Data transfers

Can transfer data on any byte-boundary combination of the PCI and Local Address spaces.

• Programmable Burst Management
• Programmable Interrupt Generator

Can assert PCI and Local interrupts from external and internal sources.

• Six programmable FIFOs for zero wait state burst operation



FIFO	Length
PCI Initiator Read	16 Lwords
PCI Initiator Write	32 Lwords
PCI Target Read	16 Lwords
PCI Target Write	32 Lwords
DMA Read	32 Lwords
DMA Write	32 Lwords

- Programmable Local Bus runs up to 50 MHz and supports non-multiplexed 32-bit address/data and slave accesses of 8-, 16-, or 32-bit Local Bus devices.

The PCI 9054 communicates using four possible Data-Transfer modes:

- Configuration Register Access
- PCI Initiator Operation
- PCI Target Operation
- DMA Operation

- Local Bus runs asynchronously to the PCI Bus

The clock on CT bus is from 66 to 105 MHz and the Bridge local clock is half the CT bus clock → 33 to 52.5 MHz. The PCI clock is 33 MHz.

- Three PCI-to-Local Address spaces

The PCI 9054 supports three PCI-to-Local Address spaces when the PCI 9054 is in PCI Target or PCI Slave mode. These spaces (Space 0, Space 1, and Expansion ROM spaces) allow any PCI Bus Master to access the Local Memory spaces with programmable wait states, bus width, burst capabilities, and so forth.

On CTPCI-CT60/63 the following space are attributed :

Space #0 = SDRAM – 32-bits on CT with NO BURST.
Space #1 = ST-RAM – 32-bits on CT with NO BURST.
Space ROM = FLASH – 32-bits on CT with NO BURST.

- Eight 32-bit Mailbox

May be accessed from the PCI or Local Bus.

- Two 32-bit Doorbell registers

One asserts interrupts from the PCI Bus to the Local Bus. The other asserts interrupts from the Local Bus to the PCI Bus.

- Performs Big Endian ↔Little Endian conversion

Supports dynamic switching between Big Endian (Address Invariance) and Little Endian (Data Invariance) operations for PCI Target, PCI Initiator, DMA, and internal register accesses on the Local Bus.
The PCI 9054 supports on-the-fly Endian conversion for Space 0, Space 1, and Expansion ROM space.
The Local Bus is BIG ENDIAN programmed with the BIGEND# input pin.
Note: The PCI Bus is always Little Endian.

- Programmable prefetch counter

The PCI 9054 can be programmed to prefetch data during PCI Target and PCI Initiator prefetches (known or unknown size). To perform burst reads, prefetching must be enabled. The prefetch size can be programmed to match the Master burst length, or can be used as Read Ahead mode data. The PCI 9054 reads single data (8, 16, or 32 bit) if the Master initiates a single cycle; otherwise, the PCI 9054 prefetches the programmed size.

- PCI-to-Local Delayed Read mode

Supports Read Ahead mode, where prefetched data can be read from the PCI 9054 internal PCI Target Read FIFO instead of from the Local Bus. The address must be subsequent to the previous address and 32-bit aligned (next address = current address + 4). This feature allows for increased bandwidth and reduced data latency.

- Posted Memory Writes.

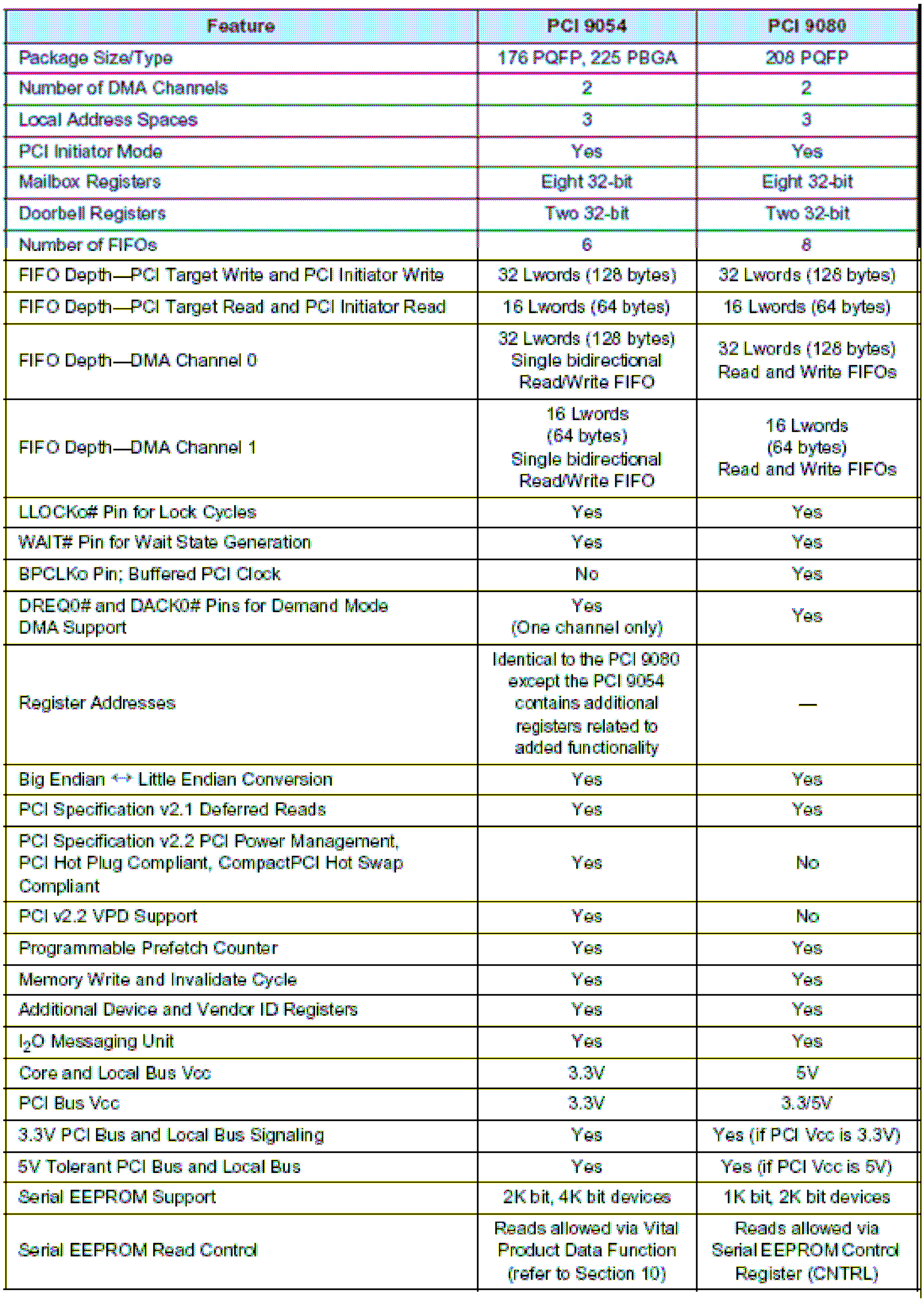
Supports the Posted Memory Writes (PMW) for maximum performance and to avoid potential deadlock situations.

- Keep Bus Mode

The PCI 9054 can be programmed to keep the PCI Bus by generating wait state(s) if the PCI Target Write FIFO becomes full.
The PCI 9054 can also be programmed to keep the Local Bus (LHOLD asserted) if the PCI Target Write FIFO becomes empty or the PCI Target Read FIFO becomes full.
The Local Bus is dropped in either case when the Local Bus Latency Timer is enabled and expires.

.

