CPCI-6713B-ISC8

Intelligent 8 channel
Serial Communications Controller
For *CompactPCI*™ systems

REFERENCE MANUAL

818-10-000-4000
Version 1.0
March 2006

ALPHI TECHNOLOGY CORPORATION

6202 S. Maple Avenue #120
Tempe, AZ 85283 USA
Tel: (480) 838-2428
Fax: (480) 838-4477

NOTICE

The information in this document has been carefully checked and is believed to be entirely reliable. While all reasonable efforts to ensure accuracy have been taken in the preparation of this manual, ALPHI TECHNOLOGY assumes no responsibility resulting from omissions or errors in this manual, or from the use of information contain herein.

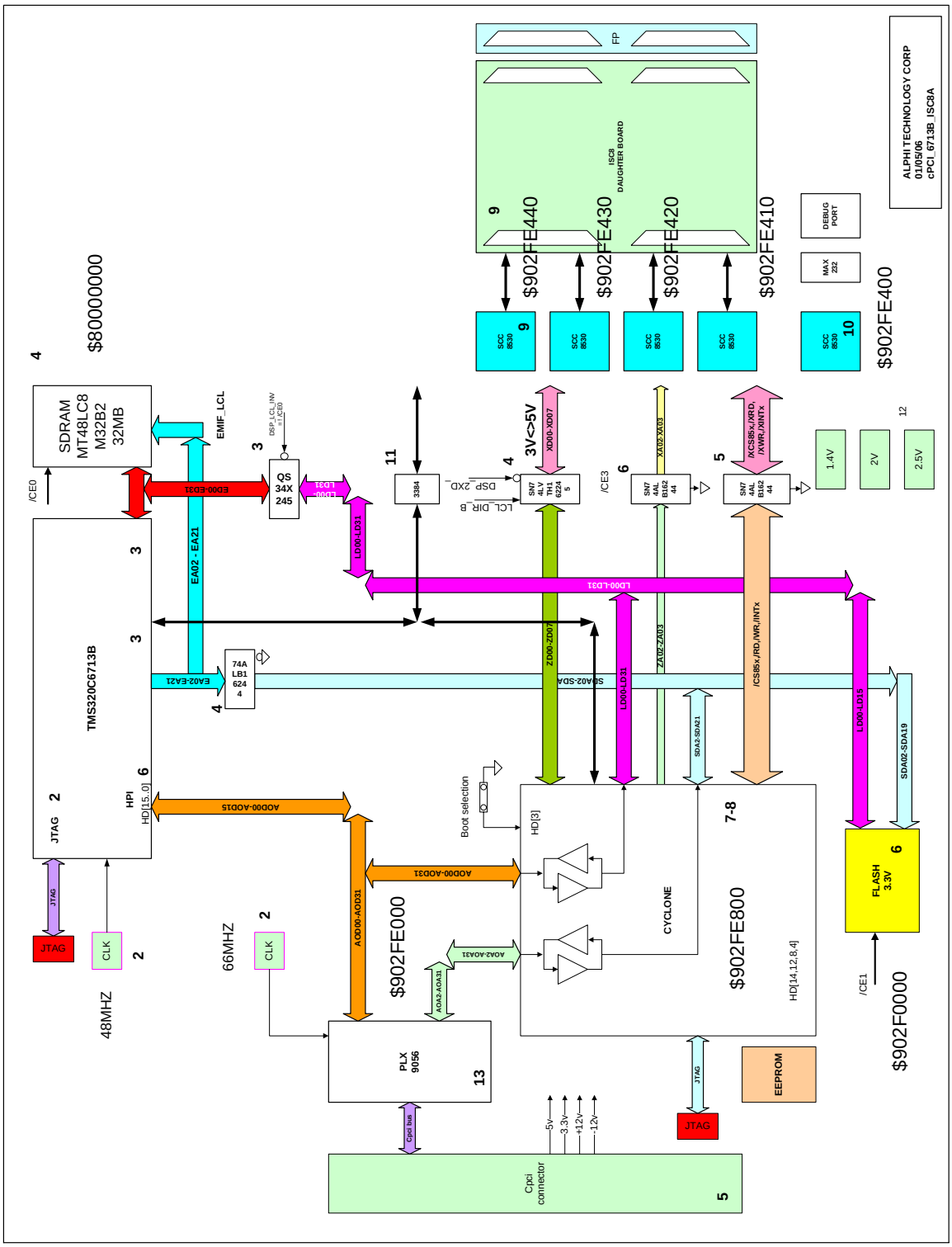
ALPHI TECHNOLOGY reserves the right to make any changes, without notice, to this or any of ALPHI TECHNOLOGY's products to improve reliability, performance, function or design.

ALPHI TECHNOLOGY does not assume any liability arising out of the application or use of any product or circuit described herein; nor does ALPHI TECHNOLOGY convey any license under its patent rights or the rights of others.

ALPHI TECHNOLOGY CORPORATION

TABLE OF CONTENTS

1. BLOCK DIAGRAM



2. ARCHITECTURE

The CPCI-6713B-ISC8 is an intelligent DSP based serial communication board.

The CPCI-6713B-ISC8 provides mechanical support and the electrical interfaces for an I/O mezzanine board. Multiple CPCI-6713B-ISC8 boards may be installed in a single system.

The main processor is a TMS320C6713B with a system clock at 288MHZ.

Other board resources include:

- Up to 32Mbytes of SDRAM
- 512KB Flash memory for the bootstrap program
- RS232C controller
- SCC8530 (5)
- Cyclone FPGA for decoding
- Support for application specific serial mezzanine I/O
- Full interrupt support of host and DSP
- Front panel I/O connectors for all mezzanine I/O
- Four (4) 16 MHz Z85230 SCC devices
- All Z85230 I/O lines routed to mezzanine connectors
- One external interrupt input

3. THE 6713B BOOT PROCESS

The TMS320C6713/13B device has two boot modes: from the HPI or from external asynchronous FLASH EEPROM. On the TMS320C6713B, boot mode is determined during the device reset.

