CHAPTER THREE

H.O.T. Works Board

The H.O.T. Works Board a complete hardware/software co-design. This chapter includes information on installation, product overview and getting started.

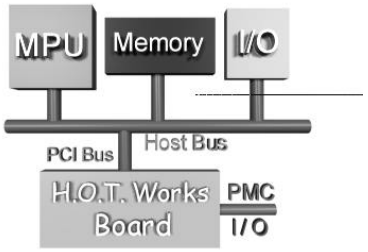
The H.O.T. Works Development System represents a major step in breaking the barriers between hardware and software; offering Hardware-On-Demand™.

CONTENTS

H.O.T. Works Board

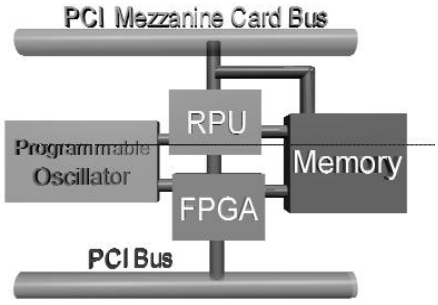
Overview

The H.O.T. Works Configurable Computer Board is a plug-in peripheral card for PCI standard slots. The H.O.T. Works Board interfaces to the host computer system through software drivers on the host side and an interface design loaded into the XC4000 FPGA on the H.O.T. Works Board. To the host computer, the H.O.T. Works Board acts as a closely coupled co-processor. This gives the standard computer architecture the added features of Run-Time Reconfigurable Hardware. At the right is a diagram of the relationship between the H.O.T. Configurable Computer Board and the host. The H.O.T. Works Board is a configurable computing system and contains various system features which enhance ease of use and flexibility. Below is a diagram of the H.O.T. Works architecture. This is a simplified representation of the various components on the H.O.T. Works Board (see Technically Speaking for details). The host and H.O.T. Works communicate over the PCI Bus through the XC4000 FPGA. The FPGA contains the PCI/RPU configuration interface based upon the LogiCORE PCI design by Xilinx. This configuration is automatically downloaded from a Boot Prom on the H.O.T. Works Board at system boot up. The FPGA may also be configured through the Xchecker connector, a Xilinx interface standard.



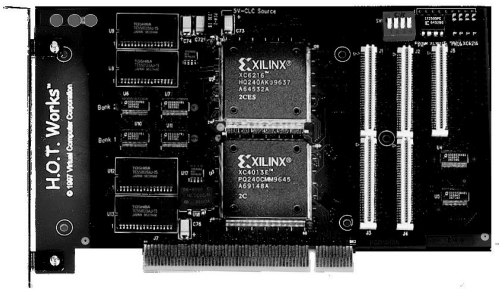
H.O.T. Board and the Host

Both, the XC6200 RPU and the XC4000 FPGA can send and receive data to/from the mezzanine connectors. The SRAM memory can send and receive data to/from RPU, FPGA and the mezzanine connectors. (see Memory - Modes of Operation). The on-board programmable oscillator can be utilized by both the RPU and FPGA. The A/D Converter tracks the electric current usage on the XC6200 RPU. It acts as a current watch dog or 'fuse' to protect the RPU.



Simple Block Diagram of H.O.T. Board

The H.O.T Board is designed for Rapid Product Development. The flexibility of the PCI Mezzanine option coupled with the power of reconfigurable logic devices offers an ideal multi purpose platform for Hardware/Software Codesigning and rapid prototyping.
The H.O.T. Works Board is a PCI standard based Development Tool providing a flexible architecture for implementing a wide variety of algorithms.



H.O.T. Works Board

FEATURES

PCI Based Development System
Up to 64K User Programmable Gates
IDD Monitoring of XC6200 RPU
Expansion via Mezzanine connectors to standard prototyping board
Device Driver implements standard Application Binary Interface
High speed Data Transfers to board memory
2 MB fast SRAM
Flexible clock generation for XC6200 RPU
Hardware implements XC6200 Computer Element Reference Design
First affordable FPGA Coprocessor platform for medium volume OEM applications

SPECIFICATIONS

General
PCI Bus Spec. R2.1

Physical
Standard PCI Single Slot
Short Form Factor

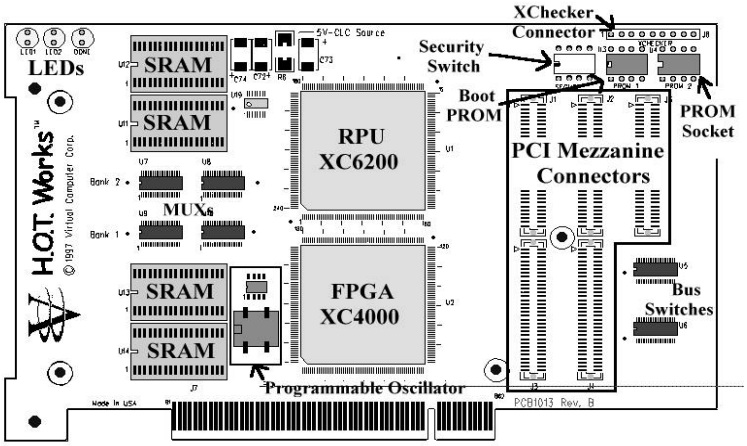
Electrical
Input Voltage (4.75 to 5.25 Volts)
Power Consm. (5 Volts @ TBD mA)

Environmental
Operating Temperature: (0° to 70° C)
Storage Temperature: (-55° to 150° C)

Timing
PCI Bus Interface (0-33MHz)

Module Interface
PCI Mezzanine Connector

Architecture

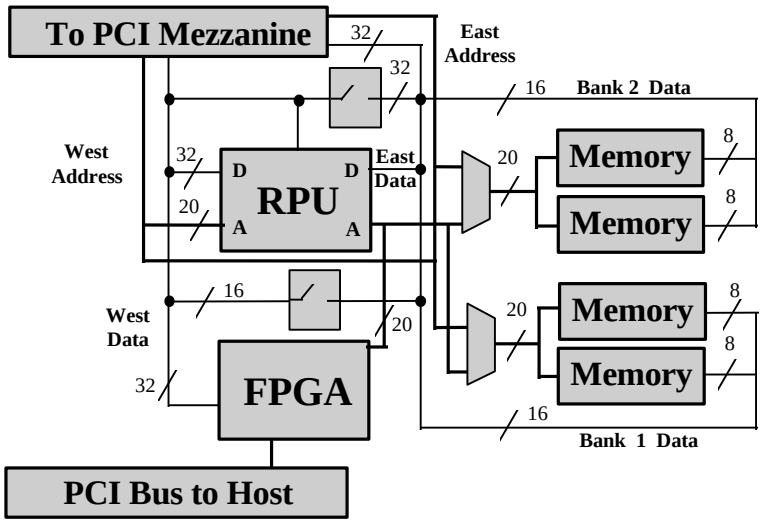


Component Layout

The H.O.T. Works Development System consists of a Xilinx 4013E and a compute element. The compute element consists of a Xilinx 6200 RPU, four 8-bit wide SRAM's and six bus controller chips to control data flow. The figure above shows the primary components of the architecture. The FPGA is used as the PCI bus interface. Approximately 50% of the chip is used for this function and the remaining area is used for card control logic. The FPGA is electrically and functionally 100% PCI compliant. For details of the PCI interface see the PCI LogiCORE product description which is available separately from Xilinx. The primary component of the compute element is the XC6200 RPU. The board architecture allows the XC6200 to be reconfigured through the PCI interface during run-time. The PCI interface provides direct access from the host PC to logic cells within the user's circuit. The output of any cell's function unit can be read and the flip-flop within any cell can be written through the PCI interface.

The compute element memory is organized into two banks. Each bank consists of a maximum of two 512K x 8 SRAM's. A bank of RAM can be accessed from either the PCI Interface or the XC6200. The banks of memory have two separate address busses and four read/write signals to control the RAMs individually. The development system provides a flexible architecture in order to implement a wide variety of algorithms. Multiple modes of operation can be set-up by selecting the muxes and bus switch in the desired manner. These modes are described in detail in the following section (Modes of Operation).