PLX 9054 and PLX 9080 (Milan) Comparison



Feature	PCI 9054	PCI 9080
Package Size/Type	176 PQFP, 225 PBGA	208 PQFP
Number of DMA Channels	2	2
Local Address Spaces	3	3
PCI Initiator Mode	Yes	Yes
Mailbox Registers	Eight 32-bit	Eight 32-bit
Doorbell Registers	Two 32-bit	Two 32-bit
Number of FIFOs	6	8
FIFO Depth—PCI Target Write and PCI Initiator Write	32 Lwords (128 bytes)	32 Lwords (128 bytes)
FIFO Depth—PCI Target Read and PCI Initiator Read	16 Lwords (64 bytes)	16 Lwords (64 bytes)
FIFO Depth—DMA Channel 0	32 Lwords (128 bytes) Single bidirectional Read/Write FIFO	32 Lwords (128 bytes) Read and Write FIFOs
FIFO Depth—DMA Channel 1	16 Lwords (64 bytes) Single bidirectional Read/Write FIFO	16 Lwords (64 bytes) Read and Write FIFOs
LLOCKo# Pin for Lock Cycles	Yes	Yes
WAIT# Pin for Wait State Generation	Yes	Yes
BPCLKo Pin; Buffered PCI Clock	No	Yes
DREQ0# and DACK0# Pins for Demand Mode DMA Support	Yes (One channel only)	Yes
Register Addresses	Identical to the PCI 9080 except the PCI 9054 contains additional registers related to added functionality	—
Big Endian ↔ Little Endian Conversion	Yes	Yes
PCI Specification v2.1 Deferred Reads	Yes	Yes
PCI Specification v2.2 PCI Power Management, PCI Hot Plug Compliant, CompactPCI Hot Swap Compliant	Yes	No
PCI v2.2 VPD Support	Yes	No
Programmable Prefetch Counter	Yes	Yes
Memory Write and Invalidate Cycle	Yes	Yes
Additional Device and Vendor ID Registers	Yes	Yes
I_2O Messaging Unit	Yes	Yes
Core and Local Bus Vcc	3.3V	5V
PCI Bus Vcc	3.3V	3.3/5V
3.3V PCI Bus and Local Bus Signaling	Yes	Yes (if PCI Vcc is 3.3V)
5V Tolerant PCI Bus and Local Bus	Yes	Yes (if PCI Vcc is 5V)
Serial EEPROM Support	2K bit, 4K bit devices	1K bit, 2K bit devices
Serial EEPROM Read Control	Reads allowed via Vital Product Data Function (refer to Section 10)	Reads allowed via Serial EEPROM Control Register (CNTRL)

PLX LOCAL PINS

LOCAL BUS pins (M mode) used on CTPCI

CCS#	Configuration Register Select	IN	Internal PCI 9054 registers are selected when CCS# is asserted low.
LCLK	Local Clock	IN	Local clock input.
LINT#	Local Interrupt	BI	As an input to the PCI 9054, when asserted low, causes a PCI interrupt. Not used on CTPCI. As an output, a synchronous level output that remains asserted as long as an interrupt condition exists. If edge level interrupt is required, disabling and then enabling Local interrupts through INTCSR creates an edge if an interrupt condition still exists or a new interrupt condition occurs.
BB#	Bus Busy	BI	As an input, monitors this signal to determine whether the external Master has ended a Bus cycle. As an output, asserts this signal after an external arbiter has granted ownership of the Local Bus and BB# is inactive from another Master.
BDIP#	Burst Data in progress	BI	As an input, driven by the Bus Master during a Burst transaction. The Master de-asserts before the last Data phase on the bus. As an output, driven by the PCI 9054 during the Data phase of a Burst transaction. The PCI 9054 de-asserts before the last Burst Data phase on the bus.
BG#	Bus Grant	I	Asserted by the Local Bus arbiter in response to BR#. Indicates the requesting Master is next.
BI#	Burst Inhibit	I	Whenever BI# is asserted, indicates that the Target device does not support Burst transactions.
BR#	Bus Request	OUT	Asserted by the Master to request use of the Local Bus. The Local Bus arbiter asserts BG# when the Master is next in line for bus ownership.
BURST#	Burst	BI	As an input, driven by the Master along with address and data indicating a Burst transfer is in progress. As an output, driven by the PCI 9054 along with address and data indicating a Burst transfer is in progress.
LA[0:31]	Address Bus	BI	Carries the 32 bits of the physical Address Bus. LA0 is most significant bit of bus address.
LD[0:31]	Data Bus	BI	All Master accesses to the PCI 9054 are 32 bits only. LD0 is most significant bit of bus address.
RD/WR#	Read/Write	BI	Asserted high for reads and low for writes.
RETRY#	Retry	OUT	Driven by the PCI 9054 when it is a Slave to indicate a Local Master must back off and restart the cycle. In Deferred Read mode, indicates a Local Master should return for the requested data.
TA#	Transfer Acknowledge	BI	As an input, when a Local Bus access is made to the PCI 9054, indicates a Write Data transfer can complete or that Read data on the bus is valid. As an output, when the PCI 9054 is a Bus Master, indicates a Write Data transfer is complete or that Read data on the bus is valid.
TEA#	Transfer Error Ack	BI	Driven by the Target device, indicating an error condition occurred during a Bus cycle.
TS#	Address Strobe	BI	Indicates the valid address and start of a new Bus access. Asserted for the first clock of a Bus access.
TSIZ[0:1]	Transfer Size	BI	Driven by the current Master along with the address, indicating the data-transfer size. TSIZ0 is most significant bit of bus address. Refer to Section 3.4.3.5 for further information.
MDREQ#/ DMPAF/ EOT#	IDMA Data Transfer Request PCI Initiator Programmable Almost Full End of Transfer for Current DMA Channel	OUT **OUT** IN	Multiplexed input or output pin : **MDREQ#:** IDMA M mode Data transfer request start. Always asserted, indicating Data transfer should start. De-asserted only when the PCI Initiator FIFO becomes full. Programmable through a Configuration register. **DMPAF**: PCI Initiator Write FIFO Almost Full status output. Programmable through a Configuration register. Connected to CPLD as INT source. **EOT#:** Terminates the current DMA transfer. *Note**:** EOT# serves as a general purpose EOT. Before asserting EOT#, user should be aware of DMA channel activity.*

LOCAL BUS pins (M mode) NOT used on CTPCI

BIGEND# WAIT#	Big Endian Select WAIT Input/Output Select	**IN** BI	Multiplexed input or output pin : Can be asserted during the Local Bus Address phase of a PCI Initiator transfer or Configuration register access to specify use of Big Endian Byte ordering. Big Endian Byte order for PCI Initiator transfers or Configuration register accesses is also programmable through the Configuration registers. PCI 9054 issues WAIT# when it is a Master on the Local Bus and has internal wait states setup. As a Slave, the PCI 9054 accepts WAIT# as an input from the Bus Master.
DP[0:3]	Data Parity	BI	Parity is even for each of up to four byte lanes on the Local Bus. Parity is checked for writes or reads to the PCI 9054. Parity is asserted for reads from or writes by the PCI 9054. DP0 is the most significant bit of the Bus address.
LRESETo#	Local Bus Reset Ou	OUT	Asserted when the PCI 9054 chip is reset. Can be used to drive RESET# input of a Local processor.

SERIAL EEPROM pins - NOT USED ON CTPCI

EECS	Serial EEPROM Chip Select	IN	Serial EEPROM Chip Select.
EEDI/ EEDO	Serial EEPROM Data IN/ Serial EEPROM Data OUT	BI	Multiplexed Write/Read data to a serial EEPROM pin. These pins have internal pull-ups.
EESK	Serial Data Clock	IN	Serial EEPROM clock pin.