Refer to the TI application note SPRA 512 for details on booting through the HPI.

When using the FLASH to boot, the first 1K of the FLASH are copied, starting to the local address 0x00000000 of the processor. After the end of the transfer, the CPU starts executing at the address 0. The small program thus loaded has the responsibility of doing at least a minimum configuration of 6713 registers and of copying a bigger program that will start executing and actually boot the board.

The board interface is configured as Big-Endian with the 8-bit data on the ED[31:24] side of the bus. HPI is always enabled

J3_ HD3	BOOT CONFIGURATION
No jumper	CE1 width 32-bit, HPI boot/Emulation boot
jumper	CE1 width 16-bit, Asynchronous external ROM boot with default timings

4. RESET SCHEME

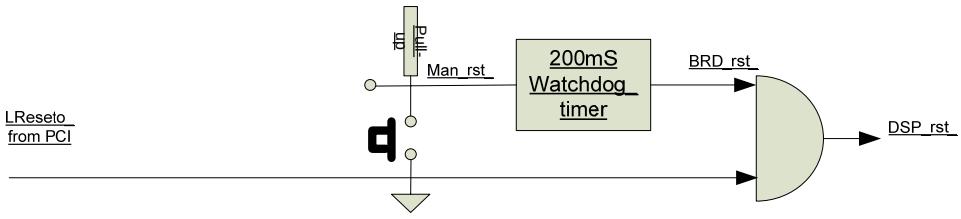
The TMS6713B-MFIO board has multiple source of reset. Upon Power-on a watch-dog timer provide a 200ms minimum delay where the DSP is maintained on reset. This gives each programmable logic chip enough time to get its program loaded.

The board can be reset by:

-Power-on reset

-PCI reset line /LRESETO from PLX

-Push-button



5. 6713 LOCAL PERIPHERALS

5.1 6713 INTERNAL REGISTERS

5.1.1 EMIF PROGRAMMING

The board uses the 6713 EMIF module (External Memory Interface) to generate on-board timings.

Using the EMIF, the memory space is divided as:

MEMORY SPACE	SETTING	RESOURCE	SIZE	ADDRESS RANGE
CE0	0xffffbf33	SDRAM	8-32MB	0x80000000-0x81FFFFFF
CE1	0x12328620	ADD_ON + PERIPHERALS	4MB	0x90000000-0x903FFFFF
CE2		Not used		0xA0000000-0xA31FFFFF
CE3	0x12328610	FLASH (16-bit wide)	4MB	0xB0000000-0xB03FFFFF

The default EMIF configuration in accordance with the PLL settings has been selected to give the maximum performance without sacrificing signal integrity. We recommend not changing it, as an incorrect setting will prevent the board from functioning properly.

Default EMIF configuration:

REGISTER	ADDRESS	SETTING
EMIF_GCTL	0x01800000	0x00003778
EMIF_CE0	0x01800008	0xffffbf33
EMIF_CE1	0x01800004	0x12328620
EMIF_CE2	0x01800010	0x00c10320
EMIF_CE3	0x01800014	0x12328610
EMIF_SDRAMCTL	0x01800018	0x46216000
EMIF_SDRAMTIM	0x0180001C	0x00000618
EMIF_SDRAMEXT	0x01800020	0x00054529

5.1.2 PLL PROGRAMMING

The outside 48MHZ clock provided to the DSP is multiplied by the programmable internal PLL (x12) generating a 576 MHZ internal clock. This clock is then divided by two to provide SYSCLK or CLKOUT2 (288 MHZ).

The outside bus timings are generated by the TMS320C6713B EMIF using CLKOUT3 (the 600MHZ clock divided by 6 by the PLL # 2) which is a 96 MHZ clock.

In any case, the EMIF clock cannot exceed 100MHZ.

We strongly recommend using the default timings.

```
/*----------------------------------------------------------------*/
/* C6713 PLL SUPPORT                                              */
/*----------------------------------------------------------------*/
#define PLL_BASE_ADDR       0x01b7c000
#define PLL_PID             ( PLL_BASE_ADDR + 0x000 )
#define PLL_CSR             ( PLL_BASE_ADDR + 0x100 )
#define PLL_MULT            ( PLL_BASE_ADDR + 0x110 )
#define PLL_DIV0            ( PLL_BASE_ADDR + 0x114 )
#define PLL_DIV1            ( PLL_BASE_ADDR + 0x118 )
#define PLL_DIV2            ( PLL_BASE_ADDR + 0x11C )
#define PLL_DIV3            ( PLL_BASE_ADDR + 0x120 )
#define PLL_OSCDIV1         ( PLL_BASE_ADDR + 0x124 )

#define CSR_PLLEN           0x00000001
#define CSR_PLLPWRDN        0x00000002
#define CSR_PLLRST          0x00000008
#define CSR_PLLSTABLE       0x00000040
#define DIV_ENABLE          0x00008000

/*----------------------------------------------------------------  */
/* init_pll()                                                       */
/* Pll Initialization                                               */
/*----------------------------------------------------------------  */
void init_pll()
{
   /*-------------------------------------------------------------------*/
   /* When PLLEN is off DSP is running with CLKIN clock    */
   /* source, currently 50MHz or 20ns clk rate.                    */
   /*-------------------------------------------------------------------*/
   *(int *)PLL_CSR  &= ~CSR_PLLEN;

   /* Reset the pll.  PLL takes 125ns to reset. */
   *(int *)PLL_CSR  |= CSR_PLLRST;

   /*----------------------------------------------------------------*/
   /* PLLOUT = CLKIN*(DIV0+1) * PLLM                                 */
```


5.2 CE0 SPACE

The SDRAM is mapped in the CE0 space. Depending on your board options, up to 32 megabytes are available (4 megabytes is standard). The SDRAM is connected to the 6713B bus. The data and address buses are isolated to the local bus by buffers.

