OPERATING MANUAL

MODEL 7642

Multi−Channel Transceiver
PCI Board

Pentek, Inc.
One Park Way
Upper Saddle River, NJ 07458
(201) 818−5900
http://www.pentek.com

Manual Part No: 800.76420 Rev: A – November 19, 2008

Manual Revision History

Date	Revision	Comments
10/19/06	Preliminary	Initial release.
2/5/07		Sect 1.15, updated Power Specifications. **For Rev C boards**: Sect 2.2, added jumper block JB2, PCI Bus Mode.
2/12/07		Sect 2.2.1 corrected jumper JB1 factory default settings.
3/26/07		New Sect 7.4, added description of ADC to DDR Memory data packing.
4/13/07		Sect 1.15, corrected Analog Signal coupling to AC per KBCase 1320. Table 2–7 reversed differential P/N for each signal pair per KBCase 1321. Sect 4.7, added Figure 4–5 timing delays.
10/15/07		Sect 1.6, 1.15, 4.3, 6.12 added Option 101, DAC5687. Sect 1.15, corrected input clock spec to '1 to 300 MHz'. Sect 4.4, corrected DAC FIFO size. **For 7142 boards with PCI7142 revision date of 10/01/07 or greater:** Sect 4.5.5, 5.7.3, 5.21, 5.22, 5.23, added FPGA Load DMA description & registers.
11/6/07		Sect 1.4, introduced new FPGA terms: PCI FPGA (XC4VFX60) & Signal FPGA (XC4VSX55). Sect 1.12, updated baseline FPGA usage percentages. Moved Vendor Data Sheets to separate document, 809.7x420.
11/15/07		Sect 6.14.2, reversed BAR2 addresses of User In/Out FIFOs.
2/12/08		Added Option 100, XC4VFX100 (PCI) FPGA, & Option 110, XC4VLX100 (Signal) FPGA. Sect 2.3, corrected Pentek part # for JTAG PCB to 004.71402.
3/3/08		Sect 2.6.1, corrected full scale input to +10 dBm.
9/30/08		Sect 1.15, changed size specs to PCI card size. Sect 5.4, corrected register BAR addresses.
11/19/08	A	Manual released, Revision A

Warranty

Pentek warrants that all products manufactured by Pentek conform to published Pentek specifications and are free from defects in materials and workmanship for a period of one year from the date of delivery when used under normal operating conditions and within the service conditions for which they were furnished. The obligation of Pentek arising from a warranty claim shall be limited to repairing or at its option, replacing without charge, any product that in Pentek's sole opinion proves to be defective within the scope of the warranty. Pentek must be notified in writing of the defect or nonconformity within the warranty period and the affected product returned to Pentek within thirty days after discovery of such defect or nonconformity. Buyer shall prepay shipping charges, taxes, duties and insurance for products returned to Pentek for warranty service. Pentek shall pay for the return of products to buyer except for products returned from another country.

Pentek shall have no responsibility for any defect or damage caused by improper installation, unauthorized modification, misuse, neglect, inadequate maintenance, or accident, or for any product that has been repaired or altered by anyone other than Pentek or its authorized representatives.

The warranty described above is buyer's sole and exclusive remedy and no other warranty, whether written or oral, is expressed or implied. Pentek specifically disclaims fitness for a particular purpose. Under no circumstances shall Pentek be liable for any direct, indirect, special, incidental, or consequential damages, expenses, losses or delays (including loss of profits) based on contract, tort, or any other legal theory.

Copyrights

Trademarks

Printed in the United States of America.

Table of Contents

Table of Contents

Table of Contents

Page

Chapter 4: Model 7642 Resource Operation (continued)

Table of Contents

Page

Chapter 5: PCI7142 Registers

Table of Contents

Page

Chapter 5: PCI7142 Registers (continued)

Table of Contents

Page

Chapter 5: PCI7142 Registers (continued)

Chapter 6: Signal FPGA Registers

Table of Contents

Page

Chapter 6: FPGA Registers *(continued)*

Table of Contents

Page

Chapter 6: FPGA Registers *(continued)*

Table of Contents

Page

Chapter 6: FPGA Registers *(continued)*

Table of Contents

Table of Contents

Page

Chapter 6: FPGA Registers *(continued)*

Table of Contents

Table of Contents

Page

Chapter 7: Data Routing and Formatting

Table of Contents

Chapter 1: Introduction

1.1 General Description

The Pentek Model 7642 is a multi–channel, high–speed data converter suitable for connection to HF or IF ports of a communications system. Using the PCI Card module format, it includes four A/D converters and one D/A converter capable of bandwidths to 40 MHz and above.

The Model 7642 is a single PCI board that attaches directly to computer boards with PCI bus slots.

1.2 Features

- ❑ Complete software radio interface solution
- ❑ Four 125–MHz, 14–bit A/D converters
- ❑ One digital upconverter with one 500–MSPS, 16–bit D/A converter
- ❑ 768 MBytes of DDR2 SDRAM
- ❑ Xilinx® Virtex™–4 FPGAs
- ❑ Up to 2.0 seconds of delay or data capture at 100 MHz
- ❑ Dual timing buses for independent input and output clock rates
- ❑ LVDS clock/sync bus for multi–module synchronization
- ❑ 32 pairs of LVDS connections to the Virtex–4 FPGA for custom I/O on P4

1.3 Block Diagram

The following is a simplified block diagram of the Model 7642 digital receiver.

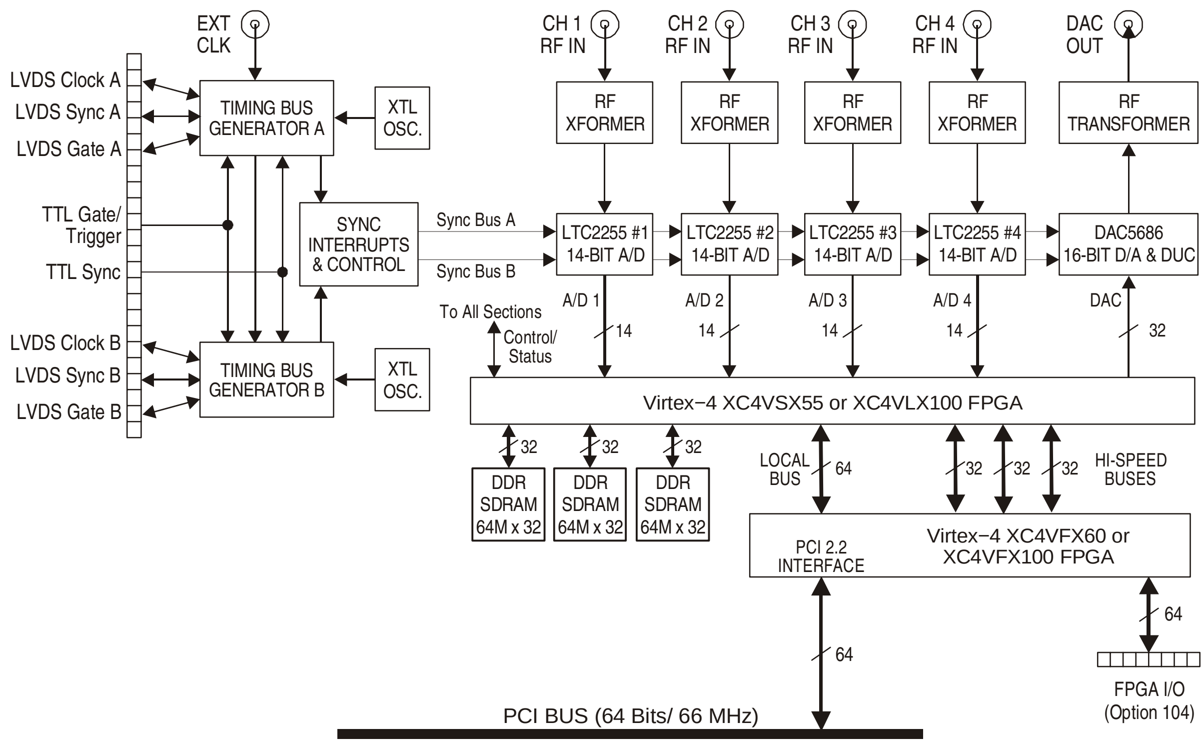


Figure 1−1: Model 7642 Block Diagram

The following defines the different uses of the term ‘Channel’ in this manual (refer also to the block diagram above).

- **Analog Input Channels** — There are four analog input channels, one for each analog RF input, identified as A/D 1, A/D 2, A/D 3, and A/D 4, corresponding to the RF inputs labeled CH 1 IN, CH2 IN, CH3 IN, and CH 4 IN on the front panel.

- **Analog Output Channel** — There is one digital upconverter output channel for the DAC5686, identified as DAC, corresponding to the RF output labeled DAC OUT on the front panel.

1.4 Principle of Operation

The Model 7642 is a complete software radio transceiver suitable for direct connection to HF or IF ports of a communications system. Using the popular PCI board format, it includes four A/D and one D/A converters.

The 7642 features a Xilinx® Virtex–4 XC4VSX55 FPGA (XC4VLX100 with Option 110) and a Xilinx Virtex–4 XC4VFX60 FPGA (XC4VFX100 with Option 100). The XC4VSX55 (or XC4VLX100) FPGA, identified as the "Signal FPGA" in this manual, can be pre–configured with one of a variety of optional IP cores to provide signal translation, pro–cessing, and time delay functions. In addition to pre–configured functions, user–cre–ated FPGA programming is supported by Pentek's GateFlow® Designer's Kit, Model 4953 Option 142. The XC4VFX60 (or XC4VFX100) FPGA, identified as the "PCI FPGA" in this manual, provides board interfaces including PCI. PCI FPGA I/O connections are provided through the optional PMC P4 connector (Option 104).

The 7642 includes a large 768–MByte block of DDR2 SDRAM. This memory is con–trolled by the Signal FPGA and is organized as three 256–MByte banks, 32 Mbyte deep by 32 bits wide. Separate banks (separate address and data per bank) allow simulta–neous access to all banks. This memory can be used as buffer memory when transfer–ring data between board resource or to off–board resources.

Four A/D converters provide input to a Signal FPGA, where the data can be formatted, processed or routed to board resources.

The D/A converter includes both interpolation filters and an upconverter stage capable of producing baseband I & Q and quadrature modulation analog output.

The 7642 includes dual onboard crystal oscillators for clocking, but can also accept external clocks through a front panel MMCX connector. The 7642 is equipped with a dual LVDS front panel clock and sync bus that can synchronize up to eight modules with built–in master/termination functions. The bus format is compatible with other Pentek PMC modules and will work with the Model 9190 Clock and Sync Distribution Amplifier for synchronizing up to 80 modules.

1.5 Analog to Digital Input Conversion

The Model 7642 is designed for a maximum input sampling frequency of 125 MHz. The front end accepts four full scale analog HF or IF inputs on front panel MMCX connectors at +10 dBm into 50 ohms with transformer coupling into four Linear Technology LTC®2255 14–bit 125 MHz A/D converters. The digital outputs are delivered into the Signal FPGA for signal processing or for routing to other module resources.

See Section 4.2 for a description of the analog to digital conversion operation.

1.6 Digital to Analog Output Conversion

The Model 7642 is designed with a maximum output sample rate of 320 MSPS in upconverter mode and 500 MSPS in D/A only mode. One Texas Instruments DAC5686 (DAC5687 with Option 101) digital upconverter and D/A accepts a baseband real or complex data stream from the Signal FPGA with signal band–widths up to 40 MHz.

When operating as an upconverter, it interpolates and translates real or complex baseband input signals to any IF center frequency between DC and 160 MHz. It delivers real or quadrature (I+Q) analog outputs through a 320–MSPS 16–bit D/A converter to a front panel MMCX connector at +4 dBm into 50 ohms. If translation is disabled, the DAC5686 acts as an interpolating 16–bit D/A with output sampling rates up to 500 MSPS.

See Section 4.3 for a description of the digital to analog output operation.

1.7 FPGA Digital Interfaces

The Model 7642 includes a Xilinx Virtex–4 XC4VSX55 FPGA (XC4VLX100 with Option 110), the Signal FPGA, which serves as a control and status engine with data and programming interfaces to each on–board resource including the A/D converters, DDR2 SDRAM memory, digital upconverter and D/A converter. The FPGA is factory programmed by Pentek to implement the standard data multiplexing, channel selection, data packing, gating, triggering, and memory control functions specified in this document.

A Xilinx Virtex–4 XC4VFX60 FPGA (XC4VFX100 with Option 100), the PCI FPGA, provides board interfaces including PCI and Serial I/O. This FPGA also includes two PowerPC® cores which can be used as local microcontrollers to create complete application engines. Option 104 adds the P4 PMC connector with 32 pairs of LVDS connections to the PCI FPGA for custom I/O. See Section 2.5 for a description of this optional interface.

The FPGAs can be reprogrammed from a PCI interface processor. Refer to Section 1.12, FPGA Configuration, for information about gate array configuration programming.

1.8 PCI Interface

The Signal FPGA is connected to the personal computer's PCI interface through a PCI Master/Slave interface (Standard PCI 2.2 Interface), the Pentek PCI7142. The PCI7142 is programmed in a PCI FPGA. This interface includes nine separate DMA controllers for efficient transfers to and from the module. Data widths of 32 or 64 bits and data rates of 33 or 66 MHz are supported.

Through this PCI interface, any PCI bus processor connected to the personal computer's PCI interface can receive data from any LTC2255. In addition, any PCI bus processor can send data to the DAC5686 for output. Any PCI Bus Master can control all programmable features on the board, the four LTC2255s, the DAC5686, the Gigabit interfaces, SDRAM memories, and all FPGA memory map registers.

1.9 Memory

Three independent banks of DDR2 (Double Data Rate 2) SDRAM are available to the Signal FPGA. Built–in memory functions include an A/D data transient capture mode with pre– and post–triggering; a D/A waveform generator mode; and an A/D data delay mode for applications like tracking receivers.

Custom user–installed functions within the Signal FPGA can take advantage of the SDRAM for many other purposes.

The SDRAMs are also available as a resource for the two PowerPC processor cores within the Signal FPGA.

See Section 4.6 for a description of the DDR2 memory operation.

1.10 Timing and Synchronization

Two independent internal timing buses (A and B) can provide either a single clock or two different clock rates for the input and output signal paths. Each timing bus includes a clock, a sync, and a gate or trigger signal. Signals from either Timing Bus A or B can be selected as the timing source for the A/Ds and the upconverter and the D/A. One external reference clock is accepted, and two internal clocks may be used for each timing bus.

A front panel 26–pin LVDS Clock/Sync connector allows multiple modules to be synchronized.

- In the slave mode, the 7642 accepts differential LVDS inputs that drive the clock, sync, and gate signals for the two internal timing buses.
- In the master mode, the LVDS bus can drive one or both sets of timing signals from the two internal timing buses for synchronizing multiple modules.

Up to seven slave 7642 modules can be driven from the LVDS bus master, supporting synchronous sampling and sync functions across all connected boards. Up to 80 boards may be synchronized with a Model 9190 Clock and Sync Generator.

See Section 4.7 for a description of timing and synchronization operation.

1.11 Interrupts

The Model 7642 has several maskable interrupt sources. PCI interrupts may be generated by A/D converter overload, DMA transfers, FIFO flags, transitions on Sync Bus gate or sync signals, clock loss on either Sync Bus, or a hardware over–temperature or power supply over–voltage.

Onboard sensors constant monitoring of critical voltages and temperatures on the Model 7642 PCB. The sensors are programmable for voltage and temperature limits. If the voltage/temperature fall outside of the set limits, an interrupt can be generated.

See Section 4.8 for a description of the interrupt operation.

1.12 FPGA Configuration

The Model 7642 includes a Xilinx Virtex–4 XC4VSX55 (or XC4VLX100) Signal FPGA and a Virtex–4 XC4VFX60 (or XC4VFX100) PCI FPGA. The 7642 is shipped with a default set of logic functions for the FPGAs, on JTAG–programmable serial EEPROMs. Upon power–up, this set of default functions is loaded into the FPGA. The baseline functionality consumes about 42.8% of the Signal FPGA, and about 46.4% of the PCI FPGA.

The Signal FPGA can be configured with user functions in one of several ways:

- The default Signal FPGA configuration is loaded from the serial EEPROM. This is the mode that the board will come up in at power up. Reprogramming may also be forced at any time through the PMC/PCI interface (see Section 5.7).

- The second method is byte wide upload from a PCI interface processor. This is done by the PCI bus processor selecting this mode, then forcing a reprogram and writing the configuration data one byte at a time to the Signal FPGA configuration data register (see Section 5.7). In this way, the Signal FPGA can be rapidly configured for the user's requirement.

- The third method is to configure the Signal FPGA directly via JTAG with a Xilinx Parallel III or MultiLINX™ cable.

- The last method is to use the serial EEPROM configuration method, but the user may overwrite the default programming by reprogramming the PROM at the JTAG interface.

  NOTE: This last method will permanently overwrite the default configuration supplied by Pentek. The default configuration is supplied with the available GateFlow FPGA Design Kit so that it can be restored if necessary.

The user may modify, add to, or replace the default logic functions on the EEPROMs with user–defined functions. In this way, additional custom hardware signal process–ing can be accomplished. Pentek has an available GateFlow® FPGA Design Kit that provides information that a user requires to modify the programmable logic functions for either FPGA. Pentek offers this as a separate development package, Model 4953 Option 142. Contact Pentek at (201) 818–5900 for details about this package.

The Model 7642 provides I/O connections from 64 FPGA spare pins through a PMC connector. These pin connections are identified in Section 2.5.

1.13 Board Support Software

The Pentek GateFlow® FPGA Design Kit (Model 4953 Option 142) facilitates user–installed FPGA functions using the Xilinx ISE Foundation tool suite. The FPGA Design Kit allows the user to configure FPGAs for implementing preprocessing functions such as convolution, framing, pattern recognition or decompression. The Pentek GateFlow IP Core Libraries include high–performance receivers, FFTs and pulse compression algorithms. Factory installed IP Cores are available.

Several Pentek software packages are available to speed development tasks for the Model 7642. Pentek's Model 4999 Option 142 ReadyFlow® Board Support Libraries contain a set of C–language routines for the Model 7642. Refer to the ReadyFlow documentation for the Model 7142, Pentek part #801.71420.

The following available software drivers allow high–level programming for various workstation platforms. Refer to the software documentation indicated for each.

- Pentek Model 4994 Option 142 Linux® Driver, Pentek document #806.71400.
- Pentek Model 4995 Option 142 Windows® Driver, Pentek document #805.71400.
- Pentek Model 4996 Option 142 VxWorks® Driver, Pentek document #803.71400.

1.14 Supporting Documentation

In addition to the operating instructions provided in this manual, copies of the following datasheets for the programmable devices on the Model 7642 are provided in the Pentek Model 7x42 Series Supplemental Manual, part number 809.7x420. This document is available on the CD provided by Pentek, in file **`8097x420.pdf`** (PDF format).

- ❑ Linear Technology Corporation – LTC2255/LTC2254 14–Bit ADCs Data Sheet
- ❑ Texas Instruments – DAC5686 and DAC5687 Digital–to–Analog Converter Data Sheets
- ❑ Analog Devices, Inc. – ADM1024 System Hardware Monitor Data Sheet
- ❑ National Semiconductor Corporation – LM83 Temperature Sensor Data Sheet

1.15 Specifications

Analog Signal Inputs

Quantity:	Four signal inputs, one per A/D converter
Connector Type:	Front panel MMCX connectors
Input Type:	Single–ended, non–inverting
Full Scale Input:	+10 dBm
Coupling:	AC
Input Impedance:	50Ω
Analog Input RF Transformers:	
Quantity:	Four (one per input channel)
Type:	Coilcraft WBC1–1TLB
3dB Passband:	250 kHz to 700 MHz
Insertion Loss:	0.58 dB max.

Analog Signal Output

Quantity:	One signal output
Connector Type:	Front panel MMCX connectors
Output Type:	Single–ended, non–inverting
Coupling:	AC
Output Impedance:	50Ω
Standard Output Transformer:	
Type:	Coilcraft WBC4–6TLB
3dB Passband:	300 kHz to 700 MHz
Output Power:	+4 dBm into 50 ohms
Option 002 Output Transformer:	
Type:	TBD

External Clock Inputs

Quantity:	One clock input
Connector Type:	Front panel MMCX connectors
Signal Type:	Sine wave, 45–55% duty cycle
Frequency Range:	1 to 300 MHz *
Voltage Range:	0 to +10 dBm
Coupling:	AC coupled
Input Impedance:	50Ω

* Limited by ADC or DAC characteristics (see next page)

Internal Clocks

Sync Bus A	125 MHz crystal oscillator
Sync Bus B	100 MHz crystal oscillator

Sync/Gate Bus Inputs/Outputs

Connector Type:	Front panel 26–pin connector
Signals:	Clock In/Out: 4 pins (two LVDS pairs, Clk A & Clk B) Sync In/Out: 4 pins (two LVDS pairs, Sync A & Sync B) Gates In/Out: 4 pins (two LVDS pairs, Gate A & Gate B) TTL Sync In: 1 pin (single–ended) TTL Gate/Trigger In: 1 pin (single–ended)

1.15 Specifications (continued)

Analog/Digital Converters

Quantity:	Four
Device:	Linear Technology LTC2255 (see Section 1.14)
Sampling Rate:	1 MHz to 125 MHz
Resolution:	14 bits
A/D Data Reduction:	Data from the A/Ds can be written directly into the Signal FPGA at a rate equal to the A/D clock decimated by any value between 1 and 4096
Clock Source:	Clock from Sync bus A or B, each bus clock source can be from onboard crystal oscillator, front panel external clock, or LVDS clock, selectable via software

Digital/Analog Upconverter

Quantity:	One
Device:	
Standard:	Texas Instruments DAC5686
Option 101:	Texas Instruments DAC5687 (see Section 1.14)
Sampling Rate:	320 MSPS in upconvert mode, 500 MSPS in D/A only mode
DAC Resolution:	16 bits
Input Data Rate:	160M 32–bit words per second
Coupling:	Transformer coupled
Output IF:	DC to 160 MHz *
Output Signal:	Analog, real or quadrature
Interpolation filters:	2x, 4x, 8x, 16x (16x not available in DAC5687)
Clock Source:	Clock from Sync bus A or B, each bus clock source can be from onboard crystal oscillator, front panel external clock, or LVDS clock, selectable via software

* Since maximum data bandwidth is 160 MHz, synthesizing IF signals at this limit is not practical

Field–Programmable Gate Arrays

Signal FPGA

Device:	
Standard:	Xilinx Virtex–4 XC4VSX55
Option 110:	Xilinx Virtex–4 XC4VLX100
Configuration:	Factory programmed by Pentek: A/D, D/A, DDR2 Memory, FIFOs

PCI FPGA

Device:	
Standard:	Xilinx Virtex–4 XC4VFX60
Option 100:	Xilinx Virtex–4 XC4VFX100
Configuration:	Factory programmed by Pentek: PCI7142 PCI interface, DMA High–speed interfaces (Option 5xx)
FPGA I/O (Option 104):	96–pin DIN connector, with 64 I/O lines to FPGA

1.15 Specifications (continued)

DDR2 SDRAM memory	
Size:	768 MBytes, organized as three banks of 64 M x 32
Interface:	Interfaced to the Signal FPGA
Bus Width:	32 bits
Clocks	
Quantity:	Two, one per Sync Bus
Clock Sources:	Separately selectable from external clocks or onboard crystal oscillator
External Clocks:	Front panel LVDS Sync Bus (one Clock input) or front panel MMCX connector inputs
Internal Clocks	125 MHz crystal oscillator, Sync Bus A 100 MHz crystal oscillator, Sync Bus B
Gates	
Quantity:	Two, one per Sync Bus
Gate Sources:	Separately selectable from external or internal gates
External Gates:	Front panel LVDS Sync Bus (two Gate inputs) or TTL Gate/Trigger input
Internal Gates:	Generated from programmable registers (one for each gate)
Gate Polarity:	Programmable polarity for external gates
Gate Disable:	Each gate can be disabled; when gate is disabled, writes default to enabled
Triggering:	Each gate can be programmed as a trigger, programmable trigger length up to 16,383 FIFO writes
Syncs	
Quantity:	Two, one per Sync Bus
Sync Source:	Separately selectable from external or internal syncs
External Sync:	Front panel LVDS Sync Bus (two Sync inputs) or TTL Sync input
Internal Sync:	Generated from programmable register
Sync Pulse Width:	2 clock cycles, minimum
Temperature and Voltage Sensors	
Temperature	
Quantity:	Three PCB temperature sensors (see Section 6.3.1)
Controller:	National Semiconductor LM83 (see Section 1.14)
Interface:	TWSI Bus
Voltage	
Quantity:	Eight PCB voltage sensors (see Section 6.3.2)
Controller:	Analog Devices ADM1024 (see Section 1.14)
Interface:	TWSI Bus

1.15 Specifications (continued)

PCI Interface

Device: Pentek PCI7142 Core (PCI FPGA)
Compliance: Standard PCI 2.2 Interface
Support for 3.3V signal logic only
PCI Bus: 64 bit, 66 MHz (can support 32 bit and/or 33 MHz)
Local Bus: 64 bit, 66 MHz
Data Transfer Modes: Direct Slave Mode and DMA Master/Slave Mode,
9–channel demand–mode and chaining controller
Control: Any PCI bus master (such as a PCI interface processor) can control both sync buses, the LTC2255s, DAC5686, the FPGAs, and all onboard registers.

PMC to PCI Adapter

Device: Technobox™ Model 5102
PCI Bridge: Intel 31154
PCI Bus Interface: 32 or 64 bit, PCI 33/66 MHz, PCI–X 33/66/100/133 MHz
PMC Interface: 32/64 bit, 33/66/100 MHz

Estimated Power Consumption

PCI Adapter
With no PMC: (TBD)
7142 PMC:
Current Draw:

+3.3V	+5V	+12V	–12V
1.35A	2.5A	0.2A *	NC

* 12V only connected to voltage test circuit

PMC Power: 16.95 Watts

Note: Power estimated with FPGA base circuitry. Power consumption increases when custom user FPGA designs are added. Different aspects of the FPGA design contribute to power consumption such as clock speed, number of logic slices used, number of DSP multipliers used, and amount of block RAM used.

Physical

Dimensions: Standard PCI card
Depth: 149.0 mm (5.87 in)
Height: 76.2 mm (3.0 in)
Weight: 127.6 grams (4.5 oz)

Environmental

Operating Temperature: 0° to 50°C
Storage Temperature: –20° to 90°C
Relative Humidity: 0 to 95%, non–condensing

Chapter 2: Installation and Connections

2.1 Inspection

After unpacking, inspect the unit carefully for possible damage to connectors or components. If any damage is discovered, contact Pentek immediately at (201) 818–5900. Please save the shipping container and packing material in case reshipment is required.

See Figure 2–1, on the next page, for a drawing of the Model 7642 assembly as shipped.

2.2 Jumper and Switch Settings

The following subsections cover user operating parameters that are set by shorting jumpers or dip switches on the Model 7642.

The term jumper (or jumper block) refers to a group of two or more pins on a circuit board that may be connected in pairs by a shorting jumper to set or change an operating characteristic of the board. A pair of pins connected in this manner are referred to as installed, or shorted. A pair of pins that are not connected are called removed, or open. The shorting jumpers used on the Model 7642 are for 0.020" (0.51 mm) square pins spaced on 0.079" (2.00 mm) centers. These jumpers are DuPont™ part number 86730–001, or equivalent. Pentek’s part number for these jumpers is 356.00010.

As shipped from the factory, several jumpers are installed in default positions on your board. The default operating parameters they select may or may not meet your requirements. Before installing your Model 7642, please review the following subsections to determine whether you need to change any of these settings.

Note

The user should not change jumpers not identified in these figures—these are reserved for factory test and setup purposes only.

The Model 7642 assembly consists of two modules, a Pentek 7142 PMC module and a PMC to PCI adapter board (Technobox™ Model 5012). See Section 2.2.2 for description of the jumper settings on the 3674 PCI adapter board.

As shipped, the 7142 PMC module is mounted onto the PCI adapter. Note that all jumpers on the PMC are located on the component side of the PMC, and are not accessible in this shipped configuration. See Section 2.2.1 for the jumper settings for the 7142 PMC. If you need to access this jumper on the 7142 PMC, you must first remove the PMC module from the PCI adapter board, as described in Section 2.2.3.

2.2 Jumper and Switch Settings (continued)

The following figure illustrates the complete Model 7642 assembly as shipped, with the 7142 PMC module mounted onto the PMC to PCI adapter.

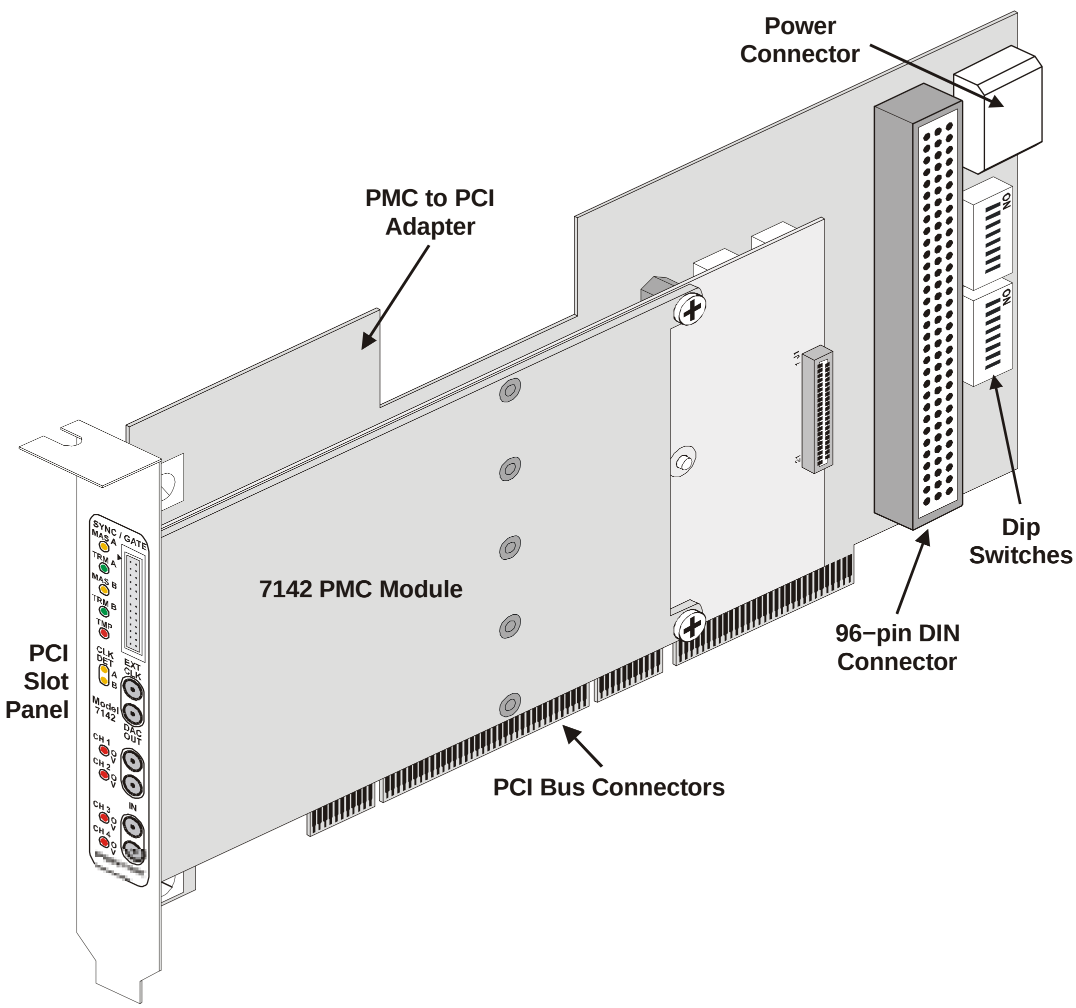


Figure 2−1: Model 7642 Assembly

Note that the Model 7642 provides a 96−pin DIN connector at the edge of the PCI adapter. The 'A' and 'C' rows of this connector are routed to the 64−pin user I/O connector (**P4**) on the PMC module. This is used with Model 7642 Option 104, described in Section 2.5.

2.2 Jumper and Switch Settings (continued)

2.2.1 7142 PMC Module Jumpers

An assembly drawing of the component side of the 7142 PCB is provided below, showing the jumpers and PMC connectors on the board.

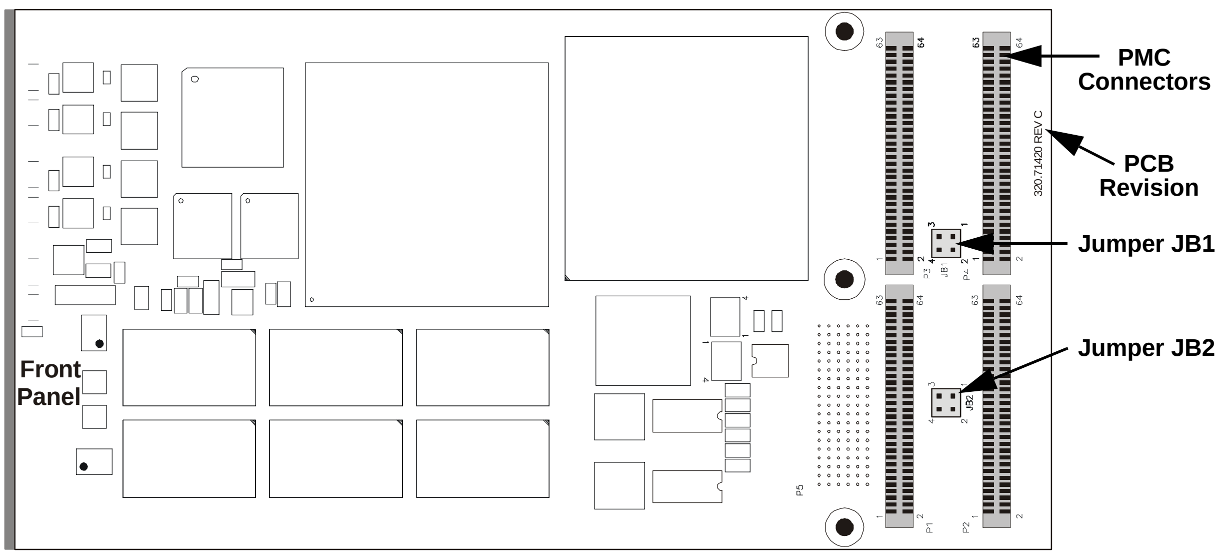


Figure 2–2: Model 7142 PCB Assembly Drawing, Component Side

Jumper blocks **JB1** and **JB2,** located on the component side of the 7142 PCB (see Figure 2–2 above), are for factory use only and should not be changed by the user.