HOT SWAP pins - NOT USED ON CTPCI

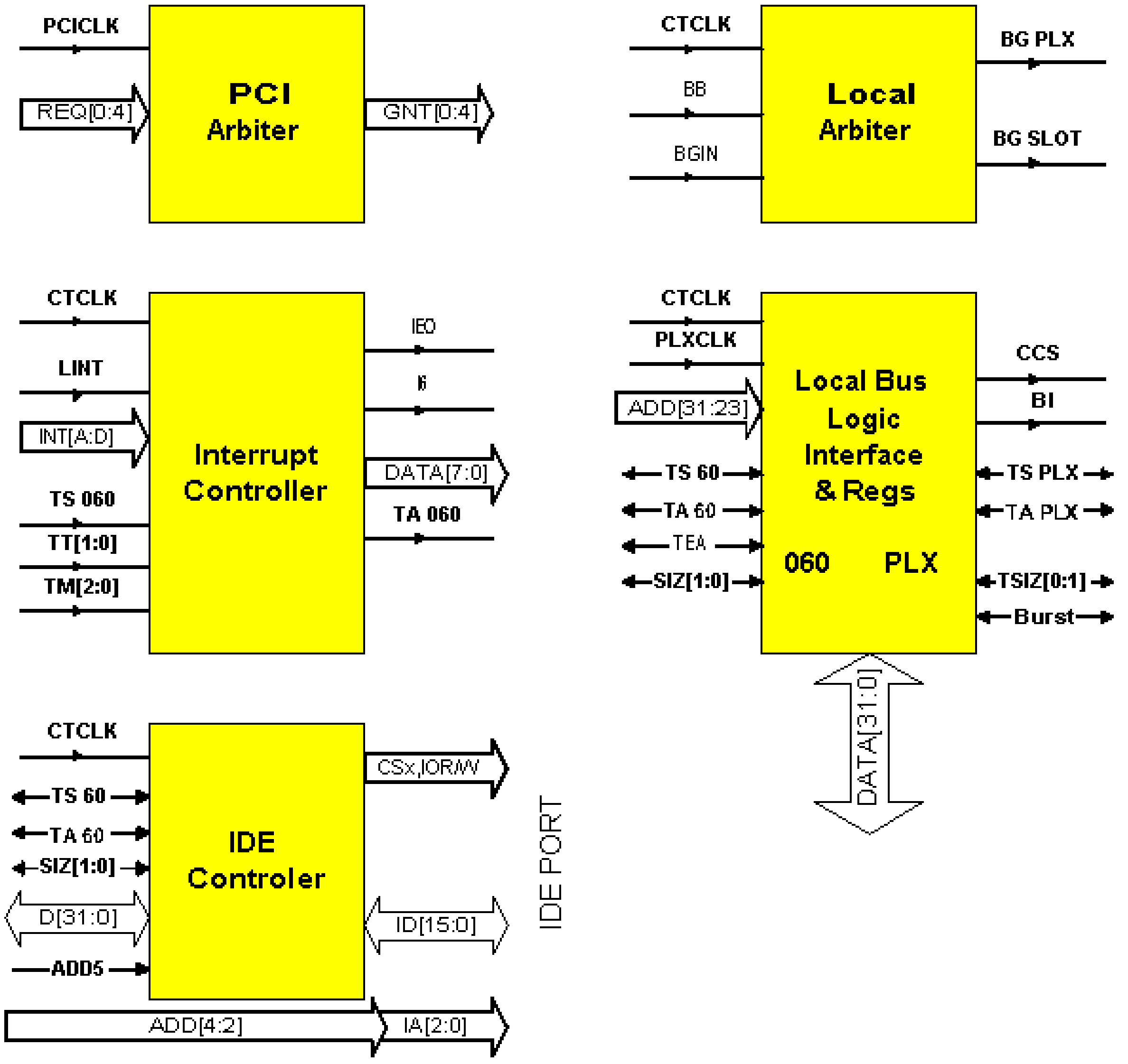
ENUM#	Enumeration	OUT	Interrupt output asserted when an adapter using PCI 9054 has been freshly inserted or is ready to be removed from a PCI slot.
LEDon/ LEDin	LEDon/LEDin	OUT/ IN	As an output, acts as the Hot Swap board indicator LED. As an input, monitors the CompactPCI board latch status.

CONFIGURATION pins

MODE[1:0]	Bus Mode	IN	Selects the PCI 9054 bus operation mode: Mode [1:0] = [1,1] → MODE M for CTPCI These pins have internal pull-ups.
TEST	Test Pin	IN	Pulled high for test and low for normal operation. When pulled high: All outputs except USERo/DREQ0#/LLOCKo# and LEDon/LEDin are placed in tri-state. USERo/DREQ0#/LLOCKo# provides NANDTREE output. The TEST pin has an internal pull-down.

CTPCI

CPLD



LOCAL BUS OPERATION (M mode (MPC860))

Burst-4 Lword Mode

If the Burst Mode bit is enabled and the Bterm Mode bit is disabled, bursting can start only on a 16-byte boundary and continue up to the next 16-byte address boundary.
After data before the boundary is transferred, the PCI 9054 asserts a new Address cycle (TS#).

Use of BURST cycle

A 68060 single cycle causes a single-cycle PCI transaction.
A 68060 Burst cycle asserts a Burst PCI Cycle transaction.
Bursts are limited to 16 bytes (four Lwords) on local 68060 bus.

DIRECT DATA TRANSFER MODES

The PCI 9054 supports three direct transfer modes:

- **PCI Initiator =** Local CPU (060) accesses PCI memory or I/O.
- **PCI Target =** PCI Master accesses Local memory (SDRAM) or I/O (ST-RAM).
- **DMA =** PCI 9054 DMA controller reads/writes PCI memory to/from Local memory (SDRAM & ST-RAM).

Local Bus Arbitration

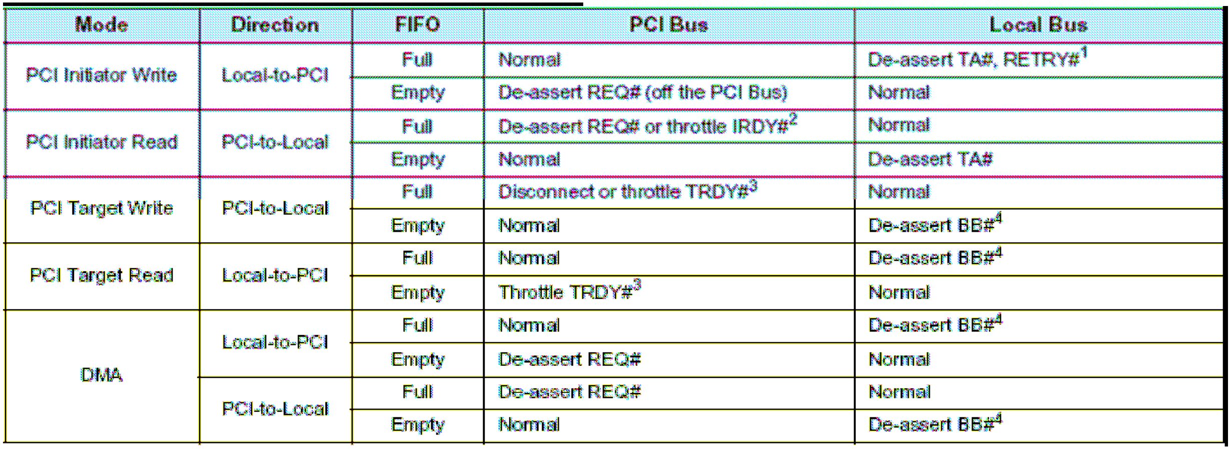
The PCI 9054 asserts BR# to request the Local Bus. It owns the Local Bus when BG# is asserted. Upon receiving BG#, the PCI 9054 waits for BB# to de-assert. The PCI 9054 then asserts BB# at the next rising edge of the Local clock after acknowledging BB# is de-asserted (no other device is acting as the Local Bus Master). The PCI 9054 continues to assert BB# while acting as the Local Bus Master (*that is*, it holds the bus until instructed to release BB#) **when the Local Bus Latency Timer is enabled and expires (MARBR[7:0])** or the transaction is complete.