5.3 CE1 SPACE

CE1 space is share between the boot flash, all peripherals including the ADD_ON module. A Cyclone FPGA is dedicated to 48 digital I/O pin buffered by a group of 6 transceivers. Each transceiver can be set as output or inputs. More specification can be found on the CYCLONE I/O module.

PERIPHERALS	SIZE	ADDRESS RANGE
FLASH	2MB	0x90000000 - 0x901F FFFF
ADD_ON (ADDA)	1MB-8KB	0x90200000 - 0x902F 7FFF
PLX_REG	1KB	0x902F E000 - 0x902F E3FF
SCC8530 controller (0)	1KB	0x902F E400 - 0x902F E40F
SCC8530 (1)	1KB	0x902F E410 - 0x902F E41F
SCC8530 (2)	1KB	0x902F E420 - 0x902F E42F
SCC8530 (3)	1KB	0x902F E430 - 0x902F E43F
SCC8530 (4)	1KB	0x902F E440 - 0x902F E44F
CYCLONE REG	1KB	0x902F E800 - 0x902F EBFF
USB CONTROLLER	1KB	Not used
CYCLONE_IO	1KB	Not used

5.3.1 FLASH EPROM

A 256Kx16 Flash memory M29F400DB is visible at the base address of CE1 (0x0x90000000) space for use during the boot process. It is also visible in CE3 (0xB0000000). The address space CE1 is 32-bit wide once the board has booted, so when accessing the FLASH, only the lower 16 bits of each 32-bit word contains actual data.

By contrast, the CE3 space is 16-bit wide, matching the FLASH actual geometry and the data is continuous.

5.3.2 CYCLONE REGISTERS

The cyclone logic has a set of locals registers used to set-up some local parameters or to communicate with the PLX bus.

DSP_CNTL0_ADR 0x902F E800

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
XCNTL1_O	*XCNTL0_O*	*XRESET*	**NMIEN**	**x**	**FLASH_PAGE**	**!/LED_O[1] L1**	**!/LED_O[0] L2**

/LED_O[1..0] : This register controls 2 LEDs. The reset position is for the LEDs to be "off". A "1" in the position corresponding to the LED will set it "on".

FLASH_PAGE: Currently unused, should be 0.

NMIEN: TMS320C6713B allows receiving the NMI interrupt from:

- LINTO# (PLX source) if LINTO_DIR = "1".
- DSPNMI (BIT # 7 PCI_CNTL0).
- The default is 0: "disabled".

XRESET: Not used .

XCNTL[1..0]: Not used .

DSP_STAT0_ADR 0x902F E804

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
XSTAT1	*XSTAT0*	*/USB_RDY*	**DSPNMI**	**RY_BY_FLASH**	**8530_IRQ**	**/LINTO**	**USERO**

XSTAT[1..0]: Not used.

/USB_RDY Not used

DSPNMI: Status of the PCI-generated bit # 7 PCI_CNTL0 to activate an NMI interrupt to the processor

RY_BY_FLASH Status of the RY_BY pin from flash memory for use in its programming.

8530_IRQ — Status of interrupt from SCC8530: when the value is “0”, the interrupt is active.

/LINTO — Status of interrupt line activated by the PCI bus through the PLX 9056.

USERO — Status of USERO line activated by the PCI bus through the PLX 9056.

DSP_CNTL1_ADR 0x902F E808

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
0	**Not used**	**Not used**	**LINTO_DIR**	**Not used**	**LOCKOUT 9056**	**Not used**	**Not used**

LINTO_DIR — A “1” direct connection of LINTO# interrupt to the DSP NMI interrupt. NMIEN BIT # 4 of DSP_CNTL0 must be set to “1”.

A “0” direct connection of LINTO# interrupt to the DSP_INT7 interrupt.

USB_DIR — Not used.

LOCKOUT 9056 — Control bit from DSP to solve a possible bottleneck arbitration scheme

Not used

WAKE_USB — Not used.

/USB_RST — Not used.

INT_VSB_EN — Not used.

INT_8530_EN — Not used.

DSP_STAT1_ADR 0x902F E80C

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
Not used	Not used	Not used	Not used	Not used	Not used	Not used	Not used

DSP_CNTL2_ADR 0x902F E810

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
D_DSP7	D_DSP6	D_DSP5	D_DSP4	D_DSP3	D_DSP2	D_DSP1	D_DSP0

This register can be used as an inter-processor communication register with the PLX 9056.

It can be read at the location BAR3 +$C

DSP_STAT2_ADR 0x902F E814

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
P_CNT7	P_CNT6	P_CNT5	P_CNT4	P_CNT3	P_CNT2	P_CNT1	P_CNT0

This register can be used as an inter-processor communication register with the PLX 9056. Its value reflects what was written in the PCI_CNTL1 register BAR3 +$8 ,lower byte of word acces

DSP_IPINT_ADR 0x902F E820-E85C

The TMS320C6713B supports 16 prioritized interrupts. In the present configuration, 5 interrupts are used by external hardware.

See Interrupt registers details in the paragraph 3.3.3.

DSP_LINTI_INTI 0x902F E820

When DSP writes at this location, an interrupt is send to the HOST through the line /LINTI.

The multiplexer must be set to “0”. It is set to “0” upon reset.

DSP_LINTI_RSTI 0x902F E824

When the interrupt /LINTI is asserted, the DSP can cleared the latch memorizing the LINTI# interrupt by writing at this address.