Table 2–1: Factory Default Jumpers

Jumper	Pins 1–2	Pins 3–4	Default Function
JB1	Removed *	Removed *	Factory Use only
JB2	Removed *	Installed *	Factory Use only

* Factory Default Setting
NOTE: JB2 is present only on boards identified as **Rev C** or greater.

DO NOT change these Factory Default jumper settings!

2.2 Jumper and Switch Settings (continued)

2.2.2 PMC to PCI Adapter Switches

The PMC to PCI Adapter has two 8–position DIP switches that the user can set, as described in the following subsections. These are located next to the DIN connector on the right side of the board, as shown in Figure 2–3 below, and are described in the following subsections.

Note that you can access these DIP switches while the PMC module is attached to the PCI adapter.

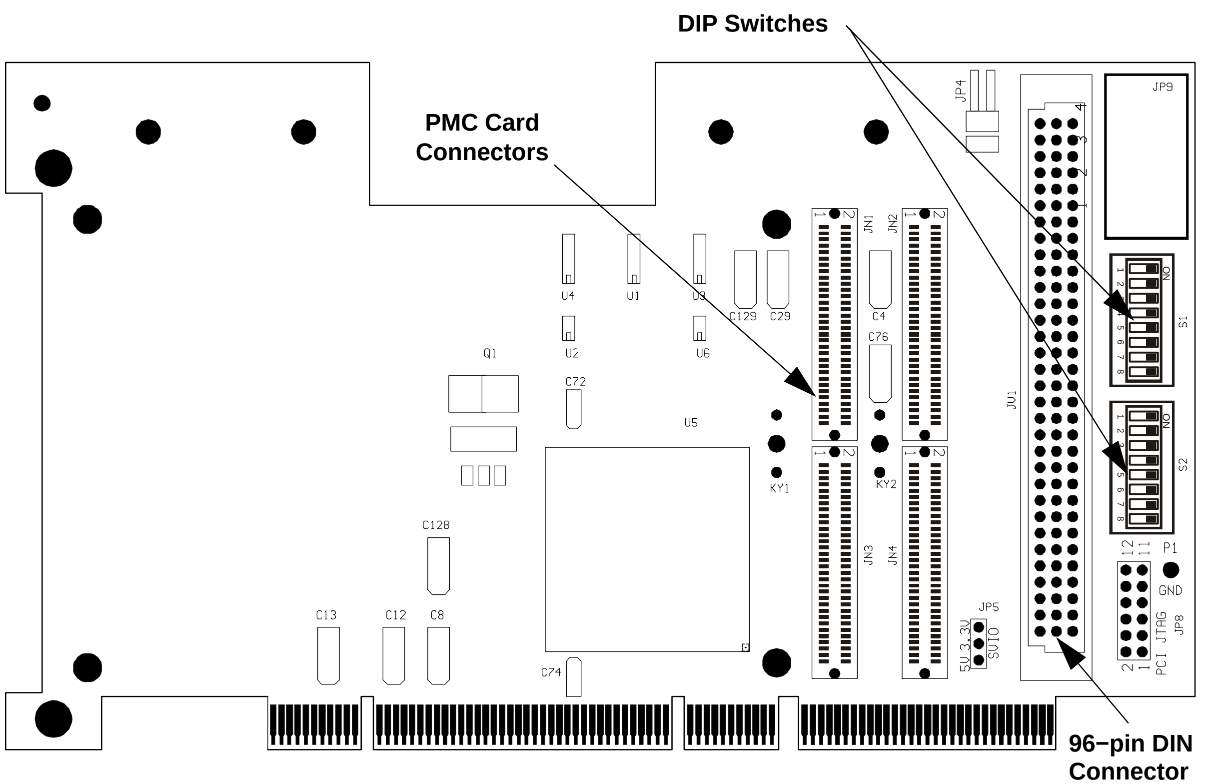


Figure 2–3: PMC to PCI Adapter, Component Side

2.2 Jumper and Switch Settings (continued)

2.2.2 PMC to PCI Adapter Switches (continued)

Two 8–position DIP switches, **S1** and **S2**, located at the back end of the PCB (see Figure 2–3 on prior page), configure certain features of the adapter in accordance with the following table. To set any of the switch positions to ON push the slider toward the **ON** printed on the body of the switch.

Table 2–2: PCI Adapter Switch Settings

Switch	Factory Default	"ON" function	"OFF" function	Description
S1–1	OFF	31154 pin G2 tied to GND	31154 pin G2 tied to Pullup Resistor	Control 31154 CLOCK outputs. Always set to OFF.
S1–2	OFF	31154 pin E2 tied to GND	31154 pin E2 tied to Pullup Resistor	Control 31154 CLOCK outputs. Always set to OFF.
S1–3	OFF	31154 pin D1 tied to GND	31154 pin D1 tied to Pullup Resistor	Control 31154 CLOCK outputs. Always set to OFF.
S1–4	OFF	31154 pin C6 tied to GND	31154 pin C6 tied to Pullup Resistor	Control 31154 CLOCK outputs. Always set to OFF.
S1–5	ON	31154 pin Y22 tied to GND. 64-bit status HIGH (1).	31154 pin Y22 tied to Pullup Resistor. 64-bit status LOW (0).	Enables 64-bit status in PCI-X config reg. Has no bridge function. See 31154 datasheet
S1–6	OFF	31154 pin AA18 tied to GND. Disable opaque mode	31154 pin AA18 tied to Pullup Resistor. Enable opaque mode	Controls "opaque" function. See 31154 datasheet.
S1–7	OFF	31154 pin V3 tied to GND. Allow max 133 MHz.	31154 pin V3 tied to Pullup Resistor. Allow max 100 MHz.	Controls "SMAX 100" for secondary PCI frequency.
S1–8	ON	31154 pin AC22 tied to GND. Enable devices 16-21	31154 pin AC22 tied to Pullup Resistor. Disable devices 16-21	Controls IDSELMASK function. See 31154 datasheet
S2–1	OFF	Ties secondary M66EN to GND (i.e., force 33MHz)	Secondary M66EN is not tied to GND. (i.e., allow 66 MHz)	Forces operation at 33 MHz on the secondary interface.
S2–2	OFF	Secondary PCI bus "X" mode disabled (PCIXCAP tied to GND)	Secondary PCI bus "X" mode enabled (PCIXCAP not tied to GND)	Enables/Disables secondary side "PCI-X" operation.
S2–3	OFF	Secondary PCI bus "X" mode forced to 66 MHz.	Secondary PCI bus can operate at 133 MHz.	When "ON", a 10K ohm resistor to GND is placed on the secondary PCIXCAP signal.
S2–4	OFF	NOT USED	NOT USED	Unconnected Switch
S2–5	OFF	Ties Primary M66EN to GND (i.e., force 33MHz)	Primary M66ENA is not tied to GND (i.e., allow 66 MHz)	Forced operation at 33 MHz on the primary PCI bus
S2–6	OFF	Primary PCI bus "X" mode disabled (PCIXCAP tied to GND)	Primary PCI bus "X" mode enabled (PCIXCAP not tied to GND)	Enables/Disables primary side "PCI-X" operation.
S2–7	OFF	Primary PCI bus "X" mode forced to 66 MHz.	Primary PCI bus can operate at 133 MHz.	When "ON", a 10K ohm resistor to GND is placed on the primary PCIXCAP signal.
S2–8	OFF	NOT USED	NOT USED	Unconnected Switch

2.2 Jumper and Switch Settings (continued)

2.2.3 Removing PMC Module from PCI Adapter

CAUTION

Perform all steps at a static−controlled work workstation.

1) From the solder side of the Model 7642 assembly (side opposite the PMC), unscrew the four pan−head Phillips mounting screws from the 7142 PMC module (see illustration below).

2) **GENTLY**, pull on the PMC card to disengage the card's connectors from the PCI adapter's PMC connectors. Slide the 7142 PMC front panel out of the opening from behind the PCI adapter panel.

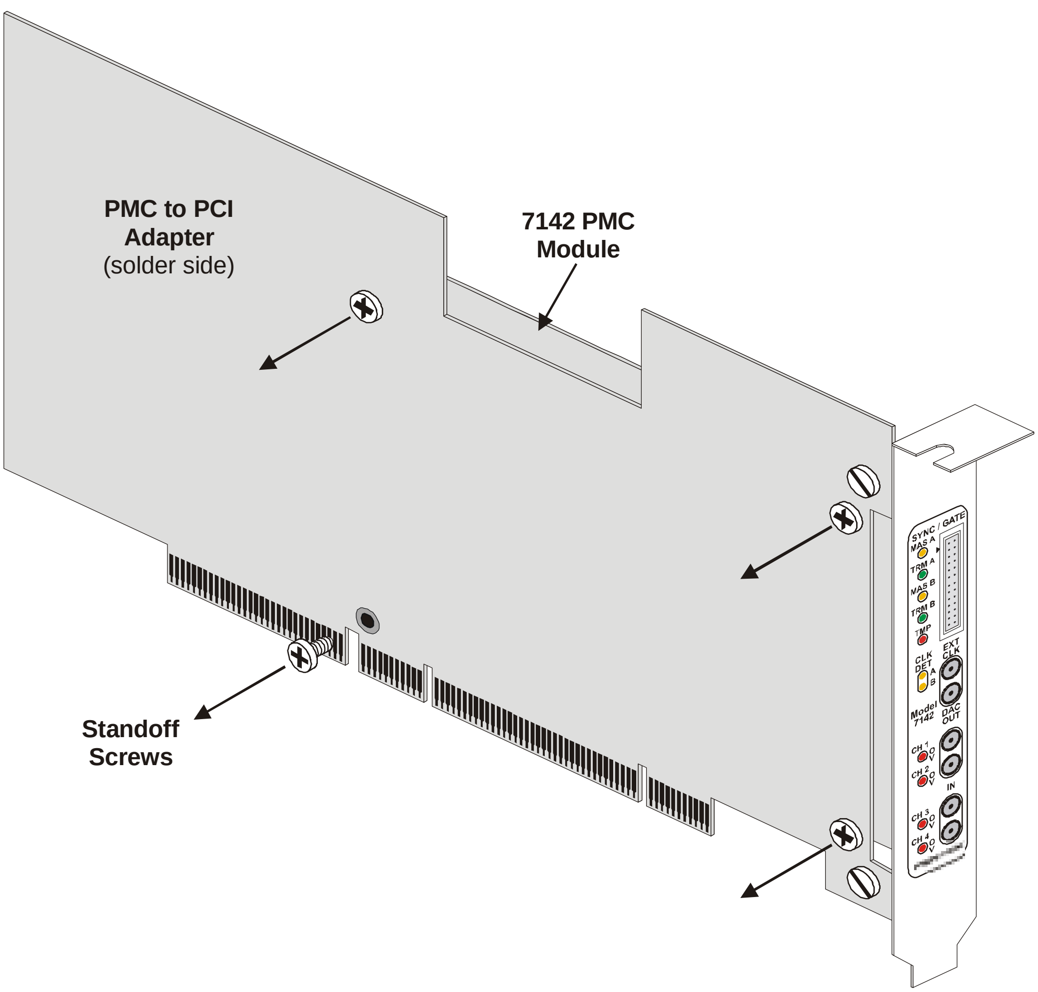


Figure 2−4: Removing PMC Module from PMC Adapter

2.2 Jumper and Switch Settings (continued)

2.2.4 Replacing PMC Module onto PCI Adapter

1) Position the 7142 PMC front panel into the opening from behind the PCI adapter slot panel (see Figure 2–7). Align the PMC module so that the PMC key hole on the PMC card is aligned over the PMC key on the PCI adapter (see illustration below).

NOTE: For proper operation of the Model 7642, the PMC key must be in the 3.3V position, as illustrated below.

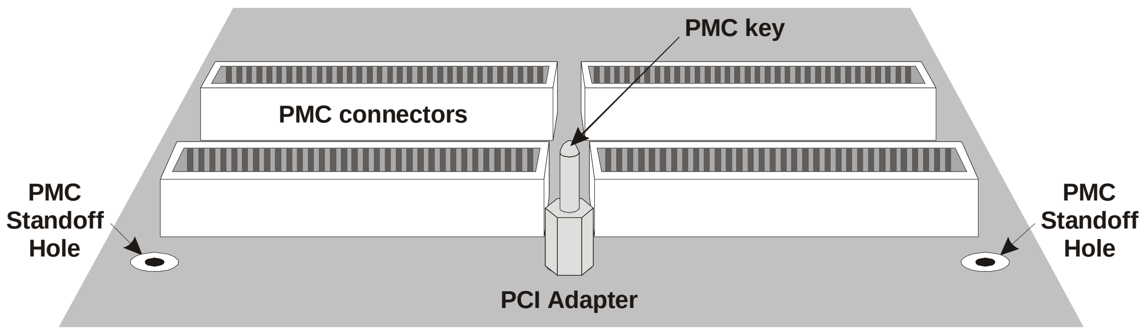


Figure 2–5: PCI Adapter PMC Connections

2) **GENTLY but firmly**, press down on the areas of the PMC opposite the connectors to fully seat the card's connectors into the adapter. The four connectors on the PMC should connect smoothly with the corresponding PMC connectors on the adapter.

3) From the solder side of the PCI adapter, secure the PMC to the PCI adapter by screwing the four pan–head Phillips mounting screws (those removed in Section 2.2.3 above) through the PCI adapter into the PMC (see illustration above and Figure 2–4 on prior page).

2.3 Installing the Pentek JTAG PC Assembly

If you need to use JTAG connections to the Model 7642 (e.g., to download code to the configuration EEPROMs or to the PowerPC core), a special JTAG connector PC assem–bly, Pentek part # 004.71402, must be mounted onto the 7142 PMC board. This unit may be mounted onto the 7142 while the PMC module is attached to the PCI adapter.

An assembly drawing of the Model 7642 is provided below, showing the JTAG PC assembly mounting orientation.

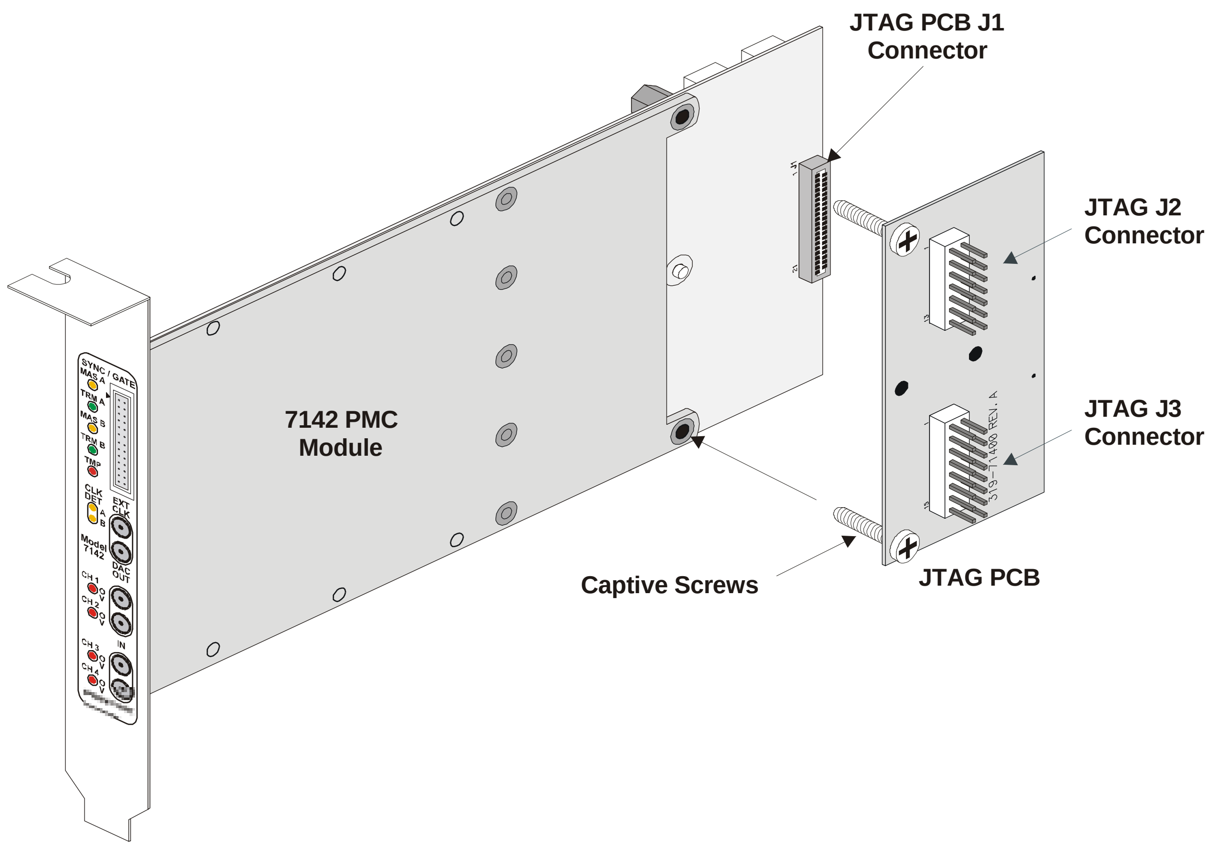


Figure 2−6: Model 7642 − JTAG PC Assembly Mounting

To mount the JTAG PC assembly onto the 7642:

1) First remove the two flat–head mounting screws that secure the Model 7642 PCB to the PCI adapter standoffs.

2) Plug the JTAG PC assembly into the connector **J1** on the 7642 (see figure above), and secure to the 7642 and PCI adapter using the two captive screws on the JTAG board.

The JTAG **J2** and **J3** connectors are described on the next page.

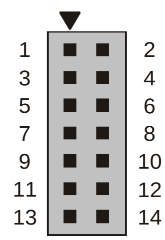
2.3 Installing the Pentek JTAG PC Assembly (continued)

The following subsections describe the JTAG connectors on the 004.71402 PC assembly. Refer to Figure 2−6 for the location of these connectors on the JTAG PC assembly.

2.3.1 JTAG J2 Connector

The JTAG **J2** In−Circuit Program Chain (ISP) connector provides a connec− tion to download programs and to perform boundary−scan tests on 7642 PCB devices. This connector is reserved for Pentek factory use only. The pinout for this 14−pin header is given in the following table.

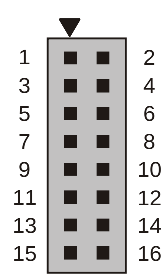
Table 2−3: JTAG J2 Connector

Signal	Pin Number		Pin Number	Signal
TMS	1		2	No Connection
TDI	3		4	GND
+3.3 V	5		6	No Connection
TDO	7		8	GND
TCK_RET	9		10	No Connection
TCK	11		12	GND
No Connection	13		14	No Connection

2.3.2 JTAG J3 Connector

The JTAG **J3** connector provides a connection to download programs and to control the execution of those programs on the PowerPC core in the FPGA. This connector is reserved for Pentek factory use only. The pinout for this 16−pin header is given in the following table.

Table 2−4: JTAG J3 Connector

Signal	Pin Number		Pin Number	Signal
PPC_TDO	1		2	No Connection
PPC_TDI	3		4	PPC_TRST_N
No Connection	5		6	+ 3.3 V
PPC_TCK	7		8	No Connection
PPC_TMS	9		10	No Connection
PPC_HALT_N	11		12	No Connection
No Connection	13		14	No Connection
No Connection	15		16	GND

2.4 Installing the Model 7642 in a Personal Computer

Pentek's Model 7642 is designed to operate in personal computers that support the Intel PCI Bus standard. This card cannot be used in computers equipped with the VESA local bus, nor in those machines that provide only ISA or EISA expansion slots.

CAUTION

Perform this installation at a static−controlled work workstation.
Disconnect all power from the PC before attempting to install this board.

Refer to Figure 2−1 on page 30 for illustration of the complete Model 7642 assembly.

1) Orient the personal computer on your static−controlled work surface such that the rear panel faces you.

2) Remove the cover from the computer, to gain access to the PC's motherboard and its local bus connectors.

3) PCI Bus connectors are usually white in color (as opposed to ISA bus connectors which are usually black, and VESA connectors which are usually brown), and are about 3½" long. Select a vacant PCI connector in which to install the Pentek 7642 assembly, and remove the blank expansion slot cover plate on the computer's rear panel located immediately to the RIGHT of the selected connector (this also differs from ISA expansion cards, where the rear panel is mounted to the left of the card).

4) Before touching the Model 7642 assembly, touch the case of your computer's power supply, to discharge any static electrical charge that may have accumulated on you. Then, remove the 7642 assembly from its anti−static packaging, and install the card's connecting edge into the selected PCI expansion socket. (See Figure 2−7 on the next page.)

   Be certain that the card edge is properly aligned with the PCI connector. Gentle downward pressure should be sufficient to fully seat the card edge in the connector. **DO NOT ATTEMPT TO FORCE THE CARD INTO THE SLOT**! If excessive force is necessary, then the card is probably misaligned. Damage to either the PC motherboard or the 7642 will be the most likely result of attempts at forced installations.

5) After the assembly is fully and properly seated in the PCI connector, secure the card's rear panel to the computer's chassis.

6) The Model 7642 board has a 4−pin power connector to supply the majority of power to the components. This is a standard disk drive power connector used in PCs. Plug a spare power connector from your PC’s power supply into this PCB connector.

2.4 Installing the Model 7642 in a Personal Computer (continued)

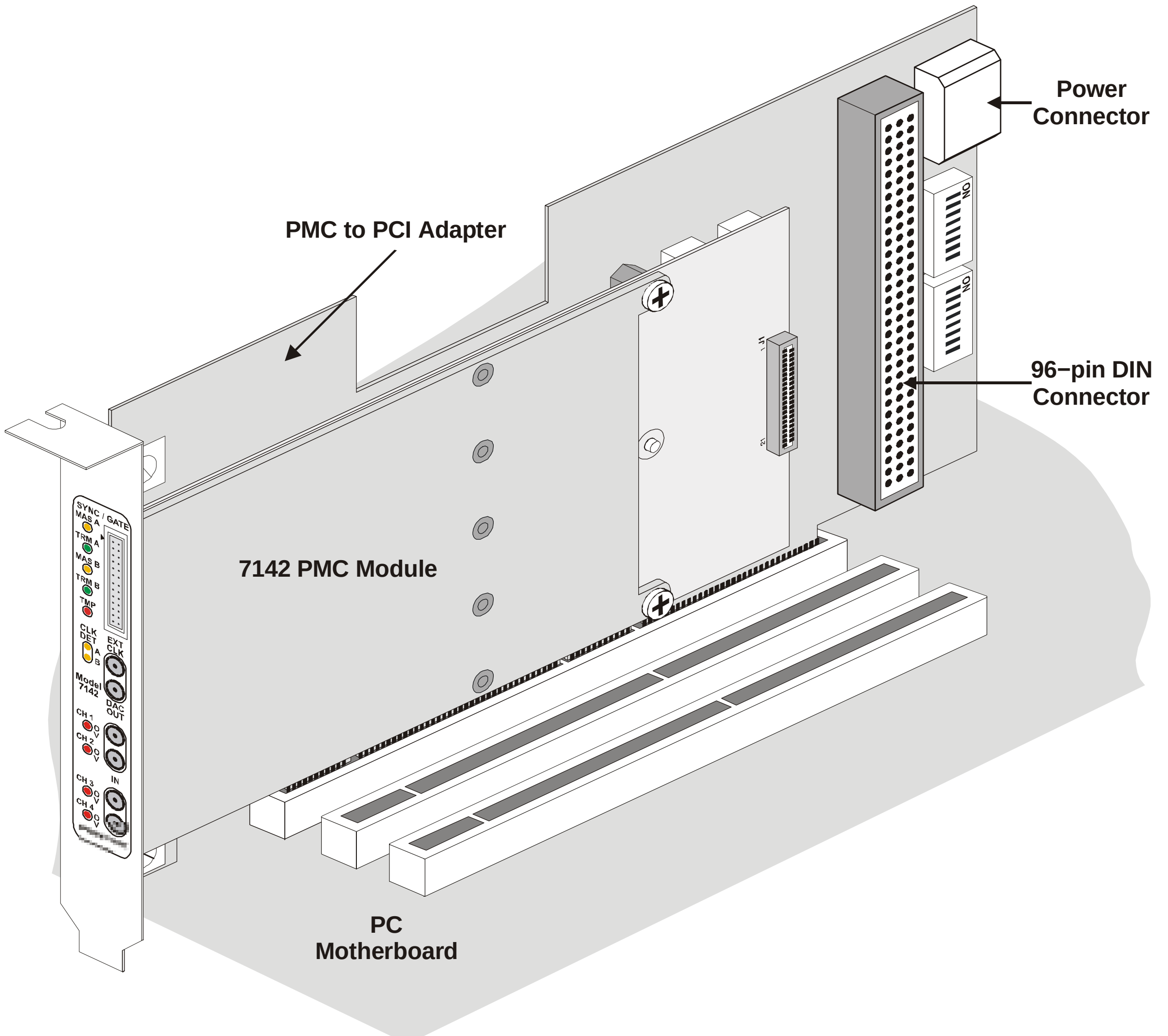


Figure 2−7: Model 7642 Mounted in PCI Slot

Note that the Model 7642 provides a 96−pin DIN connector at the edge of the PCB. The 'A' and 'C' rows of this connector are routed to PCI FPGA spare pins on the board. This is used with Model 7642 Option 104, described in Section 2.5 on the following pages.

2.5 PCI FPGA I/O Connections (Option 104)

Option 104 for the Model 7642 provides connections from PCI FPGA spare pins to a 96–pin DIN connector on the PMC to PCI adapter (see Figure 2–7 on the prior page). This connector mimics the connection of the USER I/O connector (**P4**) pins on the 7142 PMC module to the ‘A’ and ‘C’ rows of the **P2** connector on a PMC–VMEbus board. These connections are programmed for low–voltage differential signals (LVDS) in the default PCI FPGA configuration—the user can reconfigure these pins with custom FPGA programming (see Section 1.12, FPGA Configuration).

The Model 7142 PMC is connected to the PCI adapter using four 64–pin connectors, designated **P1** through **P4** on the PMC (see Figure 2–2). These connectors are defined by the PMC standard, as follows:

- **P1** and **P2** (PMC Pn1 and Pn2 respectively) contain the 32–bit PCI bus interface plus associated power, ground, reserved pins, and other necessary signals.
- **P3** (PMC Pn3) supports expansion to the 64–bit PCI bus interface.
- **P4** (PMC Pn4) supports user–defined I/O, as described below.

Model 7642 Option 104 routes spare PCI FPGA I/O pins to the PMC **P4** connector, and then to the 96–pin DIN connector on the PCI adapter (see Figure 2–7).

With Option 104, 32 pairs of LVDS connections are routed from the PCI FPGA to PMC connector **P4**, and then to the 96–pin DIN connector. These are available to the user for custom I/O. Table 2–5, on the following page, identifies the low–voltage differential signals (LVDS) that are connected from the FPGA to the PMC **P4** connector pins, and to the 96–pin DIN connector pins.

Use the FPGA Data In/Out Registers, Section 5.4, to read and write data using this connector.

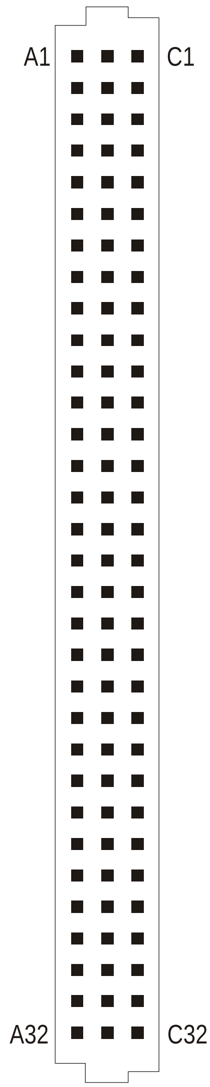
- The LVDS_OUT signals in Table 2–5 are from the FPGA Data Out Register, with LVDS_OUT_x0 from bit D0 and LVDS_OUT_x15 from bit D15.
- The LVDS_IN signals are for the FPGA Data In Register, with LVDS_IN_x0 from bit D0 and LVDS_IN_x15 from bit D15.

Note that only rows ‘A’ and ‘C’ of the 96–pin DIN connector are used for LVDS connections, providing a 64–pin path from the PCI FPGA I/O pins, through the PMC **P4** connector, to the DIN connector. Ground pins are available for custom FPGA I/O design on pins B2, B12, B22, and B32 in row ‘B’ of the DIN connector.

2.5 PCI FPGA I/O Connections (continued)

Table 2–5: Option 104 PCI FPGA I/O Pin Connections

FPGA Signal	PMC P4 Pin	DIN Conn. Pin	DIN Conn. Pin	PMC P4 Pin	FPGA Signal
LVDS_IN_P0	2	A1	C1	1	LVDS_OUT_P0
LVDS_IN_N0	4	A2	C2	3	LVDS_ OUT_N0
LVDS_IN_P1	6	A3	C3	5	LVDS_ OUT_P1
LVDS_IN_N1	8	A4	C4	7	LVDS_ OUT_N1
LVDS_IN_P2	10	A5	C5	9	LVDS_ OUT_P2
LVDS_IN_N2	12	A6	C6	11	LVDS_ OUT_N2
LVDS_IN_P3	14	A7	C7	13	LVDS_ OUT_P3
LVDS_IN_N3	16	A8	C8	15	LVDS_ OUT_N3
LVDS_IN_P4	18	A9	C9	17	LVDS_ OUT_P4
LVDS_IN_N4	20	A10	C10	19	LVDS_ OUT_N4
LVDS_IN_P5	22	A11	C11	21	LVDS_ OUT_P5
LVDS_IN_N5	24	A12	C12	23	LVDS_ OUT_N5
LVDS_IN_P6	26	A13	C13	25	LVDS_ OUT_P6
LVDS_IN_N6	28	A14	C14	27	LVDS_ OUT_N6
LVDS_IN_P7	30	A15	C15	29	LVDS_ OUT_P7
LVDS_IN_N7	32	A16	C16	31	LVDS_ OUT_N7
LVDS_IN_P8	34	A17	C17	33	LVDS_OUT_P8
LVDS_IN_N8	36	A18	C18	35	LVDS_ OUT_N8
LVDS_IN_P9	38	A19	C19	37	LVDS_ OUT_P9
LVDS_IN_N9	40	A20	C20	39	LVDS_ OUT_N9
LVDS_IN_P10	42	A21	C21	41	LVDS_ OUT_P10
LVDS_IN_N10	44	A22	C22	43	LVDS_ OUT_N10
LVDS_IN_P11	46	A23	C23	45	LVDS_ OUT_P11
LVDS_IN_N11	48	A24	C24	47	LVDS_ OUT_N11
LVDS_IN_P12	50	A25	C25	49	LVDS_ OUT_P12
LVDS_IN_N12	52	A26	C26	51	LVDS_ OUT_N12
LVDS_IN_P13	54	A27	C27	53	LVDS_ OUT_P13
LVDS_IN_N13	56	A28	C28	55	LVDS_ OUT_N13
LVDS_IN_P14	58	A29	C29	57	LVDS_ OUT_P14
LVDS_IN_N14	60	A30	C30	59	LVDS_ OUT_N14
LVDS_IN_P15	62	A31	C31	61	LVDS_ OUT_P15
LVDS_IN_N15	64	A32	C32	63	LVDS_ OUT_N15



Ground pins are available on pins B2, B12, B22, and B32 of the DIN connector.

2.6 Front Panel Connections

The Model 7642 front panel is illustrated in the figure at the right. The panel fits into the slot opening on the PCI adapter, and is accessible at the rear of the PC. Refer to Figure 2–7, on page 39, for location of the panel on the 7642 assembly. The front panel includes six MMCX coaxial connectors for input/output of analog and clock signals, and a 26–pin Sync bus connector labeled **SYNC/GATE**. These connectors are described in the following subsections.

The front panel also includes eleven LED indicators. These are described in Section 2.7, on page 44.

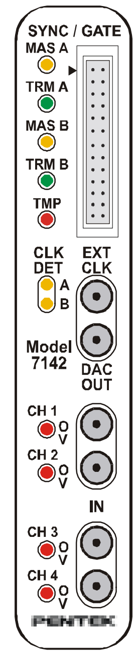


Figure 2–8: 7642 PMC Front Panel

2.6.1 Analog Input Connectors — **CH1 – CH4 IN**

The front panel has four MMCX microminiature coaxial socket receptacles for analog signal inputs, labeled **CH 1, CH 2**, **CH 3**, and **CH 4 IN**, one for each A/D input channel.

The analog input signal must be +10 dBm full scale. Each input drives an RF transformer, with 50 Ω input impedance.

2.6.2 Clock Input Connector — **EXT CLK**

The front panel has one MMCX microminiature coaxial socket receptacles, labeled **EXT CLK**, for input of an external sample clock.

The external clock signal must be a sine wave of 0 to 10 dBm, with a fre–quency range from 1 to 125 MHz for ADC use, or from 1 to 300 MHz for DAC use.

The clock can be used as the reference signal to derive the sample clock sig–nal for the A/D converters and digital receivers. This input is enabled using the Master Control Register SEL CLK bit (see Section 6.8.1.8).

2.6.3 Analog Output Connector — **DAC OUT**

The front panel has one MMCX microminiature coaxial socket receptacle for analog signal output, labeled **DAC OUT**.

The analog output signal is within the range of +4 dBm (–2 dBm with Option 002). This output is driven by an RF transformer into 50 Ω output impedance.

2.6 Front Panel Connections (continued)

2.6.4 SYNC/GATE Connector

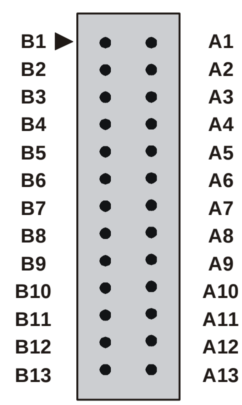
SYNC/GATE

The 26–pin **SYNC/GATE** front panel connector provides input/output pins for the low–voltage differential signal (LVDS) timing buses: two independent sets of clock, sync, and gate signals. When the Model 7642 is a bus Master, these pins output the buses to other slave units. When the 7642 is a bus Slave, these pins input all LVDS bus signals from a bus Master. This connector also provides two TTL Gate/Sync inputs. The mating 26–pin connector is Pentek part # 353.02607 (ERNI # 214346).

The following table shows the connector pin configuration.