Note: *The Local Bus Pause Timer applies only to DMA operation. It does* ***not*** *apply to PCI Target operation.*

Wait States—Local Bus

In PCI Initiator mode, when accessing the PCI 9054 registers, the PCI 9054 acts as a Local Bus Slave. The PCI 9054 asserts external wait states with the TA# signal.
In PCI Target and DMA modes, the PCI 9054 acts as a Local Bus Master. The Internal Wait States bit(s) (LBRD0[21:18, 5:2], (LBRD1[5:2]), DMAMODE0[5:2], and/or DMAMODE1[5:2]) can be used to program the number of internal wait states between the first address-to-data (and subsequent data-to-data in Burst mode).
In PCI Target and DMA modes, if TA# is enabled and active, it continues the Data transfer, regardless of the wait state counter.

RESPONSE TO FIFO FULL OR EMPTY



Mode	Direction	FIFO	PCI Bus	Local Bus
PCI Initiator Write	Local-to-PCI	Full	Normal	De-assert TA#, RETRY#[1]
		Empty	De-assert REQ# (off the PCI Bus)	Normal
PCI Initiator Read	PCI-to-Local	Full	De-assert REQ# or throttle IRDY#[2]	Normal
		Empty	Normal	De-assert TA#
PCI Target Write	PCI-to-Local	Full	Disconnect or throttle TRDY#[3]	Normal
		Empty	Normal	De-assert BB#[4]
PCI Target Read	Local-to-PCI	Full	Normal	De-assert BB#[4]
		Empty	Throttle TRDY#[3]	Normal
DMA	Local-to-PCI	Full	Normal	De-assert BB#[4]
		Empty	De-assert REQ#	Normal
	PCI-to-Local	Full	De-assert REQ#	Normal
		Empty	Normal	De-assert BB#[4]

1. Issue RETRY# depends upon the PCI Initiator Write FIFO Almost Full RETRY# Output Enable bit (LMISC[6]).
2. Throttle IRDY# depends upon the PCI Initiator PCI Read Mode bit (DMPBAM[4]).
3. Throttle TRDY# depends upon the PCI Target Write Mode bit (LBRD0[27]).
4. BB# de-assert depends upon the Local Bus PCI Target Release Bus Mode bit (MARBR[21]).

DMA OPERATIONS

The PCI 9054 supports two independent DMA channels capable of transferring data from the:

- Local-to-PCI Bus
- PCI-to-Local Bus

Each channel consists of a DMA controller and a dedicated, bidirectional FIFO. Both channels support Block transfers, and Scatter/Gather transfers, with or without End of Transfer (EOT#). Only DMA Channel 0 supports Demand mode

DMA transfers. Master mode must be enabled with the Master Enable bit (PCICR[2]) before the PCI 9054 can become a PCI Bus Master. In addition, both DMA channels can be programmed to:

- Operate in 8-, 16-, or 32-bit Local Bus width
- Use zero to 15 internal wait states (Local Bus)
- Enable/disable internal wait states (Local Bus)
- **Enable/disable Local Bus Burst capability**
- Limit Local Bus bursts to four (BTERM# enable/disable)
- Hold Local address constant (Local Target is FIFO) or increment
- Perform PCI Memory Write and Invalidate (command code = Fh) or normal PCI Memory Write (command code = 7h)
- Pause Local transfer with/without BLAST# (DMA Fast/Slow termination)
- Assert PCI interrupt (INTA#) or **Local interrupt (LINT#)** when DMA transfer is complete or Terminal Count is reached during Scatter/Gather DMA mode transfers
- Operate in DMA Clear Count mode (only if the descriptor is in Local memory)

The PCI 9054 also supports PCI Dual Address with the upper 32-bit registers (DMADAC0 and DMADAC1).
The Local Bus Latency Timer determines the number of Local clocks the PCI 9054 can burst data before relinquishing the Local Bus. The Local Pause Timer sets how soon the DMA channel can request the Local Bus.

RETRY CAPABILITY

The RETRY is not supported on CTPCI because the 68060 has a poor mechanism retry.

The 060 is not a pipelined command CPU and is only able to retry the previous cycle. It cannot go back to get the result of a posted command. This has no interest compared to the TA hold method.

Please keep the 2 following bits = 0

LMISC[6]=0 (default)
LMISC[4]=0 (default)

PCI Initiator Write FIFO Full

If FIFO Almost Full Retry is disabled (LMISC1[6]=0), the PCI 9056 de-asserts TA# until there is space in the FIFO for additional Write data.

PCI Initiator Delayed Read

It uses the RETRY mode that is not supported by the CTPCI.
When Deferred PCI Initiator Read mode is disabled, the Local Master must “keep” the Local Bus and wait for the requested data (TA# is not asserted until data is available to the Local Bus).

TEA : Transfer Error Acknowledge

TEA# output

Not supported on CTPCI

TEA# input

The PCI 9054 tolerates TEA# input assertion only during PCI Target or DMA transactions (PCI to CT60 transactions).
Only the CT60 watchdog can deliver a TEA# if the hardware on CT60 did not answered after 64us.

The PCI 9054 does not sample TEA# assertion during PCI Initiator transactions.

TEA#, Transfer Error Acknowledge, is a wired-OR signal that is asserted by a Slave device on the Local Bus. The CT60 can assert TEA# as a Master or Slave if its Bus Monitor times out. If the Bus Monitor does time out and the CT60 asserts TEA#, the device it is communicating with needs to detect this regardless of its configuration (Master or Slave) and get off the Local bus in one clock cycle. Additionally the device should terminate any active PCI bus activity via an abort and set its status bits/registers appropriately.

If TEA# is asserted by the CT60 while the PCI 9054AC is the local bus master, TEA# will preempt TA# input and terminate the current cycle. If the burst isn’t completed, the PCI 9054AC will generate a new TS# and continue. This applies for Direct Slave and DMA transfers. This case is possible if the PCI9054 accesses on CT60 at an address (bad) where there is no hardware answering (→ BUS ERROR).

IDE PORT (Secondary)

Pin out

Pin	Name	Description	Info
1	**/RESET**	**Reset**	From CT60/63 Reset signal
2	**GND**	**Ground**	
3	**DD7**	**Data 7**	From/to CPLD through 56 ohms resistor
4	**DD8**	**Data 8**	From/to CPLD through 56 ohms resistor
5	**DD6**	**Data 6**	From/to CPLD through 56 ohms resistor
6	**DD9**	**Data 9**	From/to CPLD through 56 ohms resistor
7	**DD5**	**Data 5**	From/to CPLD through 56 ohms resistor
8	**DD10**	**Data 10**	From/to CPLD through 56 ohms resistor
9	**DD4**	**Data 4**	From/to CPLD through 56 ohms resistor
10	**DD11**	**Data 11**	From/to CPLD through 56 ohms resistor
11	**DD3**	**Data 3**	From/to CPLD through 56 ohms resistor
12	**DD12**	**Data 12**	From/to CPLD through 56 ohms resistor
13	**DD2**	**Data 2**	From/to CPLD through 56 ohms resistor
14	**DD13**	**Data 13**	From/to CPLD through 56 ohms resistor
15	**DD1**	**Data 1**	From/to CPLD through 56 ohms resistor
16	**DD14**	**Data 14**	From/to CPLD through 56 ohms resistor
17	**DD0**	**Data 0**	From/to CPLD through 56 ohms resistor
18	**DD15**	**Data 15**	From/to CPLD through 56 ohms resistor
19	**GND**	**Ground**	
20	KEY	Key	
21	n/c	Not connected	
22	**GND**	**Ground**	
23	**/IOW**	**Write Strobe**	From CPLD through 56 ohms resistor
24	**GND**	**Ground**	
25	**/IOR**	**Read Strobe**	From CPLD through 56 ohms resistor
26	**GND**	**Ground**	
27	IO_CH_RDY	Not used	To CPLD
28	n/c	Not connected	
29	n/c	Not connected	
30	**GND**	**Ground**	
31	**IRQ**	**Interrupt Request**	**To Falcon IDE IRQ signal (wire to connect)**
32	n/c	Not connected	
33	**DA1**	**Address 1**	From CT60/63 ADD#3 bus through 56 ohms resistor
34	n/c	Not connected	
35	**DA0**	**Address 0**	From CT60/63 ADD#2 bus through 56 ohms resistor
36	**DA2**	**Address 2**	From CT60/63 ADD#4 bus through 56 ohms resistor
37	**/CS0**	**(1F0-1F7)**	From CPLD through 56 ohms resistor
38	**/CS1**	**(3F6-3F7)**	From CPLD through 56 ohms resistor
39	IDE_ACT	Not used	
40	**GND**	**Ground**	

IRQ from HDD

IRQ pin on CTPCI must be connected to IRQ line on the F030 (pin#1 of U25) or pin#31 of the Falcon IDE port.