A 44-bit external data path is available to XC6200 Input Output Blocks (IOB's). This data path can be used to attach daughter boards for video I/O, network connections, or sensor I/O.

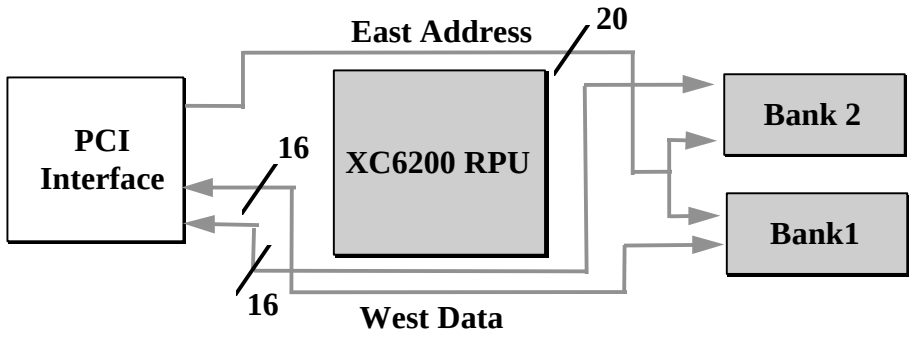


H.O.T. Works Board Bus Diagram

Memory - Modes of Operation

The on-board local memory is organized into two banks. Each bank consists of a two 512Kb x 8 SRAMs. A bank of RAM can be accessed from either the PCI Interface or the RPU (XC6200). The banks of memory have two separate address busses and four read/write signals to control the RAMs individually. Multiple modes of operation can be setup by selecting the muxes and bus switch in the desired manner. There are separate read/write and control signals for each memory bank. These modes are described in detail below.
The H.O.T. Works Board can be configured to operate with regards to memory reads and writes in many different configurations. Some of the possible configurations are outlined below.

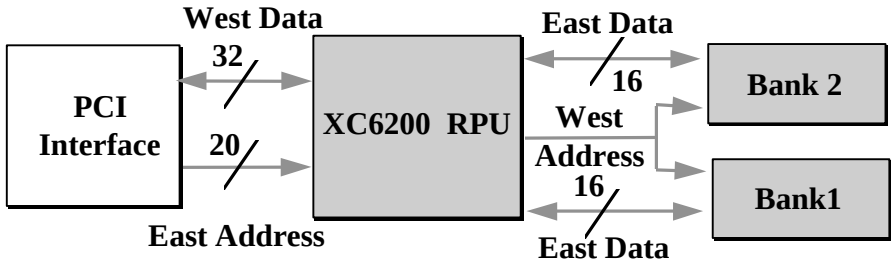
Mode 1 PCI to 32 bit RAM (Boot Default Mode)



Mode1 -- Memory Operation

- Single or Burst 32-bit read or write to local memory from PCI
- One address to both banks of memory

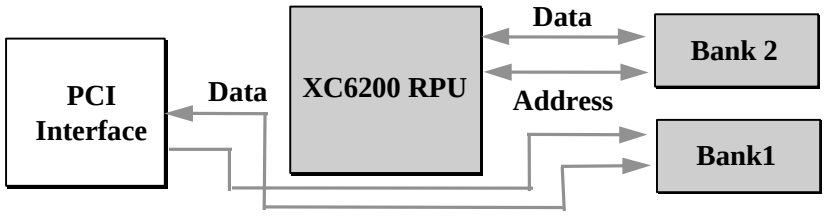
Mode 2 PCI to RPU, RPU to 32 bit RAM



Mode 2 -- Memory Operation

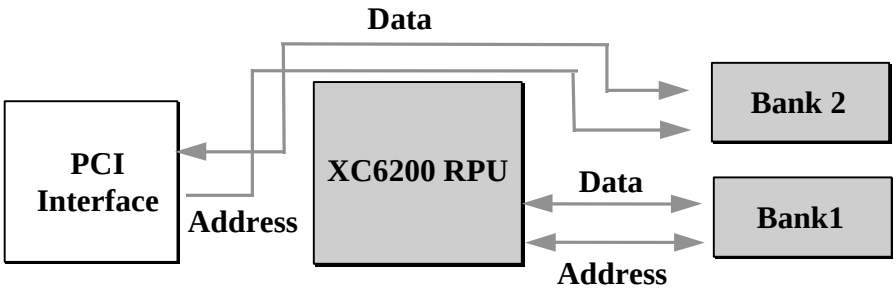
- 32-bit read or write to local memory from RPU
- One address to both banks of memory
- Concurrent PCI and local memory accesses
- Can reconfigure logic in 6200 while still processing data from local memories
- Can store data into 6200 registers and access data from local memories

Mode 3a & 3b PCI & RPU to 16 bit RAM



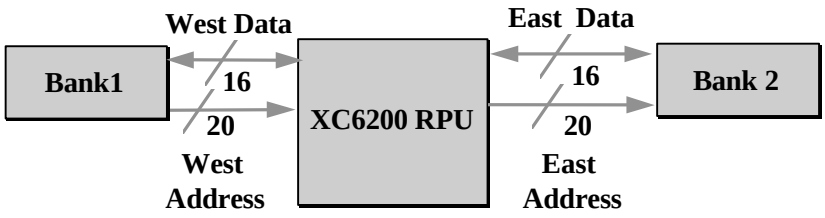
Mode 3a -- Memory Operation

- One bank of memory can be filled by the PCI interface and the other bank can be read from the 6216 so that
- real-time image processing can be performed
- One address can be generated from the FPGA XC4000 and one from the RPU XC6200



Mode 3b -- Memory Operation

Mode 4 16 bit RAM to RPU to 16 bit RAM



Mode 4 -- Memory Operation

- Use only when Isolate Bit =1
- Read or Write both banks simultaneously
- 2 addresses
- Only communication to XC6200 RPU through interrupts.

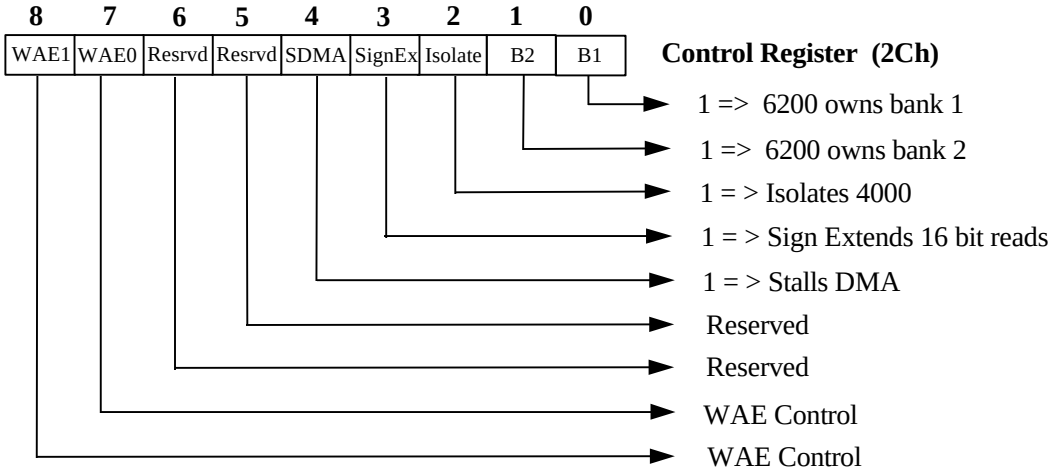
All Modes:

- Separate read/write and control signals for each memory chip.

Data Path Control Interface

The H.O.T. work Board's architecture allows for configurable data buses. These various buses are controlled by the**Control Register.**
WARNING: Great care must be taken setting these bits. One can destroy the XC4000 or the XC6200 by setting data and address paths incorrectly. You hardware warranty does not cover this type of damage. Please be careful.