The PCI can do the same activating PCI_LINTI_RST line by a write at location Base + 0x10.

The interrupt is also cleared the board reset.

DSP_CYC_REV_ADR 0x902F E860

Unique 8 bits identifiers factory programmed within the Cyclone FPGA.

Actual value : A1 hex

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
1	0	1	0	0	0	0	1

DSP_PCB_REV_ADR[] 0x902F E864

Unique 8 bits identifiers factory programmed within the Cyclone FPGA.

Actual value : 0A hex

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
0	0	0	0	1	0	1	0

DSP_PLX_REV_ADR[] 0x902F E868

Unique 8 bits identifiers factory programmed within the Cyclone FPGA

Actual value : 01 hex

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
0	0	0	0	0	0	0	1

DSP_DSP_REV_ADR[] 0x902F E86C

Unique 8 bits identifiers factory programmed within the Cyclone FPGA

Actual value : 01 hex

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
0	0	0	0	0	0	0	1

5.3.3 INTERRUPT REGISTERS

0x902F E840- 0x902F E85C

DEFAULTS INTERRUPT

External interrupt	Name source	Function
XINT[4]	SCC 8530 [4..1]	SCC controller
XINT[5]	GPS	GPS clock
XINT[6]	SCC8530[0]	Monitor
XINT[7]	/LINTO	HOST through PLX
NMI	/LINTO or DSPNMI	PCI SOURCE

The GPS signal will generate an interrupt on the positive or negative edge depending of the setting of jumper J5. No jumper will select negative edge.

NOTE: THE INTERRUPT NEEDS TO BE PROGRAMMED FOR NEGATIVE EDGE DETECTION AT TMS320C6713B LEVEL

5.3.4 INTERRUPT TO HOST

The source for /LINTI is unique and selectable using DSP_LINTI_REG register at address $ 902FE854

SOURCE	CODE	Function
DSP_LINTI_INT	0000	**DSP write generate an interrupt to PCI**
8530[1]	0001	Interrupt from SCC8530[1] is source of LINTI#
8530[2]	0002	Interrupt from SCC8530[2] is source of LINTI#
8530[3]	0003	Interrupt from SCC8530[3] is source of LINTI#
8530[4]	0004	Interrupt from SCC8530[4] is source of LINTI#
8530[0]	0005	Interrupt from SCC8530[0](mon) is source of LINTI#
GPS_int	**0006**	GPS signal
HPI_int	**0007**	Interrupt from HPI bus.

When the DSP generate an interrupt to the PCI using LINTI#, the interrupt is latched.

To reset it the DSP need to write at location $902FE824 or the PCI need to write at the address base + $10.

/RD_INT_STATUS

0x902F E858

This register provides the status of the interrupt lines going to the 6713B.

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
GPS signal	**LINTI#**	**LINTO#**	**SCC[0] MON. int**	**SCC[4] int**	**SCC[3] int**	**SCC[2] int**	**SCC[1] int**

5.3.5 INTERRUPT FROM HOST

The interrupt coming from the PCI bus is LINTO# and it is routed to the DSP_INT7 line if LINTO_DIR is = “0” or to the dsp NMI if LINTO_DIR = “1” and NMIEN is = “1”.

5.3.6 PLX 9056 REGISTERS

Base address: 0x902FFE000

The TMS320C6713B can access the PLX configurations registers at the following addresses.

NAME	PCI AD	LOCAL	FUNCTION
PCIIDR	00h	00h	**PCI Configuration ID**
PCICR	04h	04h	**PCI Command**
PCISR	06h	06h	**PCI Status**
PCIREV	08h	08h	**PCI Revision ID**
PCICCR	09-0Bh	09-0Bh	**PCI Class Code**
PCICLSR	0Ch	0Ch	**PCI Cache Line Size**
PCILTR	0Dh	0Dh	**PCI Bus Latency Timer**
PCIHTR	0Eh	0Eh	**PCI Header Type**
PCIBISTR	0Fh	0Fh	**PCI Built-In Self-Test (BIST**
PCIBAR0	10h	10h	**PCI Base Address for Memory Accesses to Local, Runtime, DMA, and Messaging Queue Registers**
PCIBAR1	14h	14h	**PCI Base Address for I/O Accesses to Local, Runtime, DMA, and Messaging Queue Registers**
PCIBAR2	18h	18h	**PCI Base Addr. for Accesses to Local Address Space 0**
PCIBAR3	1Ch	1Ch	**PCI Base Addr. for Accesses to Local Address Space 1**
PCIBAR4	20h	20h	**PCI Base Address**
PCIBAR5	24h	24h	**PCI Base Address**
PCICIS	28h	28h	**PCI Cardbus Information Structure Pointer**

PCISVID	2Ch	2Ch	**PCI Subsystem Vendor ID**
PCISID	2Eh	2Eh	**PCI Subsystem ID**
PCIERBAR	30h	30h	**PCI Base Address for Local Expansion ROM**
CAP_PTR	34h	34h	**New Capability Pointer**
PCIILR	3Ch	3Ch	**PCI Interrupt Line**
PCIIPR	3Dh	3Dh	**PCI Interrupt Pin**
PCIMGR	3Eh	3Eh	**PCI Minimum Grant**
PCIMLR3	3Fh	3Fh	**PCI Maximum Latency.**
PMCAPID	40h	180h	**Power Management Capability ID .**
PMNEXT	41h	181h	**Power Management Next Capability Pointer**
PMC	42h	182h	**Power Management Capabilities .**
PMCSR	44h	184h	**Power Management Control/Status**
PMCSR_BSE	46h	186h	**PMCSR Bridge Support Extensions**
PMDATA	47h	187h	**Power Management Data**
HS_CNTL	48h	188h	**Hot Swap Control**
HS_NEXT	49h	189h	**Hot Swap Next Capability Pointer .**
HS_CSR	4Ah	18Ah	**Hot Swap Control/Status .**
PVPDID	4Ch	18Ch	**PCI Vital Product Identification .**
PVPD_NEXT	4Dh	18Dh	**PCI Vital Product Data Next Capability Pointer**
PVPDAD	4Eh	18Eh	**PCI Vital Product Data Address**
PVPDATA	50h	190h	**PCI VPD Data .**
LAS0RR	00h	80h	**Direct Slave Local Address Space 0 Range**
LAS0BA	04h	84h	**Direct Slave Local Address Space 0 Local Base Address (Remap**
MARBR	08h or ACh	88h or 12Ch	**Mode/DMA Arbitration**