It seems this is the only moment where the IRQ is used :

IDENTIFY DEVICE command

The IDENTIFY DEVICE command enables the host to receive parameter information from the device.
Some devices may have to read the media in order to complete this command.
When the command is issued, the device sets the BSY bit to one, prepares to transfer the 256 words of device identification data to the host, sets the DRQ bit to one, clears the BSY bit to zero, and asserts INTRQ if nIEN is cleared to zero.

PIO modes

IDE port timing used by the CPLD are compatible with PIO MODE 4 (Falcon IDE port is a Mode 3 compatible).
A register can be read in the IDE devices.
PIO mode 4 is old and it seems that all HDD produced after 1998 are accepting this mode.
I did my tests with a Fujistsu 4GB MPC3043AT HDD from year 1999.

Word 64: PIO transfer modes supported

Bits (7:0) of word 64 of the IDENTIFY DEVICE data is defined as the PIO data and register transfer supported field. If this field is supported, bit 1 of word 53 shall be set to one. This field is bit significant. Any number of bits may be set to one in this field by the device to indicate the PIO modes the device is capable of supporting.

Of these bits, bits (7:2) are Reserved for future PIO modes. Bit 0, if set to one, indicates that the device supports PIO mode 3. All devices except CFA and PCMCIA devices shall support PIO mode 3 and shall set bit 0 to one. Bit 1, if set to one, indicates that the device supports PIO mode 4. If the serial interface is implemented, bits (1:0) shall be set to one.

Performances

The original Falcon Ide port has poor performances (PIO mode3.).
The new IDE port on CTPCI is using the last PIO mode 4.

Table of hardware performances

IDE port	WORD transfer	LONG transfer
F030 port **from 68030**	**4 cycles** 16 MHz → **250ns** 25 MHz → **160ns**	**8 cycles** 16 MHz → **500ns** 25 MHz → **320ns**
F030 port **from CT60-100**	***Typical (80%) = 5 cycles** 16 MHz → **312ns** 25 MHz → **200ns** *Best (20%) = 4 cycles 16 MHz → **250ns** 25 MHz → **160ns**	***Typical (80%) = 9 cycles** 16 MHz → **562ns** 25 MHz → **360ns** *Best (20%) = 8 cycles 16 MHz → **500ns** 25 MHz → **320ns**
CTPCI port	**11 cycles** 66MHz → 166ns 100MHZ → **110ns**	**20 cycles** 66MHz → 303ns 100MHZ → **200ns**

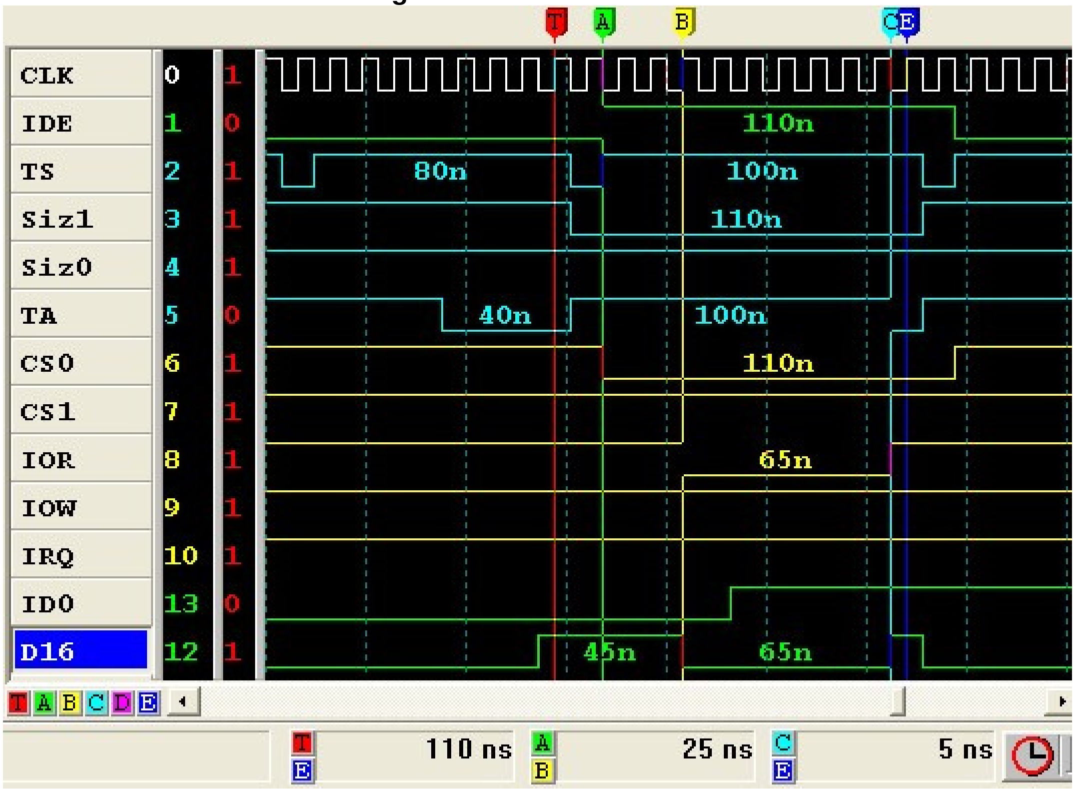
* The F030 & CT60 clocks are independent and there are variable phase alignments.

CONCLUSION :