Control Register (2Ch)



Bus Signal Behavior for 4000 FPGA

BCTL

2	1	0	WAE1#	WAE0#	Bank_ Sel1	Bank_ Sel0	WD_ [15:0]	WD_ [31:16]	EA_ [19:0]	BE1#	BE2#
0	**0**	**0**	1	1	0	0	D	D	D	0	0
0	**0**	**1**	1	0	0	1	D	T	D	0	1
0	**1**	**0**	0	1	1	0	T	D	D	1	0
0	**1**	**1**	0	0	1	1	T	T	D	1	1
1	**0**	**0**	BCTL8	BCTL7	1	1	T	T	T	0	0
1	**0**	**1**	BCTL8	BCTL7	1	1	T	T	T	0	1
1	**1**	**0**	BCTL8	BCTL7	1	1	T	T	T	1	0
1	**1**	**1**	BCTL8	BCTL7	1	1	T	T	T	1	1

Legend:

D = Driving T = Tristate

Bus Signal Behavior WARNINGS for 6200 RPU

BCTL

2	1	0		WD_ [15:0]	WD_ [31:16]	ED_ [15:0]	ED_ [31:16]	EA_ [19:0]	WA [19:0]	
0	**0**	**0**		X	X	X	X	X	OK	
0	**0**	**1**		X	DIS	X	OK	X	OK	
0	**1**	**0**		DIS	X	OK	X	X	OK	
0	**1**	**1**		DIS	DIS	OK	OK	X	OK	

1	**0**	**0**		A	B	A~	B~	OK	OK	
1	**0**	**1**		C	DIS	C~	OK	OK	OK	
1	**1**	**0**		DIS	D	OK	D~	OK	OK	
1	**1**	**1**		DIS	DIS	OK	OK	OK	OK	

Legend:

OK = OK to Drive
X = DO NOT NOT DRIVE
DIS = Disconnected

A,A~ = Only ONE to Drive
B,B~ = Only ONE to Drive
C,C~ = Only ONE to Drive

SRAM

The SRAM on the H.O.T. Works Board is mapped transparently into a region of PCI memory address space. The PCI address is mapped to the East Address bus `EA[19:0]` as follows:

```
EA[19:0] = PCI_ADDRESS[21:2]
```

The mapping of bytes within the PCI data bus `AD[31:0]` to the West Data bus `WD[31:0]` is as follows:

`AD[7:0]` corresponds to `WD[7:0]`
`AD[15:8]` corresponds to `WD[15:8]`
`AD[23:16]` corresponds to `WD[23:16]`
`AD[31:24]` corresponds to `WD[31:24]`

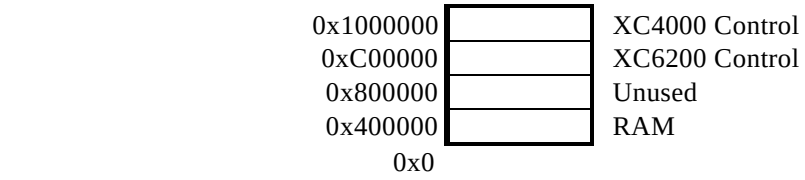
SRAM access by the PCI interface and the Xilinx 6200 is controlled by the bus switches which are enabled by the Bus Control register. The **Control Register**, can be used to effectively split the on-board SRAM into two independent 16-bit banks. Each bank may then be owned (i.e. controlled) by either the PCI Interface (4013) or the 6200. Control of a particular bank is defined to be ownership of the address, data and control signals of the two SRAM chips in the bank in question. During an SRAM access by the PCI interface, if the bus switches are enabled then the West Data bus `WD[31:0]` is connected to the East Data bus `ED[31:0]`.

When a bank is not controlled by the PCI interface, the data bus from the PCI interface to that bank is held in high impedance, allowing the 6200 to drive it. Conversely, the user of the H.O.T. Works Board must ensure that when the 6200 does not own a particular bank, it must not attempt to drive the data bus to that bank. When the 6200 owns both banks, writes to SRAM from the PCI bus will return undefined data. The arrangement of data in the host's memory, when performing a transfer to or from the SRAM, depends on the current value of the Bank Control Register. Bank1 controls data bits[15:0] and Bank2 controls data bits [31:16]:

Registers

The registers are accessed in two ways; through memory space and through I/O space.

Memory Map



Memory Space Assignments

PCI Address

bit 23	bit 22	Type	Function
0	0	R/W	Read/Write SRAM
0	1		Unused
1	0	R/W	6200 register
1	1	R/W	4000 register

Memory Offset	Type	Function
0x0	X	Used
0x4	WO	G1 Step Register
0x8	WO	Clockctrl Register
0x10	WO	Reset Register
0x14	WO	Clockgen Register
0x18	WO	Enter continuous clock mode
0xC	WO	Column Mode
0x1C	WO	Enter single clock mode
0x20	RO	Read 6200 IOBs
0x24	R	ADC Read Register
	W	ADC Threshold Register
0x28	WO	GCLK Step Register
0x2C	RW	Bank Control Register
0x30	R	Event Status Register
	W	Event Acknowledge Register
0x34	WO	Event Mask Register

I/O Space Assignments

I/O Offset	Type	Function
0x0	X	Used
0x4	WO	G1 Step Register
0x8	WO	Clockctrl Register
0x10	WO	Reset Register
0x14	WO	Clockgen Register
0x18	WO	Enter continuous clock mode
0xC	WO	Column Mode
0x1C	WO	Enter single clock mode
0x20	RO	Read 6200 IOBs
0x24	R	ADC Read Register
	W	ADC Threshold Register
0x28	WO	GCLK Step Register
0x2C	RW	Bank Control Register
0x30	R	Event Status Register
	W	Event Acknowledge Register
0x34	WO	Event Mask Register

PCI Interface

The 6200 register read/write mechanism is supported via the PCI interface, mapped into a region of PCI memory address space. A PCI address is mapped to the East Address bus `EA[19:0]` (and therefore the address bus of the 6200 register read/write interface) as follows:

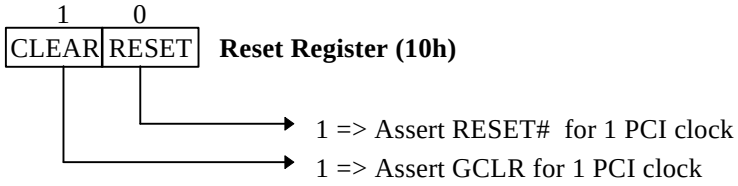
```
EA[19:0] = PCI_ADDRESS[21:2]
```

The mapping of bytes within the PCI data bus `AD[31:0]` to the West Data bus `WD[31:0]` is as follows:

`AD[7:0]` corresponds to `WD[7:0]`
`AD[15:8]` corresponds to `WD[15:8]`
`AD[23:16]` corresponds to `WD[23:16]`
`AD[31:24]` corresponds to `WD[31:24]`

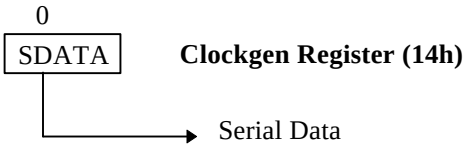
The Bank Control Register has no effect on 6200 accesses.

The 6200 may be reset, via its `RESET#` pin, or cleared, via its `GCLR` pin, by writing to the Reset Register. A write to this register can cause either, both, or none of these signals to be asserted for a single PCI clock cycle (therefore, >30ns):

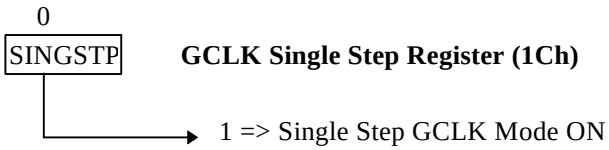


Clocking

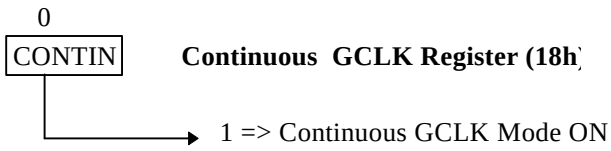
Clock signals for the XC6200 and for timing on board memory accesses are generated from an on board PLL based clock chip controlled from the XC4000E.