Table 2–6: SYNC/GATE Connector Pins

Signal	Pin	Pin	Signal
TTL GATE	B1	A1	GND
TTL SYNC	B2	A2	GND
GATE A−	B3	A3	GATE A+
GND	B4	A4	GND
SYNC A−	B5	A5	SYNC A+
GND	B6	A6	GND
CLK A−	B7	A7	CLK A+
GND	B8	A8	GND
GATE B−	B9	A9	GATE B+
GND	B10	A10	GND
SYNC B−	B11	A11	SYNC B+
GND	B12	A12	GND
CLK B−	B13	A13	CLK B+



2.7 Front Panel LEDs

The Model 7642 front panel has eleven LED indicators, as illustrated in Figure 2–8.

2.7.1 Master LEDs **MAS A, MAS B**

There are two yellow **MAS** LEDs, one for each Sync Bus (**A** or **B**). The **MAS** LED illuminates when this Model 7642 is the bus Master (MASTR bit D00 = 1, Master Bus A/B Control Register, Section 6.8.1.10 or 6.8.4.10). The bus Master generates all sync/gate/clock signals on the LVDS bus. When only a single 7642 is used, it must be a Master.

2.7.2 Terminate LEDs **TRM A, TRM B**

There are two yellow **TRM** LEDs, one for each Sync Bus (**A** or **B**). The yellow **TRM** LED illuminates when this Model 7642 bus is terminated (TERM bit D01 = 1, Master Bus A/B Control Register, Section 6.8.1.10 or 6.8.4.10). When this 7642 is the last Slave unit or only unit on the Sync Bus, you must enable bus termination.

2.7.3 Over Temperature LED **TMP**

There are several temperature/voltage sensors on the Model 7642 PCB. The sensor thresholds are set by a PCI interface processor (see TWSI Port Regis–ter, Section 6.3). When an over–temperature or over–voltage condition is indicated, the red **TMP** LED illuminates on the front panel. In addition, an over–temperature/voltage interrupt is available to any processor on the PCI interface (see Table 6–19, page 140).

<u>NOTE:</u> You must set up the temperature/voltage sensors' serial port following power on of the PCI interface (see TWSI Port Register, Section 6.3).

2.7.4 Clock LEDs **CLK DET A, CLK DET B**

There are two green **CLK DET** LEDs, one for each Sync Bus (**A** or **B**). The green **CLK DET** LEDs illuminate when a valid sample clock signal is detected on the associated Sync Bus. If this LED is not illuminated, then no clock has been detected and no data from the input stream can be processed.

2.7.5 Overload LEDs **CH1 – CH4 OV**

There are four red overload LEDs, one for each A/D input, labeled **CH1**, **CH2**, **CH3**, and **CH4 OV**. Each LED is an indicator for an A/D overload detection function in each LTC2255 A/D converter. When an overload indication is set by the LTC2255, the associated OVLD LED illuminates. In addition, an OVLD interrupt may be generated from each A/D overload indication to a PCI interface processor (see Table 6–19, page 140).

Chapter 3: Model 7642 Memory Maps

3.1 Overview

This chapter describes access to the Model 7642 resources and control registers from the personal computer's PCI interface. Memory maps to these resources are given from the PCI interface viewpoint.

3.2 Model 7642 Address Mapping

The Model 7642 is controlled from the PCI interface through a PCI Bus interface, the Pentek PCI7142 chip. All 7642 resources are configured to be available in PCI Address Space. The following table provides a memory map for these resources, relative to PCI Base Address Registers BAR0, BAR1, and BAR2.

Table 3–1: Model 7642 PCI Memory Map

PCI Bus Address *	Resource	Information
BAR0 + **0x00000** – **0x0FFFF**	PCI7142 Registers	Section 3.3.1
BAR0 + **0x10000** – **0x1FFFF**	User Space	Section 4.9
BAR1 + **0x00000** – **0x7FFFF**	Reserved	–
BAR2 + **0x0000** – **0xFFFF**	Signal FPGA Registers	Section 3.3.2

* BAR0 and BAR2 are programmed by the PCI bus driver supplied for your workstation. See Appendix A for a summary of the PCI Configuration Space Registers.

3.3 Memory Maps

3.3.1 PCI7142 Memory Map

The following is the memory map of the PCI7142 internal registers accessible from a PCI interface processor. All PCI7142 registers are 32 bits wide, however since the PCI local bus is a 64–bit data path, the register addresses are in increments of 64 bits (8 bytes). Refer to Chapter 5 for description of these registers.

NOTE: Access these registers using PCI Base Address Register 0 (**BAR0**).

Table 3–2: PCI7142 Registers Memory Map

Address Offset *	Register	Access	Information
Interrupt Control Registers			
0x0000	PCI Interrupt Flag	R/Clr	Section 5.2
0x0008	Reserved	–	–
0x0010	PCI Interrupt A Enable	R/W	Section 5.3
0x0018	PCI Interrupt B Enable	R/W	
0x0020	PCI Interrupt C Enable	R/W	
0x0028	PCI Interrupt D Enable	R/W	
0x0030 - 0x0048	Reserved	–	–
Board Option Registers			
0x0050	FPGA Data In (Option 104 only)	R.O.	Section 5.4
0x0058	FPGA Data Out (Option 104 only)	R/W	
0x0060 - 0x0068	Reserved	–	–
PCI7142 Control Registers			
0x0070	Board/Channel Reset	R/W	Section 5.5
0x0078	PCI7142 Revision	R.O.	Section 5.6
0x0080	Virtex Config Register 0	R/W	Section 5.7
0x0088	PCI DCM Control	R/W	Section 5.8
0x0090	Local DMA Request Status	R.O.	Section 5.9
0x0098	Reserved	–	–
0x00A0	PCI Bus Status	R.O.	Section 5.10
0x00A8	DMA PCI Interrupt Enable/Flag	R/Clr	Section 5.11
0x00B0	Local DMA In Base Address Remap	R/W	Section 5.12
0x00B8	Local DMA Out Base Address Remap	R/W	Section 5.13

* Offset from PCI Base Address Register 0 (**BAR0**), see Table 3–1

3.3 Memory Maps (continued)

3.3.1 PCI7142 Memory Map (continued)

NOTE: Access these registers using PCI Base Address Register 0 (**BAR0**).

Table 3–2: PCI7142 Registers Memory Map (continued)

Address Offset *	Register	Access	Information
DMA Control Registers			
0x0100	DMA Command	R/W	Section 5.14
0x0108	Reserved	–	–
0x0110	DMA0 Current Transfer Counter	R.O.	Section 5.15
0x0118	DMA0 Current PCI Address	R.O.	Section 5.16
0x0120	DMA1 Current Transfer Counter	R.O.	Section 5.15
0x0128	DMA1 Current PCI Address	R.O.	Section 5.16
0x0130	DMA2 Current Transfer Counter	R.O.	Section 5.15
0x0138	DMA2 Current PCI Address	R.O.	Section 5.16
0x0140	DMA3 Current Transfer Counter	R.O.	Section 5.15
0x0148	DMA3 Current PCI Address	R.O.	Section 5.16
0x0150	DMA4 Current Transfer Counter	R.O.	Section 5.15
0x0158	DMA4 Current PCI Address	R.O.	Section 5.16
0x0160	DMA5 Current Transfer Counter	R.O.	Section 5.15
0x0168	DMA5 Current PCI Address	R.O.	Section 5.16
0x0170	DMA6 Current Transfer Counter	R.O.	Section 5.15
0x0178	DMA6 Current PCI Address	R.O.	Section 5.16
0x0180	DMA7 Current Transfer Counter	R.O.	Section 5.15
0x0188	DMA7 Current PCI Address	R.O.	Section 5.16
0x0190	DMA8 Current Transfer Counter	R.O.	Section 5.15
0x0198	DMA8 Current PCI Address	R.O.	Section 5.16
DMA Descriptor Registers			
0x1000	DMA0 Command/Status	R/W	Section 5.17
0x1008	DMA0 Transfer Interval Counter	R/W	Section 5.18
0x1010	DMA0 Descriptor 0 Transfer Count	R/W	Section 5.19
0x1018	DMA0 Descriptor 0 PCI Address	R/W	Section 5.20
0x1020	DMA0 Descriptor1 Transfer Count	R/W	Section 5.19
0x1028	DMA0 Descriptor 1 PCI Address	R/W	Section 5.20
0x1030	DMA0 Descriptor 2 Transfer Count	R/W	Section 5.19
0x1038	DMA0 Descriptor 2 PCI Address	R/W	Section 5.20
0x1040	DMA0 Descriptor 3 Transfer Count	R/W	Section 5.19
0x1048	DMA0 Descriptor 3 PCI Address	R/W	Section 5.20

* Offset from PCI Base Address Register 0 (**BAR0**), see Table 3–1

3.3 Memory Maps (continued)

3.3.1 PCI7142 Memory Map (continued)

NOTE: Access these registers using PCI Base Address Register 0 (**BAR0**).

Table 3−2: PCI7142 Registers Memory Map (continued)

Address Offset *	Register	Access	Information
DMA Descriptor Registers (continued)			
0x1050 - 0x1FF8	Reserved	–	–
0x2000 - 0x2048	DMA1 (same as above ten DMA0 registers)	R/W	(see above)
0x2050 - 0x2FF8	Reserved	–	–
0x3000 - 0x3048	DMA2 (same as above ten DMA0 registers)	R/W	(see above)
0x3050 - 0x3FF8	Reserved	–	–
0x4000 - 0x4048	DMA3 (same as above ten DMA0 registers)	R/W	(see above)
0x4050 - 0x4FF8	Reserved	–	–
0x5000 - 0x5048	DMA4 (same as above ten DMA0 registers)	R/W	(see above)
0x5050 - 0x5FF8	Reserved	–	–
0x6000 - 0x6048	DMA5 (same as above ten DMA0 registers)	R/W	(see above)
0x6050 - 0x6FF8	Reserved	–	–
0x7000 - 0x7048	DMA6 (same as above ten DMA0 registers)	R/W	(see above)
0x7050 - 0x7FF8	Reserved	–	–
0x8000 - 0x8048	DMA7 (same as above ten DMA0 registers)	R/W	(see above)
0x8050 - 0x8FF8	Reserved	–	–
0x9000 - 0x9048	DMA8 (same as above ten DMA0 registers)	R/W	(see above)
0x9050 - 0x9FF8	Reserved	–	–
Signal FPGA Load DMA Registers			
0xA000	FPGA Load DMA Command/Status	R/W	Section 5.21
0xA008	Reserved	–	–
0xA010	FPGA Load DMA Transfer Count	R/W	Section 5.22
0xA018	FPGA Load DMA PCI Address	R/W	Section 5.23
0xA020 - 0xFFF8	Reserved	–	–

* Offset from PCI Base Address Register 0 (**BAR0**), see Table 3−1

3.3 Memory Maps (continued)

3.3.2 Signal FPGA Memory Map

The following is the memory map of the Signal FPGA registers accessible from the PCI interface. All Signal FPGA registers are 16 bits wide, however since the PCI local bus is a 64–bit data path, the register addresses are in increments of 64 bits (8 bytes). Refer to Chapter 6 for description of these registers.

NOTE: Access these registers using PCI Base Address Register 2 (**BAR2**).

Table 3–3: Signal FPGA Registers Memory Map

Address Offset *	Register	Access	Information
Global Registers			
0x8000	ID Readout	R.O.	Section 6.2
0x8008	Reserved	–	–
0x8010	Reserved	–	–
0x8018	TWSI Port	R/W	Section 6.3
0x8020	DCM Control	R/W	Section 6.4
0x8028	Miscellaneous Control	R/W	Section 6.5
0x8030	Reserved	–	–
0x8038	FPGA Revision 1	R.O.	Section 6.6
0x8040	FPGA Revision 2	R.O.	
0x8048	Core Option	R.O.	Section 6.7
0x804C - 0x8078	Reserved	–	–
Clock/Sync/Gate Registers			
0x8080	Master Bus A Control	R/W	Section 6.8.1
0x8088	Sync A Generator	R/W	Section 6.8.2
0x8090	Gate A Generator	R/W	Section 6.8.3
0x8098	Master Bus B Control	R/W	Section 6.8.4
0x80a0	Sync B Generator	R/W	Section 6.8.5
0x80a8	Gate B Generator	R/W	Section 6.8.6
0x80b0	Sync Mask	R/W	Section 6.8.7
0x80b8 - 0x80f8	Reserved	–	–

* Offset from PCI Base Address Register 2 (**BAR2**), see Table 3–1

3.3 Memory Maps (continued)

3.3.2 Signal FPGA Memory Maps (continued)

NOTE: Access these registers using PCI Base Address Register 2 (**BAR2**).

Table 3–3: Signal FPGA Registers Memory Map (continued)

Address Offset *	Register	Access	Information
Interrupt Registers			
`0x8100`	System Interrupt Enable	R/W	Section 6.9.1
`0x8108`	System Interrupt Flag	R/Clr	Section 6.9.2
`0x8110`	System Interrupt Status	R.O.	Section 6.9.3
`0x8118`	Application Interrupt Lint0 Enable	R/W	Section 6.9.4
`0x8120`	Application Interrupt Lint1 Enable	R/W	
`0x8128`	Application Interrupt Lint2 Enable	R/W	
`0x8130`	Application Interrupt Lint3 Enable	R/W	
`0x8138`	Application Interrupt Flag	R/Clr	Section 6.9.5
`0x8140`	Application Interrupt Status	R.O.	Section 6.9.6
FIFO Status Registers			
`0x8148`	ADC 1 FIFO Status	R/Clr	Section 6.10.3
`0x8150`	ADC 2 FIFO Status	R/Clr	
`0x8158`	ADC 3 FIFO Status	R/Clr	
`0x8160`	ADC 4 FIFO Status	R/Clr	
`0x8168`	User In FIFO Status	R/Clr	Section 6.14.2
`0x8170`	User Out FIFO Status	R/Clr	
`0x8178`	DAC FIFO Status	R/Clr	Section 6.11.4
`0x8180`	Reserved	–	–
`0x8188`	DDR Memory Read FIFO Status	R/Clr	Section 6.13.7
`0x8190`	DDR Memory Write FIFO Status	R/Clr	
`0x8198`	Test FIFO Status	R/Clr	Section 6.15.2
`0x81a0 - 0x83f8`	Reserved	–	–

* Offset from PCI Base Address Register 2 (**BAR2**), see Table 3–1

3.3 Memory Maps (continued)

3.3.2 Signal FPGA Memory Maps (continued)

NOTE: Access these registers using PCI Base Address Register 2 (**BAR2**).

Table 3–3: Signal FPGA Registers Memory Map (continued)			
Address Offset *	**Register**	**Access**	**Information**
D/A, DAC FIFO Registers			
`0x8400`	DAC Sync Bus Select	R/W	Section 6.11.1
`0x8408`	DAC Control / Status	R/W	Section 6.11.2
`0x8410`	DAC FIFO Control	R/W	Section 6.11.3
`0x8418`	DAC FIFO Trigger Length LSB	R/W	Section 6.11.5
`0x8420`	DAC FIFO Trigger Length MSB	R/W	
`0x8428`	DAC FIFO Interrupt Mask	R/W	Section 6.11.6
`0x8430`	DAC FIFO Almost Empty Level	R/W	Section 6.11.7
`0x8438`	DAC FIFO Almost Full Level	R/W	Section 6.11.8
`0x8440`	DAC FIFO Post Trigger Delay Length LSB	R/W	Section 6.11.9
`0x8448`	DAC FIFO Post Trigger Delay Length MSB	R/W	
`0x8450 - 0x8468`	Reserved	–	–
`0x8470`	DAC5686 Read	R.O	Section 6.12
`0x8478`	DAC5686 Write	W.O	
`0x8480 - 0x87f8`	Reserved	–	–
DDR Memory Registers			
`0x8800`	DDR Memory Control	R/W	Section 6.13.1
`0x8808`	DDR Memory Bank 0 Depth LSB	R/W	Section 6.13.2
`0x8810`	DDR Memory Bank 0 Depth MSB	R/W	
`0x8818`	DDR Memory Bank 1 Depth LSB	R/W	
`0x8820`	DDR Memory Bank 1 Depth MSB	R/W	
`0x8828`	DDR Memory Bank 2 Depth LSB	R/W	
`0x8830`	DDR Memory Bank 2 Depth MSB	R/W	
`0x8838`	DDR Memory Bank 0 R/W Address LSB	R/W	Section 6.13.3
`0x8840`	DDR Memory Bank 0 R/W Address MSB	R/W	
`0x8848`	DDR Memory Bank 1 R/W Address LSB	R/W	
`0x8850`	DDR Memory Bank 1 R/W Address MSB	R/W	
`0x8858`	DDR Memory Bank 2 R/W Address LSB	R/W	
`0x8860`	DDR Memory Bank 2 R/W Address MSB	R/W	
* Offset from PCI Base Address Register 2 (**BAR2**), see Table 3–1			

3.3 Memory Maps (continued)

3.3.2 Signal FPGA Memory Maps (continued)

NOTE: Access these registers using PCI Base Address Register 2 (**BAR2**).

Table 3–3: Signal FPGA Registers Memory Map (continued)

Address Offset *	Register	Access	Information
DDR Memory Registers (continued)			
0x8868	DDR Memory Bank 0 Capture End Address LSB	R.O.	Section 6.13.4
0x8870	DDR Memory Bank 0 Capture End Address MSB	R.O.	
0x8878	DDR Memory Bank 1 Capture End Address LSB	R.O.	
0x8880	DDR Memory Bank 1 Capture End Address MSB	R.O.	
0x8888	DDR Memory Bank 2 Capture End Address LSB	R/W	
0x8890	DDR Memory Bank 2 Capture End Address MSB	R/W	
0x8898	DDR Memory Bank 0 Trigger Address LSB	R.O.	Section 6.13.5
0x88a0	DDR Memory Bank 0 Trigger Address MSB	R.O.	
0x88a8	DDR Memory Bank 1 Trigger Address LSB	R.O.	
0x88b0	DDR Memory Bank 1 Trigger Address MSB	R.O.	
0x88b8 - 0x88f8	Reserved	–	–
ADC FIFO Registers			
0x8900	ADC 1 Post Trigger Delay Length LSB	R/W	Section 6.10.4
0x8908	ADC 1 Post Trigger Delay Length MSB	R/W	
0x8910	ADC 2 Post Trigger Delay Length LSB	R/W	
0x8918	ADC 2 Post Trigger Delay Length MSB	R/W	
0x8920	ADC 3 Post Trigger Delay Length LSB	R/W	
0x8928	ADC 3 Post Trigger Delay Length MSB	R/W	
0x8930	ADC 4 Post Trigger Delay Length LSB	R/W	
0x8938	ADC 4 Post Trigger Delay Length MSB	R/W	
0x8940	ADC 1 Pre Trigger Count Capture LSB	R.O.	Section 6.10.5
0x8948	ADC 1 Pre Trigger Count Capture MSB	R.O.	
0x8950	ADC 2 Pre Trigger Count Capture LSB	R.O.	
0x8958	ADC 2 Pre Trigger Count Capture MSB	R.O.	
0x8960	ADC 3 Pre Trigger Count Capture LSB	R.O.	
0x8968	ADC 3 Pre Trigger Count Capture MSB	R.O.	
0x8970	ADC 4 Pre Trigger Count Capture LSB	R.O.	
0x8978	ADC 4 Pre Trigger Count Capture MSB	R.O.	
0x8980 - 0x8bf8	Reserved	–	–

* Offset from PCI Base Address Register 2 (**BAR2**), see Table 3–1

3.3 Memory Maps (continued)

3.3.2 Signal FPGA Memory Maps (continued)

NOTE: Access these registers using PCI Base Address Register 2 (**BAR2**).

Table 3–3: Signal FPGA Registers Memory Map (continued)			
Address Offset *	**Register**	**Access**	**Information**
ADC FIFO Registers (continued)			
0x8c00	ADC Sync Bus Select	R/W	Section 6.10.1
0x8c08	ADC 1 FIFO Control	R/W	Section 6.10.2
0x8c10	ADC 1 Trigger Length LSB	R/W	Section 6.10.6
0x8c18	ADC 1 Trigger Length MSB	R/W	
0x8c20	ADC 1 FIFO Interrupt Mask	R/W	Section 6.10.7
0x8c28	ADC 1 FIFO Almost Empty Level	R/W	Section 6.10.8
0x8c30	ADC 1 FIFO Almost Full Level	R/W	Section 6.10.9
0x8c38	ADC 1 FIFO Decimation Divider	R/W	Section 6.10.10
0x8c40	ADC 2 FIFO Control	R/W	Section 6.10.2
0x8c48	ADC 2 Trigger Length LSB	R/W	Section 6.10.6
0x8c50	ADC 2 Trigger Length MSB	R/W	
0x8c58	ADC 2 FIFO Interrupt Mask	R/W	Section 6.10.7
0x8c60	ADC 2 FIFO Almost Empty Level	R/W	Section 6.10.8
0x8c68	ADC 2 FIFO Almost Full Level	R/W	Section 6.10.9
0x8c70	ADC 2 FIFO Decimation Divider	R/W	Section 6.10.10
0x8c78	ADC 3 FIFO Control	R/W	Section 6.10.2
0x8c80	ADC 3 Trigger Length LSB	R/W	Section 6.10.6
0x8c88	ADC 3 Trigger Length MSB	R/W	
0x8c90	ADC 3 FIFO Interrupt Mask	R/W	Section 6.10.7
0x8c98	ADC 3 FIFO Almost Empty Level	R/W	Section 6.10.8
0x8ca0	ADC 3 FIFO Almost Full Level	R/W	Section 6.10.9
0x8ca8	ADC 3 FIFO Decimation Divider	R/W	Section 6.10.10
0x8cb0	ADC 4 FIFO Control	R/W	Section 6.10.2
0x8cb8	ADC 4 Trigger Length LSB	R/W	Section 6.10.6
0x8cc0	ADC 4 Trigger Length MSB	R/W	
0x8cc8	ADC 4 FIFO Interrupt Mask	R/W	Section 6.10.7
0x8cd0	ADC 4 FIFO Almost Empty Level	R/W	Section 6.10.8
0x8cd8	ADC 4 FIFO Almost Full Level	R/W	Section 6.10.9
0x8ce0	ADC 4 FIFO Decimation Divider	R/W	Section 6.10.10
0x8ce8 - 0x8d30	Reserved	–	–
* Offset from PCI Base Address Register 2 (**BAR2**), see Table 3–1			

3.3 Memory Maps (continued)

3.3.2 Signal FPGA Memory Maps (continued)

NOTE: Access these registers using PCI Base Address Register 2 (**BAR2**).

Table 3–3: Signal FPGA Registers Memory Map (continued)

Address Offset *	Register	Access	Information
Test FIFO Registers			
`0x8d38`	Test FIFO Control	R/W	Section 6.15.1
`0x8d40`	Test FIFO Flags Interrupt Mask	R/W	Section 6.15.3
`0x8d48`	Test FIFO Almost Empty Level	R/W	Section 6.15.4
`0x8d50`	Test FIFO Almost Full Level	R/W	Section 6.15.5
DDR Memory FIFO Registers			
`0x8d58`	DDR Memory Read FIFO Control	R/W	Section 6.13.6
`0x8d60 - 0x8d68`	Reserved	–	–
`0x8d70`	DDR Memory Read FIFO Flags Interrupt Mask	R/W	Section 6.13.8
`0x8d78`	DDR Memory Read FIFO Almost Empty Level	R/W	Section 6.13.9
`0x8d80`	DDR Memory Read FIFO Almost Full Level	R/W	Section 6.13.10
`0x8d88`	DDR Memory Write FIFO Control	R/W	Section 6.13.6
`0x8d90 - 0x8d98`	Reserved	–	–
`0x8da0`	DDR Memory Write FIFO Flags Interrupt Mask	R/W	Section 6.13.8
`0x8da8`	DDR Memory Write FIFO Almost Empty Level	R/W	Section 6.13.9
`0x8db0`	DDR Memory Write FIFO Almost Full Level	R/W	Section 6.13.10
`0x8db8 - 0x8df8`	Reserved	–	–
User Registers			
`0x8e00`	Reserved	–	–
`0x8e08`	User Out FIFO Control	R/W	Section 6.14.1
`0x8e10`	User Out FIFO Flags Interrupt Mask	R/W	Section 6.14.3
`0x8e18`	User Out FIFO Almost Empty Level	R/W	Section 6.14.4
`0x8e20`	User Out FIFO Almost Full Level	R/W	Section 6.14.5
`0x8e28`	User In FIFO Control	R/W	Section 6.14.1
`0x8e30`	User In FIFO Flags Interrupt Mask	R/W	Section 6.14.3
`0x8e38`	User In FIFO Almost Empty Level	R/W	Section 6.14.4
`0x8e40`	User In FIFO Almost Full Level	R/W	Section 6.14.5
`0x8e48 - 0x8e78`	Reserved	–	–
`0x8e80 - 0x8f78`	User Signal FPGA Address Space	R/W	Section 4.9
`0x8f80 - 0x8ff8`	Reserved	–	–

* Offset from PCI Base Address Register 2 (**BAR2**), see Table 3–1

Chapter 4: Model 7642 Resource Operation

4.1 Overview

This chapter describes operation of the Model 7642 resources from the personal computer's PCI interface.

4.2 Analog to Digital Input Conversion

The Model 7642 is designed for a maximum input sampling frequency of 125 MHz. It accepts four analog RF inputs at +4dBm full scale into 50 ohms on front panel MMCX connectors. Each of the four inputs is transformer coupled, and digitized by a Linear Technology LTC2255 14–bit, 125–MHz A/D converter (ADC).

Refer to the Linear Technology LTC2255 data sheet (see Section 1.14) for description of the A/D converter operation. There are no user–programmable registers for the LTC2255 on the Model 7642.

The outputs of the A/Ds are delivered, in parallel, to the Signal FPGA through 14–bit paths. Each of these data paths is routed to an PCI Interface FIFO, as follows:

- A/D 1 output to ADC 1 FIFO
- A/D 2 output to ADC 2 FIFO
- A/D 3 output to ADC 3 FIFO
- A/D 4 output to ADC 4 FIFO

See Section 7.3, ADC Data Routing and Formatting, for additional information about the formatting of ADC FIFO data. See Section 4.4 for description of FIFO Operation.

Each A/D output may also be routed to the DDR2 Memory, see Section 4.6.

When TEST RAMP is enabled, all LTC2255 outputs to the ADC FIFOs and DDR Memory are replaced by a test pattern. See TEST RAMP in the Miscellaneous Control Register, Section 6.5.

The clock source for the A/D converters can be selected from one of two different sources, Sync Bus A or Sync Bus B. See Section 4.7 for description of the board's Timing and Synchronization.

4.3 Digital To Analog Output

The Model 7642 is designed with a maximum output sample rate of 320 MHz in upconverter mode and 500 MSPS in D/A only mode. One Texas Instruments 16–bit DAC5686 (DAC5687 with Option 101) is used to produce an analog output capable of operating in D/A only or quadrature modulation modes. The D/A has built–in interpolation filters settable to 2x, 4x, 8x, and 16x (16x is not provided in the DAC5687).

NOTE: The following refer only to the DAC5686. In all instances, the same operating descriptions apply to the DAC5687 with Option 101.

When operating as an upconverter, the DAC5686 interpolates and translates real or complex baseband input signals to any IF center frequency between DC and 160 MHz. It delivers real or quadrature (I+Q) analog outputs through a 320–MHz 16–bit D/A converter to a front panel MMCX connector at +4dBm full scale into 50 ohms.

In D/A only mode, the DAC5686 acts as a dual interpolating 16–bit D/A producing one output with sampling rates up to 500 MSPS. In modes that include frequency translation, the DAC5686 accepts real or complex inputs and produces an upconverted and interpolated, real or complex single output.

NOTE: The Model 7642 routes only the channel B output of the DAC5686 to the front panel DAC OUT connector. However, depending on the mode of operation, both DAC input data channels A and B may be used.

Refer to the Texas Instruments DAC5686 data sheet (see Section 1.14) for description of DAC operation. Setup and operation of the DAC5686 is controlled by a set of Signal FPGA registers, described in Section 6.12.

The Signal FPGA routes data from the DAC FIFO to DAC5686 channels A and B, depending on the packing mode selected. See Section 7.5, DAC Data Routing and Formatting, for additional information about these data packing modes and formats.

The clock and sync signals for the D/A converter can be selected from one of two different sources, Sync Bus A or Sync Bus B. See Section 4.7 for description of the board's Timing and Synchronization.

4.3 Digital To Analog Output (continued)

4.3.1 DAC5686 Digital Input Modes

Two digital data input modes are defined by the DAC5686. Data (I and Q) can be input to the DAC5686 from the Signal FPGA as separate parallel streams on two data buses (**Dual Bus Mode**), or as a single interleaved data stream on one data bus (**Interleaved Bus Mode**). Refer to the Texas Instruments DAC5686 data sheet (see Section 1.14) for detailed descriptions of these input modes.

The Signal FPGA provides different packing modes of DAC FIFO data that support these digital data input modes. Refer to Section 7.5, DAC Data Routing and Formatting, for additional information about these data packing modes and formats. See Section 4.4 for description of FIFO Operation.

4.3.2 DAC5686 Operating Modes

The DAC5686 provides two modes of operation: **Dual–channel Baseband I&Q Transmission** and **Quadrature Modulation**.

- In dual–channel baseband I&Q transmission mode, the DAC5686 provides two independent digital to analog conversion paths. In this mode, interpolation filtering increases the DAC update rate, which reduces sinx/x roll–off and enables the use of relaxed analog post–filtering.
- In quadrature modulation mode, on–chip mixing provides baseband–to–IF up–conversion. Mixing frequencies are flexibly chosen with a 32–bit programmable NCO. Channel carrier selection is performed at baseband by complex mixing in the Signal FPGA. Baseband I and Q from the FPGA are input to the DAC5686, which interpolates the low date rate signal to higher data rates. The single DAC output (channel B) is the final IF single–sideband spectrum presented to RF.

Refer to the Texas Instruments DAC5686 data sheet (see Section 1.14) for detailed descriptions of these operating modes.

4.4 FIFO Operation

There are ten user–programmable PCI Interface FIFOs in the Signal FPGA, which provides direct high–speed data transfer between the PCI interface and the various resources on the 7642.

NOTE: PCI bus interface FIFOs are 64 bits wide. In this manual, data formats are described in reference to 32–bit words. Thus, whenever data is read from or written to the FIFOs, it is formatted as two 32–bit words for each FIFO word.

- Each ADC, User, Memory, and Test FIFO is 8 KBytes deep (1K x 64 on PCI bus side, 2K x 32 on device side).
- The DAC FIFO is 64 KBytes deep (8K x 64 on PCI bus side, 16K x 32 on DAC side).

The following block diagram illustrates the interfaces and data flow for these FIFOs.

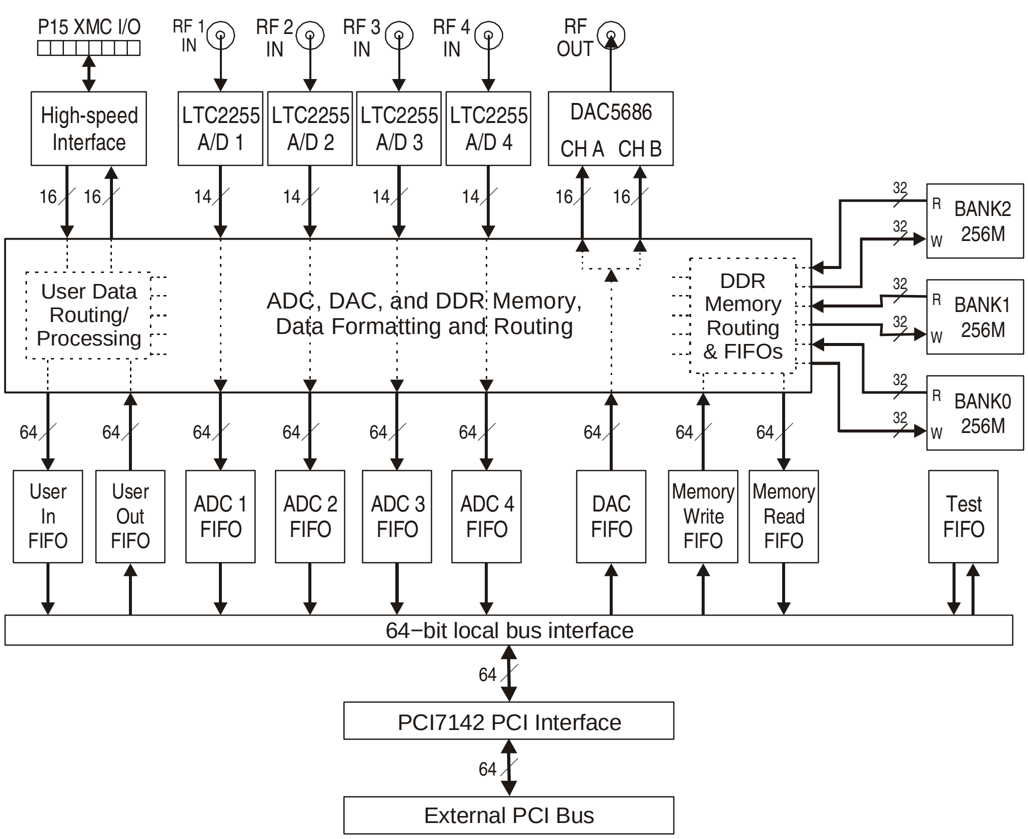


Figure 4–1: FIFO Block Diagram

The following pages identify the 7642 registers used for setting the FIFO boundary flag levels, enabling interrupts from FIFO flags, and monitoring FIFO status bits.

4.4 FIFO Operation (continued)

4.4.1 Initializing the FIFOs

1) First, set the programmable Almost Full and Almost Empty boundary flag levels using the applicable FIFO AE/AF Level Registers, as listed below.

Table 4−1: FIFO Almost Empty/Almost Full Count Registers

FIFO	Register	Reference
ADC 1 FIFO	ADC 1 FIFO AE/AF Levels	Sections 6.10.8, 6.10.9
ADC 2 FIFO	ADC 2 FIFO AE/AF Levels	
ADC 3 FIFO	ADC 3 FIFO AE/AF Levels	
ADC 4 FIFO	ADC 4 FIFO AE/AF Levels	
DAC FIFO	DAC FIFO E/AF Levels	Sections 6.11.7, 6.11.8
DDR Memory Read FIFO	DDR Memory Read FIFO AE/AF Levels	Sections 6.13.9, 6.13.10
DDR Memory Write FIFO	DDR Memory Write FIFO AE/AF Levels	
User Out FIFO	User Out FIFO AE/AF Levels	Sections 6.14.4, 6.14.5
User In FIFO	User In FIFO AE/AF Levels	
Test FIFO	Test FIFO AE/AF Levels	Sections 6.15.4, 6.15.5

2) Next, reset the FIFO using bit D01 of the applicable FIFO Control Register, listed below.