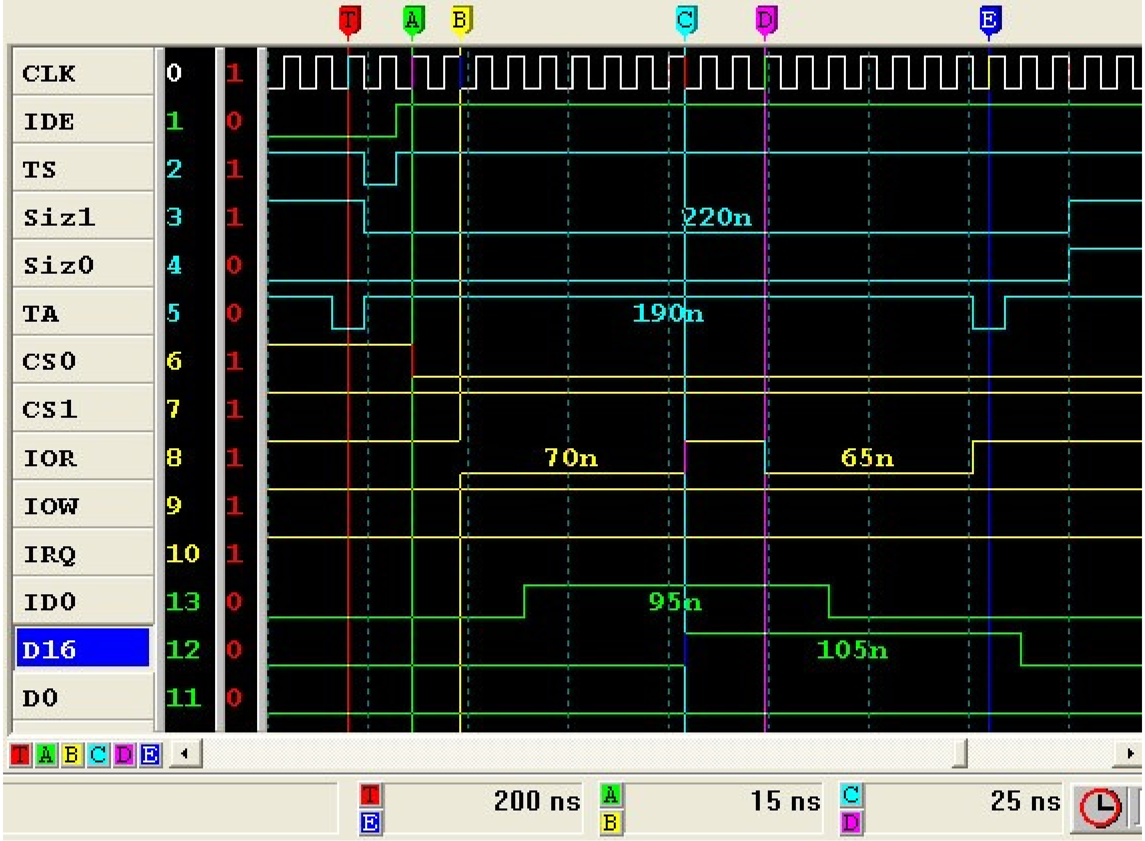
The new IDE port is 2.5 times faster that the Falcon 030 port (030 mode & CT60 mode) in the pure hardware point of view.

But this new bandwidth is not used (or not usable) because HDD TEST in KRONOS 2.01 gives only 16% of speed increase (tested with HDDriver 7.80).

IDE Read WORD Access Timings



IDE Read LONG Access Timings



T-E = IDE access cycle.
A-B = addresses (ADD bus & CSx signals) to IOR/ or IOW/ time.
C-D = Recovery time between 2 IOR/ or IOW/ pulse.

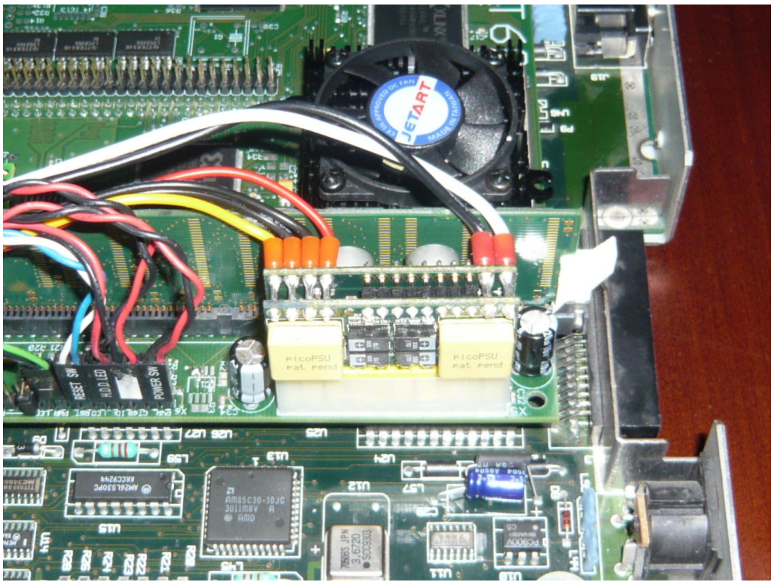
CT60/63 power consumptions

Configuration : - Falcon mb with 14MB ST-RAM + atari keyboard
- CT63 with 128 MB SDRAM with 16 chips.
- Boot 16colors - Desk 16colors - TOS system.
Not included (+5V) : Floppy, HDD, CD-ROM

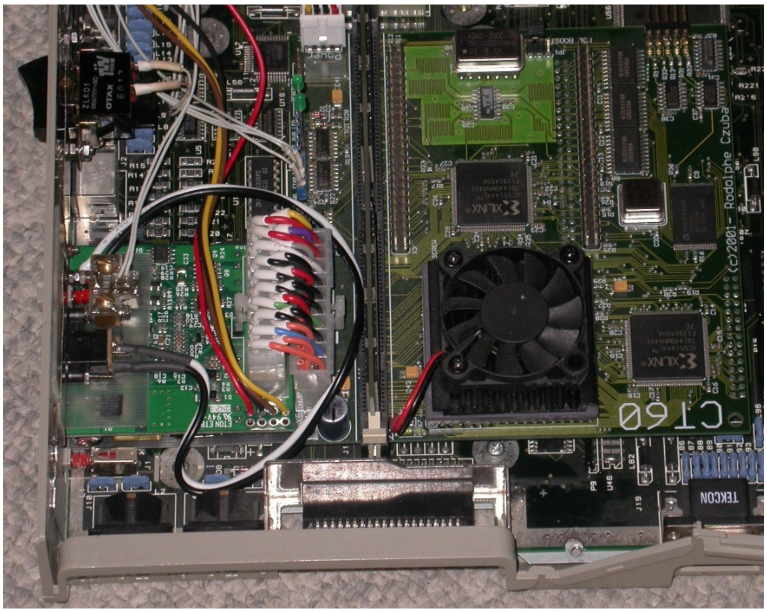
+3.3V	**66MHz**	**95Mhz**
Boot	1.26A	1.56A
Desk	1.13A	1.39A
Kronos 1.71 CPU test	1.31A	**1.60A**
Anyplayer 2.21 .MOV	1.23A	1.51A

+5V	**66MHz**	**95Mhz**
Boot	1.90A	**1.92A**
Desk	1.89A	1.85A
Kronos 1.71 CPU test	1.77A	1.82A
Anyplayer 2.21 .MOV	1.85A	1.86A

PicoPSU on CT60



Micro PSU board near CT60 in the Falcon case



3. Delayed Read Mode bit (MARBR[24])

Changes to PCI 9054 Data Book revision 2.1: The name for the Mode/DMA Arbitration register bit 24 (MARBR[24]) is changed from “Delayed Read Mode” to “PCI r2.1 Features Enable”. PCI 9054 Data Book sections 3.4.3.2 for M mode, and the MARBR[24] register bit description, are revised as follows:

Table 4-35. (MARBR; PCI:08h or ACh, LOC: 88h or 12Ch)
Mode/Arbitration Register

Bit	Description	Read	Write	Value after Reset
24	**PCI r2.1 Features Enable.** When set to 1, the PCI 9054 performs all PCI Read and Write transactions in compliance with PCI r2.1. Setting this bit enables Delayed Reads, 2^{15} PCI Clock timeout on Retries, 16- and 8-clock PCI latency rules, and enables the option to select PCI Read No Write Mode (Retries for writes) (bit [25]). Refer to Sections 3.4.3.2 and 5.4.2.2 for additional information. Value of 0 causes TRDY# to remain de-asserted on reads until Read data is available. If Read data is not available before the PCI Target Retry Delay Clocks counter (LBRD0[31:28]) expires, a PCI Retry is issued.	Yes	Yes	0

3.4.3.2 PCI r2.1 Features Enable

The PCI 9054 can be programmed through the PCI r2.1 Features Enable bit (MARBR[24]) to perform all PCI Read/Write transactions in compliance to PCI r2.1 (and PCI r2.2). The following PCI 9054 behavior occurs when MARBR[24] = 1.