The 6200 can also be run in a single stepping mode where each clock transition is initiated by a host software access.

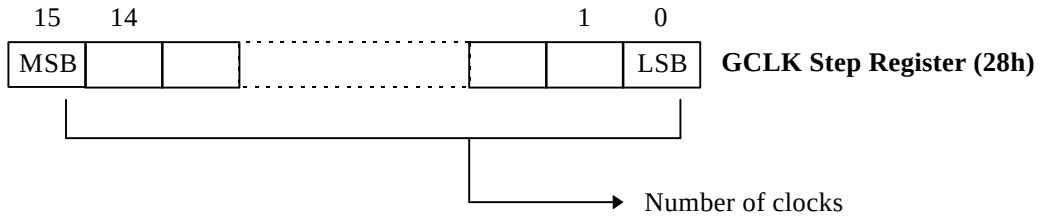


The 6200 can also be run in a continuous stepping mode.



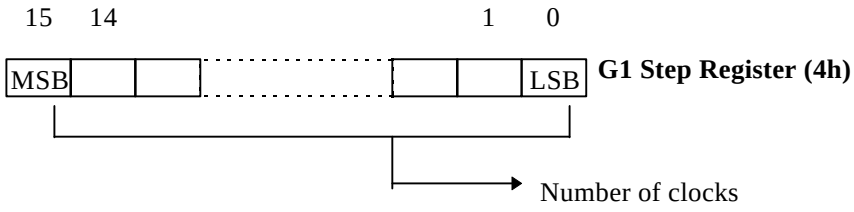
Programmable Clock Generator

The H.O.T. Works Board allows the 6200 to be run in what are termed `continuous clock mode' and `single clock mode'. In continuous clock mode, the global clock driving the 6200 is simply run continuously. In single clock mode, the global clock is stopped, and individual clock pulses or individual trains of clock pulses may be sent to the 6200, under control of the user's software. The PCI/6200 board is put into continuous clock mode by PCI reset, or by performing a write (with dummy data) to a specific location in I/O space. To put the board into single clock mode, a write is made to another location in I/O space. Once the board is in single clock mode, the user may send trains of clock pulses to the 6200 by writing values to the **Step Register**. The value written to the Step Register is an unsigned 16-bit integer, representing the number of clock pulses to send to the 6200.



At the end of a sequence of clock pulses, the **E_STEP** bit in the Event Status Register is set high, remaining high until acknowledged, and optionally, an interrupt can be caused on the host .

When a 6200 read is performed while the board is in single clock mode, three 6200 clock pulses are required. Two are required for the data read and the third for switching off the data bus. A 6200 write requires a minimum of two clock pulses (never depend on the number of clock cycles to clock your design. These accesses never cause an interrupt, or alter the Event Status Register.



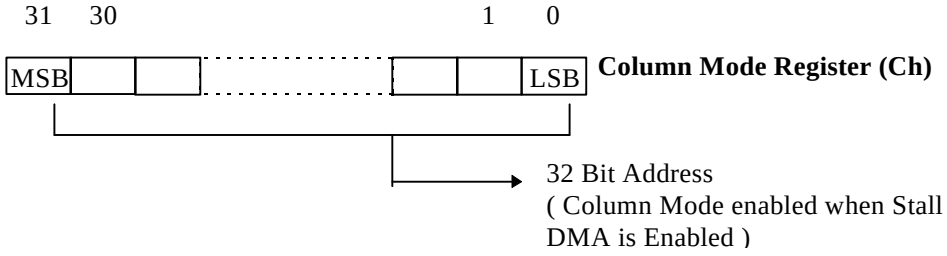
At the end of a sequence of clock pulses, the **E_STEP** bit in the Event Status Register is set high, remaining high until acknowledged, and optionally, an interrupt can be caused on the host .

Programmable Clock

As well as stopping and starting the 6200 clock, the PCI/6200 Evaluation Board permits any clock frequency in the range 1MHz to 100MHz to be sent to the XC6200 global clock. A single-bit register allows the configuration of the clock generator chip in bit-serial mode. Refer to the data sheet for ICD2053B for details on the bitstream format.

Column Mode

The Column Mode Register activates the option of writing data to a single column on the XC6200. This mode works only when the Stall DMA (Bit4) in the Bank Control Register = 1. All data is read and written to the address indicated by the Column Mode Register.



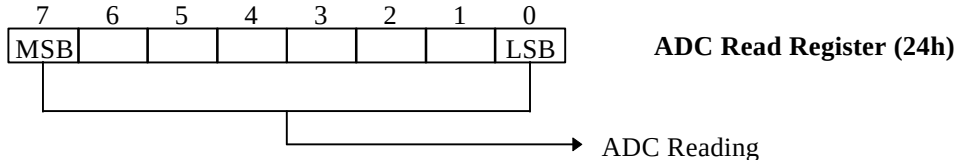
A/D Converter

The purpose of the A/D converter is to monitor power supply current (IDD) to the XC6200. If power consumption exceeds a preset value the XC4000E will reset (if enabled) the XC6200, clearing the faulty user configuration. Users can also monitor power consumption via software to help track down less dramatic or intermittent faults in their designs or to evaluate power consumption for different 6200 designs.

Circuitry on the H.O.T. Works Board measures the supply current drawn by the 6200. The **ADC Read Register** is an eight bit register in I/O space which returns the last reading made by the analogue-to-digital converter. The equation for converting ADC readings to a supply current values is as follows:

IDD = (ADC reading) * ~5.0 (mA)

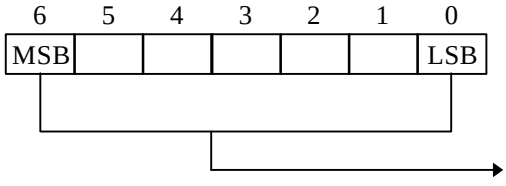
At all times, the last reading from the analogue-to-digital converter is available via the ADC Read Register as an unsigned eight bit value:



ADC Read Register (24h)

The host can be interrupted when the 6200 supply current exceeds a user-programmable threshold. In addition, the 6200 may be automatically reset when this event occurs. The **ADC Threshold Register** is written with a value consisting of:

- An unsigned eight bit threshold value, and
- This function is always enabled.



ADC Threshold Register (24h)

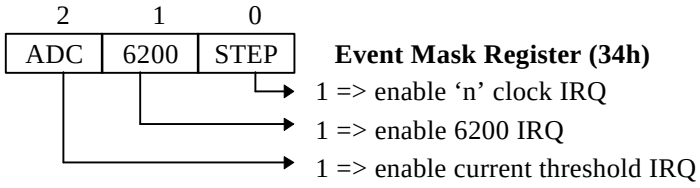
The threshold is exceeded when the ADC reading exceeds the threshold value written into the ADC Threshold Register. When exceeded, the **E_ADC** bit in the Event Status Register is set high, and remains high until acknowledged, even if the current falls below the threshold.