NAME	PCI AD	LOCAL	FUNCTION
BIGEND	0Ch	8Ch	**Big/Little Endean Descriptor.**
LMISC1	0Dh	8Dh	**Local Miscellaneous Control**
PROT_AREA	0Eh	8Eh	**Serial EEPROM Write-Protected Address Boundary**
LMISC2	0Fh	8Fh	**Local Miscellaneous Control 2**
EROMRR	10h	90h	**Direct Slave Expansion ROM Range**
EROMBA	14h	94h	**Direct Slave Expansion ROM Local Base Address Re BREQo Control**
LBRD0	18h	98h	**Local Address Space 0/Expansion ROM Bus Region Descriptor**
DMRR	1Ch	9Ch	**Local Range for Direct Master-to-PCI**
DMLBAM	20h	A0h	**Local Base Address for Direct Master-to-PCI Memory**
DMLBAI	24h	A4h	**Local Base Address for Direct Master-to-PCI I/O Configuration**
DMPBAM	28h	A8h	**PCI Base Address (Remap for Direct Master-to-PCI Memory)**
DMCFGA	2Ch	Ach	**PCI Configuration Address for Direct Master-to-PCI I/O Configuration**
LAS1RR	F0h	170h	**Direct Slave Local Address Space 1 Range**
LAS1BA	F4h	174h	**Direct Slave Local Address Space 1 Local Base Address (Remap**
LBRD1	F8h	178h	**Local Address Space 1 Bus Region Descriptor**
DMDAC	FCh	17Ch	**Direct Master PCI Dual Address Cycles Upper Address**
PCIARB	100h	1A0h	**PCI Arbiter Control**
PABTADR	104h	1A4h	**PCI Abort Address**
MBOX0	40h or 78h	C0h	**Mailbox 0**
MBOX1	44h or 7Ch	C4h	**Mailbox 1**
MBOX2	48h	8Ch	**Mailbox 2**
MBOX3	4Ch	CCh	**Mailbox 3**
MBOX4	50h	D0h	**Mailbox 4**
MBOX5	54h	D4h	**Mailbox 5**
MBOX6	58h	D8h	**Mailbox 6**
MBOX7	5Ch	DCh	**Mailbox 7**
P2LDBELL	60h	E0h	**PCI-to-Local Doorbell**

L2PDBELL	64h	E4h	**Local-to-PCI Doorbell**
INTCSR	68h	E8h	**Interrupt Control/Status**
CNTRL	6Ch	ECh	**Serial EEPROM Control, PCI Command Codes,User I/O Control, and Init Control**
PCIHIDR	70h	F0h	**PCI Hardwired Configuration ID**
PCIHREV	74h	F4h	**sPCI Hardwired Revision ID**

NAME	PCI AD	LOCAL	FUNCTION
DMAMODE0	80h	100h	**DMA Channel 0 Mode**
DMAPADR0	84h	104h	**when DMAMODE0[20]=0**
Or	88h	108h	**when DMAMODE0[20]=1 DMA Channel 0 PCI Address**
DMALADR0	88h	108h	**when DMAMODE0[20]=0**
Or	8Ch	0Ch	**when DMAMODE0[20]=1 DMA Channel 0 Local Address**
DMASIZ0	8Ch	10Ch	**when DMAMODE0[20]=0**
Or	84h	4h	**when DMAMODE0[20]=1 DMA Channel 0 Transfer Siz Bytes**
DMADPR0	90h	110h	**DMA Channel 0 Descriptor Pointer**
DMAMODE	14h	114h	**DMA Channel 1 Mode**
DMAPADR1	98h	118h	**when DMAMODE1[20]=0**
Or	9Ch	11Ch	**when DMAMODE1[20]=1 DMA Channel 1 PCI Address**
DMALADR1	9Ch	11Ch	**when DMAMODE1[20]=0**
Or	A0h	120h	**when DMAMODE1[20]=1 DMA Channel 1 Local Address.**
DMASIZ1	A0h	120h	**when DMAMODE1[20]=0**
Or	98h	118h	**when DMAMODE1[20]=1 DMA Channel 1 Transfer Size Bytes**
DMADPR1	A4h	124h	**DMA Channel 1 Descriptor Pointer**
DMACSR0	A8h	128h	**DMA Channel 0 Command/Status**
DMACSR1	A9h	129h	**DMA Channel 1 Command/Status.**
DMAARB	Ach	12Ch	**DMA Arbitration**
DMATHR	B0h	130h	**DMA Threshold**
DMADAC0	B4h	134h	**DMA Channel 0 PCI Dual Address Cycles Upper Address**
DMADAC1	B8h	138h	**DMA Channel 1 PCI Dual Address Cycle Upper**

			Address
OPQIS	30h	B0h	**Outbound Post Queue Interrupt Status**
OPQIM	34h	B4h	**Outbound Post Queue Interrupt Mask.**
IQP	40h		**Inbound Queue Port.**
OQP	44h		**Outbound Queue Port**
MQCR	C0h	140h	**Messaging Queue Configuration**
QBAR	C4h	144h	**Queue Base Address**
IFHPR	C8h	148h	**Inbound Free Head Pointer**
IFTPR	CCh	14Ch	**Inbound Free Tail Pointer**
IPHPR	D0h	150h	**Inbound Post Head Pointer**
IPTPR	D4h	154h	**Inbound Post Tail Pointer**
OFHPR	D8h	158h	**Outbound Free Head Pointer**
OFTPR	DCh	15Ch	**Outbound Free Tail Pointer**
OPHPR	E0h	160h	**Outbound Post Head Pointer**
OPTPR	E4h	164h	**Outbound Post Tail Pointer**
QSR	E8h	168h	**Queue Status/Control**

Detailed information can be found into the PLX 9056 data sheet.