Table 4−2: FIFO Control Registers

FIFO	Register	Reference
ADC 1 FIFO	ADC 1 FIFO Control	Section 6.10.2
ADC 2 FIFO	ADC 2 FIFO Control	
ADC 3 FIFO	ADC 3 FIFO Control	
ADC 4 FIFO	ADC 2 FIFO Control	
DAC FIFO	DAC FIFO Control	Section 6.11.3
DDR Memory Read FIFO	DDR Memory Read FIFO Control	Section 6.13.6
DDR Memory Write FIFO	DDR Memory Write FIFO Control	
User Out FIFO	User Out FIFO Control	Section 6.14.1
User In FIFO	User In FIFO Control	
Test FIFO	Test FIFO Control	Section 6.15.1

NOTE: While the FIFO is in reset, you cannot read or write to the FIFO. You must clear the FIFO Reset bit (bit D01 of the applicable register, above) and then wait four clock cycles before you can write to the FIFO.

4.4 FIFO Operation (continued)

4.4.1 Initializing the FIFOs (continued)

3) Next, enable the desired boundary flag interrupts using the applicable FIFO Interrupt Mask Register, as listed below.

<table>
<tr><th colspan="3">Table 4−3: FIFO Interrupt Mask Registers</th></tr>
<tr><th>FIFO</th><th>Register</th><th>Reference</th></tr>
<tr><td>ADC 1 FIFO</td><td>ADC 1 FIFO Interrupt Mask</td><td rowspan="4">Section 6.10.7</td></tr>
<tr><td>ADC 2 FIFO</td><td>ADC 2 FIFO Interrupt Mask</td></tr>
<tr><td>ADC 3 FIFO</td><td>ADC 3 FIFO Interrupt Mask</td></tr>
<tr><td>ADC 4 FIFO</td><td>ADC 4 FIFO Interrupt Mask</td></tr>
<tr><td>DAC FIFO</td><td>DAC FIFO Interrupt Mask</td><td>Section 6.11.6</td></tr>
<tr><td>DDR Memory Read FIFO</td><td>DDR Memory Read FIFO Interrupt Mask</td><td rowspan="2">Section 6.13.8</td></tr>
<tr><td>DDR Memory Write FIFO</td><td>DDR Memory Write FIFO Interrupt Mask</td></tr>
<tr><td>User Out FIFO</td><td>User Out FIFO Interrupt Mask</td><td rowspan="2">Section 6.14.3</td></tr>
<tr><td>User In FIFO</td><td>User In FIFO Interrupt Mask</td></tr>
<tr><td>Test FIFO</td><td>Test FIFO Interrupt Mask</td><td>Section 6.15.3</td></tr>
</table>

4) Next, select the source and characteristics of the gates used to control the FIFO writes using the applicable FIFO Control Register, as listed below.

<table>
<tr><th colspan="3">Table 4−4: FIFO Control Registers</th></tr>
<tr><th>FIFO</th><th>Register</th><th>Reference</th></tr>
<tr><td>ADC 1 FIFO</td><td>ADC 1 FIFO Control</td><td rowspan="4">Section 6.10.2</td></tr>
<tr><td>ADC 2 FIFO</td><td>ADC 2 FIFO Control</td></tr>
<tr><td>ADC 3 FIFO</td><td>ADC 3 FIFO Control</td></tr>
<tr><td>ADC 4 FIFO</td><td>ADC 4 FIFO Control</td></tr>
<tr><td>DAC FIFO</td><td>DAC FIFO Control</td><td>Section 6.11.3</td></tr>
<tr><td>DDR Memory Read FIFO</td><td>DDR Memory Read FIFO Control</td><td rowspan="2">NA</td></tr>
<tr><td>DDR Memory Write FIFO</td><td>DDR Memory Write FIFO Control</td></tr>
<tr><td>User Out FIFO</td><td>User Out FIFO Control</td><td rowspan="2">NA</td></tr>
<tr><td>User In FIFO</td><td>User In FIFO Control</td></tr>
<tr><td>Test FIFO</td><td>Test FIFO Control</td><td>NA</td></tr>
</table>

4.4 FIFO Operation (continued)

4.4.1 Initializing the FIFOs (continued)

5) Next, if using trigger mode, set the Trigger Length using the applicable FIFO Trigger Length Registers, as listed below.

Table 4–5: FIFO Trigger Length Registers

FIFO	Register	Reference
ADC 1 FIFO	ADC 1 FIFO Trigger Length	Section 6.10.6
ADC 2 FIFO	ADC 2 FIFO Trigger Length	
ADC 3 FIFO	ADC 3 FIFO Trigger Length	
ADC 4 FIFO	ADC 4 FIFO Trigger Length	
DAC FIFO	DAC FIFO Trigger Length	Section 6.11.5
DDR Memory Read FIFO	DDR Memory Read FIFO Trigger Length	NA
DDR Memory Write FIFO	DDR Memory Write FIFO Trigger Length	
User Out FIFO	User Out FIFO Trigger Length	NA
User In FIFO	User In FIFO Trigger Length	
Test FIFO	Test FIFO Trigger Length	NA

6) Finally, enable the FIFO operation using the applicable FIFO Control Register, as listed below.

Table 4–6: FIFO Control Registers

FIFO	Register	Reference
ADC 1 FIFO	ADC 1 FIFO Control	Section 6.10.2
ADC 2 FIFO	ADC 2 FIFO Control	
ADC 3 FIFO	ADC 3 FIFO Control	
ADC 4 FIFO	ADC 4 FIFO Control	
DAC FIFO	DAC FIFO Control	Section 6.11.3
DDR Memory Read FIFO	DDR Memory Read FIFO Control	Section 6.13.6
DDR Memory Write FIFO	DDR Memory Write FIFO Control	
User Out FIFO	User Out FIFO Control	Section 6.14.1
User In FIFO	User In FIFO Control	
Test FIFO	Test FIFO Control	Section 6.15.1

4.4 FIFO Operation (continued)

4.4.2 FIFO Data Transfers

Following FIFO Initialization as described in Section 4.4.1, the FIFO is ready for data transfers. The PCI interface performs FIFO reads and writes using DMA operation. Refer to Section 4.5 on the next page for description of 7642 DMA Operation.

PCI interface data reads occur from an Incoming FIFO and writes occur to an Outgoing FIFO. Read and write operations will cause the appropriate FIFO status flags to be set and cleared when the data reaches the pro–grammed boundary conditions.

You can monitor the FIFO operation using the applicable FIFO Status Regis–ter, as listed below.

Table 4–7: FIFO Status Registers

FIFO	Register	Reference
ADC 1 FIFO	ADC 1 FIFO Status	Section 6.10.3
ADC 2 FIFO	ADC 2 FIFO Status	
ADC 3 FIFO	ADC 3 FIFO Status	
ADC 4 FIFO	ADC 4 FIFO Status	
DAC FIFO	DAC FIFO Status	Section 6.11.4
DDR Memory Read FIFO	DDR Memory Read FIFO Status	Section 6.13.7
DDR Memory Write FIFO	DDR Memory Write FIFO Status	
User Out FIFO	User Out FIFO Status	Section 6.14.2
User In FIFO	User In FIFO Status	
Test FIFO	Test FIFO Status	Section 6.15.2

Read and write operations will also cause the appropriate FIFO status flags to assert PCI interrupts if enabled. See Section 4.8 for description of Inter–rupt Operation using the FIFO flag bits.

See Section 7.3, ADC Data Routing and Formats, for additional information about the formatting of ADC FIFO data. See Section 7.4, ADC FIFO to DDR Memory Routing, for additional information about DDR Memory formats. See Section 7.5, DAC Data Routing and Formats, for additional information about DAC FIFO packing modes and formats.

NOTE: The User Out and In FIFOs are not connected to any board resources in the default 7642 configuration. The user may route FPGA signals through these FIFOs with custom FPGA logic using the available Pentek GateFlow® FPGA Design Kit, Model 4953 Option 142 (see Section 1.12).

4.5 DMA Operation

The PCI interface performs FIFO reads and writes using DMA operation. The PCI7142 incorporates nine high–performance DMA channels. The nine DMA channels (DMA0 through DMA8) are assigned to the PCI Interface FIFOs (Section 4.4) as shown in the following table:

Table 4–8: DMA Use by FIFOs

DMA Channel	FIFO	Direction *
DMA0	ADC 1 FIFO	Incoming
DMA1	ADC 2 FIFO	Incoming
DMA2	ADC 3 FIFO	Incoming
DMA3	ADC 4 FIFO	Incoming
DMA4	User In FIFO	Incoming
DMA5	Reserved	Incoming
DMA6	DAC FIFO	Outgoing
DMA7	User Out FIFO	Outgoing
DMA8	DDR Memory Read FIFO	Incoming

* The FIFO direction, Incoming or Outgoing, is labeled in reference to the controlling PCI interface processor; that is, 'Incoming' refers to data flow from the 7642 devices to the PCI interface

Each DMA channel has the capability to transfer data between PCI devices, SDRAM, or 7642 resources. The DMA uses the FIFOs for temporary DMA data storage. These FIFOs allows all DMA channels to work concurrently since each channel utilizes a separate FIFO. For example, DMA0 transfers data from ADC channel 1 to the PCI bus using one FIFO, while DMA6 transfers data from the PCI bus to the DAC output channel using another FIFO.

The DMA channels are programmable from the PCI bus using DMA Descriptors. Each DMA channel has four DMA Descriptors that can be chained. Each DMA Descriptor is loaded by the DMA controller into the channel's working registers while the DMA transaction is active.

DMA Demand mode is the recommended mode of DMA operation for the FIFOs, as described in Section 4.5.1.

<u>NOTE:</u> In addition to the nine DMA channels identified above, a separate DMA channel exists for DMA transfers from the PCI bus to the Signal FPGA. This FPGA Load DMA operation is described in Section 4.5.5.

4.5 DMA Operation (continued)

4.5.1 DMA Demand Mode Operation

DMA demand mode operation uses the FIFO flags to gate the DMA controller on and off to ensure no loss of data.

To use DMA Demand Mode, set the following programmable flags and values:

- In the DMA Command Status Register, Section 5.17, set Demand Mode to 1 (Enable), set Burst Mode to 1 (Enable), and set the Maximum Burst Count greater than 0.
- In the applicable FIFO Almost Full register (Section 6.10.9 for ADC FIFOs, Section 6.11.8 for DAC FIFOs), set the FIFO Almost Full level greater than the burst length.
- In the applicable FIFO Almost Empty register (Section 6.10.8 for ADC FIFOs, Section 6.11.7 for DAC FIFOs), set the FIFO Almost Empty level to the Almost Full level (above) minus 32.

The following shows the Pentek–recommended settings for each FIFO.

Table 4–9: Recommended Demand Mode Settings

FIFO	Max. Burst Count	Almost Full	Almost Empty
ADC FIFOs	512	544	512
DAC FIFOs	2048	6176	6144

Demand Mode operates as follows when set up as described above:

a) Reading from an ADC FIFO – When the ADC FIFO Almost Full condition occurs, a DMA Request is asserted to transfer data from the FIFO to the programmed destination address. When the Almost Empty occurs, the DMA Request is cleared.

b) Writing to a DAC FIFO – When the DAC FIFO Almost Empty condition occurs, a DMA Request is asserted to transfer data from the programmed source address to the FIFO. When the Almost Full occurs, the DMA Request is cleared.

Once the DMA transfers starts it will run until the Burst Count completes.

Figure 4–2 on the next page illustrates the DMA operation signal timing.

4.5 DMA Operation (continued)

4.5.1 DMA Demand Mode Operation (continued)

Figure 4–2 below illustrates the DMA operation signal timing.

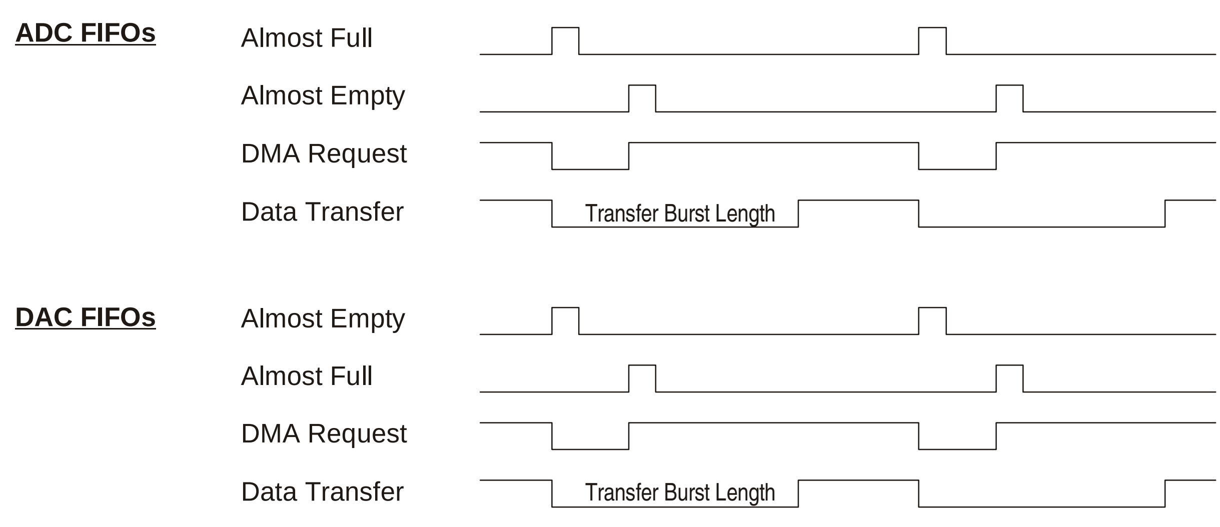


Figure 4–2: FIFO DMA Timing (2 cycles)

4.5.2 DMA Non–Demand Mode Operation

To use non–demand mode DMA operation, set the following programmable flags and values:

- In the DMA Command Status Register, Section 5.17, set Demand Mode to 0 (Disable), set Burst Mode to 1 (Enable), and set the Transfer Count Register, Section 5.19, equal to or less than the size of the applicable FIFO.
- Set the Maximum Burst Count in the DMA Command Status Register, Section 5.17, for non–demand mode as follows:
  - If the Maximum Burst Count = 0, the burst length will be the size of the Transfer Count.
  - If the Maximum Burst Count is greater than 0, the Burst Count (and Transfer Count) must be equal to or less than the size of the applicable FIFO.

4.5 DMA Operation (continued)

4.5.3 DMA Channel Programming

Each DMA channel can programmed for separate, independent DMA transfers using the associated DMA Command/Status Register, Section 5.17, DMA Transfer Interval Counter Register, Section 5.18, and DMA Descriptor Registers. There are four Descriptors for each DMA channel, identified as Descriptor 0 through 3, and there are two registers for each descriptor, the DMA Descriptor Transfer Count Register, Section 5.19, and the DMA Descriptor PCI Address Register, Section 5.20.

To setup a DMA channel for operation (note that each set of registers used must be for the same DMA channel):

1) Use the DMA Command/Status Register, Section 5.17, to set the operating mode for the channel.

2) Use the DMA Descriptor Transfer Count Registers, Section 5.19, and the DMA Descriptor PCI Address Registers, Section 5.20, to specify the transfer count and PCI (source or destination) address for each Descriptor to be used for this data transfer. If using several descriptors, be sure to set the Chain bit D31=1 in each Descriptor Transfer Count Register. Setting bit D31=1 in the last Descriptor will cause the chain to go back to Descriptor 0.

3) If DMA interrupts are desired, enable the applicable interrupt bits in the PCI Interrupt Enable Register, Section 5.3. The types of DMA interrupts that can be generated by a DMA Channel are described in Section 4.5.4.

   If a DMA Chain Finish interrupt is desired, you must also enable the end of chain interrupt (bit D30) for that DMA channel in the DMA Descriptor Transfer Count Registers, Section 5.19.

4) Start the DMA operation using the DMA Command/Status Register, Section 5.17.

5) Use the DMA Current Transfer Count Register, Section 5.15, and the DMA Current PCI Address Register, Section 5.16, to monitor the progress of the data transfer.

4.5 DMA Operation (continued)

4.5.4 DMA Interrupts

Three types of DMA completion interrupts can be generated during DMA operation. Each DMA channel (0 through 8) has a separate set of bits for generating, enabling, and monitoring each of these interrupts.

- DMA Chain Finish

  This interrupt is generated at the end of the last Descriptor transfer specified in the DMA Descriptor Transfer Count Registers, Section 5.19, when the Chain bit D31 is set to '0', End of Chain. This signifies completion of the entire linked chain of data transfers.

- DMA Descriptor Finish

  This interrupt is generated at the end of the Descriptor transfer count specified in each DMA Descriptor Transfer Count Register, Section 5.19. This signifies completion of just one of the Descriptor's data transfer.

- DMA Xfer Interval Count Finish

  This interrupt is generated at the end of the transfer interval count specified in the DMA Transfer Interval Counter Register, Section 5.18. This signifies completion of a designated portion of the data transfer. The DMA Transfer Interval Counter Register records a count of these interrupts.

  Transfer Interval Count is specified in words. Words are either 32 or 64 bits wide, as specified by Data Width, bit D07, in the DMA Command/Status Register, Section 5.17.

When any of one of these conditions occurs, the applicable Flag bit is set in the PCI Interrupt Flag Register, Section 5.2, and an interrupt is generated if enabled by the associated PCI Interrupt Enable Register, Section 5.3.

In addition, an interrupt can be generated during DMA operation if a buffer overrun has occurred on any DMA channel. When this condition occurs, the applicable Flag bit is set in the DMA PCI Interrupt Enable/Flag Register, Section 5.11, and a PCI interrupt is generated to the PCI bus if enabled by the associated Enable bit in this register.

See Section 4.8 for description of the board's Interrupt Operation.

4.5 DMA Operation (continued)

4.5.5 FPGA Load DMA Operation

NOTE: FPGA Load DMA operation is available only in 7142 boards with a PCI FPGA revision date greater that 10/01/07 (see Section 5.6).

In addition to the nine DMA channels described in the previous sections, a separate DMA channel is provided for DMA transfers from the PCI bus to the Signal FPGA. This DMA channel receives 32–bits words from the PCI bus, each containing four 8–bit FPGA configuration data words. The DMA channel uses a 1K FIFO for temporary data storage. The FIFO receives 32–bit data from the DMA engine and transmits 8–bit words to the Signal FPGA at the same data rate as the PCI bus.

The following figure illustrates the flow of data from the PCI interface to the Signal FPGA using this DMA channel.

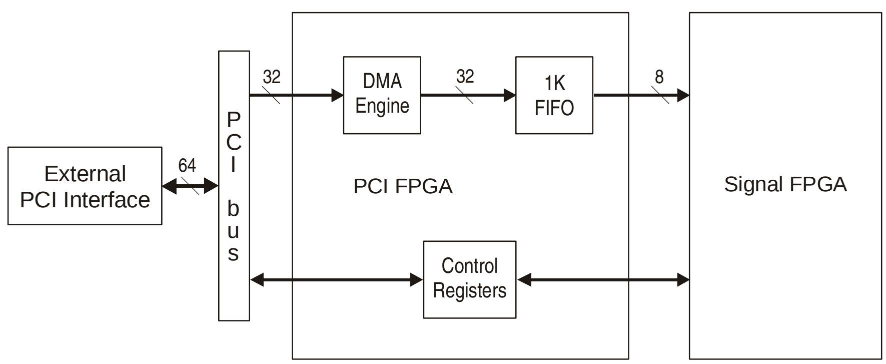


Figure 4–3: FPGA Load DMA Block Diagram

To program the FPGA Load DMA channel for operation:

1) Use the Virtex Config Register, Section 5.7, to set the Signal FPGA DATA SRC to the DMA FIFO.

2) Use the DMA Transfer Count Register, Section 5.22, and the FPGA DMA PCI Address Register, Section 5.23, to specify the transfer count and PCI address of the data to be loaded to the Signal FPGA.

(Continued on the next page.)

4.5 DMA Operation (continued)

4.5.5 FPGA Load DMA Operation (continued)

3) If a DMA interrupt is desired, enable the FPGA DMA DONE EN bit in the DMA PCI Interrupt Flag/Enable Register, Section 5.11.

4) Start the DMA operation using the FPGA Load DMA Command/Status Register, Section 5.21.

5) Use the DMA Status bit of the FPGA Load DMA Command/Status Register, Section 5.21 to monitor the progress of the data transfer.

For a single data block transfer, the Signal FPGA upload is complete. For multiple block transfers, continue on the next page.

If you use multiple blocks, you must wait until each transfer has completed before starting the next block DMA transfer, as follows (continuing from Step 5 above).

6) Wait until the prior DMA transfer has completed, as indicated by DMA Status = 1 (Done), bit D04 in the FPGA Load DMA Command/Status Register, Section 5.21.

7) Use the DMA Transfer Count Register, Section 5.22, and the FPGA DMA PCI Address Register, Section 5.23, to specify the transfer count and PCI address of the next block of data to be loaded to the FPGA.

8) Start the DMA operation using the FPGA Load DMA Command/Status Register, Section 5.21.

9) Use the DMA Status bit of the FPGA Load DMA Command/Status Register, Section 5.21 to monitor the progress of the data transfer.

If there are more blocks to transfer, repeat the above steps starting at Step 6.

NOTE: All other DMA channels should be idle while you are transferring data to the Signal FPGA with this DMA channel.

4.6 DDR Memory Operation

Three independent banks of DDR2 SDRAM are available from the Signal FPGA to sup– port transient capture or popular applications such as tracking receivers. Each bank is 256 MBytes for a total of 768 MBytes of available memory.

The following is a simplified block diagram of the DDR Memory interfaces and control logic. Use the DDR Memory Registers, described in Section 6.13, to setup and control the operation of all DDR Memory data transfers.

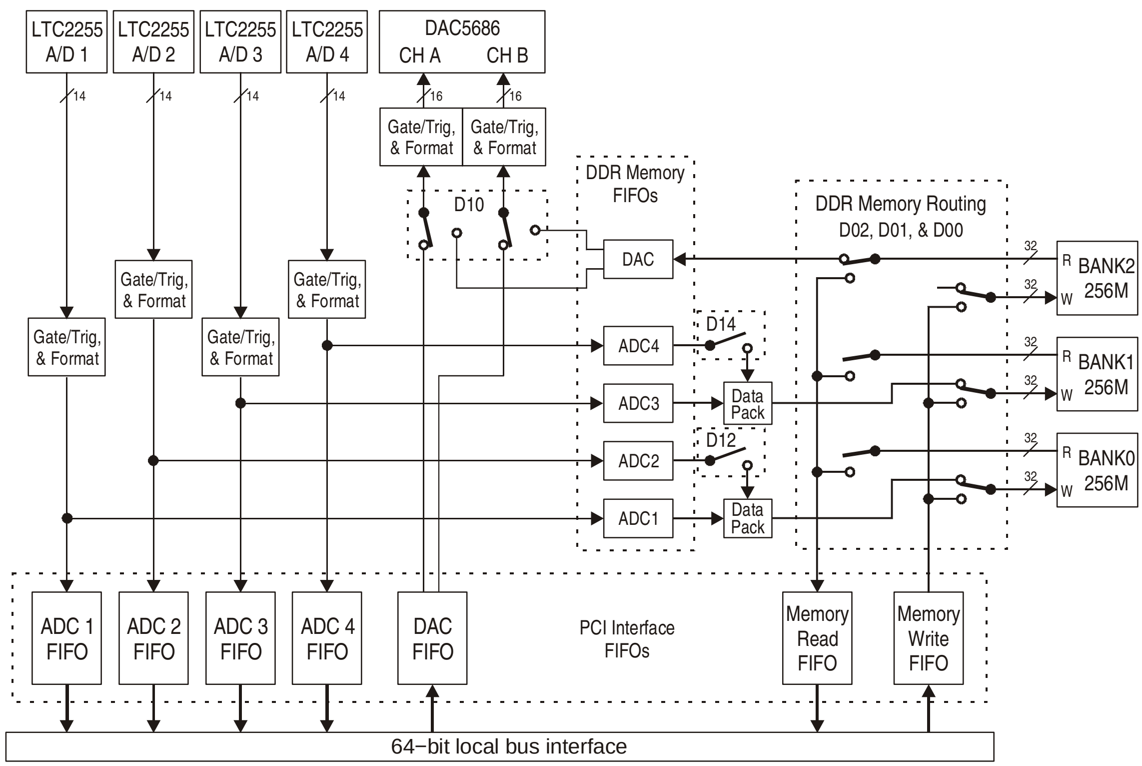


Figure 4−4: DDR Memory Control Logic
(DDR Memory Control Register bits indicated by Dxx − default position shown)

NOTE: The registers used for programming the PCI Interface ADC and DAC FIFOs (Section 4.4) also set the same programming parameters for the associated DDR Memory ADC and DAC FIFOs.

4.6 DDR Memory Operation (continued)

4.6.1 DDR Memory FIFOs

There are five DDR Memory FIFOs; each is 8 KBytes (2 KWord by 32 bits). These FIFOs provide direct high–speed data transfers between the ADC inputs and DDR Memory, or between DDR Memory and the DAC output, when enabled by the DDR Memory Control Register, Section 6.13.1.

NOTE: The DDR Memory interface FIFOs are 32 bits wide. Thus, whenever data is read from or written to these FIFOs, it is formatted as a single 32–bit word.

Figure 4–4 on the prior page illustrates the data flow for the DDR Memory FIFOs. The following table lists the DDR Memory FIFOs and their use.

Table 4–10: DDR Memory FIFOs Use

DDR Memory FIFO	FIFO Use
ADC1 FIFO	From LTC2255 A/D Channel 1 to Bank 0
ADC2 FIFO	From LTC2255 A/D Channel 2 to Bank 0
ADC3 FIFO	From LTC2255 A/D Channel 3 to Bank 1
ADC4 FIFO	From LTC2255 A/D Channel 4 to Bank 1
DAC FIFO	From Bank 2 to DAC5686

When the ADC2 and/or ADC4 channels are enabled by the BANK x PACK bits of the DDR Memory Control Register, Section 6.13.1, data packing is per–formed by interleaving successive 16–bit words from the two associated ADC FIFOs to the respective DDR Memory Bank (ADC1 and 2 to BANK 0, or ADC3 and 4 to BANK 1). See Section 7.4, ADC FIFO to DDR Memory Routing, for additional information about data packing from the DDR Memory ADC FIFOs to DDR Memory.

The registers used for programming the PCI Interface ADC and DAC FIFOs (see Section 4.4) also set the same programming parameters for the associ–ated DDR Memory ADC and DAC FIFOs.

4.6.2 DDR Memory Triggering

There are two types of triggering modes provided for data capture using the DDR Memory FIFOs, Pre–Trigger and Post–Trigger. These triggering modes are described in Section 4.7.4.

4.7 Timing and Synchronization

The Model 7642 provides two internal timing buses, Sync Bus A and Sync Bus B, which provide two independent sets of clock, sync, and gate signals. The clock, sync, and gate signals can go out or in on the front panel LVDS Sync Bus connector. This connector also provides two TTL Gate/Sync inputs that can be used by either Sync Bus. The front panel also has one MMCX microminiature coaxial socket receptacle for input of an external sample clock.

See Figure 4–5 on the next page for a simplified logic diagram of the clock, sync, and gate signals. See Figure 4–6 on page 74 for the timing delays introduced by the circuit elements in these signals' paths.

The front panel LVDS Sync Bus includes two independent sets of clock, sync, and gate signals. The multi–pin connector (Section 2.6.4) has signals for Bus A on the upper pins and Bus B on the lower pins. It allows one Model 7642 to act as a bus master, driving the sample clock, sync, and gate signals out using LVDS differential signaling. Additionally, the front panel cable can be split to allow board–to–board cabling of two independent buses: Bus A and Bus B. Up to seven slave 7642's can be driven on each bus, supporting synchronous sampling and sync functions across all connected boards. If more than seven boards need to be synchronized, Pentek's Model 9190 Clock and Sync Generator allows synchronization of up 80 I/O modules.

Each of the front panel sync buses is connected through LVDS drivers and receivers to the board's internal sync buses. On each board, clock and sync signals for the A/D can be driven from either Bus A or Bus B. Similarly the D/A section of each board can be driven from either sync bus, allowing the A/D, receiver, and upconverter sections to operate asynchronously if a different sync bus is used for each, or for the A/D, receiver, and upconverter sections to operate synchronously if the same bus is chosen for both.

Setup and operation of the Sync Buses is controlled by a set of Signal FPGA registers, described in Section 6.8.

4.7.1 Syncs

A Sync signal is routed to the Phase Synchronization input (PHSTR) on the DAC5686. This sync can be driven directly from the front panel LVDS Sync Bus SYNC A or B signal (Section 2.6.4) if the Model 7642 is a Sync Bus Slave, or from a sync signal generated on the board.

Each onboard sync can be selected from a register write (Sync A/B Generator Register, Section 6.8.2 or Section 6.8.5), or the external TTL SYNC input (Section 2.6.4) can be selected for both syncs. The polarity of the external TTL SYNC input is programmable.

When the Model 7642 is a Sync Bus Master, the onboard generated sync is output to the respective LVDS Sync Bus.

4.7 Timing and Synchronization (continued)

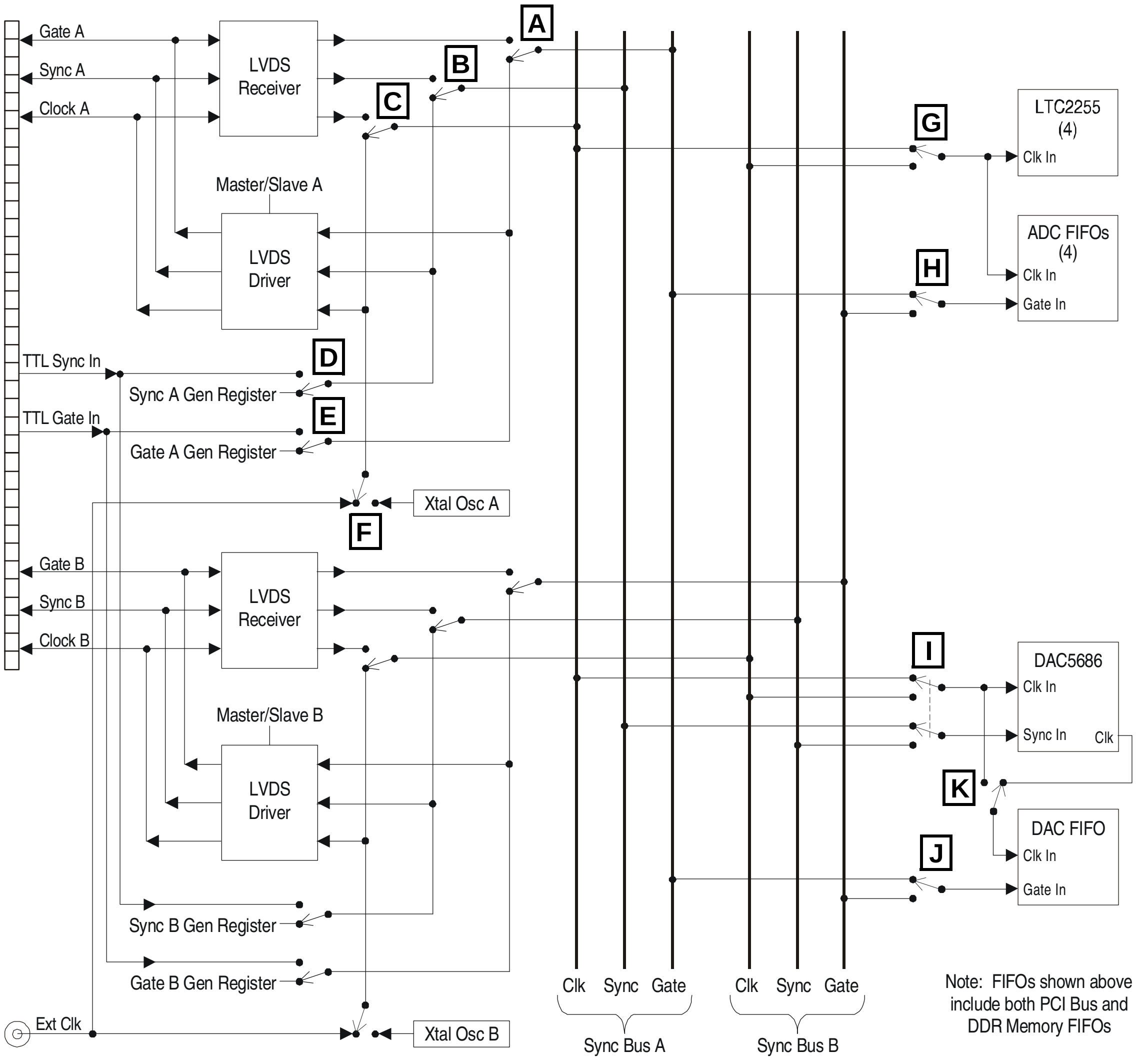


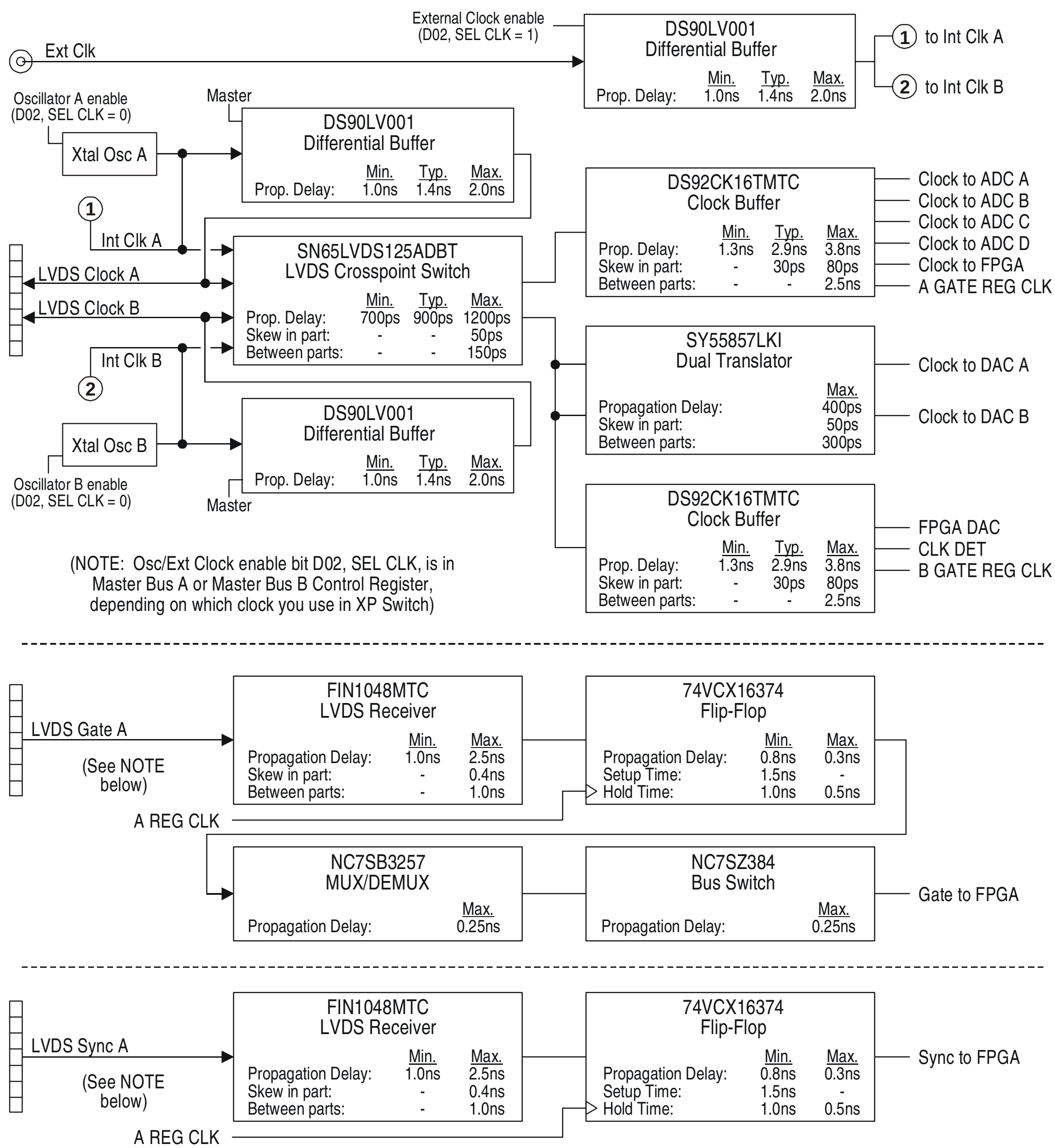
Figure 4−5: Gate/Sync/Clock Logic Diagram

Switches:
- **A** − Master Bus Control Register, Section 6.8.1, bit D09, GATE SRC
- **B** − Master Bus Control Register, Section 6.8.1, bit D06, SYNC SRC
- **C** − Master Bus Control Register, Section 6.8.1, bit D04, CLK SRC
- **D** − Master Bus Control Register, Section 6.8.1, bit D08, SEL GATE
- **E** − Master Bus Control Register, Section 6.8.1, bit D05, SEL SYNC
- **F** − Master Bus Control Register, Section 6.8.1, bit D02, SEL CLK
- **G** − ADC Sync Bus Select Register, Section 6.10.1, bit D00, SEL
- **H** − ADC FIFO Control Register, Section 6.10.2, bit D07, GATE SEL
- **I** − DAC Sync Bus Select Register, Section 6.11.1, bit D00, SEL
- **J** − DAC FIFO Control Register, Section 6.11.3, bit D07, GATE SEL
- **K** − DAC Control/Status Register, Section 6.11.2, bit D03, DAC CLK

(Only Sync Bus A references illustrated − See Section 6.8.4 for Sync Bus B control bits)

4.7 Timing and Synchronization (continued)

The following diagram shows the timing delays introduced by the circuit elements in the Clock, Gate, and Sync signal paths.