Interrupt Logic

There are three possible sources of interrupts from the H.O.T. Works Board:

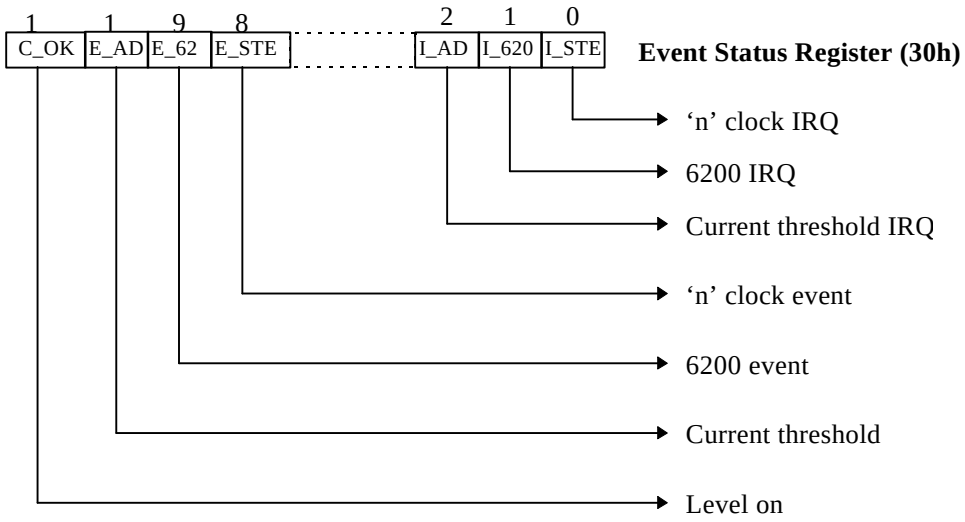
- The last clock pulse in a sequence has been sent to the 6200.
- 6200 requests an interrupt.
- 6200 supply current threshold is exceeded.

Each of the above events may be masked using the **Event Mask Register**. An event may cause an interrupt on the host when the corresponding bit in the Event Mask is a 1. When it is 0, the event may not cause an interrupt on the host. To disable interrupts entirely, the Event Mask Register should be written with 0.



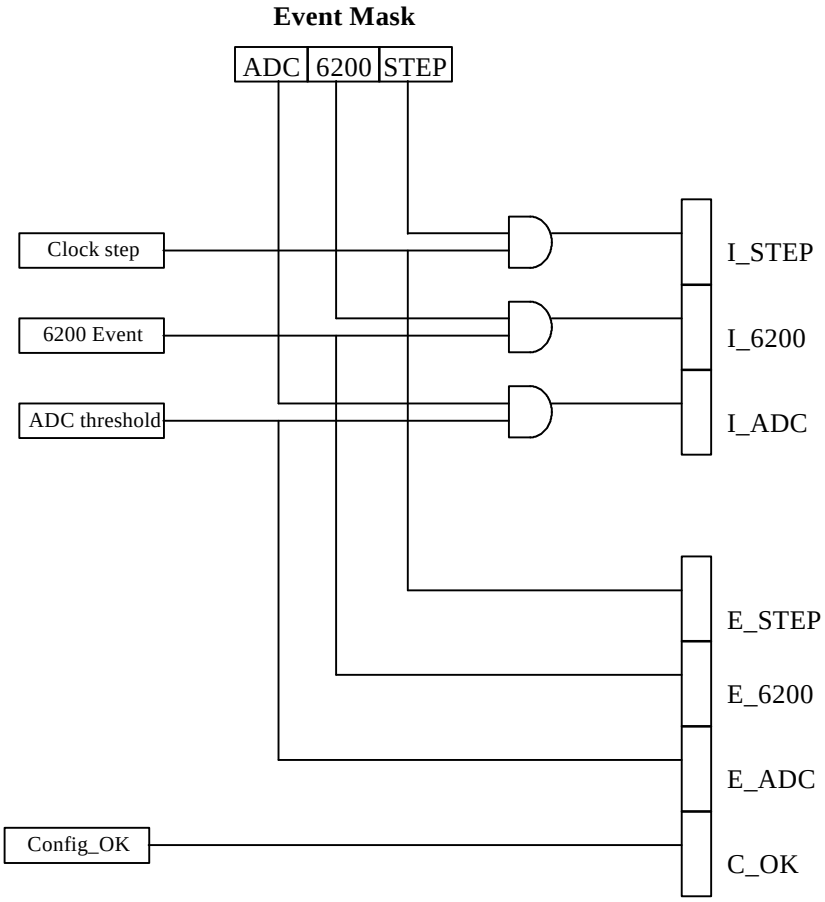
Whether masked or not, the event status is available via the **Event Status Register**. Even if masked by the Event Mask Register, when the 6200 requests an interrupt, for example, a corresponding bit will be set in the Event Status Register. In addition, the masked event status is available in the Event Status Register, which consists of the bitwise product of the set of detected events and the Event Mask.

The 6200 requests an interrupt on the host by pulsing the signal CON1 low if using GCLK OR/ CON2 low if using G1CLK. The PCI interface senses the falling edge of this signal, and sets the E_6200 bit in the Event Status Register. Provided that it is not masked by the Event Mask Register, the I_6200 bit will also be set.

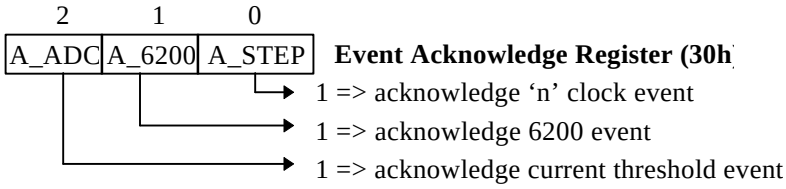


The bit C_OK is simply the level of the signal CONFIG_OK from the 6200, and the discussion below does not apply to this flag.

Those bits prefixed with E_ (to denote event) appear in the Event Status Register regardless of the Event Mask Register. Those bits prefixed with I_ (to denote interrupt) appear as 1 in the Event Status Register only if the corresponding event (E_) bit is 1 and the corresponding bit in the Event Mask Register is 1. The following diagram illustrates the relationship between event bits and interrupt bits:



Once an event has occurred, in order to clear the event condition in the Event Status Register, it is necessary to acknowledge it by performing an I/O write to the **Event Acknowledge Register**. Writing a 1 to a particular bit clears the event condition associated with that bit. A 0 written to a particular bit has no effect.



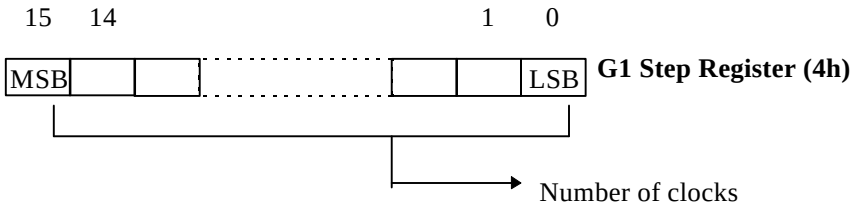
Expandability

Access to the complete range of on board signals and busses is provided through the five mezzanine connectors. Details of the mezzanine pinouts can be found later in this chapter.
User defined designs can be connected via the mezzanine connectors allowing complete configurability and testing of the design through the PCI bus. The mezzanine connectors can also be used to connect several H.O.T. Works Boards together to create a high performance system. Any Xilinx chips on the daughter boards can have their configuration loaded as slaves from the H.O.T. Works Board.

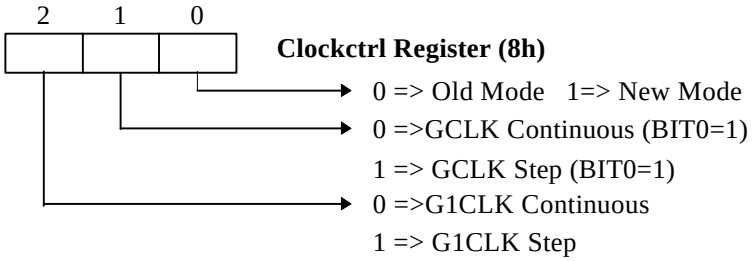
The XCHECKER connector supports the use of a Xilinx XCHECKER and XC4000 JTAG circuitry.