5.3.7 RS-232C PORT

Base address: 0x902FFE400

Port A Control Register: 0x902FFE408

Port A Data Register: 0x902FFE40C

The board uses the port A of a SCC85C230 to provide a serial interface. It use a 16 MHz clock as default clock. This clock is the result of the EMIF clock (96 MHz) divided by 3 then by 2 inside the Cyclone.

Optional clock selected by J7 is a 7.372800 MHz local clock oscillator. The port B has no outside connection.

The interrupt from the serial controller is routed to the Cyclone to be shared with others interrupts source.

5.4 CE2 SPACE

Not used.

5.5 CE3 SPACE

The CE3 space allows access to the boot FLASH as a 16-bit space. This allows to for instance running a program stored into it, or in general to access the data continuously. By contrast when accessing the same FLASH at the address 0x90000000, only 2 bytes are valid in each 4 bytes.

6. PCI-SLAVE ADDRESS MAP

6.1 PLX ADDRESS MAP

6.2 LOCAL RESOURCES

There are 4 base address regions available on the PLX PCI9056. The **PCI-6713MFC** uses all 4 regions.

- BAR0: Memory access the PLX operation registers.
- BAR1: I/O access the PLX operation registers.
- BAR2: Not used.
- BAR3: Pass-thru for the HPI (offset 0x20000) and cyclone (offset 0x00000) accesses

  BAR3 is mapped in 16 bits.

RESOURCE	REGION	SPACE	ADDRESS RANGE
PLX registers	BAR0	AS0	0x000-0x1FF
Dual-Ported SRAM	BAR2	Not Used	0x000000-0x3FFFFF
Cyclone registers	BAR3	AS15	0x00000
DSP HPI registers	BAR3	AS15	0x20000-0x2000F

6.2.1 CYCLONE REGISTERS

The HOST can access several registers located inside the CYCLONE FPGA.

These registers are used mainly to monitor the behavior of the board or to control and/or generate interrupt to the DSP. Details are provided above.

PCI_CNTL0 ***Base + 0x0***

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
DSPNMI	USERI_O	HINTEN	0	0		DSPRST	0

DSPNMI A “1” generates an NMI interrupt to the TMS320C6713B.

NMIEN bit must be set to “1”..

HINTEN Host interrupt enable. This bit, when set to “1”, allows the LINTI# signal to generate an interrupt on the PCI. The default is “disabled”

DSPRST Not used.

When set to “1” by PCI will maintain the TMS320C6713B on reset mode. It needs to be set back to “0” to remove the DSP reset.

PCI_STAT0 ***Base + 0x4***

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
0	N/A_	0	/LINTI	XCNTL1	XCNTL0	/LED_1_	/LED_0_

/LINTI Status of the interrupt request line LINTI to the PCI bus. When this bit is "0", an interrupt is requested. HINTEN must be enabled for this line to actually generate an interrupt on the PCI bus.

DSP_HINT Status of TMS320C6713B generated HPI interrupt

XCNTL1, XCNTL0 Not used.

/LED_0 Status of the LEDs controlled by the DSP. At reset, the LEDS are "off". A "1" will turn the corresponding LED "on".

/LED_1 Status of the LEDs controlled by the DSP. At reset, the LEDS are "off". A "1" will turn the corresponding LED "on".

PCI_CNTL1 ***Base + 0x8***

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
P_CNT7	**P_CNT6**	**P_CNT5**	**P_CNT4**	**P_CNT3**	**P_CNT2**	**P_CNT1**	**P_CNT0**

The value written by the host in the PCI_CNTL1 register can be read by the DSP in the DSP_STAT2 register at address $902FE814

PCI_STAT1 ***Base + 0xC***

ED7	ED6	ED5	ED4	ED3	ED2	ED1	ED0
D_DSP7	**D_DSP6**	**D_DSP5**	**D_DSP4**	**D_DSP3**	**D_DSP2**	**D_DSP1**	**D_DSP0**

The value written in the *DSP_CNTL2* register by the DSP can be read by the host in this register.

PCI_LINTI_RST ***Base + 0x10***

A write at this address by the host will de-assert the line */LINTI* that generates an interrupt on the PCI bus.

PCI_DSP_INT ***Base + 0x14***

Not used

6.2.2 DSP HPI REGISTERS

The HOST through the PLX 9056 can access to all the resource of the board through the Host Port Interface bus or H.P.I. of the DSP. It is a 16 bit bus with only four (4) registers locations. As the TMS320C6713B is a 32bit machine, every access consists of two consecutive 16 bit half words.

The two DSP signals *HCNTL1* and *HCNTL0* that select the type of access are automatically generated by the board logic from the address used.