NOTE: LVDS Gate and Sync signal paths illustrated only for ‘A’ signal inputs.
LVDS ‘B’ signal inputs are similar, with ‘B’ notation in place of ‘A’.

Figure 4−6: Gate/Sync/Clock Signal Path Timing

4.7 Timing and Synchronization (continued)

4.7.2 Clocks

The clocks for all board functions (LTC2255 A/Ds, DAC5686, and Signal FPGA FIFOs) can be driven from the front panel LVDS Sync Bus CLK A or B signal (Section 2.6.4) if the Model 7642 is a Sync Bus Slave, or from a clock generated on the board.

Each onboard clock can be selected from the onboard crystal oscillators (selected by the Master Bus A/B Control Register, Section 6.8.1 or 6.8.4), or from the external MMCX EXT CLK input (Section 2.6.2).

When the Model 7642 is a Sync Bus Master, the onboard clock is output to the respective LVDS Sync Bus.

4.7.3 Gates

Gates are used to enable writes to the PCI bus interface FIFOs and the DDR Memory FIFOs. Gate selection is controlled by the Master Bus A/B Control Register, Section 6.8.1 or 6.8.4, and by the individual FIFO Control Registers, Sections 6.11.3 (DAC FIFO) and 6.10.2 (ADC FIFOs). Gates may be individually disabled, in which case FIFO writes are always enabled.

Each gate can be driven directly from the front panel LVDS Sync Bus GATE A or B signal (Section 2.6.4) if the Model 7642 is a Sync Bus Slave, or from a gate signal generated on the board.

Each onboard generated gate can be selected from a register write (Gate A/B Generator Register, Section 6.8.3 or 6.8.6), or the external TTL GATE input (Section 2.6.4) can be selected for both gates. The polarity of each external TTL GATE input is programmable.

When the Model 7642 is a Sync Bus Master, the gate generated onboard is output to the respective LVDS Sync Bus.

Each gate may be programmed to act as a trigger (Trigger mode) using the FIFO Control Registers, Sections 6.11.3 and 6.10.2. In this case, the gate is generated after a desired polarity logic transition on the external gate source or register write, and the resulting gate continues either indefinitely (Hold mode) or for a programmed number of FIFO writes up to 16,383 (Trigger Length). At any time the triggered gate may be asynchronously disabled by a control register write.

When Trigger mode is selected for a Gate signal, Pre– or Post–Triggering can be used for certain FIFOs, as described in the following subsection.

4.7 Timing and Synchronization (continued)

4.7.4 Pre/Post–Triggering

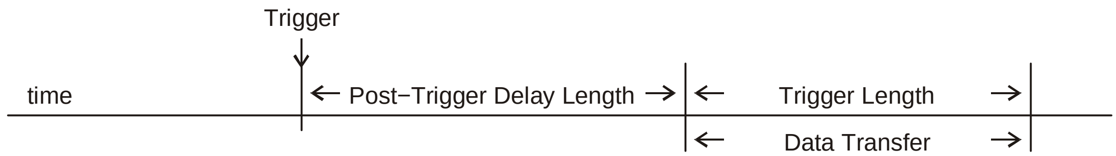
When Trigger mode is selected for a Gate signal, Pre– or Post–Triggering can be used for certain FIFO data transfers. Use the applicable Trigger Length Registers, Sections 6.10.6 and 6.11.5, to specify the trigger length.

Do not use Trigger HOLD mode with these functions.

- ❑ Post–Triggering (the default mode, ADC and DAC FIFOs)

  Post–Triggering allows you to specify a number of data samples to delay, after receipt of a trigger, before the data is actually transferred (written to or read from the associated FIFO). At the end of this delay, the data samples are transferred until the number of samples defined by the Trigger Length are transferred. Use the Post Trigger Delay Length Registers, Section 6.10.4 or 6.11.9, to specify the post–trigger delay.

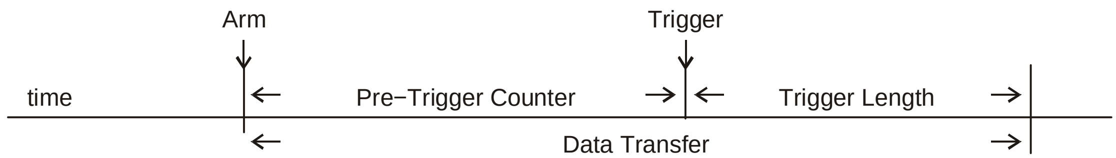
  Post–Triggering is illustrated below.



- ❑ Pre–Triggering (ADC FIFOs only)

  Pre–Triggering allows you to transfer data samples (write to the ADC FIFO) before receipt of the trigger. This transfer starts when you ARM the trigger with the ADC FIFO Control Register, Section 6.10.2. After the trigger is received, the data samples continue to be transferred until the number of samples defined by the Trigger Length are transferred, in addition to the Pre–Trigger count. Use the Pre Trigger Count Capture Registers, Section 6.10.5, to determine the actual number of A/D data samples stored in the ADC FIFO before receipt of the trigger.

  Pre–Triggering is illustrated below.



  Pre–Triggering is provided only for the ADC FIFOs, and should only be used for ADC writes to the DDR Memory, Section 4.6.

4.8 Interrupt Operation

The 7642 PMC module can generate four PCI interrupt outputs, INTA through INTD, to the PCI interface. Refer to the Operating Manual supplied with your PCI interface processor for description of the interrupt response operation.

The 7642 can generate four Local bus interrupt outputs, LINTo0 through LINTo3, on the local PCI bus from 7642 resources.

The following figure illustrates the interrupt signal routing on the Model 7642.

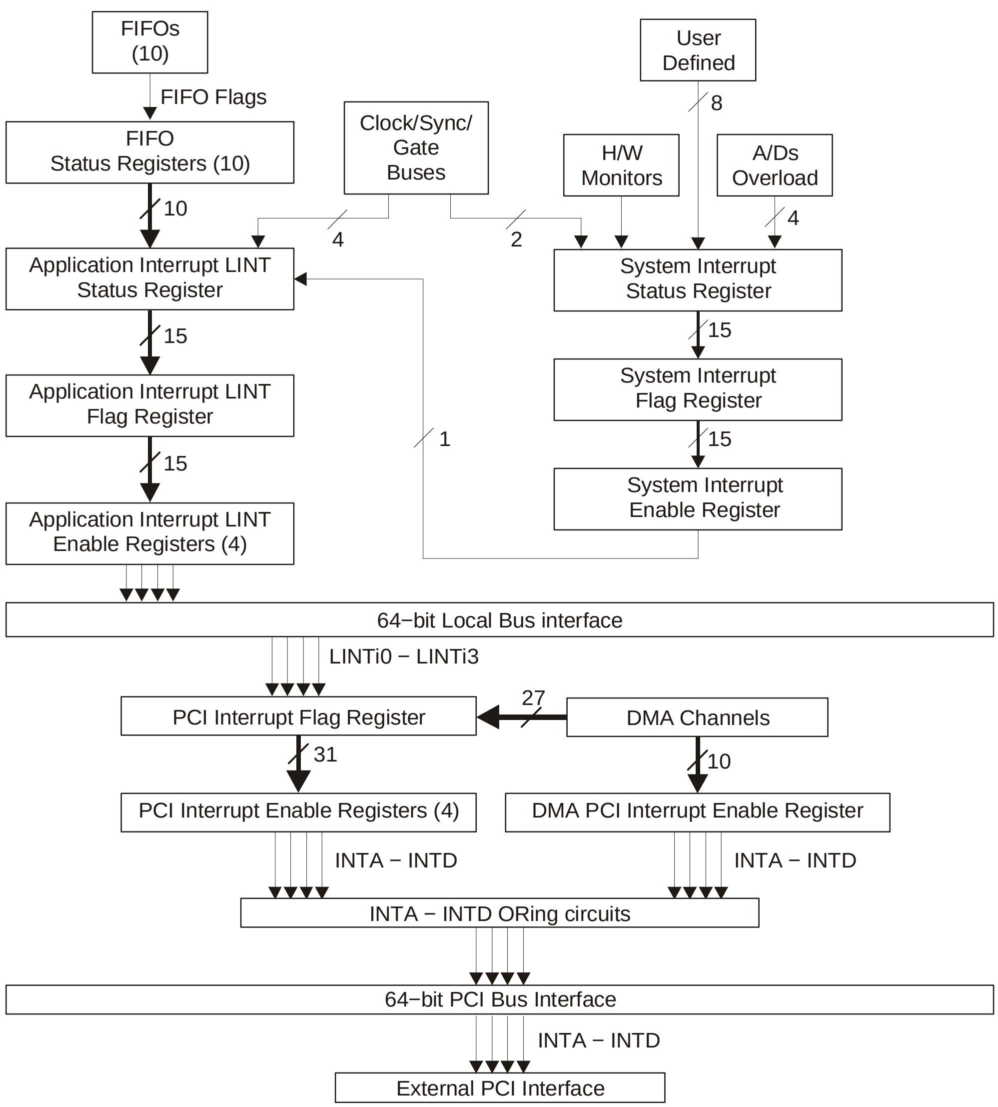


Figure 4−7: Interrupt Signal Logic Diagram

4.8 Interrupt Operation (continued)

4.8.1 System Interrupts

The System Interrupt Registers (Sections 6.9.1 to 6.9.3) contain interrupt status, flag, and enable bits for several system interrupt conditions. The System Interrupts include the following signals/conditions:

- VOLT/TEMP – This interrupt is associated with a temperature/voltage limit interrupt from the LM83 or ADM1024 sensors. (To determine the sensor causing the interrupt, use the TWSI Port Register, Section 6.3.)
- CLK A LOSS, CLK B LOSS – Each of these interrupts is associated with a Clock loss interrupt from one of the 7642 Sync Buses, A or B.
- OVLD AD1, OVLD AD2, OVLD AD3, OVLD AD4 – Each of these interrupts is associated with an analog input overload interrupt from one of the four A/D converters.
- User Defined – There are eight user–defined interrupts that can be programmed for custom FPGA core applications.

In response to these interrupt signals, the System Interrupt Enable Register, Section 6.9.1, can be enabled to assert an interrupt to the Application Interrupt Registers, identified in Section 4.8.2 below.

4.8.2 Application Interrupts

The set of Application Interrupt Registers (Sections 6.9.4 to 6.9.6) contain interrupt status, flag, and enable bits for the FIFO Flags and the System Interrupt conditions. The Application Interrupts include the following signals/conditions:

- SYS INT – This interrupt is associated with an interrupt from the System Interrupt Enable Register, identified in Section 4.8.1 above.
- GATE B, GATE A – Each of these interrupts is associated with a Sync Bus GATE (A or B) interrupt. The interrupt occurs at the start of the respective gate signal.
- SYNC B, SYNC A – Each of these interrupts is associated with a Sync Bus SYNC (A or B) interrupt. The interrupt occurs at the start of the respective Sync pulse.

4.8 Interrupt Operation (continued)

4.8.2 Application Interrupts (continued)

- TRIGGER CAPTURE – This interrupt is associated with an interrupt from the OR'ed result of the ADC and DAC FIFO Trigger Capture bits. Use the applicable FIFO Status Registers, Sections 6.10.3 and 6.11.4, to determine the interrupting FIFO.

- DDR MEMORY READ FIFO – This interrupt is associated with a FIFO flag condition from the DDR Memory Read FIFO. Use the DDR Memory Read FIFO Status Registers, Section 6.13.7, to determine the interrupting condition from this FIFO.

- DDR MEMORY WRITE FIFO – This interrupt is associated with a FIFO flag condition from the DDR Memory Write FIFO. Use the DDR Memory Write FIFO Status Registers, Section 6.13.7, to determine the interrupting condition from this FIFO.

- DAC FIFO – This interrupt is associated with a FIFO flag condition from the DAC FIFO. Use the DAC FIFO Status Register, Section 6.11.4, to determine the interrupting condition from the FIFO.

- ADC 1 FIFO, ADC 2 FIFO, ADC 3 FIFO, ADC 4 FIFO – Each of these interrupts is associated with a FIFO flag condition from one of the ADC FIFOs. Use the ADC FIFO Status Registers, Section 6.10.3, to determine the interrupting condition from the associated FIFO.

- User Out FIFO, User In FIFO – Each of these interrupts is associated with a FIFO flag condition from one of the User FIFOs (Out or In). Use the User FIFO Status Registers, Section 6.14.2, to determine the interrupting condition from the associated FIFO.

In response to these interrupt signals, any of the four Application Interrupt LINT Enable Registers, Section 6.9.4, can be enabled to assert any of the four local bus interrupt inputs, LINTi0 to LINTi3, to the PCI Interrupt Flag Registers, identified in Section 4.8.3 below.

4.8 Interrupt Operation (continued)

4.8.3 Local Bus PCI Interrupts

The PCI Interrupt Registers (Sections 5.2 and 5.3) contain interrupt flag and enable bits for the DMA Channels and the Application Interrupt conditions. The PCI Interrupts include the following signals/conditions:

- LINTix – Each of these interrupts is associated with one of the PCI local bus interrupt inputs, LINTi0 through LINTi3, from the Application Interrupt Enable Register, identified in Section 4.8.2 above.

- DMAn Chain Finish – Each of these interrupts is associated with a DMA Chain finish interrupt from the associated DMA channel. This interrupt is generated at the end of the last descriptor transfer specified in the DMA Descriptor Transfer Count Registers, Section 5.19, when the Chain bit D31 is set to ‘0’, End of Chain.

- DMAn Descriptor Finish – Each of these interrupts is associated with a DMA Descriptor finish interrupt from the associated DMA. This interrupt is generated at the end of the descriptor transfer count specified in each DMA Descriptor Transfer Count Register, Section 5.19.

- DMAn Xfer Interval Count Finish – Each of these interrupts is associated with a DMA Xfer Interval Count Finish interrupt from that DMA. This interrupt is generated at the end of the transfer interval count specified in the DMA Transfer Interval Counter Register, Section 5.18.

  (See also Section 4.5.4 for further description of these DMA Interrupts.)

In response to these interrupt signals, any of the four PCI Interrupt Enable Registers, Section 5.3, can be enabled to assert any of the PCI bus interrupt outputs, INTA to INTD, to the PCI Bus, as identified in Section 4.8.5 below.

4.8.4 DMA PCI Interrupts

The DMA PCI Interrupt Enable/Flag Register (Section 5.11) contains inter–rupt flag and enable bits for DMA channel overrun conditions. In response to these conditions, this register can enabled assertion of any PCI bus inter–rupt output, INTA to INTD, to the PCI Bus, as described in Section 4.8.5 below.

Note that the PCI bus interrupts from this register are ORed with the PCI interrupts from the PCI Interrupt Enable Registers, Section 4.8.3, on the applicable PCI interrupt signal line (INTA to INTD).

4.8 Interrupt Operation (continued)

4.8.5 PCI Bus Interrupt Outputs

Four PCI Bus interrupts, INTA to INTD, are associated with interrupt signals from the PCI Interrupt Flag Register, identified in Section 4.8.3 above, OR'ed with interrupt signals from the DMA PCI Interrupt Enable/Flag Register identified in Section 4.8.4 above.

Any of the four PCI Interrupt Enable Registers, Section 5.3, can be enabled to assert an interrupt from the OR'ed result of the flag bits, to the applicable PCI interrupt output, INTA to INTD. In addition any of the four PCI Bus interrupts can be enabled from the DMA PCI Interrupt Enable/Flag Register, Section 5.11. The PCI bus interrupt signals from both sets of registers are OR'ed on the applicable PCI interrupt line (INTA to INTD).

4.9 User FPGA Address Space

Each 7642 FPGA (the Signal FPGA and PCI FPGA) has a block of addresses reserved for user applications.

- The XC4VFX60 (or XC4VFX100) PCI FPGA board interfaces including PCI and Serial I/O. The PCI FPGA also includes two PowerPC cores which can be used as local microcontrollers to create complete application engines. The default Pentek FPGA configuration provides related control registers, which are described in Chapter 5. See Section 3.3.1 for the memory map for these registers.

  This PCI FPGA has a block of user registers reserved for custom processing of interface data. Access these registers using PCI Base Address Register 0 (**BAR0**) at offset addresses **0x10000** to **0x1FFFF**.

- The XC4VSX55 (or XC4VLX100) Signal FPGA serves as a control and status engine with data and programming interfaces to each of the on–board resources including the A/D converters, DDR2 memory, digital upconverter and D/A converters. The default Pentek FPGA configuration provides related control registers, which are described in Chapter 6. See Section 3.3.2 for the memory map for these registers.

  This Signal FPGA has a block of user registers reserved for custom processing of related data from or to these resources. Access these registers using PCI Base Address Register 2 (**BAR2**) at offset addresses **0x8E80** to **0x8F7F**.

In the default FPGA configurations, these registers perform no functions on the board, but are available as read/write temporary storage. You can read and write any type of data to each register.

NOTE: If you have ordered a Pentek–defined custom FPGA Option, these user registers may be programmed for that option's processing. Any such use of these User FPGA Address Space registers is defined in the applicable Pentek documentation for that option.

Chapter 5: PCI7142 Registers

5.1 Overview

The Signal FPGA is connected to the PCI interface through a PCI Master/Slave inter– face (Standard PCI 2.2), the Pentek PCI7142 (programmed in the XC4VFX60 or XC4VFX100 PCI FPGA). This interface provides direct Slave Mode and DMA Master/ Slave Mode, 9–channel demand mode, and chaining controller. Through this interface, any PCI Bus Master can control all programmable features on the board, including the four LTC2255s, the DAC5686, and all FPGA memory map registers. Data widths of 32 or 64 bits and data rates of 33 or 66 MHz guarantee compatibility with PMC, cPCI, and PCI platforms.

This chapter describes the PCI7142 internal registers accessible from the PCI interface. All descriptions are given from the PCI bus device's viewpoint. All PCI7142 registers are 32 bits wide—however since the PCI local bus is a 64–bit data path, the register addresses are in increments of 64 bits (8 bytes). Refer to Section 3.3.1 for the memory map of all PCI7142 registers.

The following is a simplified block diagram of the PCI7142 interfaces, indicating the location of the PCI7142 internal registers. (See Figure 6–1 for a simplified block diagram of the Signal FPGA interfaces and its registers.)

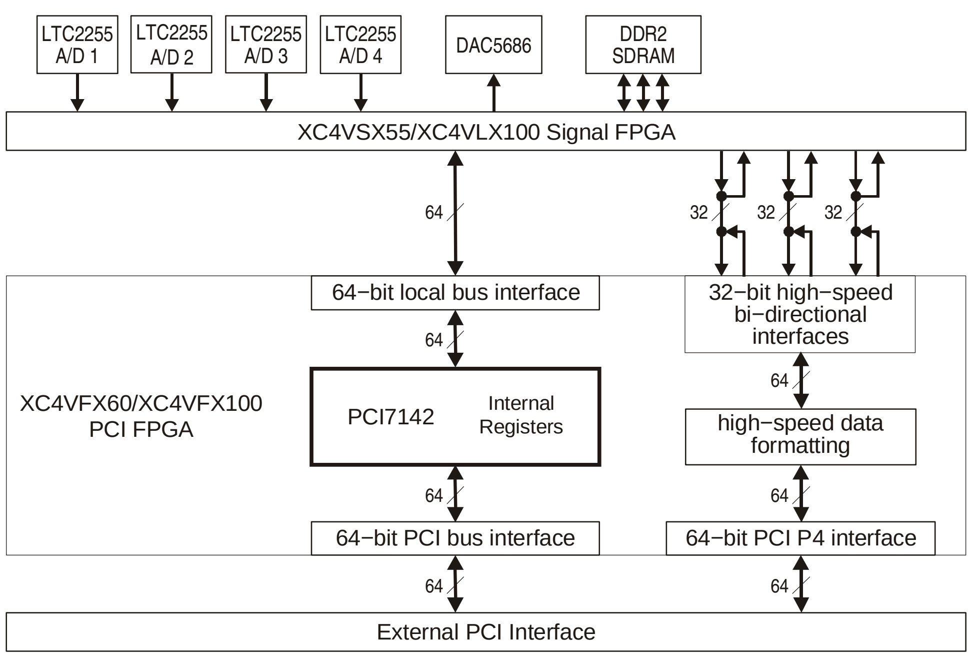


Figure 5−1: PCI7142 Interface Diagram

5.2 PCI Interrupt Flag Register

The PCI Interrupt Flag Register has one read/clear bit associated with each interrupt condition from various 7642 resources.

The following table shows which bit in this register is associated with each interrupt condition. The paragraphs following the table provide descriptions of these bits.

Table 5−1: PCI Interrupt Flag Register
R/Clr @ BAR0+0x0000

	D31	D30	D29	D28	D27	D26	D25	D24
Bit Name	Reserved	LINTi3	LINTi2	LINTi1	LINTi0	DMA8 Xfer Interval Count Finish	DMA8 Descriptor Finish	DMA8 Chain Finish
Function	**Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch							
	D23	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
Bit Name	DMA7 Xfer Interval Count Finish	DMA7 Descriptor Finish	DMA7 Chain Finish	DMA6 Xfer Interval Count Finish	DMA6 Descriptor Finish	DMA6 Chain Finish	DMA5 Xfer Interval Count Finish	DMA5 Descriptor Finish
Function	**Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch							
	D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D09**	**D08**
Bit Name	DMA5 Chain Finish	DMA4 Xfer Interval Count Finish	DMA4 Descriptor Finish	DMA4 Chain Finish	DM3 Xfer Interval Count Finish	DMA3 Descriptor Finish	DMA3 Chain Finish	DMA2 Xfer Interval Count Finish
Function	**Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch							
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	DMA2 Descriptor Finish	DMA2 Chain Finish	DMA1 Xfer Interval Count Finish	DMA1 Descriptor Finish	DMA1 Chain Finish	DMA0 Xfer Interval Count Finish	DMA0 Descriptor Finish	DMA0 Chain Finish
Function	**Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch							

IMPORTANT! When reset, including power−up, the state of this register is unknown. You should clear this register before using it, by writing ‘1’s into all bits.

5.2 PCI Interrupt Flag Register (continued)

Each bit of this register latches an interrupt occurrence. A logic '1' in any bit in this register indicates that an interrupt has occurred.

Since these bits latch in response to an interrupt, to detect subsequent interrupts, you must clear the bits in this register. To clear any bit in this register that is set to logic '1' you must write a '1' to that bit.

The following table describes the interrupt condition associated with each bit of the PCI Interrupt Flag register.

Table 5−2: PCI Interrupt Flag Register Bits

Bit Name	Bit Positions	Interrupt Function
DMAn Chain Finish	D0, D3, D6, D9, D12 D15, D18, D21, D24	Each of these flag bits is associated with a DMA Chain finish interrupt on the PCI bus from the associated DMA. This interrupt is generated at the end of the last descriptor transfer specified in the DMA Descriptor Transfer Count Register, Section 5.19, when the Chain bit D31 is set to '0', End of Chain.
DMAn Descriptor Finish	D1, D4, D7, D10, D13 D16, D19, D22, D25	Each of these flag bits is associated with a DMA Descriptor finish interrupt on the PCI bus from the associated DMA. This interrupt is generated at the end of the descriptor transfer count specified in each DMA Descriptor Transfer Count Register, Section 5.19.
DMAn Xfer Interval Count Finish	D2, D5, D8, D11, D14 D17, D20, D23, D26	Each of these flag bits is associated with a DMA Transfer Interval Count finish interrupt on the PCI bus from the associated DMA. This interrupt is generated at the end of the transfer interval count specified in the DMA Transfer Interval Counter Register, Section 5.18.
LINTin	D27, D28, D29, D30	Each of these flag bits is associated with one of the PCI local bus interrupt inputs, LINTi0 through LINTi3. The PCI local bus interrupts are associated with the Application Interrupt LINT Enable Registers, Section 6.9.4.

Refer to Interrupt Operation, Section 4.8, for further description of interrupt routing and operation on the Model 7642.

5.3 PCI Interrupt Enable Registers

The PCI Interrupt Enable Registers contains enable bits for each interrupt condition defined in the PCI Interrupt Flag Register, Section 5.2. Each bit enables or disables the generation of a PCI interrupt output to the PCI interface in response to these interrupts. There are four PCI Interrupt Enable Registers, one for each PCI bus interrupt (INTA to INTD). Each register enables interrupts for the respective PCI bus interrupt to the PCI interface.

The following table shows which bit in this register is associated with each interrupt condition. The paragraphs following the table provide descriptions of these bits.

Table 5−3: PCI Interrupt Enable Registers
INTA: R/W @ BAR0+0x0010
INTB: R/W @ BAR0+0x0018
INTC: R/W @ BAR0+0x0020
INTD: R/W @ BAR0+0x0028

	D31	D30	D29	D28	D27	D26	D25	D24
Bit Name	PCI INTx	LINTi3	LINTi2	LINTi1	LINTi0	DMA8 Xfer Interval Count Finish	DMA8 Descriptor Finish	DMA8 Chain Finish
Function	0 = Disable Interrupt 1 = Enable Interrupt							
	D23	**D22**	**D21**	**D20**	**D19**	**D18**	**D17**	**D16**
Bit Name	DMA7 Xfer Interval Count Finish	DMA7 Descriptor Finish	DMA7 Chain Finish	DMA6 Xfer Interval Count Finish	DMA6 Descriptor Finish	DMA6 Chain Finish	DMA5 Xfer Interval Count Finish	DMA5 Descriptor Finish
Function	0 = Disable Interrupt 1 = Enable Interrupt							
	D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D09**	**D08**
Bit Name	DMA5 Chain Finish	DMA4 Xfer Interval Count Finish	DMA4 Descriptor Finish	DMA4 Chain Finish	DM3 Xfer Interval Count Finish	DMA3 Descriptor Finish	DMA3 Chain Finish	DMA2 Xfer Interval Count Finish
Function	0 = Disable Interrupt 1 = Enable Interrupt							
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	DMA2 Descriptor Finish	DMA2 Chain Finish	DMA1 Xfer Interval Count Finish	DMA1 Descriptor Finish	DMA1 Chain Finish	DMA0 Xfer Interval Count Finish	DMA0 Descriptor Finish	DMA0 Chain Finish
Function	0 = Disable Interrupt 1 = Enable Interrupt							
All bits default to the logic '0' state at power on and reset								

5.3 PCI Interrupt Enable Register (continued)

Each bit of this register enables or disables the generation of a PCI bus interrupt (INTA to INTD, depending on the register used) to the PCI interface. Setting the bit associated with a given interrupt to logic '1' enables the generation of a PCI bus interrupt when that interrupt condition occurs. When a bit is cleared to logic '0' (its default state), the PCI bus interrupt will not be generated by the defined interrupt condition.

The following table describes the interrupt condition associated with each bit of the PCI Interrupt Enable register.

Table 5–4: PCI Interrupt Enable Register Bits

Bit Name	Bit Positions	Interrupt Function
DMAn Chain Finish	D0, D3, D6, D9, D12 D15, D18, D21, D24	Each of these enable bits is associated with a DMA Chain finish interrupt on the PCI bus from the associated DMA. This interrupt is generated at the end of the last descriptor transfer specified in the DMA Descriptor Transfer Count Register, Section 5.19, when the Chain bit D31 is set to ‘0’, End of Chain.
DMAn Descriptor Finish	D1, D4, D7, D10, D13 D16, D19, D22, D25	Each of these enable bits is associated with a DMA Descriptor finish interrupt on the PCI bus from the associated DMA. This interrupt is generated at the end of the descriptor transfer count specified in each DMA Descriptor Transfer Count Register, Section 5.19.
DMAn Xfer Interval Count Finish	D2, D5, D8, D11, D14 D17, D20, D23, D26	Each of these enable bits is associated with a DMA Transfer Interval Count finish interrupt on the PCI bus from the associated DMA. This interrupt is generated at the end of the transfer interval count specified in the DMA Transfer Interval Counter Register, Section 5.18.
LINTin	D27, D28, D29, D30	Each of these enable bits is associated with one of the local bus input interrupts, LINTi0 through LINTi3, from the Signal FPGA. The PCI local bus interrupts are generated using the Application Interrupt LINT Enable Registers, Section 6.9.4.
PCI INTx	D31	This bit enables the PCI Interrupt output for the associated register, INTA through INTD.

Refer to Interrupt Operation, Section 4.8, for further description of interrupt routing and operation on the Model 7642.

5.4 FPGA Data In/Data Out Registers

The FPGA Data In/Data Out registers allow reading from or writing to the XC4VFX60 (or XC4VFX100) PCI FPGA spare pins. The Model 7642 routes these I/O signals from the PCI FPGA through PMC **P4** connector to the 96–pin DIN connector on the PCI adapter.

NOTE: This capability is provided only with Option 104 (see Section 2.5).

The following table shows the contents of these registers.

Table 5–5: FPGA Data In/Data Out Registers
Data In: R.O. @ BAR0+0x0050
Data Out: R/W @ BAR0+0x0058

	D31 – D16															
Bit Name	Not used															
Function	Write with zeros, Mask when reading															
	D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D09**	**D08**	**D07**	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
Function	0 = ‘0’ 1 = ‘1’															
All bits default to the logic '0' state at power on and reset																

Bits D0 through D15 correspond to the LVDS input and output signals described for Option 104 in Section 2.5. The LVDS_OUT signals shown in Table 2–5 are from the FPGA Data Out Register, with LVDS_OUT_x0 from bit D0 and LVDS_OUT_x15 from bit D15. The LVDS_IN signals are for the FPGA Data In Register, with LVDS_IN_x0 from bit D0 and LVDS_IN_x15 from bit D15.

5.5 Board/Channel Reset Register

The Board/Channel Reset Register provides a reset for the entire 7142 PMC module.

The following table shows the contents of this register. The paragraphs following the table provide descriptions of the bits in this register.

Table 5–6: Board/Channel Reset Register
R/W @ BAR0+0x0070

	D31 – D16	
Bit Name	Not used	
Function	Write with zeros, Mask when reading	
	D15	**D14 – D00**
Bit Name	Board Reset	Reserved
Function	0 =Run 1 =Reset	Write with zeros, Mask when reading
All bits default to the logic '0' state at power on and reset		

Bit D15 issues a reset to the entire 7142 PMC module. When the bit is set to logic '1', the 7142 is in reset. When the bit is cleared to logic '0' (its default state) the 7142 is in a nor–mal run state.

5.6 PCI7142 Revision Register

The PCI7142 Revision Register identifies the Pentek version number and the date of the PCI7142 code currently installed in the XC4VFX60 (or XC4VFX100) PCI FPGA.

The following table shows the contents of this register. The paragraphs below PCI7142 Revision Register describe the contents of the register.