3.4.3.2.1 Direct Slave Delayed Read Mode

PCI Bus single cycle aligned or unaligned 32-bit Direct Slave Read transactions always result in a 1-Lword single cycle transfer on the Local Bus, with corresponding Local Address and TSIZ[0:1] asserted to reflect the PCI Byte Enables (C/BE[3:0]#), unless the PCI Read No Flush Mode bit is enabled (MARBR[28] = 1) (refer to Section 3.4.3.3). This causes the PCI 9054 to Retry all PCI Bus Read requests that follow, until the original PCI Address and Byte Enables (C/BE[3:0]#) are matched.

3.4.3.2.2 2_{15} PCI Clock Timeout

If a PCI Master does not complete its originally requested Direct Slave Delayed Read transfer, the PCI 9054 flushes the Direct Slave Read FIFO after 2_{15} PCI clocks and will grant an access to a new Direct Slave Read access. The PCI 9054 Retries all other Direct Slave Read accesses that occur before the 2_{15} PCI clock timeout.

3.4.3.2.3 PCI r2.1 16- and 8- clock rule

The PCI 9054 guarantees that if the first Direct Slave Write data cannot be accepted by the PCI 9054 and/or the first Direct Slave Read data cannot be returned by the PCI 9054 within 16 PCI clocks from the beginning of the Direct Slave cycle (FRAME# asserted), the PCI 9054 issues a Retry (STOP# asserted) to the PCI Bus.

During successful Direct Slave Read and/or Direct Slave Write accesses, the subsequent data after the first access is accepted for writes or returned for reads in 8 PCI clocks (TRDY# asserted). Otherwise, the PCI 9054 issues a PCI disconnect (STOP# asserted) to the PCI Master.

In addition, setting the PCI r2.1 Features Enable bit (MARBR[24] = 1) allows optional enabling of the following PCI r2.1 function:

- No write while delayed read is pending (PCI Retries for writes) (MARBR[25])

The following PCI 2.1 optional function can be activated except if MARBR[25,24] = 11b:

- Write and flush pending delayed read (MARBR[26])

9. Retried Direct Slave Single Read completed as Burst Read

Design Issue: If a Direct Slave Single Read request that has been retried by the PCI 9054 is completed as a Burst Read transaction, the PCI 9054 will stall the PCI bus with continuous retries on the second read data. If the PCI r2.1 Features Enable bit is set (MARBR[24] = 1), the PCI 9054 will issue a Target Abort following expiration of 32K PCI clock timeout. If this bit is clear (MARBR[24] = 0), the 32K PCI clock discard timer is not enabled.

The PCI Specification does not allow a PCI master to extend a read transaction beyond its original intended length after it has been terminated with Retry. Accordingly, the scenario described above can only occur if a second PCI master requests the same transaction (address, command, byte enables and parity) that is being retried for the original PCI master request. The PCI 9054 cannot and need not distinguish between the two and will simply attempt to complete the access.

Solutions/Workarounds: To avoid stalling the PCI bus under this scenario:

1. Set the PCI r2.1 Features Enable bit (MARBR[24] = 1) to enable 32K PCI clock timeout for retries.
2. Enable both the Read Ahead Mode feature (MARBR[28] = 1) and local Prefetch (LBRD0[8] = 0 for Space 0, LBRD1[9] = 0 for Space 1, and/or LBRD0[9] = 0 for Expansion ROM). This will allow the PCI 9054 to keep the read data in the FIFO and transfer it on the PCI request.
   This workaround is only applicable to memory-mapped address spaces. Although burst I/O transactions are legal in PCI, processors typically do not perform burst I/O and x86 processors cannot. Additionally the PCI 9054 will disconnect with the first data of a PCI I/O transaction. The PCI 9054 does not prefetch I/O-mapped address spaces, and therefore Read Ahead mode is not applicable to I/O-mapped spaces.
3. Limit the number of retries allowed from the PCI 9054 before discarding an uncompleted transaction and reporting retry timeout error.

13. PCI Target Abort during DMA Transfer

Design Issue: During a PCI-to-Local DMA transfer, if a Target Abort occurs on the last DMA data transfer cycle, the PCI 9054 will generate an unknown data cycle for the last data to the Local bus. A PCI Target Abort at any other time during the DMA transaction will be successfully completed. This is a rare case condition. If a Target Abort during a DMA transaction occurs, the system should repeat the operation.

Recommendation: A Target Abort is by definition an error condition, and if a Target Abort occurs the last data should be assumed to be invalid. After a DMA transaction is complete, software should check the Received Target Abort status bit (PCISR[12]). If the bit indicates that a PCI Target Abort occurred, software should repeat the DMA transaction.

14. Interrupt Control/Status register (INTCSR) indication of a Master Abort or Target Abort condition

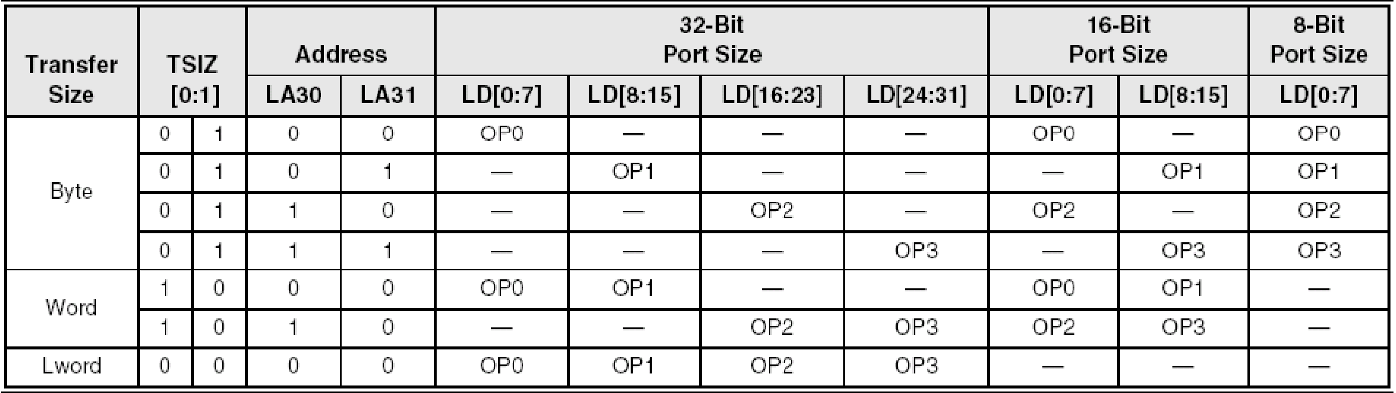
Design Issue: When a Master Abort or Target Abort condition is detected, status bits INTCSR[27:24] reflect the most recent error condition depending on the abort received. Clearing the abort condition will not reset these status bits to their default values of 1.

Recommendation: The INTCSR[27:24] status bits indicating that a Master Abort or Target Abort was signaled or detected are updated when either another abort condition occurs, or a PCI reset is applied to the PCI 9054 (a software reset via CNTRL[30] will not change INTCSR register contents). Otherwise, these bits reflect the status of the last abort condition received. If monitoring of these bits is necessary for error recovery, monitor the equivalent PCI configuration register error bits in the PCI Status register (PCISR[13:11]). PCI 9054 issuance of a Master Abort is signaled in PCISR[13], PCI 9054 issuance of a Target Abort is signaled in PCISR[11], and receipt of a Target Abort from another device while PCI 9054 is master is signaled in PCISR[12].

17. Direct Slave Transfer Size

Issue: In PCI 9054 Data Book version 2.1, Table 3-3 lists how data is transferred onto the M-mode Local Bus during Direct Slaves Writes. The table is incorrect and is replaced by the Table 3-3 shown below.