BAR3 offset	HCNTL1	HCNTL0	Description
0x20000	0	0	Host reads from or writes to the HPI **control** register (HPIC).
0x20004	0	1	Host reads from or writes to the HPI **address** register (HPIA).
0x20008	1	0	Host reads from or writes to the HPI **data** register (HPID) in *auto increment* mode. The HPI address register (HPIA) is post incremented by a word address (four bytes addresses)
0x2000C	1	1	Host reads from or writes to the HPI **data** register (HPID) in *fixed* address mode. The HPI address register (HPIA) is not affected.

```
typedef struct hpi_t {
        long ctrl;
        long addr;
        long data;
        long autoinc;
} hpi_t;
hpi_t *hpi;

ULONG readHPI(ULONG addr)
{
        int i = 10000;

        hpi->addr = addr;
        hpi->ctrl = 0x00110011;
```



```
        while ((hpi->ctrl & 8)&& (i--)) ;              // wait for ready
        if (i == 0) printf("\nTimeout accessing the HPI and waiting for ready\n");
        return hpi->data;
}

void writeHPI(ULONG addr, ULONG data)
{
        int i = 10000;

        hpi->addr = addr & 0xfffffffc;                 // force a 32 bit boundary
        hpi->data = data;
        hpi->ctrl = 0x00110011;
        while ((hpi->ctrl & 8)&& (i--)) ;              // wait for ready
        if (i == 0) printf("\nTimeout accessing the HPI and waiting for ready\n");
}
```

6.3 JTAG MODE

Pin	Signal name	Description
1	TCK	Clock Signal
2	GND	Ground
3	TDO	Data from device
4	VCC	Power supply
5	TMS	JTAG state machine control
6	NC	Power supply
7	NC	No connect
8	NC	No connect
9	TDI	Data to Device
10	GND	Ground

6.4 PCI ACCESS

6.4.1 RESETTING THE BOARD IN A WINDOWS XP ENVIRONMENT

```
HRESULT PlxDevice::Reset()
{
        HRESULT result;

        if (!m_pPlxRegisters)
                return ALPHI_E_BAD_CARD_CONFIGURATION;

        m_ProgramStartingAddress = 0;

        m_pPlxRegisters->cntrl = 0xc0017676;                    // assert the board local reset

        DWORD Expiration = QueryTimeInMs() + scm_resetdelay;
        DWORD Expiration2 = QueryTimeInMs() + scm_resetdelay/2;

        while (QueryTimeInMs() <= Expiration2);

        GetMboxStatus(0xffffffff);

        // Since the DSP is not running yet, it has not set
        // the MBOX empty bits yet.
        m_pPlxRegisters->p2ldbell = 0xf0;

        while (QueryTimeInMs() <= Expiration);

        // Turn off PLX related interrupts on the DSP.
        m_pPlxRegisters->intcsr &= ~0x10000;
        // Remove the reset, turning off access from HOST
        m_pPlxRegisters->cntrl  = 0x17676;                      // remove the local reset

        if (m_DeviceCaps.ProcessorType == None)
                return ALPHI_S_OK;

        Expiration = QueryTimeInMs() + scm_startupdelay;

        // wait for DSP to clear the status bits
        while (m_pPlxRegisters->p2ldbell && (QueryTimeInMs() <= Expiration));
```



```
        if (m_pPlxRegisters->p2ldbell) {
                printf("The board did not clear up the doorbell register\n");
                return ALPHI_E_DEVICE_FAILED_RESET;
        }

        // Reset the empty bits
        m_pPlxRegisters->p2ldbell = 0xf0;

        if (FAILED(ReadMbox(2, true))){
                printf("Failed to read the mailbox\n");
                return ALPHI_E_DEVICE_FAILED_RESET;
        }

        if ((result = IsKernelRunning()) == ALPHI_E_DSP_NOT_RUNNING_KERNEL) {
                printf("The kernel test failed!\n");
                return ALPHI_E_DSP_NOT_RUNNING_KERNEL;
        }
        else return result;
}
```

6.4.2 ACCESSING THE LOCAL PERIPHERAL USING HPI ACCESS

The 6713 HPI interface allows accessing any address on the board, through the processor itself. This snippet of code gives an example of using the interface. It can be much improved by using the auto-increment access for instance.

```
typedef struct hpi_t {
        long ctrl;
        long addr;
        long data;
        long autoinc;
} hpi_t;
hpi_t *hpi;

ULONG readHPI(ULONG addr)
{
        int i = DELAY;

        hpi->addr = addr;
        hpi->ctrl = 0x00110011;
```



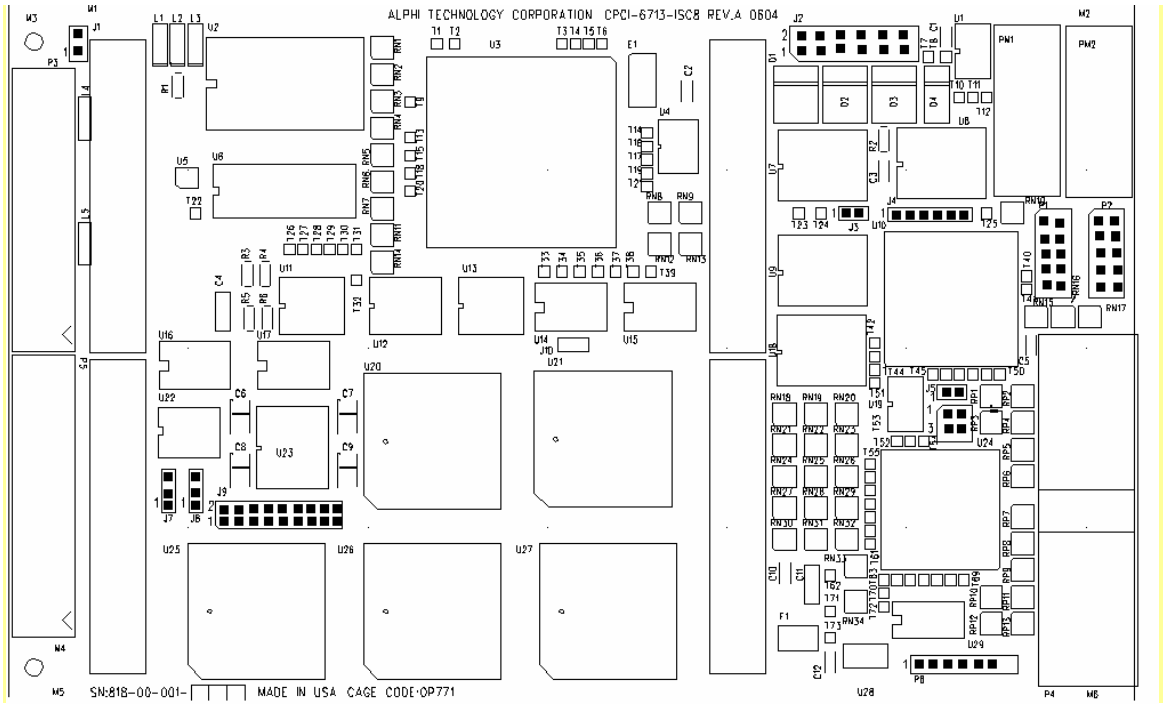
```
        while ((hpi->ctrl & 8)&& (i--)) ;                    // wait for ready
        if (i == 0) printf("\nTimeout accessing the HPI and waiting for ready\n");
        return hpi->data;
}

void writeHPI(ULONG addr, ULONG data)
{
        int i = DELAY;

        hpi->addr = addr & 0xfffffffc;                       // force a 32 bit boundary
        hpi->data = data;
        hpi->ctrl = 0x00110011;
        while ((hpi->ctrl & 8)&& (i--)) ;                    // wait for ready
        if (i == 0) printf("\nTimeout accessing the HPI and waiting for ready\n");
}
```