Table 5−7: PCI7142 Revision Register
R.O. @ BAR0+0x0078

	D31 − D28	D27 − D24	D23 − D20	D19 − D16
Bit Name	YEAR		MONTH	
Function	3rd Digit of Year	4th Digit of Year	1st Digit of Month	2nd Digit of Month
	D15 − D12	**D11 − D08**	**D07 − D00**	
Bit Name	DAY		PCI7142 CODE REVISION	
Function	1st Digit of Day	2nd Digit of Day	PCI FPGA Pentek Code Revision Number	

The PCI7142 revision date is coded as six hexadecimal characters, as follows:

Year: Bits D31 – D28, 3rd digit of Year
Bits D27 – D24, 4th digit of Year

Month: Bits D23 – D20, 1st digit of Month
Bits D19 – D16, 2nd digit of Month

Day: Bits D15 – D12, 1st digit of Day
Bits D11 – D08, 2nd digit of Day

The PCI7142 Code Revision number (bits D07 – D00) is a binary value that identifies the Pentek version number of the code in the Pentek PCI7142 PCI interface core, starting at revision 0.

For example, the date of **July 15, 2008** (7/15/08) would appear as follows in the PCI7142 Revision Register, assuming a code revision of ‘1’.

PCI7142 Revision Register = **`0x08071501`**

5.7 Virtex Config Register

The Virtex Config Register is used to reconfigure/reprogram the XC4VSX55 (or XC4VLX100) Signal FPGA. The bits in this register allow you to read and set the status of the Signal FPGA configuration cycle, and to control uploading the configuration to the Signal FPGA from a PCI interface processor. Refer to Section 1.12, FPGA Configuration, for additional information about configuring the Signal FPGA.

The following table shows the contents of the Virtex Config Register. The subsections following the table provide descriptions of the bits in this register.

Table 5–8: Virtex Config Register
R/W @ BAR0+0x0080

	D31 – D16							
Bit Name	Not used							
Function	Write with zeros, Mask when reading							
	D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D09**	**D08**
Bit Name	D7	D6	D5	D4	D3	D2	D1	D0
Function	Signal FPGA Configuration Data							
	D07	**D06**	**D05**	**D04**	**D03 ***	**D02 ***	**D01**	**D00**
Bit Name	WRITE STROBE	DATA SRC	DIRECTION	Reserved	$\overline{\text{INIT}}$	DONE	PRGM	LD SRC
Function	0 = Disable 1 = Enable	0 = Register 1 = DMA	0 = Read 1 = Write	Write 0s, Mask read	0 = Initializing 1 = Done	0 = Configuring 1 = Done	0 = Disable 1 = Reprogram	0 = onboard 1 = Upload

* These bits are Read Only
All bits default to the logic '0' state at power on and reset

5.7.1 Configuration Data [D7:D0] Bits D15 – D08

These eight bits contain configuration data to be written to the Signal FPGA. These bits are valid only when DATA SRC, bit D06, Section 5.7.3, is cleared to 0, and LD SRC, bit D00, Section 5.7.8, is set to 1.

5.7.2 WRITE STROBE Bit D07

This bit enables the write strobe to the Virtex Config Data Register. When you set this bit to logic '1', you can write configuration data to the Signal FPGA. Clear the bit to logic '0' (its default state) to disable writing configuration data to the Signal FPGA.

5.7 Virtex Config Register (continued)

5.7.3 DATA SRC Bit D06

This bit selects the source of the Signal FPGA upload data when LD SRC, bit D00, Section 5.7.8, is set to 1. Clear the bit to logic '0' (its default state) to write configuration data to the FPGA using the Configuration Data bits (bits D15 – D08, Section 5.7.1) of this register. When you set this bit to logic '1', you can upload configuration data to the FPGA from the PCI bus using the FPGA Load DMA channel (see Section 4.5.5).

5.7.4 DIRECTION Bit D05

This bit sets the read/write access of the Virtex Config Data Register. Clear the bit to logic '0' (its default state) to read configuration data from the Signal FPGA. When you set this bit to logic '1', you can write configuration data to the FPGA. To write with the Virtex Config Data Register, you must also enable configuration upload (LD SRC bit D00 = 1, Section 5.7.8).

5.7.5 $\overline{\text{INIT}}$ Bit D03

This read–only bit indicates the status of the Signal FPGA's '$\overline{\text{INIT}}$' pin. At initialization of a configuration cycle this bit goes to logic '0', then logic '1'. It remains at logic '0' if an error in configuration data is detected. When read as logic '1', initialization is done.

5.7.6 DONE Bit D02

This read–only bit indicates the status of the Signal FPGA's 'DONE' pin. When read as logic '0', the FPGA is in the configuration cycle. When read as logic '1', the FPGA has completed configuration.

5.7.7 PRGM Bit D01

This bit starts the Signal FPGA configuration reprogramming. The Signal FPGA begins a configuration reprogramming cycle after you transition this bit from logic '0', to logic '1', then to logic '0', while in Serial EEPROM or Processor Upload mode (see LD SRC, below).

5.7.8 LD SRC Bit D00

This bit selects the source of the Signal FPGA configuration data. When you clear this bit to logic '0' (its default state), the FPGA selects its configuration data from the onboard Serial EEPROMs. When you set this bit to logic '1', you can upload the configuration from a PMC processor using the Configu–ration Data bits (D15 – D08) in this register, or from the FPGA Load DMA channel, depending on the DATA SRC bit D06, Section 5.7.3.

5.8 PCI DCM Control Register

The PCI DCM Control Register controls each Digital Clock Manager (DCM) on the XC4VFX60 (or XC4VFX100) PCI FPGA.

The following table shows the contents of the DCM Register. The subsections following the table provide descriptions of each control bit in this register.

Table 5–9: PCI DCM Control Register
R/W @ BAR0+0x0088

	D31 – D08							
Bit Name	Reserved							
Function	Write with zeros, Mask when reading							
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01 ***	**D00 ***
Bit Name	Reserved					DCM RST	LOCKD 1	LOCKD 0
Function	Write with zeros, Mask when reading					0 =Run 1 =Reset	0 =Not locked 1 =Locked	0 =Not locked 1 =Locked

* These bits are Read Only
All bits default to the logic '0' state at power on and reset

5.8.1 DCM RST — Bit D02

This bit resets both DCM clocks 0 and 1. Set the bit to logic '1' to reset the DCM clocks. Clear the bit to logic '0' (its default state) to return the DCM clocks to normal operation.

5.8.2 LOCKD 1/0 — Bits D01, D00

Each of these read–only bits indicates the locked status of a DCM clock. When read as logic '0' the DCM clock output is not locked. When read as logic '1' the DCM clock output is locked.

5.9 Local DMA Request Status Register

The read–only Local DMA Request Status Register reads the DMA Request status of each DMA request. The following table shows the contents of this register.

NOTE: This register is for Pentek factory use only.

Table 5–10: Local DMA Request Status Register
R.O. @ BAR0+0x0090

	D31 – D16							
Bit Name	Not used							
Function	Write with zeros, Mask when reading							
	D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D09**	**D08**
Bit Name	Reserved				TEST FIFO Write	TEST FIFO Read	DDR Write FIFO	DDR Read FIFO
Function	Write with zeros, Mask when reading				Factory use only			
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	User Out FIFO	DAC FIFO	Reserved	User In FIFO	ADC 4 FIFO	ADC 3 FIFO	ADC 2 FIFO	ADC 1 FIFO
Function	Factory use only							
All bits default to the logic '0' state at power on and reset								

Register Bit	DMA Request Number	FIFO
D00	DMA Request 0	ADC 1 FIFO
D01	DMA Request 1	ADC 2 FIFO
D02	DMA Request 2	ADC 3 FIFO
D03	DMA Request 3	ADC 4 FIFO
D04	DMA Request 4	User In FIFO
D05	DMA Request 5	Reserved
D06	DMA Request 6	DAC FIFO
D07	DMA Request 7	User Out FIFO
D08	DMA Request 8	DDR Memory Read FIFO
D09	DMA Request 9	DDR Memory Write FIFO
D10	DMA Request 10	Test FIFO Read
D11	DMA Request 11	Test FIFO Write

5.10 PCI Bus Status Register

The PCI Bus Status Register is a read−only register that indicates the speed of the PCI bus that the 7642 is connected to on the personal computer's PCI interface, 33 MHz or 66 MHz.

The following table shows the contents of this register.

Table 5−11: PCI Bus Status Register R.O. @ BAR0+0x00A0		
	D31 − D01	**D00**
Bit Name	Reserved	Bus Speed
Function	Write with zeros, Mask when reading	0 = 33 MHz 1 = 66 MHz
All bits default to the logic '0' state at power on and reset		

5.11 DMA PCI Interrupt Enable/Flag Register

The DMA PCI Interrupt Enable/Flag Register enables or disables assertion of a PCI interrupt to the PCI interface in response to a DMA channel buffer overrun. The fol−lowing table shows the contents of this register. The subsections following the table provide descriptions of the bits in this register.

Table 5−12: DMA PCI Interrupt Enable/Flag Register
R/Clr @ BAR0+0x00A8

	D31	D30	D29	D28	D27	D26	D25	D24
Bit Name	Reserved							FPGA DMA FIFO Status
Function	Write with zeros, Mask when reading							0 = OK 1 = Underflow
	D23 *	**D22**	**D21**	**D20**	**D19**	**D18**	**D17 ***	**D16**
Bit Name	FPGA DMA DONE FLAG	FPGA DMA DONE EN	INTD	INTC	INTB	INTA	DMA8 FLAG	DMA8 EN
Function	0 = Not done 1 = Done	0 = Disable 1 = Enable	0 = Disable 1 = Enable				0 = OK 1 = Overrun	0 = Disable 1 = Enable
	D15 *	**D14**	**D13 ***	**D12**	**D11 ***	**D10**	**D09 ***	**D08**
Bit Name	DMA7 FLAG	DMA7 EN	DMA6 FLAG	DMA6 EN	DMA5 FLAG	DMA5 EN	DMA4 FLAG	DMA4 EN
Function	0 = OK 1 = Overrun	0 = Disable 1 = Enable	0 = OK 1 = Overrun	0 = Disable 1 = Enable	0 = OK 1 = Overrun	0 = Disable 1 = Enable	0 = OK 1 = Overrun	0 = Disable 1 = Enable
	D07 *	**D06**	**D05 ***	**D04**	**D03 ***	**D02**	**D01 ***	**D00**
Bit Name	DMA3 FLAG	DMA3 EN	DMA2 FLAG	DMA2 EN	DMA1 FLAG	DMA1 EN	DMA0 FLAG	DMA0 EN
Function	0 = OK 1 = Overrun	0 = Disable 1 = Enable	0 = OK 1 = Overrun	0 = Disable 1 = Enable	0 = OK 1 = Overrun	0 = Disable 1 = Enable	0 = OK 1 = Overrun	0 = Disable 1 = Enable

* These bits are Read/Clear Flag bits. Each bit can be cleared only by writing a ‘1’ to that bit.
IMPORTANT! When reset, including power−up, the state of the Read/Clear bits are unknown. You should clear this register before using it, by writing ‘1’s into all FLAG bits.

Register Bits	DMA Channel	Associated FIFO
D00, D01	DMA0	ADC 1 FIFO
D02, D03	DMA1	ADC 2 FIFO
D04, D05	DMA2	ADC 3 FIFO
D06, D07	DMA3	ADC 4 FIFO
D08, D09	DMA4	User In FIFO
D10, D11	DMA5	Reserved
D12, D13	DMA6	DAC FIFO
D14, D15	DMA7	User Out FIFO
D16, D17	DMA8	DDR Memory Read FIFO
D22, D23, D24	FPGA Load DMA	FPGA Load DMA FIFO

5.11 DMA PCI Interrupt Enable/Flag Register (continued)

NOTE: Bits D24, D23, and D22 are available only in 7142 boards with a PCI FPGA revision date greater that 10/01/07 (see Section 5.6).

5.11.1 FPGA DMA FIFO Status — Bit D24

This read–only bit indicates the status of the FPGA Load DMA FIFO. When read as logic '1', the Signal FPGA has a data underflow condition. When read as logic '0', the FIFO indicates no underflow.

5.11.2 FPGA DMA DONE FLAG — Bit D23

This bit latches a DMA done condition for the FPGA Load DMA channel. A logic '1' in this bit indicates the end of the transfer count specified in the FPGA Load DMA Transfer Count Register, Section 5.22. A logic '0' indicates the DMA is not done. When this bit changes to '1' a PCI bus interrupt can be asserted if the associated enable bit (Section 5.11.3 below) is enabled and one of the INTA/D bits is enabled (Section 5.11.4 below).

Since this bit latches in response to an interrupt, to detect a subsequent FPGA Load DMA done condition, you must clear the FLAG bit. To clear the FLAG bit, write a '1' to the bit.

See Section 4.5.5 for description of FPGA Load DMA channel operation.

5.11.3 FPGA DMA DONE EN — Bit D22

This bit enables or disables an FPGA Load DMA done interrupt to the PCI Interrupt line enabled (INTD to INTA, Section 5.11.4 below). Setting this bit to '1' enables the interrupt at the end of the FPGA Load DMA Transfer Count. When the bit is cleared to logic '0' (its default state), this interrupt will not be generated.

5.11.4 INTD/C/B/A — Bits D21 – D18

Each of these four bits enables or disables the generation of a PCI bus interrupt (INTA to INTD, depending on the bit) to the PCI interface in response to a DMA channel buffer overrun. Setting a bit to logic '1' enables the generation of the associated PCI bus interrupt when any DMA FLAG bit in this register (Section 5.11.5 below) is set to '1' and the associated DMA EN bit in this register (Section 5.11.6 below) is set to '1'. When a bit is cleared to logic '0' (its default state), that PCI bus interrupt will not be generated.

Note that the PCI bus interrupt is ORed with the PCI interrupt from the PCI Interrupt Enable Registers, Section 5.3, on the applicable PCI interrupt signal line (INTA to INTD).

5.11 DMA PCI Interrupt Enable/Flag Register (continued)

5.11.5 DMAn FLAG Bits D17/D15/D13/D11/D09/D07/D05/D03/D01

Each of these bits latches a buffer overrun condition for the respective DMA channel. A logic '1' in any bit in this register indicates that a channel buffer overrun has occurred. A '0' indicates no buffer overrun. When one of these bits changes to '1' a PCI bus interrupt can be asserted if the associated DMA EN bit (Section 5.11.6 below) is enabled and one of the INTA/D bits is enabled (Section 5.11.4 above).

Since these bits latch in response to an interrupt, to detect a subsequent channel buffer overrun, you must clear the FLAG bits in this register. To clear any FLAG bit in this register that is set to logic '1' you must write a '1' to that bit.

5.11.6 DMAn EN Bits D16/D14/D12/D10/D08/D06/D04/D02/D00

Each of these bits enables or disables a DMA buffer overrun interrupt to the PCI Interrupt line enabled (INTD to INTA, Section 5.11.4 above). Setting the bit associated with a given DMA channel to '1' enables the interrupt when that channel overrun occurs. When a bit is cleared to '0' (its default state), the interrupt will not be generated by the channel overrun.

5.12 Local DMA In Address Remap Register

The Local DMA In Address Remap Register permits remapping each incoming DMA channel for use with a different incoming FIFO.

The following table shows the contents of this register.

Table 5−13: Local DMA In Address Remap Register R/W @ BAR0+0x00B0								
	D31 – D28	D27 – D24	D23 – D20	D19 – D16	D15 – D12	D11 – D08	D07 – D04	D03 – D00
Bit Name	Reserved	DMA8	DMA5	DMA4	DMA3	DMA2	DMA1	DMA0
Function	Write 0s, Mask read	FIFO Codes, see below						
All bits default to the default FIFOs at power on and reset (see Table 5−14, below)								

Each incoming DMA channel has a four−bit field in this register that maps an incoming FIFO to that channel. The mapping code for the four bits in each field is as follows, in hex:

0x0 = ADC 1 FIFO
0x1 = ADC 2 FIFO
0x2 = ADC 3 FIFO
0x3 = ADC 4 FIFO
0x4 = User In FIFO
0x5 = (Reserved)
0x8 = DDR Memory Read FIFO
0xC = Test FIFO Read

The default mapping is shown in the following table:

Table 5−14: Default DMA In Mapping

DMA Channel	FIFO	Register Bits	Code
DMA0	ADC 1 FIFO	D03 − D00	0
DMA1	ADC 2 FIFO	D04 − D07	1
DMA2	ADC 3 FIFO	D08 − D11	2
DMA3	ADC 4 FIFO	D12 − D15	3
DMA4	User In FIFO	D16 − D19	4
DMA5	Reserved	D20 − D23	5
DMA8	DDR Memory Read FIFO	D24 − D27	8

5.13 Local DMA Out Address Remap Register

The Local DMA Out Address Remap Register permits remapping each outgoing DMA channel for use with a different outgoing FIFO.

The following table shows the contents of this register.

Table 5−15: Local DMA Out Address Remap Register R/W @ BAR0+0x00B8			
	D31 − D08	**D07 − D04**	**D03 − D00**
Bit Name	Reserved	DMA7	DMA6
Function	Write with zeros, Mask when reading	FIFO Codes, see below	
All bits default to the default FIFOs at power on and reset (see Table 5−16, below)			

Each outgoing DMA channel has a four−bit field in this register that maps an outgoing FIFO to that channel. The mapping code for the four bits in each field is as follows, in hex:

0x6 = DAC FIFO
0x7 = User Out FIFO
0x9 = DDR Memory Write FIFO
0xD = Test FIFO Write

The default mapping is shown in the following table:

Table 5−16: Default DMA Out Mapping			
DMA Channel	**FIFO**	**Register Bits**	**Code**
DMA6	DAC FIFO	D03 − D00	6
DMA7	User Out FIFO	D07 − D04	7

5.14 DMA Command Register

The DMA Command Register specifies the type of PCI Bus read and write commands used for DMA operations.

The following table shows the contents of this register.

Table 5−17: DMA Command Register R/W @ BAR0+0x0100			
	D31 − D08	**D07 − D04**	**D03 − D00**
Bit Name	Reserved	PCI Write Command Code for DMA	PCI Read Command Code for DMA
Function	Write with zeros, Mask when reading	0111	1100
All bits default to the state indicated above at power on and reset			

The PCI Write command (bits D07 – D04) is fixed at code ‘0111’, Memory Write.

The PCI Read command (bits D03 – D00) defaults to code ‘1100’, Memory Read Multi– ple. The following codes are valid for the PCI Read command:

0110 – Memory Read
1100 – Memory Read Multiple
1110 – Memory Read Line

5.15 DMAx Current Transfer Counter Register

The read−only DMA Current Transfer Counter Register indicates the current status and data transfer count for the applicable DMA channel. There are nine DMA Current Transfer Count registers, one for each DMA channel (0 through 8), which support nine data transfer FIFOs.

The following table shows the contents of this DMA Current Transfer Counter Register. The subsections following the table provide descriptions of the bits in this register.

Table 5−18: DMAx Current Transfer Counter Register

DMA0: R.O. @ BAR0+0x0110
DMA1: R.O. @ BAR0+0x0120
DMA2: R.O. @ BAR0+0x0130
DMA3: R.O. @ BAR0+0x0140
DMA4: R.O. @ BAR0+0x0150
DMA5: R.O. @ BAR0+0x0160
DMA6: R.O. @ BAR0+0x0170
DMA7: R.O. @ BAR0+0x0180
DMA8: R.O. @ BAR0+0x0190

	D31	D30	D29 − D28	D27 − D00
Bit Name	DMA Active	Reserved	Descriptor Pointer	Transfer Count
Function	0 = DMA Active 1 = No DMA	Write 0s, Mask read	Current Descriptor number	Current data transfer count, in words (32−bit or 64−bit, see Section 5.15.3)
When reset, including power−up, the state of this register is unknown.				

In the default configuration, the nine DMA channels are assigned to the following FIFOs (note that the Direction given is from the PCI interface):

DMA Channel	FIFO	Direction
DMA0	ADC 1 FIFO	Incoming
DMA1	ADC 2 FIFO	Incoming
DMA2	ADC 3 FIFO	Incoming
DMA3	ADC 4 FIFO	Incoming
DMA4	User In FIFO	Incoming
DMA5	Reserved	Incoming
DMA6	DAC FIFO	Outgoing
DMA7	User Out FIFO	Outgoing
DMA8	DDR Memory Read FIFO	Incoming

5.15 DMAx Current Transfer Status Register (continued)

5.15.1 DMA Active Bit D31

This bit indicates the current active status of the applicable DMA channel, depending on the register accessed. When read as logic '0' the DMA channel is active (a data transfer is running). When read as logic '1' the DMA channel is not active.

5.15.2 Descriptor Pointer Bits D29 – D28

These two bits indicate the number of the current Descriptor that is active for the applicable DMA channel, depending on the register accessed. There are four Descriptors for each DMA channel, identified as 0 through 3.

The four Descriptors can programmed for separate DMA transfers using the DMA Command/Status Register, Transfer Interval Counter Register, four Descriptor Transfer Count Registers, and four Descriptor PCI Address Registers (see Sections 5.17 to 5.20).

5.15.3 Transfer Count Bits D27 – D00

This field indicates the current transfer count of the applicable DMA channel, depending on the register accessed. The Transfer Count is a 28–bit binary value, with bit D27 the MSB. Words are either 32 or 64 bits wide, as specified by Data Width, bit D07, in the DMA Command/Status Register, Section 5.17.10.

This register field is loaded with the programmed Transfer Interval Count, Section 5.18.3, at the start of the DMA, and decrements to zero during the transfer.

5.16 DMAx Current PCI Address Register

The read−only DMA Current PCI Address indicates the current PCI address for the applicable DMA channel. There are nine DMA Current PCI Address registers, one for each DMA channel (0 through 8).

The following table shows the contents of this register.

Table 5−19: DMAx Current PCI Address Register

DMA0: R.O. @ BAR0+0x0118
DMA1: R.O. @ BAR0+0x0128
DMA2: R.O. @ BAR0+0x0138
DMA3: R.O. @ BAR0+0x0148
DMA4: R.O. @ BAR0+0x0158
DMA5: R.O. @ BAR0+0x0168
DMA6: R.O. @ BAR0+0x0178
DMA7: R.O. @ BAR0+0x0188
DMA8: R.O. @ BAR0+0x0198

	D31 − D00
Bit Name	PCI Address
Function	Current PCI Address for DMAx

When reset, including power−up, the state of this register is unknown.

In the default configuration, the DMA channels are mapped to the following FIFOs:

DMA Channel	FIFO
DMA0	ADC 1 FIFO
DMA1	ADC 2 FIFO
DMA2	ADC 3 FIFO
DMA3	ADC 4 FIFO
DMA4	User In FIFO
DMA5	Reserved
DMA6	DAC FIFO
DMA7	User Out FIFO
DMA8	DDR Memory Read FIFO

5.17 DMAx Command/Status Register

The DMA Command/Status Register is used to control DMA operation for the appli– cable DMA channel. There are nine DMA Command/Status Count registers, one for each DMA channel (0 through 8). The bits in this register allow you to set and monitor the operating mode of the DMA channel and the PCI bus data transfer associated with that DMA operation.

Refer to Section 4.5 for description of DMA operation.

The following table shows the contents of the DMA Command/Status Register. The subsections following the table provide descriptions of the bits in this register.

Table 5–20: DMAx Command/Status Register

DMA0: R/W @ BAR0+0x1000
DMA1: R/W @ BAR0+0x2000
DMA2: R/W @ BAR0+0x3000
DMA3: R/W @ BAR0+0x4000
DMA4: R/W @ BAR0+0x5000
DMA5: R/W @ BAR0+0x6000
DMA6: R/W @ BAR0+0x7000
DMA7: R/W @ BAR0+0x8000
DMA8: R/W @ BAR0+0x9000

	D31 – D16
Bit Name	Maximum Burst Count
Function	Burst Count, in words

	D15 ***	D14	D13 **	D12 **	D11 **	D10 **	D09 **	D08 **
Bit Name	DAC Buffering	PCI Address	Detected Parity Error Flag	Signaled System Error Flag	Received Master Abort Flag	Received Target Abort Flag	Signaled Target Abort Flag	Data Parity Error Flag
Function	0 = Enable 1 = Disable	0 = Increment 1 = Hold	0 = No Error 1 = Error	0 = No Error 1 = Error	0 = No Abort 1 = Abort	0 = No Abort 1 = Abort	0 = No Abort 1 = Abort	0 = No Parity 1 = Parity

	D07	D06	D05	D04 **	D03	D02	D01 *	D00
Bit Name	Data Width	Burst Mode	Demand Mode	DMA Status	DMA Channel Buffer Reset	DMA Abort	DMA Start	DMA Enable
Function	0 = 32 1 = 64	0 = N/A 1 = Enable	0 = Disable 1 = Enable	0 = DMA Active 1 = Done	0 = Run 1 = Reset	0 = N/A 1 = Abort DMA	0 = Stop 1 = Start	0 = Disable 1 = Enable

* Bit D01 is Write Only
** These bits are Read Only
*** Bit D15 applies only to DMA Channels 6 and 7 (DMA6 & DMA7).
All bits default to the logic '0' state at power on and reset

5.17 DMAx Command/Status Register (continued)

5.17.1 Maximum Burst Count — Bits D31 – D16

The burst count is a 16–bit binary value, with bit D31 the MSB. This value specifies the maximum data transfer burst length, in words, for the applicable DMA channel. A value of '0' specifies the maximum burst count. See Data Width, Section 5.17.10 below, for the word size of each transfer, 32–bit or 64–bit.

NOTE: When in 32–bit mode, the burst count must be an even value.

5.17.2 DAC Buffering — Bit D15

This bit enables or disables buffering of the data transfer for the applicable DMA channel PCI bus data transfer. When set to logic '0' buffering is enabled to the output device. When set to logic '1' the DMA channel does not buffer the data transfer to this device. Typically, this should be set to '1' to disable buffering.

NOTE: This bit is applicable only to DMA channels 6 and 7 (Out FIFOs).

5.17.3 PCI Address Hold — Bit D14

This bit sets the PCI address to increment or hold for the applicable DMA channel PCI bus data transfer. When set to logic '0' the PCI address increments. When set to logic '1' the DMA address holds and does not increment.

5.17.4 Detected Parity Error Flag — Bit D13

This read–only bit indicates the PCI bus Parity Error status for the applicable DMA channel PCI bus data transfer. When read as logic '0' the DMA channel did not detect a PCI bus parity error. When read as logic '1' the DMA channel detected a PCI bus parity error.

5.17.5 Signaled System Error Flag — Bit D12

This read–only bit indicates the PCI bus SERR pin status for the applicable DMA channel PCI bus data transfer. When read as logic '0' the DMA channel did not detect a PCI bus SERR error. When read as logic '1' the DMA channel detected a PCI bus SERR error.

5.17.6 Received Master Abort Flag — Bit D11

This read–only bit indicates the current Master Abort status for the applicable DMA channel PCI bus data transfer. When read as logic '0' a PCI Bus Master did not terminate the data transfer. When read as logic '1' a PCI Bus Master did terminate the data transfer by a bus Master Abort.

5.17 DMAx Command/Status Register (continued)

5.17.7 Received Target Abort Flag Bit D10

This read–only bit indicates the current Target Abort status for the applicable DMA channel PCI bus data transfer. When read as logic '0' a Target Abort did not terminate the data transfer. When read as logic '1' a Target Abort has terminates the bus master's data transfer.

5.17.8 Signaled Target Abort Flag Bit D09

This read–only bit indicates the current Target Abort status of the Target device for the applicable DMA channel PCI bus data transfer. When read as logic '0' the Target device did not terminate the transaction with a Target Abort. When read as logic '1' the Target device terminated the transaction with a Target Abort.

5.17.9 Data Parity Error Flag Bit D08

This read–only bit indicates the enabled PCI bus PERR pin status for the applicable DMA channel PCI bus data transfer. When read as logic '0' the no PCI bus parity errors have occurred. When read as logic '1' the DMA channel detected a PERR parity error from the bus Master. This error flag can only be set if the Parity Error Response (bit 6 of the bus master's Command Register) is enabled.

5.17.10 Data Width Bit D07

This bit sets the PCI bus data width of the DMA transfer for the applicable DMA channel. When set to logic '0', the PCI bus data width is 32–bits. When set to logic '1', the PCI bus data width is 64–bits.

The setting of this bit affects the Maximum Burst Count, Section 5.17.1. When this bit is cleared to 0, the burst count is the number of 32–bit words; when this bit is set to 1, the burst count is the number of 64–bit words.

5.17.11 Burst Mode Bit D06

This bit sets Burst Mode DMA operation for the applicable DMA channel. When set to logic '1', each data transfer is in multiple bursts, and Maximum Burst Count, Section 5.17.1, specifies the maximum transfer of each burst.

NOTE: This bit should not be cleared to logic '0'.

See Sections 4.5.1 and 4.5.2 for description of Burst Mode DMA operation.

5.17 DMAx Command/Status Register (continued)

5.17.12 Demand Mode Bit D05

This bit sets Demand Mode DMA operation for the applicable DMA chan–nel. When set to logic '0', the Demand Mode is not enabled. When set to logic '1', Demand Mode is enabled. See Section 4.5.1 for description of Demand Mode Operation.

5.17.13 DMA Status Bit D04

This read–only bit indicates the current active status for the applicable DMA channel. When read as logic '0' the DMA channel is active (a data transfer is running). When read as logic '1' the DMA channel is not active.

5.17.14 DMA Channel Buffer Reset Bit D03

This bits issues a reset to one of the DMA channel buffers in the PCI7142. This reset flushes any remaining data from the respective buffer in the PCI7142 (this reset does not clear or reset the associated FIFO in the Signal FPGA). When the bit is set to logic '1', the channel is in reset. When the bit is cleared to logic '0' (its default state), the channel is in a normal run state.

5.17.15 DMA Abort Bit D02

This bit aborts DMA operation for the applicable DMA channel. When this bit is set to logic '1' the DMA channel aborts the current transfer. The user cannot clear this bit to logic '0'—the DMA channel logic clears this bit when it aborts the channel.

5.17.16 DMA Start Bit D01

This bit starts DMA operation for the applicable DMA channel. When this bit is set to logic '1', the channel starts transferring data if the channel is enabled.

5.17.17 DMA Enable Bit D00

This bit enables the applicable DMA channel. Writing a logic '1' enables the channel to transfer data. Writing a logic '0' disables the channel from start–ing a DMA transfer, and if in the process of transferring data, suspends the transfer (pause).

5.18 DMAx Transfer Interval Counter Register

The DMA Transfer Interval Counter Register is used to setup and monitor DMA Count Interrupt operation for the applicable DMA channel. There are nine DMA Transfer Interval Counter registers, one for each DMA channel (0 through 8). The following table shows the contents of the DMA Transfer Interval Counter Register. The subsec– tions following the table provide descriptions of the bits in this register.

Table 5–21: DMAx Transfer Interval Counter Register

DMA0: R/W @ BAR0+0x1008
DMA1: R/W @ BAR0+0x2008
DMA2: R/W @ BAR0+0x3008
DMA3: R/W @ BAR0+0x4008
DMA4: R/W @ BAR0+0x5008
DMA5: R/W @ BAR0+0x6008
DMA6: R/W @ BAR0+0x7008
DMA7: R/W @ BAR0+0x8008
DMA8: R/W @ BAR0+0x9008

	D31	D30 – D28	D27 – D24 *	D23 – D00
Bit Name	Reset	Reserved	Interrupt Counter	Transfer Interval Count
Function	0 =Run 1 =Reset	Write with zeros, Mask when reading	Current Interrupt Count	Transfer Count, in words

* These bits are Read Only
All bits default to the logic '0' state at power on and reset

5.18.1 Reset — Bit D31

This bit resets both the Interrupt Counter and the Transfer Interval Counter. When the bit is set to logic '1', the counters are in reset (reset does not clear this register). When the bit is cleared to logic '0' (its default state), the counters are released for counting (run state).

5.18.2 Interrupt Counter — Bits D27 – D24

The Interrupt Counter is a 4–bit read–only binary value, with bit D27 the MSB. This value indicates the current count of the number of DMA Count interrupts that has been generated.

5.18.3 Transfer Interval Count — Bits D23 – D00

The Transfer Interval Count is a 24–bit binary value, with bit D23 the MSB. This value specifies the data transfer count, in words, that will generate a DMA Transfer Interval Count Finish interrupt to the PCI Interrupt Flag Register, Section 5.2. Words are either 32 or 64 bits wide, as specified by Data Width, bit D07, in the DMA Command/Status Register, Section 5.17.10.

NOTE: When in 32–bit mode, the transfer count must be an even value.

5.19 DMAx Descriptor Transfer Count Registers

The DMA Descriptor Transfer Count Registers are used to setup the DMA operation Descriptors of the applicable DMA channel. There are nine DMA channels, DMA (0 through 8), and four Descriptors for each DMA channel, for a total of 36 DMA Descriptor Transfer Count Registers. Each of these registers has an associated DMA Descriptor PCI Address Register, Section 5.20, that must be programmed at the same time.

The following table shows the contents of the DMA Transfer Count Register. The sub–sections following the table provide descriptions of the bits in this register.

Table 5−22: DMAx Descriptor Transfer Count Registers

	Descriptor 0	Descriptor 1	Descriptor 2	Descriptor 3
DMA0: R/W @	BAR0+0x1010	BAR0+0x1020	BAR0+0x1030	BAR0+0x1040
DMA1: R/W @	BAR0+0x2010	BAR0+0x2020	BAR0+0x2030	BAR0+0x2040
DMA2: R/W @	BAR0+0x3010	BAR0+0x3020	BAR0+0x3030	BAR0+0x3040
DMA3: R/W @	BAR0+0x4010	BAR0+0x4020	BAR0+0x4030	BAR0+0x4040
DMA4: R/W @	BAR0+0x5010	BAR0+0x5020	BAR0+0x5030	BAR0+0x5040
DMA5: R/W @	BAR0+0x6010	BAR0+0x6020	BAR0+0x6030	BAR0+0x6040
DMA6: R/W @	BAR0+0x7010	BAR0+0x7020	BAR0+0x7030	BAR0+0x7040
DMA7: R/W @	BAR0+0x8010	BAR0+0x8020	BAR0+0x8030	BAR0+0x8040
DMA8: R/W @	BAR0+0x9010	BAR0+0x9020	BAR0+0x9030	BAR0+0x9040

	D31	D30	D29− D28	D27 − D00
Bit Name	Chain	Interrupt	Reserved	Transfer Count
Function	0 = End 1 = Next	0 = Disable 1 = Enable	Write with zeros, Mask when reading	Transfer Count, In Bytes (see Section 5.19.3 for bits D02 − D00)
All bits default to the logic '0' state at power on and reset				

5.19.1 Chain

Bit D31

This bit enables the next Descriptor in the chain for the applicable DMA channel, depending on the register accessed. When you clear this bit to logic '0' (its default state), the descriptor chain ends at this Descriptor. Set the bit to logic '1' to continue (link) the data transfer to the next Descriptor.