Register Summary

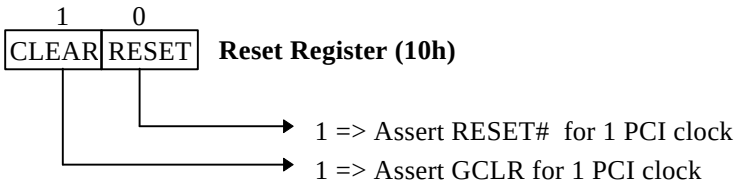
G1 Step Register



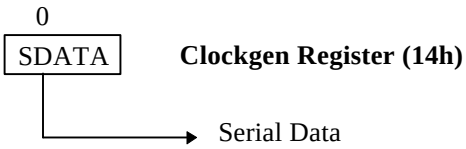
Clockctrl Register



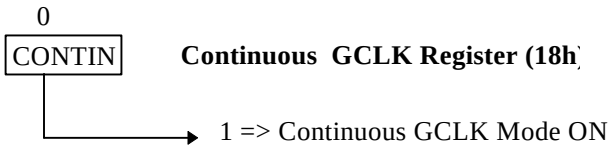
Reset Register



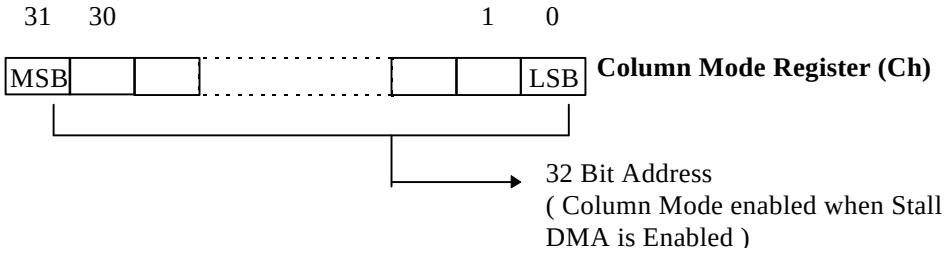
Clockgen Register



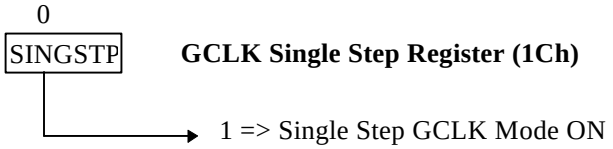
Continuous GCLK Register



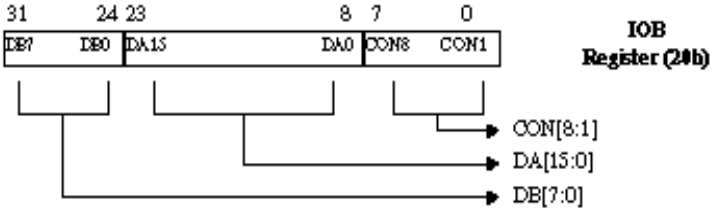
Column Mode Register



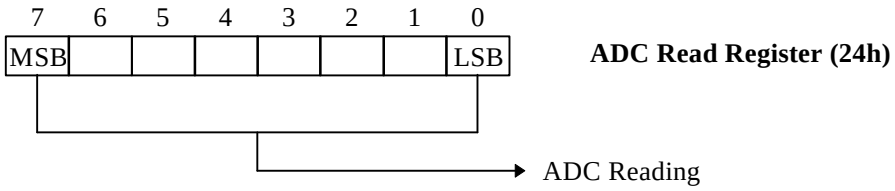
GCLK Single Step Register



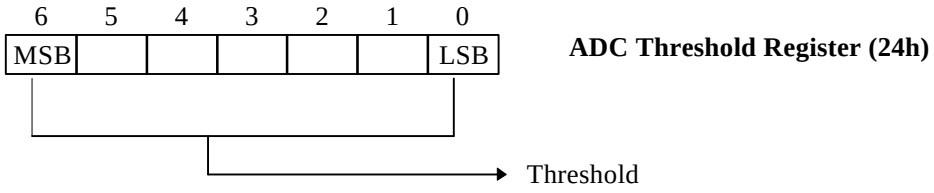
Read IOB Register



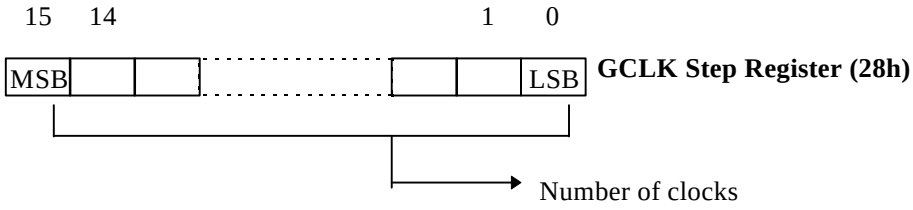
ADC Read Register



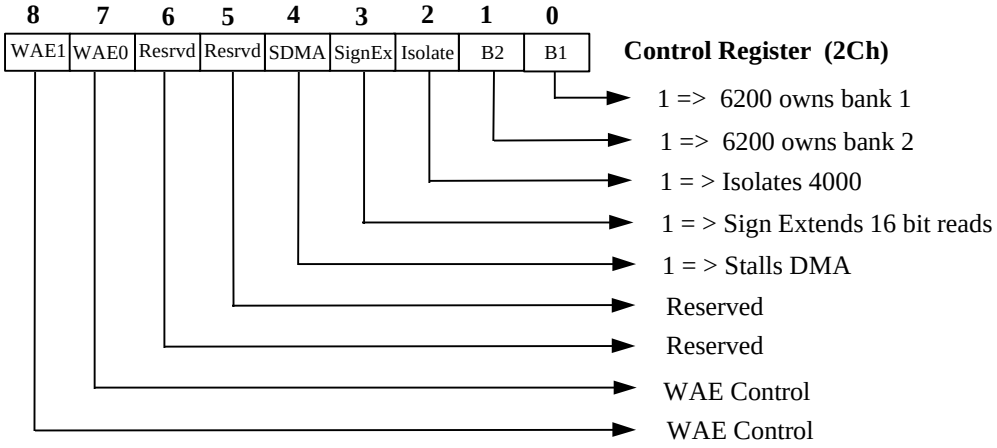
ADC Threshold Register



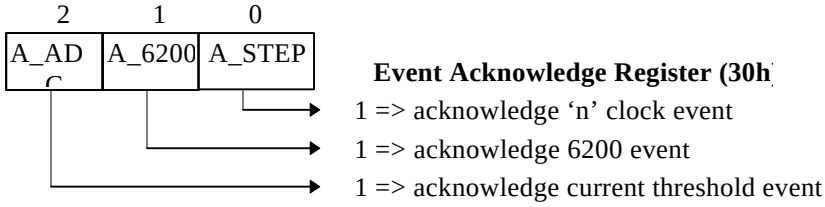
GCLK Step Register



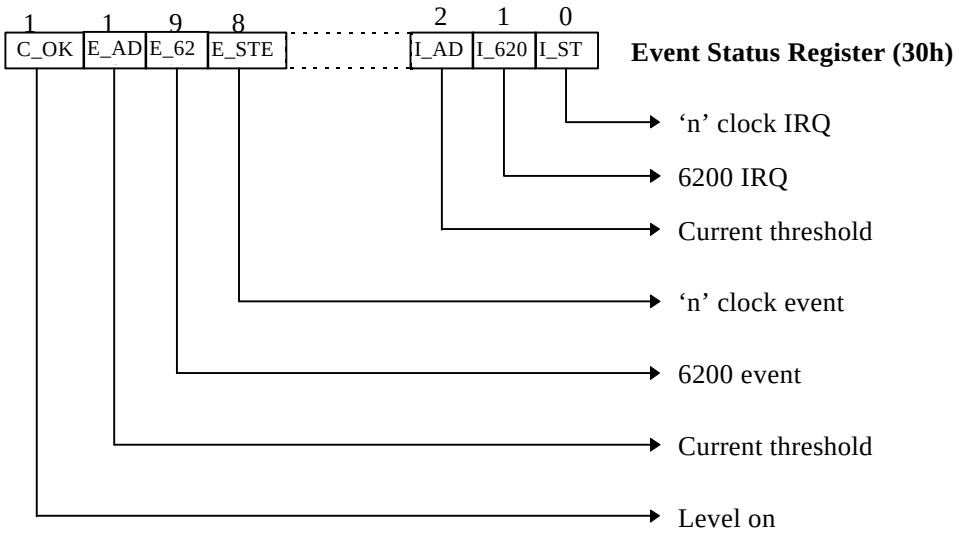
Bank Control Register



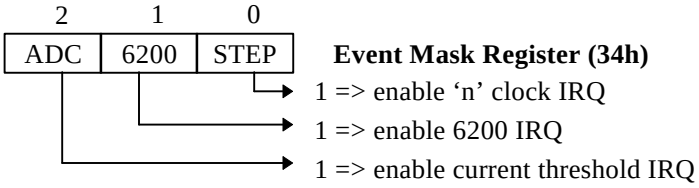
Event Acknowledge Register



Event Status Register



Event Mask Register



XC6200 Pin Descriptions

XC6200 Pinouts - West Side

PinOut	HQ240	XC6200DS
D0/W1	P60	WD_0
GND	P59	GND
W14	P58	WA_7
NC	P57	
D1/W3	P56	WD_1
W16	P55	WA_8
D2/W5	P54	WD_2
W18	P53	WA_9
D3/W7	P52	WD_3
W20	P51	WA_10
D4/W9	P50	WD_4
W22	P49	WA_11
W24	P48	WA_12
W26	P47	WA_13
D5/W11	P46	WD_5
GND	P45	GND
W28	P44	WA_14
W30	P43	WA_15
W32	P42	WA_16
D6/W13	P41	WD_6
VCC	P40	VCC
D16/W33	P39	WD_16
W34	P38	WA_17
GND	P37	GND
D7/W15	P36	WD_7
D17/W35	P35	WD_17
W36	P34	WA_18
D18/W37	P33	WD_18
D8/W17	P32	WD_8
W38	P31	WA_19

PinOut	HQ240	XC6200DS
VCC	P30	VCC
GND	P29	GND
D19/W39	P28	WD_19
D9/W19	P27	WD_9
D20/W41	P26	WD_20
D21/W43	P25	WD_21
D10/W21	P24	WD_10
D22/W45	P23	WD_22
GND	P22	GND
W48	P21	C29
W50	P20	C30
VCC	P19	VCC
D24/W49	P18	WD_24
D11/W23	P17	WD_11
D23/W47	P16	WD_23
D25/W51	P15	WD_25
GND	P14	GND
W56	P13	C31
D12/W25	P12	WD_12
W58	P11	C32
D26/W53	P10	WD_26
D27/W55	P9	WD_27
D13/W27	P8	WD_13
D28/W57	P7	WD_28
D14/W29	P6	WD_14
D29/W59	P5	WD_29
D15/W31	P4	WD_15
D30/W61	P3	WD_30
D31/W63	P2	WD_31
GND	P1	GND

XC6200 Pinouts - South Side

PinOut	HQ240	XC6200DS
VCC	P61	VCC
RdWr/S1	P62	RDWR#
CS/S3	P63	CS
W12/S0	P64	WA_6
OE/S5	P65	OE#
W10/S2	P66	WA_5
Reset/S7	P67	RESET#
W8/S4	P68	WA_4
W6/S6	P69	WA_3
W4/S8	P70	WA_2
W2/S10	P71	WA_1
W0/S12	P72	WA_0
S14	P73	6K_WE1#
S16	P74	6K_WE2#
GND	P75	GND
S26	P76	6K_WE3#
S33	P77	6K_WE4#
Serial/S9	P78	SERIAL#
Wait/S11	P79	WAIT
VCC	P80	VCC
S28	P81	6K-OE1#
S35	P82	6K-OE2#
GND	P83	GND
GClk/S13	P84	GCLK
S38	P85	6K-OE3#
GClr/S15	P86	GCLR
S37	P87	LED1
S40	P88	6K-OE4#
S39	P89	LED2
VCC	P90	VCC

PinOut	HQ240	XC6200DS
GND	P91	GND
G1/S17	P92	G1
G2/S19	P93 *	P_LCK *
S42	P94	C12
S41	P95	C13
E0/S50	P96	ED_0