7. CONNECTIONS

7.1 JUMPER LOCATION



7.2 JUMPER DESCRIPTION

JUMPER	FACTORY SETTING	DESCRIPTION
J1	None	J1 1-2 Asserts Local Reset
J3	1-2	Boot Select
J4	None	Factory test use
J5	None	Select falling or rising edge external interrupt mode. Use this jumper in conjunction with J8 as required. None Rising Edge 1-2 Falling Edge
J6	1-2 3-4	Factory use Select JTAG download for FPGA
J7	2-3	Selects 16 MHz from Cyclone or Oscillator U17 as PCLK source for all 8530 devices J7 1-2 selects U17source 7.3728MHZ J7 2-3 selects Cyclone divider source (default)
J8	2-3	Connects the external interrupt signal EX_GPS to pull-up or pull-down resistors. J8 1-2 selects pull-up J8 2-3 selects pull-down
J9	None	Routes the buffered U16 clock signal to each 85230 Receive Clock (RTXC) pin J9 1-2 RTXCA System Console Port J9 3-4 RTXCB SCC4 J9 5-6 RTXCA SCC4 J9 7-8 RTXCB SCC3

		J9 9-10 RTXCA SCC3 J9 10-11 RTXCB SCC2 J9 12-13 RTXCA SCC2 J9 14-15 RTXCB SCC1 J9 16-17 RTXCA SCC1

Table 7.1 Jumper Descriptions

7.3 CONNECTORS DESCRIPTION

P1	Factory use FPGA JTAG
P2	RS-232 consol port
P3	I/O connector Ch 1-4
P4	CPCI connector
P5	I/O connector Ch 5-8
P6	JTAG for PLX9056
J2	JTAG for TMS3206713B
P4	CPCI connector

7.4 IP I/O CONNECTORS (P3, P5)

Connector	I/O for
P3	Mezzanine I/O 1 1-50
P5	Mezzanine I/O 2 1-50

50 pin high density flat cable connectors are used to route all the IP I/O signals off the card. ALPHI Technology can supply transition modules and cable assemblies to meet any special requirements. Contact the factory for additional information.

Connectors are manufactured by HIROSE.

The I/O signals for the mezzanine are directly routed off the card through the front panel.

Use	Model
On PC Board	HIF6A-50PA-1.27DS
Suggested Plug	H1F6-50D-1.27R

Table 7.2: I/O Connector Model Numbers

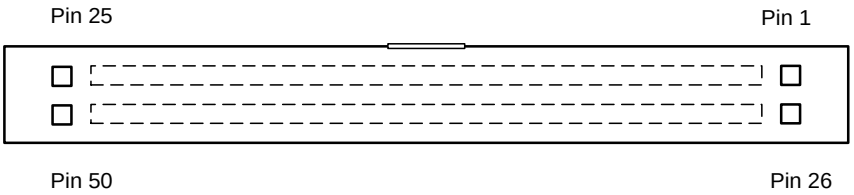


Figure 7.1: IP I/O CONNECTORS

7.5 SERIAL RS232 PORT (P3)

An 8 pin header is located on the PCB and is used to route the RS232 signals off the card.

Port A of the 8530 is configured as an RS232 port, and serves as the console output for the bootloader and any DSP programs linked with the ALPHI_IO.LIB library discussed in the DSP Support Library. Port B is not connected.

The connector is a standard 2x8 pin header on 0.100 inch centers.

The pin out is described in the table below.

Pin	Description	Pin	Description
1	Request To Send	2	Transmit Data
3	Receive Data	4	Clear To Send
5	Ground	6	Ground
7	No Connection	8	No Connection

Table 7.3 Serial port pin out

8. LED INDICATORS

There are two LED indicators visible at the CPCI card bracket. They are marked with a legend on the bracket, and they are labeled on the PCB as L1 – L2 where L1 is at the top of the card.

The LEDs have the following meanings:

LED	LEGEND	Meaning
L1	P	PLX access Local bus
L2	D	DSP access Local bus
L3		User programmable Led #0
L4		User programmable Led #1
L5		5 Volt power supply indicator

Table 8.1 LED Description Table