When you set this bit to '1' for Descriptor 3, the data transfer continues, restarting at Descriptor 0, for a continuous transfer. A continuous transfer can only be stopped using the DMA Abort bit, Section 5.17.15, for this DMA channel.

NOTE: If you setup a DMA channel for a continuous data transfer (Chain bit D31 = 1 in all four Descriptors), you must ensure that the gating signal used for the transfer is not stopped before you stop the transfer or the channel may hang up.

5.19 DMAx Descriptor Transfer Count Register (continued)

5.19.2 Interrupt

Bit D30

This bit enables an end of chain interrupt for the applicable DMA channel. When set to logic '0', the end of chain interrupt is not enabled. When set to logic '1', an end of chain interrupt is enabled.

When enabled, at the end of the last descriptor transfer count specified for this channel a DMA Chain Finish interrupt will be generated to the PCI Interrupt Flag Register, Section 5.2. You must enable the applicable DMA Chain Finish bit in the PCI Interrupt Enable Register, Section 5.3, to route this interrupt to the PCI or Local bus.

5.19.3 Transfer Count

Bits D27 – D00

The Transfer Count is a 28–bit binary value, with bit D27 the MSB. This value indicates the data transfer count, in bytes, for this Descriptor.

The Transfer Count must be an even number of bytes for the word size specified, 32–bit or 64–bit. Thus, if 32–bit transfers are specified, Transfer Count bits D01 and D00 must be '0'; if 64–bit transfers are specified, bits D02, D01, and D00 must be '0'. See Data Width, Section 5.17.10, for the word size.

5.20 DMAx Descriptor PCI Address Registers

The DMA Descriptor PCI Address Registers are used to setup the DMA operation Descriptors of the applicable DMA channel. There are nine DMA channels, DMA (0 through 8), and four Descriptors for each DMA channel, for a total of 36 DMA Descriptor PCI Address Registers. Each of these registers has an associated DMA Descriptor Transfer Count Register, Section 5.19, that must be programmed at the same time.

The following table shows the contents of the DMA Descriptor PCI Address Register. The paragraphs following the table provide descriptions of the bits in this register.

Table 5–23: DMAx Descriptor PCI Address Registers

	Descriptor 0	Descriptor 1	Descriptor 2	Descriptor 3
DMA0: R/W @	BAR0+0x1018	BAR0+0x1028	BAR0+0x1038	BAR0+0x1048
DMA1: R/W @	BAR0+0x2018	BAR0+0x2028	BAR0+0x2038	BAR0+0x2048
DMA2: R/W @	BAR0+0x3018	BAR0+0x3028	BAR0+0x3038	BAR0+0x3048
DMA3: R/W @	BAR0+0x4018	BAR0+0x4028	BAR0+0x4038	BAR0+0x4048
DMA4: R/W @	BAR0+0x5018	BAR0+0x5028	BAR0+0x5038	BAR0+0x5048
DMA5: R/W @	BAR0+0x6018	BAR0+0x6028	BAR0+0x6038	BAR0+0x6048
DMA6: R/W @	BAR0+0x7018	BAR0+0x7028	BAR0+0x7038	BAR0+0x7048
DMA7: R/W @	BAR0+0x8018	BAR0+0x8028	BAR0+0x8038	BAR0+0x8048
DMA8: R/W @	BAR0+0x9018	BAR0+0x9028	BAR0+0x9038	BAR0+0x9048

	D31 – D00
Bit Name	PCI Address
Function	Source/Destination PCI Address

All bits default to the logic '0' state at power on and reset

The PCI Address specifies the source or destination PCI bus address (depending on the associated FIFO's direction) for the data transfer. The PCI address is a 32–bit address.

NOTE: Each DMA channel has four possible Descriptors that can be linked to provide a continuous data transfer. There are two registers for each Descriptor, the DMA Descriptor PCI Address Register, above, and the DMA Descriptor Transfer Count Register, Section 5.19.

5.21 FPGA Load DMA Command/Status Register

NOTE: FPGA Load DMA operation is available only in 7142 boards with a PCI FPGA revision date greater that 10/01/07 (see Section 5.6).

The FPGA Load DMA Command/Status Register controls operation of the FPGA Load DMA channel. The bits in this register allow you to start and monitor the DMA channel and the PCI bus data transfer to the Signal FPGA. See Section 4.5.5 for description of FPGA Load DMA channel operation.

The following table shows the contents of the FPGA Load DMA Command/Status Register. The subsections following the table describe the bits in this register.

Table 5−24: FPGA Load DMA Command/Status Register
R/W @ BAR0+0xA000

	D31 − D08
Bit Name	Reserved
Function	Write with zeros, Mask when reading

	D07	D06	D05	D04 **	D03	D02	D01 *	D00
Bit Name	Reserved			DMA Status	FIFO Reset	Reserved	DMA Start	Reserved
Function	Write with zeros, Mask when reading			0 = DMA Active 1 = Done	0 = Run 1 = Reset	Write 0s, Mask read	0 = Stop 1 = Start	Write 0s, Mask read

* Bit D01 is Write Only
** Bit D04 is Read Only
All bits default to the logic '0' state at power on and reset

5.21.1 DMA Status — Bit D04

This read−only bit indicates the current active status for the FPGA Load DMA channel. When read as logic '0' the DMA channel is active (a PCI bus data transfer is running). When read as logic '1' the DMA channel is not active.

5.21.2 FIFO Reset — Bit D03

This bits issues a reset to the FPGA Load DMA channel FIFO in the PCI7142. This reset flushes any remaining data from the FIFO. When the bit is set to logic '1', the channel is in reset. When the bit is cleared to logic '0' (its default state), the channel is in a normal run state.

5.21.3 DMA Start — Bit D01

This bit starts FPGA Load DMA operation. When this bit is set to logic '1', the FPGA Load DMA channel starts transferring data.

5.22 FPGA Load DMA Transfer Count Register

NOTE: FPGA Load DMA operation is available only in 7142 boards with a PCI FPGA revision date greater that 10/01/07 (see Section 5.6).

The FPGA Load DMA Transfer Count Register is used to setup DMA operation of the FPGA Load DMA channel. This register has an associated FPGA Load DMA PCI Address Register, Section 5.20, that must be programmed at the same time. Refer to Section 4.5.5 for description of FPGA Load DMA channel operation.

The following table shows the contents of this register.

Table 5–25: FPGA Load DMA Transfer Count Register
R/W @ BAR0+0xA010

	D31– D28	D27 – D00
Bit Name	Reserved	Transfer Count
Function	Write with zeros, Mask when reading	Transfer Count, In words
All bits default to the logic '0' state at power on and reset		

The Transfer Count is a 28–bit binary value, with bit D27 the MSB. This value indicates the data transfer count, in 32–bit words, for the FPGA Load DMA channel.

5.23 FPGA Load DMA PCI Address Register

The FPGA Load DMA PCI Address Register is used to setup the DMA operation of the FPGA Load DMA channel. This register has an associated FPGA Load DMA Transfer Count Register, Section 5.19, that must be programmed at the same time.

The following table shows the contents of this register.

Table 5–26: FPGA Load DMA PCI Address Register
R/W @ BAR0+0xA018

	D31 – D00
Bit Name	PCI Address
Function	Source PCI Address
All bits default to the logic '0' state at power on and reset	

The PCI Address is a 32–bit value that specifies the source PCI bus address of the data transfer, for the FPGA Load DMA channel.

Chapter 6: Signal FPGA Registers

6.1 Overview

The XC4VSX55 (or XC4VLX100) Signal FPGA is connected to the PCI interface through the Pentek PCI7142 core. Through this PCI interface, any processor on the PCI interface can receive data from any LTC2255, and/or send data to the DAC5686 for two channels of output. Any PCI Bus Master can control all programmable features on the board, including the four LTC2255s, the DAC5686, and all Signal FPGA memory map regis–ters.

This chapter describes the Signal FPGA registers accessible from a PCI interface pro–cessor. All descriptions are given from the PCI interface processor's viewpoint. All FPGA registers are 16 bits wide, however since the PCI local bus is a 64–bit data path, the register addresses are in increments of 64 bits (8 bytes). Refer to Section 3.3.2 for the memory map of all FPGA registers in the Model 7642.

Below is a simplified block diagram of the Signal FPGA interfaces. (See Figure 6–5 on page 179 for illustration of the FPGA 'Switching and Routing' between the FIFOs and the A/Ds, DAC, and DDR Memory. See Figure 5–1 for a simplified block diagram of the PCI7142 interfaces.)

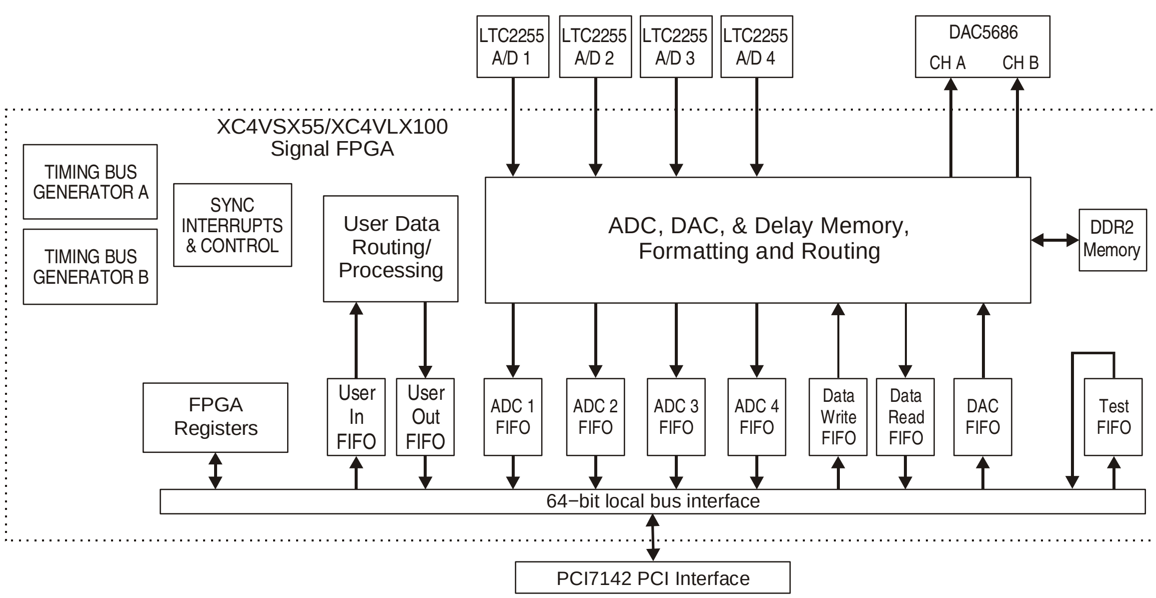


Figure 6–1: Signal FPGA Interface Diagram

6.2 ID Readout Register

The ID Readout Register is a read−only register that provides the Pentek module identification. When read, it always returns a value of **0x7142**, which identifies the 7142 PMC module used on the Model 7642 board.

Table 6−1: ID Readout Register
R.O. @ BAR2+0x8000

	D15 − D12	D11 − D08	D07 − D04	D03 − D00
Bit Name	Board Identification			
Function	0 1 1 1	0 0 0 1	0 1 0 0	0 0 1 0

6.3 TWSI Port Register

The TWSI Port Register allows you to program the voltage/temperature sensor thresh–olds over the board's serial TWSI (Two–Wire Serial Interface) bus.

The board's voltage/temperature sensors provide constant monitoring of critical volt–ages and temperatures on the PCB. When an over–temperature or over–voltage con–dition is indicated, the red **TMP** LED is illuminated on the front panel, and an over–temperature/voltage interrupt is available to a PCI bus device (see Table 6–19).

The following table shows this register's bit layout. The paragraphs following this table describe these bits.

Table 6–2: TWSI Port Register
R/W @ BAR2+0x8018

	D15 – D03	D02	D01	D00
Bit Name	Reserved	SER DIR	SER CLK	SER DAT
Function	Write with zeros, Mask when reading	0 = Write 1 = Read	Serial Clock	Serial Data
All bits default to the logic '0' state at power on and reset				

The TWSI Port Register contains three bits: one bit sets the read/write direction of the register (SER DIR) and the other two provide the serial clock (SER CLK) and data (SER DATA) to/from the serial TWSI bus.

NOTE: You must set up the TWSI bus devices following power on of the Model 7642. The following paragraphs describe the setup of these devices. Routines for programming these devices are provided in the ReadyFlow board support software for the Model 7642 (Pentek part #801.76420).

The devices on the TWSI bus include an Analog Devices ADM1024 Hardware Monitor and a National Semiconductor LM83 Temperature Sensor. You can program these devices from the Temperature/Voltage Sense Register, using internal registers in each device, as described in the following paragraphs.

6.3 TWSI Port Register (continued)

Each device on the TWSI bus has a separate bus address, as listed in the following table. The bus address is sent first, as serial data.

Table 6−3: TWSI Bus Addresses

Device	Write Address	Read Address	Information
LM83 Temperature Sensor	`0x30`	`0x31`	Section 6.3.1
ADM1024 Hardware Monitor	`0x58`	`0x59`	Section 6.3.2

The TWSI bus addresses consist of the 7−bit serial bus address of the device in the first seven most significant bits, plus the least significant bit = 0/1 for write/read.

6.3.1 LM83 Temperature Sensor

There are several temperature sensors in the LM83 that can be programmed on the serial TWSI bus, using LM83 internal registers (refer to the LM83 data sheet, see Section 1.14, for description of these registers).

The following table lists the programmable sensors on the LM83:

Table 6−4: LM83 Programmable Sensors

Monitored Condition	LM83 Input	ReadyFlow Limits *
PCI7142 FPGA Temperature	D1	60° C
Signal FPGA Temperature	D2	70° C
PCB Temperature	D3	60° C
LM83 Local	(internal)	50° C

* These limits are programmed if you run the Pentek ReadyFlow 7642 Hardware Monitor routines for the LM83 using the default values.

A red front panel LED labeled **TEMP** (Section 2.7.3) is illuminated by the LM83 during an over−temperature condition—when any of the four inputs exceeds the HIGH setpoint limit. A separate over−temperature interrupt signal is provided from the LM83 to the Interrupt Status Register, Section 6.9.3, based on over−temperature of the four programmed limits.

6.3 TWSI Port Register (continued)

6.3.2 ADM1024 Hardware Monitor

The board uses several voltage sensor inputs in the ADM1024 that can be programmed on the serial TWSI bus, using ADM1024 internal registers (refer to the ADM1024 data sheet, see Section 1.14, for description of the registers associated with these inputs).

The following table lists the programmable voltage sensors on the ADM1024:

Table 6–5: ADM1024 Programmable Sensors

Monitored Condition	ADM1024 Input	ReadyFlow Limits *
+5 Volt supply	+5 V_{IN}	+5 Volts, ±10%
+12 Volt supply	+12 V_{IN}	+12 Volts, ±10%
+3.3 Volt supply	V_{CC}	+3.3 Volts, ±10%
+1.2 Volt supply A	+V_{CCP1}	+1.2 Volts, ±10%
+1.25 Volt supply	AIN1	+1.25 Volt, ±10%
+2.5 Volt RapidIO supply	AIN2	+2.5 Volt, ±10%
+2.5 Volt supply	+2.5 V_{IN}	+2.5 Volt, ±10%
+1.8 Volt supply	V_{CCP2}	+1.8 Volt, ±10%

* These limits are programmed if you run the Pentek ReadyFlow 7642 Hardware Monitor routines for the ADM1024 using the default values.

An interrupt indicating voltage out of limit is also provided from the ADM1024 to the Interrupt Status Register, Section 6.9.3.

6.4 DCM Control Register

The DCM Control Register controls the Digital Clock Managers (DCM) on the Signal FPGA. Two DCMs in the Signal FPGA are used to control the clock signal provided to the DAC FIFO. See Figure 6–2 on the following page for illustration of the DCM logic for the DAC FIFO clock.

The following table shows the contents of the DCM Register. The subsections following the table provide descriptions of each control bit in this register.

Table 6–6: DCM Control Register
R/W @ BAR2+0x8020

	D15	D14	D13	D12	D11	D10	D09	D08
Bit Name	Reserved							
Function	Write with zeros, Mask when reading							
	D07	**D06 ***	**D05 ***	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	Reserved	PLLLOCK DCM LOCKED	DAC CLK DCM LOCKED	DCM RST	DCM SEL	Reserved		
Function	Write 0, Mask read	0 =Not locked 1 =Locked	0 =Not locked 1 =Locked	0 =Run 1 =Reset	0 =Input clock 1 =DCM	Write with zeros, Mask when reading		

* These bits are Read Only
All bits default to the logic '0' state at power on and reset

6.4.1 DCM LOCKED — Bits D06, D05

Each of these read–only bits indicates the locked status of one of the DCM clocks. Bit D06 is the status of the DCM for the PLLLOCK clock; bit D05 is the status of the DCM for the DAC5686 clock input. When read as logic '0' the DCM is not locked. When read as logic '1' the DCM is locked.

6.4.2 DCM RST — Bit D04

This bit resets both DCMs. Set the bit to logic '1' to reset the DCMs. Clear the bit to logic '0' (its default state) to return the DCMs to normal operation.

6.4.3 DCM SEL — Bit D03

This bit selects or bypasses the DCM operation on both clock lines. Clear the bit to logic '0' (its default state), to bypass the DCM and use the clock source directly (use for clocks up to 80 MHz). Set the bit to logic '1', to use the DCMs on both clock sources (use for clocks of 32 MHz and up).

6.4 DCM Control Register (continued)

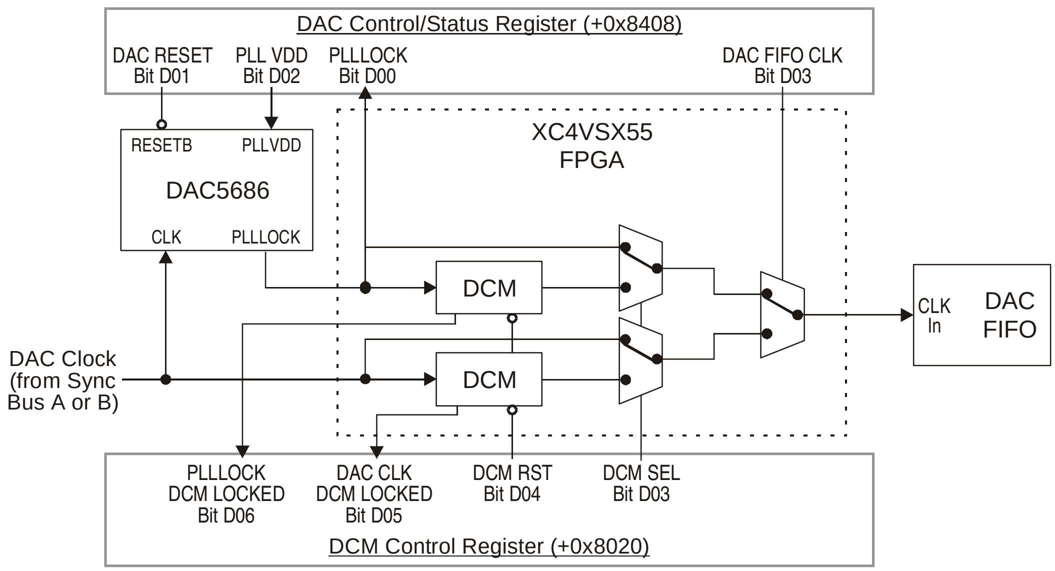


Figure 6–2: DAC FIFO Clock Logic

There are two clock signals that can be selected for clocking writes to the DAC FIFO.

- The clock output from the DAC5686 PLLLOCK pin (if the DAC PLL is disabled by the PLL VDD bit, DAC Control/Status Register, Section 6.11.2) or
- The DAC Clock (selected by the DAC Sync Bus Select Register, Section 6.11.1)

When the DAC FIFO CLK bit D03 (DAC Control/Status Register, Section 6.11.2) is cleared to logic '0' (the default state), the clock output from the DAC5686 PLLLOCK pin is routed to the DAC FIFO. When set to logic '1', the DAC input clock is routed to the DAC FIFO.

To ensure that the DAC FIFO clock is compensated for Signal FPGA circuit delays, both signals can be controlled by DCMs in the FPGA using the DCM SEL bit D03. Clear the bit to logic '0' (its default state), to bypass the DCM and use the clock source directly (use for clocks up to 80 MHz). Set the bit to logic '1', to use the DCMs on both clock sources (use for clocks of 32 MHz and up).

6.5 Miscellaneous Control Register

The Miscellaneous Control Register sets the Endian byte swap mode of the board and controls test pattern generation for the ADC data.

The following table shows the contents of this register. The subsections following the table provide descriptions of each control bit in this register.

Table 6−7: Miscellaneous Control Register
R/W @ BAR2+0x8028

	D15 − D03	D02	D01	D00
Bit Name	Reserved	ENDIANNESS	TEST RAMP	Reserved
Function	Write with zeros, Mask when reading	0 = Little Endian 1 = Big Endian	0 = Disable 1 = Enable	Write 0s, Mask read
All bits default to the logic '0' state at power on and reset				

6.5.1 ENDIANNESS — Bit D02

This bit configures all FIFO transfers across the PCI bus to operate in big or little endian mode. When the bit is cleared to logic '0' (its default state), the FIFO transfers are in little Endian mode. When this bit is set to logic '1', the FIFO transfers are in big Endian mode and the eight bytes of each FIFO 64−bit word are reversed in order.

6.5.2 TEST RAMP — Bit D01

This bit enables or disables test pattern generation for ADC data. When the bit is cleared to logic '0' (its default state), the test pattern is disabled and the LTC2255 outputs are used. When this bit is set to logic '1', the test pattern replaces the LTC2255 data.

The test pattern is a repetitive, incrementing digital signal that can be used for ADC data instead of the LTC2255 outputs. When TEST RAMP is enabled, the output of each LTC2255 to the ADC FIFOs and DDR Memory is replaced by this test pattern. The test pattern signal can be controlled by the Trigger/Gate and Formatting functions applicable to the ADC FIFOs and data routing formats.

6.6 FPGA Revision Registers

The two FPGA Revision Registers identify the Pentek version number and the date of the FPGA code currently installed in the Signal FPGA.

The following tables show the contents of these two registers. The paragraphs below FPGA Revision Register describe the contents of these registers.

Table 6−8: FPGA Revision Register 1
R.O. @ BAR2+0x8038

	D15 − D12	D11 − D08	D07 − D04	D03 − D00
Bit Name	YEAR		MONTH	
Function	3rd Digit of Year	4th Digit of Year	1st Digit of Month	2nd Digit of Month

Table 6−9: FPGA Revision Register 2
R.O. @ BAR2+0x8040

	D15 − D12	D11 − D08	D07 − D04	D03 − D00
Bit Name	DAY		FPGA CODE REVISION	
Function	1st Digit of Day	2nd Digit of Day	Signal FPGA Pentek Code Revision Number	

The Signal FPGA code revision date is coded as six hexadecimal characters in the two registers as follows:

Year: FPGA Revision Register 1, bits D15 – D12, 3rd digit of Year
FPGA Revision Register 1, bits D11 – D08, 4th digit of Year

Month: FPGA Revision Register 1, bits D07 – D04, 1st digit of Month
FPGA Revision Register 1, bits D03 – D00, 2nd digit of Month

Day: FPGA Revision Register 2, bits D15 – D12, 1st digit of Day
FPGA Revision Register 2, bits D11 – D08, 2nd digit of Day

The Signal FPGA Code Revision number (FPGA Revision Register 2, bits D07 – D00) is a binary value, starting at revision 0.

For example, the date of **July 15, 2008** (7/15/08) would appear as follows in the two registers, assuming a code revision of ‘1’.

FPGA Revision Register 1 = **`0x0807`**
FPGA Revision Register 2 = **`0x1501`**

6.7 Core Option Register

The read−only Core Option Register identifies the Pentek Option number of the FPGA IP Core programmed in the Signal FPGA.

The following table shows the contents of this register. The paragraphs below describe the contents of this register.

Table 6−10: Core Option Register
R.O. @ BAR2+0x8048

	D15 − D12	D11 − D08	D07 − D04	D03 − D00
Bit Name	D15 − D12	D11 − D8	D7 − D4	D3 − D0
Function	4th digit of Option Number	3rd digit of Option Number	2nd digit of Option Number	1st digit of Option Number
All bits default to the number of the installed Core at power on and reset				

The Option Number is coded as four hexadecimal characters as follows:

Bits D15 – D12, 4th digit of Option Number
Bits D11 – D08, 3rd digit of Option Number
Bits D07 – D04, 2nd digit of Option Number
Bits D03 – D00, 1st digit of Option Number

For example, Option 430 would appear as follows in this register: **`0x0430`**. If no Pentek FPGA IP Core is provided with the board, this register will read **`0x0000`**.

6.8 Sync Bus Registers

The following subsections describe the Signal FPGA registers that control Sync Buses A and B. Refer to Section 4.7, Timing and Synchronization, for additional information about the sync, gate, and clock signals. The following illustrates the Sync Bus logic.

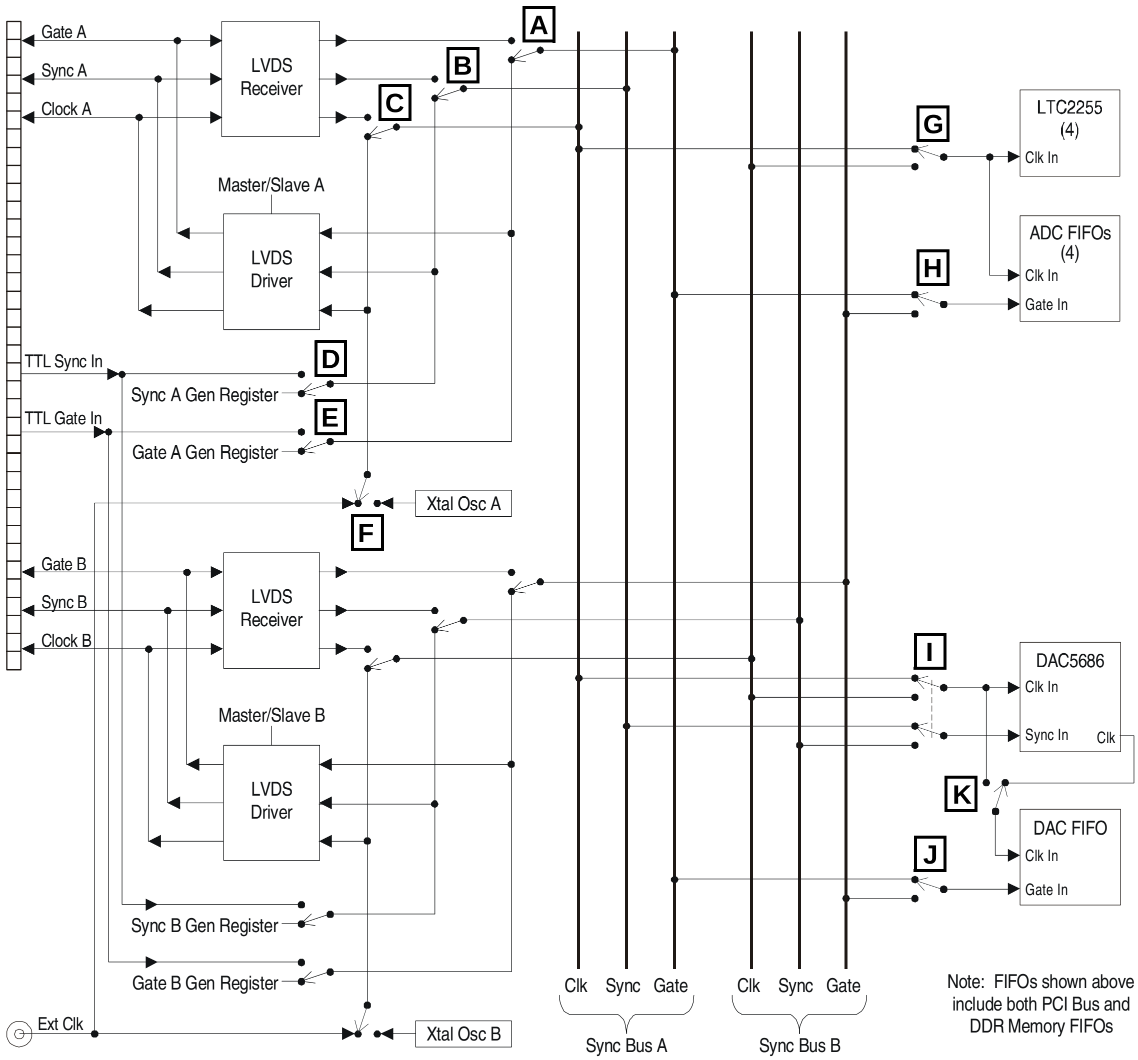


Figure 6−3: Gate/Sync/Clock Logic Diagram

Switches:

- **A** − Master Bus Control Register, Section 6.8.1, bit D09, GATE SRC
- **B** − Master Bus Control Register, Section 6.8.1, bit D06, SYNC SRC
- **C** − Master Bus Control Register, Section 6.8.1, bit D04, CLK SRC
- **D** − Master Bus Control Register, Section 6.8.1, bit D08, SEL GATE
- **E** − Master Bus Control Register, Section 6.8.1, bit D05, SEL SYNC
- **F** − Master Bus Control Register, Section 6.8.1, bit D02, SEL CLK
- **G** − ADC Sync Bus Select Register, Section 6.10.1, bit D00, SEL
- **H** − ADC FIFO Control Register, Section 6.10.2, bit D07, GATE SEL
- **I** − DAC Sync Bus Select Register, Section 6.11.1, bit D00, SEL
- **J** − DAC FIFO Control Register, Section 6.11.3, bit D07, GATE SEL
- **K** − DAC Control/Status Register, Section 6.11.2, bit D03, DAC CLK

(Only Sync Bus A references illustrated − See Section 6.8.4 for Sync Bus B control bits)

6.8 Sync Bus Registers (continued)

6.8.1 Master Bus A Control Register

The Master Bus A Control Register allows you to configure the Model 7642 Sync Bus A as a Master or Slave on the LVDS Sync Bus, toggle the onboard sync bus termination, select the source of the clock, select the source and polarity of the sync, and enable the onboard oscillator.

The following table shows the contents of the Master Control Register. The subsections following the table provide descriptions of each control bit in this register.

Table 6−11: Master Bus A Control Register
R/W @ BAR2+0x8080

	D15 – D11					D10	D09	D08
Bit Name	Reserved					GATE POL	GATE SRC	SEL GATE
Function	Write with zeros, Mask when reading					0 = Negative 1 = Positive	0 = Use SEL GATE (D08) 1= LVDS Bus	0 = Register 1 = Ext TTL Gate
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	SYNC POL	SYNC SRC	SEL SYNC	CLK SRC	Reserved	SEL CLK	TERM	MASTR
Function	0 = Negative 1 = Positive	0 = Use SEL SYNC (D05) 1= LVDS Bus	0 = Register 1 = Ext TTL Sync	0 = Use SEL CLK (D02) 1= LVDS Bus	Write 0, mask read	0 = Oscillator 1 = Ext Clock	0 = None 1 = Terminated	0 = Slave 1 = Master
All bits default to the logic '0' state at power on and reset								

6.8.1.1 GATE POL Bit D10

This bit selects the polarity of the External TTL GATE input selected for sync bus A (see GATE SRC, below). When the bit is cleared to logic '0' (its default state), a negative input enables FIFO input in Gate mode or triggers on a negative−going edge in Trigger mode. When the bit is set to logic '1', a positive input enables FIFO inputs in Gate mode or triggers on a positive−going edge in Trigger mode.

Polarity does not apply to the LVDS Gate/Trigger input nor to a gate generated by the onboard register (Section 6.8.3).

6.8 Sync Bus Registers (continued)

6.8.1 Master Bus A Control Register (continued)

6.8.1.2 GATE SRC — Bit D09

This bit selects the source of the sync bus A GATE signal from either the LVDS Sync Bus or an on–board sync signal. When this bit is cleared to logic '0' (its default state), the gate is selected by SEL GATE, bit D08. When the bit is set to logic '1', the LVDS Sync Bus (SYNC/GATE connector, Section 2.6.4) is the source.

6.8.1.3 SEL GATE — Bit D08

When Sync Bus A on this 7642 is the Sync Bus Master (MASTR bit D00 = 1) or when the LVDS Sync Bus' gate input is not used (GATE SRC bit D09 = 0), this bit selects the gate signal for sync bus A from either a register or the external TTL GATE input. When this bit is cleared to logic '0' (its default state), the Gate A Generator Register, Section 6.8.3, is the source of the gate. When the bit is set to logic '1', the external TTL gate input (SYNC/GATE connector, Section 2.6.4) is the source of the gate.

6.8.1.4 SYNC POL — Bit D07

This bit selects the polarity of the External TTL SYNC input selected for sync bus A (see SYNC SRC, below). Polarity does not apply to the LVDS Sync input nor to a gate generated by the onboard register (Section 6.8.2). When this bit is cleared to logic '0' (its default state), a negative sync is a reset. When the bit is set to logic '1', a positive sync is a reset.

6.8.1.5 SYNC SRC — Bit D06

This bit selects the source of the sync bus A SYNC signal from either the LVDS Sync Bus or an on–board sync signal. When this bit is cleared to logic '0' (its default state), the sync is selected by SEL SYNC bit D05. When the bit is set to logic '1', the LVDS Sync Bus (SYNC/GATE connector, Section 2.6.4) is the source.

6.8 Sync Bus Registers (continued)

6.8.1 Master Bus A Control Register (continued)

6.8.1.6 SEL SYNC Bit D05

When Sync Bus A on this 7642 is the Sync Bus Master (MASTR bit D00 = 1) or when the LVDS Sync Bus' sync input is not used (SYNC SRC bit D06 = 0), this bit selects the sync bus A sync signal from either a register or the external TTL SYNC input. When this bit is cleared to logic '0' (its default state), the Sync A Generator Register, Section 6.8.2, is the source of the sync. When the bit is set to logic '1', the external TTL sync input (SYNC/GATE con– nectors, Section 2.6.4) is the source of the sync.

6.8.1.7 CLK SRC Bit D04

This bit selects the source of the sync bus A clock from either the LVDS Sync Bus or an onboard clock source. When the bit is cleared to logic '0' (its default state), the LVDS Sync Bus is bypassed and the clock is selected by SEL CLK bit D02, below. When this bit is set to logic '1', the EXT LVDS CLK A input (SYNC/GATE connector, Section 2.6.4) is the source of the clock.