Table 3-3. Data Bus TSIZ[0:1] Contents for Single Write Cycles



Transfer Size	TSIZ [0:1]		Address		32-Bit Port Size				16-Bit Port Size		8-Bit Port Size
			LA30	LA31	LD[0:7]	LD[8:15]	LD[16:23]	LD[24:31]	LD[0:7]	LD[8:15]	LD[0:7]
Byte	0	1	0	0	OP0	—	—	—	OP0	—	OP0
	0	1	0	1	—	OP1	—	—	—	OP1	OP1
	0	1	1	0	—	—	OP2	—	OP2	—	OP2
	0	1	1	1	—	—	—	OP3	—	OP3	OP3
Word	1	0	0	0	OP0	OP1	—	—	OP0	OP1	—
	1	0	1	0	—	—	OP2	OP3	OP2	OP3	—
Lword	0	0	0	0	OP0	OP1	OP2	OP3	—	—	—

Note: The "—" symbol indicates that a valid byte is not required during that Write Cycle.
However, the PCI 9054 drives these byte lanes, although the data is not used.

Errata Rev. 1.7 for Silicon AC
May 2005
Used for CTPCI & software design.

2. Simultaneous Access to Queue Register in Messaging Unit

Erratum Issue: The PCI 9054 updates one of four queue pointers automatically each time there is a Read or Write to the messaging unit Inbound (Port 40h) or Outbound (Port 44h) ports. The four registers are the inbound free tail pointer (IFTPR), the inbound post head pointer (IPHPR), the outbound free head pointer (OFHPR), and outbound post tail pointer (OPTPR). If a local master initiates a write to the PCI 9054 messaging unit queue registers simultaneous to a PCI access to the Inbound or Outbound ports, the PCI 9054 will fail to automatically increment the appropriate pointers to reflect this. This can result in overwriting a previous message or retrieving a previously read message from the queue. These pointers cannot be 'corrected' due to the fact that the IOP cannot determine if an increment failure has occurred. The failure only occurs when the PCI 9054 returns TA# to the local master simultaneous with the PCI 9054 returning TRDY# to the PCI master on an Inbound or Outbound port access that has been retried.

This erratum does not affect the I_2O protocol. It only affects custom messaging unit implementations.

Solutions/Workarounds (use any):

1. Disable the PCI r2.1 Features Enable bit by clearing MARBR[24]. With this bit clear, the PCI Target Retry Delay Clocks register bits (LBRD0[31:28]) should be set to a value of 3h or greater.

2. **For Multiple Initiator Implementations:**
   a. Implement a semaphore using two on-chip mailboxes or any shared memory region, so that the local master can access the messaging unit queue registers only when the PCI master is not accessing them. Use one mailbox to signal that the PCI bus wants to access the messaging unit and the other mailbox to signal that the Local bus wants to access the messaging unit. Before accessing the messaging unit read the status of the opposite side's mailbox. If the other side has access then Backoff (attempt access later) or else write a flag in the mailbox to claim access. Then read the other mailbox to ensure that both the PCI and Local sides did not write their flags simultaneously. If they did each side will need to Backoff, otherwise access the messaging unit and then clear the flag in the mailbox.
   b. Update the interfering queue registers only from the PCI side, via messages passed through the PCI 9054 internal mailbox registers.

For Single Initiator Implementations:

Implement a single request/reply message protocol for custom message passing. This is recommended for applications where there is a single host and the host waits for a reply before initiating a new request.

3. Direct Master Read with PCI Initiator Cache Enabled

Erratum Issue: PCI 9054 PCI Initiator (Direct Master) reads with PCI Initiator Cache enabled (DMPBAM[2] = 1) will result in 32-bit data reads intermittently returning incorrect data. The incorrect data returned will be the Lword value at the 32-bit aligned address that immediately precedes the correct value. In other words, data that has been previously cached and read out of the PCI Initiator Read FIFO will be repeated on a subsequent read.

Solution/Workaround:

Disable PCI Initiator Cache by clearing the PCI Initiator Cache Enable bit (DMPBAM[2] = 0).

11. PQFP Package LA[0:5] Signal Noise

Note. This erratum only pertains to the PQFP packaged PCI 9054AC part (PLX part number PCI 9054-AC50PI).
It does not pertain to the PBGA packaged part (PLX part number PCI 9054-AC50BI).

Erratum Issue: For the PCI 9054-AC50PI, noise may be injected on the Local Bus causing incorrect values to be output on address bits LA[0:5] if the following occur simultaneously:

1. The PCI 9054-AC50PI is driving the PCI bus with patterns that maximize the number of simultaneously switching outputs on AD[31:0], and
2. The PCI 9054-AC50PI is driving a Local Bus address during Direct Slave or DMA data transfers.

The Local bus signals affected LA[0:5] that should be logic 0's might be incorrectly driven to up to 0.8V for as long as 5 nsec. The amplitude of the noise is proportional to the loading and signal amplitude/charge on the PCI AD[5:0] signals when these are driven low by the PCI 9054. Additionally, only Local Bus devices that detect a logic one near the bottom of the switching range are affected.

Solutions/Workarounds (do one of the following):

1. Do not use the PCI 9054-AC50PI LA[0:5] signals in designs that perform Direct Slave or DMA data transfers. This will limit PCI 9054-AC50PI designs to the lower 64 Mbytes of Local Bus address space
2. Use the PCI 9054-AC50BI or PCI 9056BA66BI parts instead which do not have this issue.

15. Direct Slave Disconnect Without Data

Erratum issue: When the PCI 2.2 Mode bit is set (MARBR[24] = 1), during a Direct Slave burst read, if the first datum is not available for transfer within 16 PCI clocks, the PCI 9054 will issue a PCI Retry, and latch the PCI Command, Address and Byte Enables into its Read FIFO. A PCI master that is target terminated with Retry must unconditionally repeat the same request until it completes.

If instead at least one datum is transferred to the PCI bus, and if the next datum for the PCI burst read is not available within 8 PCI clocks, the PCI 9054 will issue a Disconnect Without Data (TRDY# de-asserted and STOP# asserted, same as Retry signaling). A PCI master that is target terminated with Disconnect is not required to repeat the same request.

The PCI 9054 latches the Command, Address and Byte Enables into its Read FIFO and does not flush the FIFO when issuing a Disconnect Without Data, the same as it does when issuing a Retry (which is a special case of Disconnect Without Data). Therefore, the PCI 9054 requires that the PCI master repeat any transaction that the PCI 9054 terminates with a Disconnect Without Data.

If a PCI Master does not complete its originally requested Direct Slave Delayed Read transfer, the PCI 9054 flushes the Direct Slave Read FIFO after 2^15 PCI clocks and will grant an access to a new Direct Slave Read access. The PCI 9054 retries all other Direct Slave Read accesses that occur before the 2^15 PCI clock timeout.

Typically, if a PCI master receives a Disconnect Without Data in response to its issuance of a read command, the master will repeat the transaction (because the master still wants the data), and the PCI 9054 will respond normally. A situation in which a PCI master might instead read a different address would be an upstream bridge that is 64-bit architecture receiving the Disconnect Without Data on a non-Qword aligned address, with the PCI 9054 Local Address Space being read mapped as Prefetchable (the space's LASxRR[3] and PCIBAR[3] register bits are set). In this case, the bridge might repeat the read starting one address earlier, discarding the last Lword that it read in order to align its repeated read to a Qword-aligned address. The bridge is permitted to discard data only if the target's address space is mapped as Prefetchable memory (LASxRR[3,0] = 10b). It is legal for a bridge to not discard the data but repeat the read starting with the Qword-aligned address with all Byte Enables de-asserted, however, such behavior by design is unlikely.

This erratum does not occur for reads of PCI 9054 registers, and is unlikely to occur for I/O-mapped Local Address Spaces since typical CPUs do not perform burst I/O reads (a Disconnect Without Data can only occur on burst transfers).

Workaround:
Configure memory-mapped Local Address Spaces as non-Prefetchable (LASxRR[3] = 0 in the serial EEPROM). Changing Prefetchable mapping to non-Prefetchable may reduce performance, since the upstream bridge is no longer allowed to prefetch beyond the amount of data that it was commanded to fetch.