E2/S52	P97	ED_1
GND	P98	GND
E4/S54	P99	ED_2
S45	P100	C14
VCC	P101	VCC
ShiftDOut/S21	P102	C15
SEData/S23	P103	SEDATA
E6/S56	P104	ED_3
S47	P105	C16
GND	P106	GND
E8/S58	P107	ED_4
S49	P108	C17
S51	P109	CON1
S53	P110	CON2
E10/S60	P111	ED_5
E12/S62	P112	ED_6
SECE/S25	P113	CON3
SEReset/S27	P114	CON4
S59	P115 *	C_CLK *
S61	P116 *	DONE *
SEClk/S29	P117 *	INIT# *
ConfigOK/S31	P118	C_OK
GND	P119	GND
S63	P120 *	PROG# *

* Pin 93 was G2 on orginal Xilinx Reference Design
* Pin 115 was CON5 on original Xilinx Reference Design
* Pin 116 was CON6 on original Xilinx Reference Design
* Pin 117 was CON7 on original Xilinx Reference Design
* Pin 120 was CON8 on original Xilinx Reference Design

XC6200 Pinouts - East Side

PinOut	HQ240	XC6200DS
VCC	P121	VCC
E14	P122	ED_7
A0/E1	P123	EA_0
E16	P124	ED_8
A1/E3	P125	EA_1
E18	P126	ED_9
A2/E5	P127	EA_2
E20	P128	ED_10
A3/E7	P129	EA_3
E22	P130	ED11
E24	P131	ED12
E26	P132	ED13
A4/E9	P133	EA_4
E28	P134	ED_14
GND	P135	GND
E30	P136	ED_15
E32	P137	ED_16
A5/E11	P138	EA_5
A16/E33	P139	EA_16
VCC	P140	VCC
E34	P141	ED_17
A17/E35	P142	EA_17
GND	P143	GND
A6/E13	P144	EA_6
E36	P145	ED_18
E37	P146	EA_18
E38	P147	ED_19
A7/E15	P148	EA_7
E39	P149	EA_19
VCC	P150	VCC

PinOut	HQ240	XC6200DS
GND	P151	GND
A8/E17	P152	EA_8
E40	P153	ED_20
A9/E19	P154	EA_9
E41	P155	C18
E42	P156	ED_21
E43	P157	ED_22
GND	P158	GND
A10/E21	P159	EA_10
E50	P160	ED_26
VCC	P161	VCC
E45	P162	ED_23
E52	P163	ED_28
A11/E23	P164	EA_11
E47	P165	ED_24
GND	P166	GND
E58	P167	C19
E49	P168	ED_25
A12/E25	P169	EA_12
E51	P170	ED_27
E53	P171	ED_29
E55	P172	ED_30
A13/E27	P173	EA_13
E57	P174	ED_31
A14/E29	P175	EA_14
E59	P176	C20
A15/E31	P177	EA_15
E61	P178	C21
E63	P179	C22
VCC	P180	VCC

XC6200 Pinouts - North Side

PinOut	HQ240	XC6200DS
VCC	P240	VCC
N1	P239	DA0
N3	P238	DA1
N0	P237	C0
N2	P236	C1
N4	P235	C2
N6	P234	C3
N5	P233	DA2
N7	P232	DA3
N12	P231	C4
N14	P230	C5
N16	P229	C6
N18	P228	C7
GND	P227	GND
N24	P226	C8
N26	P225	C9
N33	P224	DB0
N28	P223	C10
VCC	P222	VCC
N35	P221	DB1
N9	P220	DA4
GND	P219	GND
N11	P218	DA5
N38	P217	C11
N37	P216	DB2
N40	P215	C23
N39	P214	DB3
N13	P213	DA6
VCC	P212	VCC
GND	P211	GND

PinOut	HQ240	XC6200DS
N15	P210	DA7
N17	P209	DA8
N19	P208	DA9
N42	P207	C24
N41	P206	DB4
N44	P205	C25
GND	P204	GND
N43	P203	DB5
N21	P202	DA10
VCC	P201	VCC
N23	P200	DA11
N54	P199	C26
N45	P198	DB6
N56	P197	C27
GND	P196	GND
N47	P195	DB7
N58	P194	C28
N49	P193	DB8
N51	P192	DB9
N53	P191	DB10
N55	P190	DB11
N57	P189	DB12
N59	P188	DB13
N25	P187	DA12
N27	P186	DA13
N61	P185	DB14
N63	P184	DB15
N29	P183	DA14
GND	P182	GND
N31	P181	DA15

Mezzanine Connector Pinout

Mezzanine Pinout - J1

Pin#	SignalName	SignalName	Pin#
1	C25/TCK	-12V	2
3	GND	C21/INTA#	4
5	C22/INTB#	C23/INTC#	6
7	C14/BUSMODE1#	5V	8
9	C24/INTD#	PCI-RSVD*	10
11	GND	PCI-RSVD*	12
13 *	G1/CLK *	GND	14
15	GND	C12/GNT#	16
17	C28/REQ#	5V	18
19	5V/V(I/O)	DB15/AD[31]	20
21	DB12/AD[28]	DB11/AD[27]	22
23	DB9/AD[25]	GND	24
25	GND	C3/C-BE[3]#	26
27	DB6/AD[22]	DB5/AD[21]	28
29	DB3/AD[19]	5V	30
31	5V/V(I/O)	DB1/AD[17]	32
33	C4/FRAME#	GND	34
35	GND	C6/IRDY#	36
37	C10/DEVSEL#	5V	38
39	GND	C11/LOCK#	40
41	C30/SDONE#	C29/SBO#	42
43	C8/PAR	GND	44
45	5V/V(I/O)	DA15/AD[15]	46
47	DA12/AD[12]	DA11/AD[11]	48
49	DA9/AD[09]	5V	50
51	GND	C0/C-BE[0]#	52
53	DA6/AD[06]	DA5/AD[05]	54
55	DA4/AD[04]	GND	56
57	5V/V(I/O)	DA3/AD[03]	58
59	DA2/AD[02]	DA1/AD[01]	60
61	DA0/AD[00]	5V	62
63	GND	C31/REQ64#	64

*** Pin 13 was G2/CLK on original Xilinx Reference Design**

Mezzanine Pinout - J2

Pin#	Signal Name	Signal Name	Pin #
1	12V	C27/TRST#	2
3	C26/TMS	C19/TDO	4
5	C18/TDI	GND	6
7	GND	PCI-RSVD*	8
9	PCI-RSVD*	PCI-RSVD*	10
11	C15/BUSMODE2#	3.3V	12
13	C20/RST#	C16/BUSMODE3#	14
15	3.3V	C17/BUSMODE4#	16
17	PCI-RSVD*	GND	18
19	DB14/AD[30]	DB13/AD[29]	20
21	GND	DB10/AD[26]	22
23	DB8/AD[24]	3.3V	24
25	C13/IDSEL	DB7/AD[23]	26
27	3.3V	DB4/AD[20]	28
29	DB2/AD[18]	GND	30
31	DB0/AD[16]	C2/C-BE[2]#	32
33	GND	PCI-RSVD	34
35	C5/TRDY#	3.3V	36
37	GND	C7/STOP#	38
39	C9/PERR#	GND	40
41	3.3V	C_OK/C_OK	42
43	C1/C-BE[1]#	GND	44
45	DA14/AD[14]	DA13/AD[13]	46
47	GND	DA10/AD[10]	48
49	DA8/AD[08]	3.3V	50
51	DA7/AD[07]	PCI-RSVD	52
53	3.3V	PCI-RSVD	54
55	PCI-RSVD	GND	56
57	PCI-RSVD	PCI-RSVD	58
59	GND	PCI-RSVD	60
61	C32/ACK64#	3.3V	62
63	GND	PCI-RSVD	64

Mezzanine Pinout - J3

Pin#	Signal Name	Signal Name	Pin#
1	PCI-RSVD*	GND	2
3	GND	ED_16/C-BE[7]#	4
5	BANK1_SEL/C-BE[6]#	BANK2_SEL/C-BE[5]#	6
7	LED1/C-BE[4]#	GND	8
9	5V/V(I/O)	LED2/PAR64	10
11	WD_31/AD[63]	WD_30/AD[62]	12
13	WD_29/AD[61]	GND	14
15	GND	WD_28/AD[60]	16
17	WD_27/AD[59]	WD_26/AD[58]	18
19	WD_25/AD[57]	GND	20
21	5V/V(I/O)	WD_24/AD[56]	22
23	WD_23/AD[55]	WD_22/AD[54]	24
25	WD_21/AD[53]	GND	26
27	GND	WD_20/AD[52]	28
29	WD_19/AD[51]	WD_18/AD[50]	30
31	WD_17/AD[49]	GND	32
33	GND	WD_16/AD[48]	34
35	WD_15/AD[47]	WD_14/AD[46]	36
37	WD_13/AD[45]	GND	38
39	5V/V(I/O)	WD_12/AD[44]	40
41	WD_11/AD[43]	WD_10/AD[42]	42
43	WD_9/AD[41]	GND	44
45	GND	WD_8/AD[40]	46
47	WD_7/AD[39]	WD_6/AD[38]	48
49	WD_5/AD[37]	GND	50
51	GND	WD_4/AD[36]	52
53	WD_3/AD[35]	WD_2/AD[34]	54
55	WD_1/AD[33]	GND	56
57	5V/V(I/O)	WD_0/AD[32]	58
59	PCI-RSVD*	PCI-RSVD*	60
61	PCI-RSVD*	GND	62
63	GND	PCI-RSVD*	64

Mezzanine Pinout - J4

Pin#	Signal Name	Signal Name	Pin#
1	ED_7	ED_6	2
3	ED_5	ED_4	4
5	ED_3	ED_2	6
7	ED_1	ED_0	8
9	WA_19	WA_18	10
11	WA_17	WA_16	12
13	WA_15	WA_14	14
15	WA_13	WA_12	16
17	WA_11	WA_10	18
19	WA_9	WA_8	20
21	WA_7	WA_6	22
23	WA_5	WA_4	24
25	WA_3	WA_2	26
27	WA_1	WA_0	28
29	EA_19		30
31	EA_17	EA_18	32
33	EA_15	EA_16	34
35	EA_13	EA_14	36
37	EA_11	EA_12	38
39	EA_9	EA_10	40
41	EA_7	EA_8	42
43	EA_5	EA_6	44
45	EA_3	EA_4	46
47	EA_1	EA_2	48
49	ED_31	EA_0	50
51	ED_29	ED_30	52
53	ED_27	ED_28	54
55	ED_25	ED_26	56
57	ED_23	ED_24	58
59	ED_21	ED_22	60
61	ED_19	ED_20	62
63	ED_17	ED_18	64

Mezzanine Pinout - J5

Pin #	Signal Name	Signal Name	Pin #
1	6K_WE3#	6K_WE4#	2
3	6K_WE1#	6K_WE2#	4
5	6K_OE3#	6K_OE4#	6
7	6K_OE1#	6K_OE2#	8
9	CLKIN	DONE	10
11	GND	WAE1#	12
13	WAE0#	OE4#	14
15	OE3#	OE2#	16
17	OE1#	CS4#	18
19	CS3#	CS2#	20
21	CS1#	WE4#	22
23	WE3#	WE2#	24
25	WE1#	BE4#	26
27	BE3#	BE2#	28
29	BE1#	B2A_19*****	30 *****
31 *	B1A_19 *	SEDATA	32
33	WAIT	GCLR	34
35	CON8	SERIAL#	36
37	CON6	CON7	38
39	CON4	CON5	40
41	CON2	CON3	42
43	ED_15	CON1	44
45	ED_14	GND	46
47	GND	*****	48 *****
49 *****	*****	ED_13	50
51	ED_11	ED_12	52
53	ED_9	ED_10	54
55	RESET#	ED_8	56
57	CS	OE#	58
59	GND	RDWR#	60
61	5V	GND	62
63	GND	5V-CLC	64

*** Pin 30 & 31 was not used on original Xilinx Reference Design**

*** Pin 48 & 49 G1 & GCLK used on original Xilinx Reference Design**