6.8.1.8 SEL CLK Bit D02

When Sync Bus A on this 7642 is the Sync Bus Master (MASTR bit D00 = 1) or when the LVDS Sync Bus clock input is not used (CLK SRC bit D04 = 0), this bit selects the sync bus A clock signal from either the onboard oscillator or the external clock input. When this bit is cleared to logic '0' (its default state), the onboard Oscillator A is selected and this oscillator is enabled. When the bit is set to logic '1', the external clock input is used (MMCX EXT CLK connector, Section 2.6.2) and Oscillator A is disabled.

6.8.1.9 TERM Bit D01

This bit enables termination of LVDS Sync Bus A by a Slave, or by a Master if it is the only unit on the bus. When the bit is cleared to logic '0' (its default state), the sync bus is not terminated. When the bit is set to logic '1', the bus is terminated. The sync bus must be terminated when this Model 7642 is the last (or only) Slave on the bus, or if it is a Master or the only unit on the bus.

6.8 Sync Bus Registers (continued)

6.8.1 Master Bus A Control Register (continued)

6.8.1.10 MASTR Bit D00

This bit sets this Model 7642 as either Master or Slave on LVDS Sync Bus A. When the bit is cleared to logic '0' (its default state), this 7642 is a Slave. When bit is set to logic '1', this 7642 is the Sync Bus A Master. When only one 7642 is used it must be set as a Master.

When set as a Sync Bus Master, this Model 7642 Sync Bus A is the source of the sync, clock, and gate signals output to the LVDS Sync Bus. You must select these sources using control bits SYNC SRC (bits D05, D06), CK SRC (bit D04), and GATE SRC (bits D09, D08), and if needed, generate the sync and/or gate signals using the Gate A Generator Register (Section 6.8.3) and Sync A Generator Register (Section 6.8.2).

6.8 Sync Bus Registers (continued)

6.8.2 Sync A Generator Register

The Sync A Generator Register is used on a 7642 that is configured as a Sync Bus Master (MASTR = 1, Master Bus A Control Register, Section 6.8.1.10), or on a 7642 that is not connected to an external LVDS Sync Bus.

The following table shows the contents of the Sync Generator Register.

Table 6−12: Sync A Generator Register
R/W @ BAR2+0x8088

	D15 − D01	D00
Bit Name	Reserved	$\overline{\text{SYNC}}$
Function	Write with zeros, Mask when reading	0 = Reset 1 = Release
Bit D00 defaults to the logic '1' state at power on and reset		

If the sync source is set to the onboard register (SYNC SRC= 00, Master Bus A Control Register, Section 6.8.1.5), the $\overline{\text{SYNC}}$ bit in this register creates the sync signal (SYNC A) for the board, and for output to the LVDS Sync Bus (SYNC/GATE connector, Section 2.6.4) for a Sync Bus Master.

The Sync A signal can be enabled and routed to different devices on the board using the Sync Mask Register, Section 6.8.7, DAC Sync Bus Select Register, Section 6.11.1, and ADC Sync Bus Select Register, Section 6.10.1.

When the bit is cleared to logic '0', the device is in reset. When the bit is set to logic '1' (its default state), the device is released for normal operation.

6.8 Sync Bus Registers (continued)

6.8.3 Gate A Generator Register

The Gate A Generator Register is used on a 7642 that is configured as a Sync Bus Master (MASTR = 1, Master Bus A Control Register, Section 6.8.1.10), or on a 7642 that is not connected to an external LVDS Sync Bus.

The following table shows the contents of the Gate Generator Register.

Table 6−13: Gate A Generator Register
R/W @ BAR2+0x8090

	D15 − D01	D00
Bit Name	Reserved	$\overline{\text{GATE}}$
Function	Write with zeros, Mask when reading	0 = Enable 1 = Disable
Bit D00 defaults to the logic '1' state at power on and reset		

If the gate source is set to the onboard register (GATE SRC= 00, Master Bus A Control Register, Section 6.8.1.2), the $\overline{\text{GATE}}$ bit in this register creates the gate signal (Gate A) for FIFO writing control, and for output to the LVDS Sync Bus (SYNC/GATE connector, Section 2.6.4) for a Sync Bus Master.

When the bit is cleared to logic '0', FIFO writes are enabled. When the bit is set to logic '1' (its default state), FIFO writes are disabled.

6.8 Sync Bus Registers (continued)

6.8.4 Master Bus B Control Register

The Master Bus B Control Register allows you to configure the Model 7642 sync bus B as a Master or Slave on the LVDS Sync Bus, toggle the onboard sync bus termination, select the source of the clock, select the source and polarity of the sync, and enable the onboard oscillator.

The following table shows the contents of the Master Control Register. The subsections following the table provide descriptions of each control bit in this register.

Table 6−14: Master Bus B Control Register
R/W @ BAR2+0x8098

	D15 – D11					D10	D09	D08
Bit Name	Reserved					GATE POL	GATE SRC	SEL GATE
Function	Write with zeros, Mask when reading					0 = Negative 1 = Positive	0 = Use SEL GATE (D08) 1= LVDS Bus	0 = Register 1 = Ext TTL Gate
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	SYNC POL	SYNC SRC	SEL SYNC	CLK SRC	Reserved	SEL CLK	TERM	MASTR
Function	0 = Negative 1 = Positive	0 = Use SEL SYNC (D05) 1= LVDS Bus	0 = Register 1 = Ext TTL Sync	0 = Oscillator 1= LVDS Bus	Write 0, mask read	0 = Oscillator 1 = Ext Clock	0 = None 1 = Terminated	0 = Slave 1 = Master
All bits default to the logic '0' state at power on and reset								

6.8.4.1 GATE POL — Bit D10

This bit selects the polarity of the External TTL GATE input selected for sync bus B (see GATE SRC, below). When the bit is cleared to logic '0' (its default state), a negative input enables FIFO input in Gate mode or triggers on a negative−going edge in Trigger mode. When the bit is set to logic '1', a positive input enables FIFO inputs in Gate mode or triggers on a positive−going edge in Trigger mode.

Polarity does not apply to the LVDS Gate/Trigger input nor to a gate generated by the onboard register (Section 6.8.6).

6.8 Sync Bus Registers (continued)

6.8.4 Master Bus B Control Register (continued)

6.8.4.2 GATE SRC — Bit D09

This bit selects the source of the sync bus B GATE signal from either the LVDS Sync Bus or an on–board sync signal. When this bit is cleared to logic '0' (its default state), the gate is selected by SEL GATE, bit D08. When the bit is set to logic '1', the LVDS Sync Bus (SYNC/GATE connector, Section 2.6.4) is the source.

6.8.4.3 SEL GATE — Bit D08

When Sync Bus B on this 7642 is the Sync Bus Master (MASTR bit D00 = 1) or when the LVDS Sync Bus' gate input is not used (GATE SRC bit D09 = 0), this bit selects the gate signal for sync bus B from either a register or the external TTL SYNC input. When this bit is cleared to logic '0' (its default state), the Gate B Generator Register, Section 6.8.6, is the source of the gate. When the bit is set to logic '1', the external TTL gate input (SYNC/GATE connector, Section 2.6.4) is the source of the gate.

6.8.4.4 SYNC POL — Bit D07

This bit selects the polarity of the External TTL SYNC input selected for sync bus B (see SYNC SRC, below). Polarity does not apply to the LVDS Sync input nor to a gate generated by the onboard register (Section 6.8.5). When this bit is cleared to logic '0' (its default state), a negative sync is a reset. When the bit is set to logic '1', a positive sync is a reset.

6.8.4.5 SYNC SRC — Bit D06

This bit selects the source of the sync bus B SYNC signal from either the LVDS Sync Bus or an on–board sync signal. When this bit is cleared to logic '0' (its default state), the sync is selected by SEL SYNC bit D05. When the bit is set to logic '1', the LVDS Sync Bus (SYNC/GATE connector, Section 2.6.4) is the source.

6.8 Sync Bus Registers (continued)

6.8.4 Master Bus B Control Register (continued)

6.8.4.6 SEL SYNC Bit D05

When Sync Bus B on this 7642 is the Sync Bus Master (MASTR bit D00 = 1) or when the LVDS Sync Bus' sync input is not used (SYNC SRC bit D06 = 0), this bit selects the sync bus B sync signal from either a register or the external TTL SYNC input. When this bit is cleared to logic '0' (its default state), the Sync A Generator Register, Section 6.8.5, is the source of the sync. When the bit is set to logic '1', the external TTL sync input (SYNC/GATE con– nectors, Section 2.6.4) is the source of the sync.

6.8.4.7 CLK SRC Bit D04

This bit selects the source of the sync bus A clock from either the LVDS Sync Bus or an onboard oscillator. When the bit is cleared to logic '0' (its default state), the onboard oscillator is selected (for this setting, OSC DSBL bit D03, below, must be cleared to logic '0' to enable the oscillator). When this bit is set to logic '1', the EXT LVDS CLK A input (SYNC/GATE connector, Section 2.6.4) is the source of the clock.

6.8.4.8 SEL CLK Bit D02

When Sync Bus B on this 7642 is the Sync Bus Master (MASTR bit D00 = 1) or when the LVDS Sync Bus clock input is not used (CLK SRC bit D04 = 0), this bit selects the sync bus B clock signal from either the onboard oscillator or the external clock input. When this bit is cleared to logic '0' (its default state), the onboard Oscillator B is selected and this oscillator is enabled. When the bit is set to logic '1', the external clock input is used (MMCX EXT CLK connector, Section 2.6.2) and Oscillator B is disabled.

6.8.4.9 TERM Bit D01

This bit enables termination of LVDS Sync Bus B by a Slave, or by a Master if it is the only unit on the bus. When the bit is cleared to logic '0' (its default state), the sync bus is not terminated. When the bit is set to logic '1', the bus is terminated. The sync bus must be terminated when this Model 7642 is the last (or only) Slave on the bus, or if it is a Master or the only unit on the bus.

6.8 Sync Bus Registers (continued)

6.8.4 Master Bus B Control Register (continued)

6.8.4.10 MASTR Bit D00

This bit sets this Model 7642 as either Master or Slave on LVDS Sync Bus B. When the bit is cleared to logic '0' (its default state), this 7642 is a Slave. When bit is set to logic '1', this 7642 is the Sync Bus B Master. When only one 7642 is used it must be set as a Master.

When set as a Sync Bus Master, this Model 7642 Sync Bus B is the source of the sync, clock, and gate signals output to the LVDS Sync Bus. You must select these sources using control bits SYNC SRC (bits D05, D06), CK SRC (bit D04), and GATE SRC (bits D09, D08), and if needed, generate the sync and/or gate signals using the Gate B Generator Register (Section 6.8.6) and Sync B Generator Register (Section 6.8.5).

6.8 Sync Bus Registers (continued)

6.8.5 Sync B Generator Register

The Sync B Generator Register is used on a 7642 that is configured as a Sync Bus Master (MASTR = 1, Master Bus B Control Register, Section 6.8.4.10), or on a 7642 that is not connected to an external LVDS Sync Bus.

The following table shows the contents of the Sync Generator Register.

Table 6−15: Sync B Generator Register R/W @ BAR2+0x80A0		
	D15 − D01	**D00**
Bit Name	Reserved	$\overline{\text{SYNC}}$
Function	Write with zeros, Mask when reading	0 = Reset 1 = Release
Bit D00 defaults to the logic '1' state at power on and reset		

If the sync source is set to the onboard register (SYNC SRC= 00, Master Bus B Control Register, Section 6.8.4.5), the $\overline{\text{SYNC}}$ bit in this register creates the sync signal (SYNC B) for the board, and for output to the LVDS Sync Bus (SYNC/GATE connector, Section 2.6.4) for a Sync Bus Master.

The Sync B signal can be enabled and routed to different devices on the board using the Sync Mask Register, Section 6.8.7, DAC Sync Bus Select Register, Section 6.11.1, and ADC Sync Bus Select Register, Section 6.10.1.

When the bit is cleared to logic '0' the device is in reset. When the bit is set to logic '1' (its default state) the device is released for normal operation.

<u>**NOTE:**</u> When using Sync B, the DAC clock must be set up properly using the DAC Control/Status Register, Section 6.11.2.

6.8 Sync Bus Registers (continued)

6.8.6 Gate B Generator Register

The Gate B Generator Register is used on a 7642 that is configured as a Sync Bus Master (MASTR = 1, Master Bus B Control Register, Section 6.8.4.10), or on a 7642 that is not connected to an external LVDS Sync Bus.

The following table shows the contents of the Gate Generator Register.

Table 6−16: Gate B Generator Register
R/W @ BAR2+0x80A8

	D15 – D01	D00
Bit Name	Reserved	$\overline{\text{GATE}}$
Function	Write with zeros, Mask when reading	0 = Enable 1 = Disable
Bit D00 defaults to the logic '1' state at power on and reset		

If the gate source is set to the onboard register (GATE SRC= 00, Master Bus B Control Register, Section 6.8.4.2), the $\overline{\text{GATE}}$ bit in this register creates the gate signal (Gate B) for FIFO writing control, and for output to the LVDS Sync Bus (SYNC/GATE connector, Section 2.6.4) for a Sync Bus Master.

When the bit is cleared to logic '0' FIFO writes are enabled. When the bit is set to logic '1' (its default state) FIFO writes are disabled.

NOTE: When using Gate B, the DAC clock must be set up properly using the DAC Control/Status Register, Section 6.11.2.

6.8 Sync Bus Registers (continued)

6.8.7 Sync Mask Register

This register enables a sync signal to be applied to the DAC5686 Upconverter. The sync signal is derived from the either Sync Bus A or B depending on the setting of the DAC Sync Bus Select Register (Section 6.11.1).

The following table shows the contents of the Sync Mask Register.

Table 6−17: Sync Mask Register
R/W @ BAR2+0x80B0

	D15 − D02	D01	D00
Bit Name	Reserved	DAC	Reserved
Function	Write with zeros, Mask when reading	0 = Disable 1 = Enable	Write 0, mask read

Bit D00 defaults to the logic '1' state at power on and reset

When the DAC bit, D01, is set to logic '1', the sync signal is enabled to the Phase synchronization input (PHSTR) for both DAC channels on the DAC5686. When the bit is cleared to logic '0', the sync is not enabled for the DAC5686.

6.9 Interrupt Registers

The following subsections describe the Signal FPGA registers that control the FGGA interrupts. Refer to Section 4.8, Interrupt Operation, for additional information about the use and programming of interrupts.

6.9.1 System Interrupt Enable Register

The System Interrupt Enable Register contains enable bits for several system interrupt conditions. Each bit enables or disables the generation of an inter– rupt to the Application Interrupt Status Registers, Section 6.9.6. The four Application Interrupt LINT Enable Registers, Section 6.9.4, can be config– ured to generate any of the four local bus interrupt inputs (LINTi0 to LINTi3) to the PCI7142 in response to these interrupts.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–18: System Interrupt Enable Register
R/W @ BAR2+0x8100

	D15 – D08							
Bit Name	D15 – D08							
Function	User–Defined Read/Write							
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	Reserved	OVLD AD4	OVLD AD3	VOLT/TEMP	CLK B LOSS	CLK A LOSS	OVLD AD2	OVLD AD1
Function	Write 0, Mask read	0 = Disable Interrupt 1 = Enable Interrupt						
All bits default to the logic '0' state at power on and reset								

Each bit of this register enables or disables an interrupt to the Application Interrupt Status Registers for a particular system interrupt condition. Set– ting a bit to logic '1' enables generation of a local interrupt when that inter– rupt condition occurs. When a bit is cleared to logic '0' (its default state), the interrupt will not be generated by the associated interrupt condition.

6.9 Interrupt Registers (continued)

6.9.1 System Interrupt Enable Register (continued)

The following table describes the interrupt condition associated with each bit of the System Interrupt register.

Table 6−19: System Interrupt Register Bits

Bit Name	Bit Position	Interrupt Function
User−Defined	D15 − D08	These bits are reserved for user−defined interrupt functions
VOLT/TEMP	D04	This bit is associated with a temperature/voltage limit interrupt from the LM83 or ADM1024 sensors. (To determine the sensor causing the interrupt, use the TWSI Port Register, Section 6.3.)
CLK A LOSS CLK B LOSS	D03 D02	This bit is associated with a Clock loss interrupt from one of the Sync Buses, A or B.
OVLD AD4 OVLD AD3 OVLD AD2 OVLD AD1	D06 D05 D01 D00	Each of these bits is associated with an analog input overload interrupt from one of the four A/D converters.

Note: These bit assignments and definitions apply to System Interrupt Enable Register, Section 6.9.1, System Interrupt Flag Register, Section 6.9.2, and System Interrupt Status Register, Section 6.9.3

6.9 Interrupt Registers (continued)

6.9.2 System Interrupt Flag Register

The System Interrupt Flag Register has one read/clear bit associated with each system interrupt condition (the same bit associations as in the System Interrupt Status Register, Section 6.9.3).

The following table shows which bit in this register is associated with each interrupt condition. Table 6−19, on the prior page, provides description of the interrupt condition associated with each of these bits.

Table 6−20: System Interrupt Flag Register
R/Clr @ BAR2+0x8108

	D15 − D08							
Bit Name	D15 − D08							
Function	User−Defined Read/Clear							
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	Reserved	OVLD AD4	OVLD AD3	VOLT/TEMP	CLK B LOSS	CLK A LOSS	OVLD AD2	OVLD AD1
Function	Write 0, Mask read	**Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch						

IMPORTANT! When reset, including power−up, the state of this register is unknown. You should clear this register before using it, by writing ‘1’s into all bits.

__Read__:

Each bit of this register latches an interrupt occurrence. Logic '1' in any bit in this register indicates that an interrupt has occurred. Note that when any bit in the System Interrupt Status Register (Section 6.9.3) changes to logic '1', the corresponding bit in this register will also be set to logic '1'. However, when a bit in the System Interrupt Status Register changes from logic '1' to logic '0', the corresponding latched bit in this register does not clear, but remains at the logic '1' state.

__Clear__:

Since these bits latch in response to an interrupt, to detect subsequent inter−rupts, you must clear the bits in this register. To clear any bit in this register that is set to logic '1' you must write a '1' to that bit.

Refer to Interrupt Operation, Section 4.8, for further description of interrupt routing and operation on the Model 7642.

6.9 Interrupt Registers (continued)

6.9.3 System Interrupt Status Register

The System Interrupt Status Register has one read−only bit associated with each system interrupt condition (the same bit associations as in the System Interrupt Flag Register, Section 6.9.2).

The following table shows which bit in this register is associated with each interrupt condition. Table 6−19, on page 140, provides description of the interrupt condition associated with each of these bits.

Table 6−21: System Interrupt Status Register
R.O. @ BAR2+0x8110

	D15 − D08							
Bit Name	D15 − D08							
Function	User−Defined Read Only							
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	Reserved	OVLD AD4	OVLD AD3	VOLT/TEMP	CLK B LOSS	CLK A LOSS	OVLD AD2	OVLD AD1
Function	Mask read	0 = No interrupt 1 = Overload	0 = No interrupt 1 = Overload	0 = No interrupt 1 = Overtemp	0 = No interrupt 1 = No Clock	0 = No interrupt 1 = No Clock	0 = No interrupt 1 = Overload	0 = No interrupt 1 = Overload
When reset, including power−up, the state of this register is unknown.								

Each bit in this register changes whenever the associated interrupt indica− tion changes.

- A status bit changes to logic '1' when the source interrupt occurs. Note that when a status bit in this register changes to '1' the corresponding bit in the System Interrupt Flag Register, Section 6.9.2, is set to '1'.
- A status bit in this register clears to '0' when that interrupt condition clears, whereas the associated bit in the System Interrupt Flag Register remains latched at logic '1' until it is cleared by that register's clear action, Section 6.9.2.

If the corresponding bit has been set to logic '1' in the System Interrupt Enable Register (Section 6.9.1), then an System Interrupt Status Register bit transition from logic '0' to logic '1' will also generate a local bus interrupt to the PCI interface. If you do not want an interrupt to occur and have set the corresponding interrupt enable bit to '0', you may poll the System Interrupt Status Register bit to determine if such an event has occurred.

6.9 Interrupt Registers (continued)

6.9.4 Application Interrupt LINT Enable Registers

The Application Interrupt LINT Enable Registers contain enable bits for each application interrupt condition defined. Each bit enables or disables the generation of a local bus interrupt input (LINTix) to the PCI7142.

There are four Application Interrupt LINT Enable Registers, one for each local bus interrupt input (LINTi0 to LINTi3). Each register enables interrupts for the respective local bus interrupt. The PCI7142 can be configured, using the PCI Interrupt Enable Registers, Section 5.3, to route any of the four local bus interrupts (LINTi0 to LINTi3) to PCI interrupts (INTA to INTD).

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–22: Application Interrupt LINT Enable Registers
LINTi0: R/W @ BAR2+0x8118
LINTi1: R/W @ BAR2+0x8120
LINTi2: R/W @ BAR2+0x8128
LINTi3: R/W @ BAR2+0x8130

	D15	D14	D13	D12	D11	D10	D09	D08
Bit Name	SYS INT	GATE B	GATE A	SYNC B	SYNC A	TRIGGER CAPTURE	DDR MEMORY READ FIFO	DDR MEMORY WRITE FIFO
Function	0 = Disable Interrupt 1 = Enable Interrupt							
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	Reserved	DAC FIFO	User Out FIFO	User In FIFO	ADC 4 FIFO	ADC 3 FIFO	ADC 2 FIFO	ADC 1 FIFO
Function	Write 0, Mask read	0 = Disable Interrupt 1 = Enable Interrupt						
All bits default to the logic '0' state at power on and reset								

Each bit of this register enables or disables the generation of a local bus interrupt (LINT0 to LINT3, depending on the register used) for a particular interrupt condition. Setting the bit associated with a given interrupt to logic '1' enables the generation of a local bus interrupt when that interrupt condition occurs. When a bit is cleared to logic '0' (its default state), the local bus interrupt will not be generated by the defined interrupt condition.

Table 6–23, on the next page, provides descriptions of the interrupt condition associated with each of these bits.

6.9 Interrupt Registers (continued)

6.9.4 Application Interrupt LINT Enable Registers (continued)

The following table describes the interrupt condition associated with each bit of the four Application Interrupt Enable registers.

Table 6–23: Application Interrupt Register Bits

Bit Name	Bit Position	Interrupt Function
SYS INT	D15	This bit is associated with an interrupt from the OR'ed result of any of the enabled interrupts in the System Interrupt Enable Register (Section 6.9.1).
GATE B GATE A	D14 D13	Each of these bits is associated with an Sync Bus GATE (A or B) interrupt. The interrupt occurs at the start of the respective gate signal.
SYNC B SYNC A	D12 D11	Each of these bits is associated with an Sync Bus SYNC (A or B) interrupt. The interrupt occurs at the start of the respective Sync pulse.
TRIGGER CAPTURE	D10	This bit is associated with an interrupt from the OR'ed result of the ADC and DAC FIFO Trigger Capture bits. Use the applicable FIFO Status Registers, Sections 6.10.3 or 6.11.4 to determine the interrupting FIFO.
DDR MEMORY READ FIFO	D09	This bit is associated with an interrupt from the DDR Memory Read FIFO status flags (Full, Almost Full, Almost Empty, Empty). Use the DDR Memory Read FIFO Status Registers, Section 6.13.7, to determine the interrupting condition from this FIFO.
DDR MEMORY WRITE FIFO	D08	This bit is associated with an interrupt from the DDR Memory Write FIFO status flags (Full, Almost Full, Almost Empty, Empty). Use the DDR Memory Write FIFO Status Registers, Section 6.13.7, to determine the interrupting condition from this FIFO.
DAC FIFO	D06	This bit is associated with an interrupt from the DAC FIFO status flags (Full, Almost Full, Almost Empty, Empty). Use the DAC FIFO Status Register, Section 6.11.4, to determine the interrupting condition from the FIFO.
User Out FIFO User In FIFO	D05 D06	Each of these bits is associated with an interrupt from a User FIFO status flag (Full, Almost Full, Almost Empty, Empty). Use the User FIFO Status Registers, Section 6.14.2, to determine the interrupting condition from the associated FIFO.
ADC 4 FIFO ADC 3 FIFO ADC 2 FIFO ADC 1 FIFO	D03 D02 D01 D00	Each of these bits is associated with an interrupt from the ADC n FIFO status flags (Full, Almost Full, Almost Empty, Empty). Use the ADC FIFO Status Registers, Section 6.10.3, to determine the interrupting condition from the associated FIFO.
These bit assignments and definitions apply to Application Interrupt Enable Register, Section 6.9.4, Interrupt Flag Register, Section 6.9.5, and Interrupt Status Register, Section 6.9.6		

6.9 Interrupt Registers (continued)

6.9.5 Application Interrupt Flag Register

The Application Interrupt Flag Register has one read/clear bit associated with each application interrupt condition (the same bit associations as the Application Interrupt LINT Enable Register, Section 6.9.4).

The following table shows which bit in this register is associated with each interrupt condition. Table 6–23, on page 144, provides descriptions of the interrupt condition associated with each of these bits.

Table 6–24: Application Interrupt Flag Register
R/Clr @ BAR2+0x8138

	D15	D14	D13	D12	D11	D10	D09	D08
Bit Name	SYS INT	GATE B	GATE A	SYNC B	SYNC A	TRIGGER CAPTURE	DDR MEMORY READ FIFO	DDR MEMORY WRITE FIFO
Function	**Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch							
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	Reserved	DAC FIFO	User Out FIFO	User In FIFO	ADC 4 FIFO	ADC 3 FIFO	ADC 2 FIFO	ADC 1 FIFO
Function	Write 0, Mask read	**Read:** 0 = No interrupt 1 = Interrupt latched **Clear:** 1 = Clear latch						
IMPORTANT! When reset, including power−up, the state of this register is unknown. You should clear this register before using it, by writing '1's into all bits.								

Read: Each bit of this register latches an interrupt occurrence. Logic '1' in any bit in this register indicates that an interrupt has occurred. Note that when any bit in the Application Interrupt Status Register (Section 6.9.6) changes to logic '1', the corresponding bit in this register will also be set to logic '1'. However, when a bit in the Application Interrupt Status Register changes from logic '1' to logic '0', the corresponding latched bit in this register does not clear, but remains at the logic '1' state.

Clear: Since these bits latch in response to an interrupt, to detect subsequent interrupts, you must clear the bits in this register. To clear any bit in this register that is set to logic '1' you must write a '1' to that bit.

6.9 Interrupt Registers (continued)

6.9.6 Application Interrupt Status Register

The Application Interrupt Status Register has one read–only bit associated with each application interrupt condition (the same bit associations as the Application Interrupt LINT Enable Register, Section 6.9.4).

The following table shows which bit in this register is associated with each interrupt condition. Table 6–23, on page 144, provides descriptions of the interrupt condition associated with each of these bits.

Table 6–25: Application Interrupt Status Register
R.O. @ BAR2+0x8140

	D15	D14	D13	D12	D11	D10	D09	D08
Bit Name	SYS INT	GATE B	GATE A	SYNC B	SYNC A	TRIGGER CAPTURE	DDR MEMORY READ FIFO	DDR MEMORY WRITE FIFO
Function	0 = No Interrupt 1 = Interrupt condition asserted							
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	Reserved	DAC FIFO	User Out FIFO	User In FIFO	ADC 4 FIFO	ADC 3 FIFO	ADC 2 FIFO	ADC 1 FIFO
Function	Mask read	0 = No Interrupt 1 = Interrupt condition asserted						
When reset, including power–up, the state of this register is unknown.								

Each bit in this register changes whenever the associated interrupt indica–tion changes.

- A status bit changes to logic '1' when the source interrupt occurs. Note that when a status bit in this register changes to '1' the corresponding bit in the Application Interrupt Flag Register, Section 6.9.5, is set to '1'.
- A status bit in this register clears to logic '0' when that interrupt condition clears, whereas the associated bit in the Application Interrupt Flag Register remains latched at '1' until it is cleared by that register's clear action, Section 6.9.5.

If the corresponding bit in any of the Application Interrupt LINT Enable Registers (Section 6.9.4) has been set to logic '1', then a transition from logic '0' to logic '1' will also generate that local bus interrupt to the PCI interface. If you do not want an interrupt to occur and have set the corresponding interrupt enable bit to '0', you may poll the Application Interrupt Status Register bit to determine if such an event has occurred.

6.10 ADC Registers

The following subsections describe the Signal FPGA registers that control analog to digital input data routing. There are no user–programmable registers for the LTC2255 device.

6.10.1 ADC Sync Bus Select Register

The ADC Sync Bus Select Register selects either Sync Bus A or Sync Bus B to be used as the clock and sync signal source for the analog–to–digital input processing, including the LTC2255s and ADC FIFOs.

Refer to the Master Bus A/B Control Register, Section 6.8.1 or 6.8.4, to configure these signals for the selected Sync Bus.

The following table shows the contents of this register.

Table 6–26: ADC Sync Bus Select Register
R/W @ BAR2+0x8C00

	D15 – D01	D00
Bit Name	Reserved	SEL
Function	Write with zeros, Mask when reading	0 = Bus A 1 = Bus B
Bit D00 defaults to the logic '1' state at power on and reset		

Setting this bit to logic '0' selects Sync Bus A, setting to '1' selects Sync Bus B.

6.10 ADC Registers (continued)

6.10.2 ADC FIFO Control Registers

There are four ADC FIFO Control Registers, one for each ADC channel, 1, 2, 3, and 4. Each register contains bits that select the data packing modes and the source and characteristics of the gate/trigger used to control writing data samples from the LTC2255's to the PCI Interface ADC FIFOs and the DDR Memory ADC FIFOs, and bits that specify the format of the data samples written to these FIFOs.

NOTE: The ADC FIFO register bit settings also control the same packing modes and operating characteristics of the associated DDR Memory ADC FIFOs (see Figure 6–5 on page 179).

The following table shows the contents of these registers. The subsections following the table provide descriptions of the bits in this register.

Table 6–27: ADC FIFO Control Registers
ADC 1: R/W @ BAR2+0x8C08
ADC 2: R/W @ BAR2+0x8C40
ADC 3: R/W @ BAR2+0x8C78
ADC 4: R/W @ BAR2+0x8CB0

	D15 – D14		D13	D12	D11	D10	D09	D08
Bit Name	Reserved		ARM	PRE/POST TRIGGER	Reserved	WORD SWAP	Reserved	PACK MODE
Function	Write with zeros, Mask when reading		0 = Not Armed 1 = Arm	0 = Post 1 = Pre	Write 0s, Mask read	0 = No Swap 1 = Swap	Write 0s, Mask read	0 = No Pack 1 = Pack
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	GATE SEL	Reserved	GATE CONTROL	TRIG CLEAR	HOLD MODE	GATE/TRIG	FIFO RESET	FIFO ENABLE
Function	0 = Gate A 1 = Gate B	Write 0s, Mask read	0 = Enable 1 = Disable	0 = None 1 = Force Gate Inactive	0 = None 1 = Hold	0 = Gate 1 = Trigger	0 = Run 1 = Reset	0 = Disable 1 = Enable

All bits default to the logic '0' state at power on and reset

6.10 ADC Registers (continued)

6.10.2 ADC FIFO Control Registers (continued)

6.10.2.1 ARM Bit D13

This bit arms the trigger for data transfers from the ADC channel (1, 2, 3, or 4 depending on the register used) to the associated ADC PCI Interface and DDR Memory FIFOs. When the bit is cleared to logic '0' (its default state), the trigger is not armed. When the bit is set to logic '1', the trigger is armed and the data transfer can be controlled by the next trigger.

Use the PRE/POST TRIGGER bit D12, Section 6.10.2.2, to specify the trigger delay mode for data transfers. When you select PRE trigger counting mode, you must ARM the trigger to start the transfer of data.

6.10.2.2 PRE/POST TRIGGER Bit D12

This bit selects a trigger mode for data transfers from the ADC channel (1, 2, 3, or 4 depending on the register used) to the associated ADC PCI Interface and DDR Memory FIFOs. When the bit is cleared to logic '0' (its default state), POST trigger delay mode is selected. When the bit is set to logic '1', PRE trigger counting mode is selected.

- For POST trigger delay mode, use the Post Trigger Delay Length Registers, Section 6.10.4, to specify the POST trigger delay length.

- For PRE trigger counting mode, you must ARM the trigger, bit D13 Section 6.10.2.1, to start the transfer of data. Use the Pre Trigger Count Capture Registers, Section 6.10.5, to return the number of PRE trigger samples stored.

When using PRE/POST trigger modes, you must select Trigger mode using the GATE/TRIG bit D02, Section 6.10.2.9, and you must set a trigger length using the associated Trigger Length Register, Section 6.10.6. See Section 4.7.4 for additional information on Pre/Post Triggering.

6.10 ADC Registers (continued)

6.10.2 ADC FIFO Control Registers (continued)

6.10.2.3 WORD SWAP Bit D10

This bit selects the 16–bit word swapping mode for writing data samples to the ADC FIFO (1, 2, 3, or 4). When the bit is cleared to logic '0' (its default state), the data is not swapped, and the first data sample (in A/D packed mode, see below) is in the lowest 16 bits of the 64–bit FIFO word. When the bit is set to logic '1', word swapping is enabled and each pair of 16–bit samples (in packed mode) are swapped in the 64–bit FIFO word, as follows:

Bits D00:15 are swapped with bits D16:31,
Bits D32:47 are swapped with bits D48:63.

6.10.2.4 PACK MODE Bit D08

This bit selects the data packing mode for writing data samples to the ADC FIFO (1, 2, 3, or 4). When the bit is cleared to logic '0' (its default state), the data is unpacked; when the bit is set to logic '1', the A/D data is packed.

Refer to Section 7.3, ADC Data Routing and Formats, for additional information about the ADC data packing modes.

6.10.2.5 GATE SEL Bit D07

This bit selects the source of the gate for writes to the ADC FIFO (1, 2, 3, or 4). When the bit is cleared to logic '0' (its default state), Gate A is the source; when the bit is set to logic '1', Gate B is the source.

6.10.2.6 GATE CONTROL Bit D05

This bit enables Gate operation on the ADC FIFO (1, 2, 3, or 4). When the bit is cleared to '0' (the default state), the FIFO is controlled by the Gate (gate control is Enabled). When the bit is set to '1', the FIFO is free running (gate control Disabled).

6.10 ADC Registers (continued)

6.10.2 ADC FIFO Control Registers (continued)

6.10.2.7 TRIG CLEAR Bit D04

This bit forces the selected gate to the inactive state in Trigger mode (GATE/TRIG bit D02 = 1, below). When this bit is cleared to logic '0' (its default state), there is no effect on the gate. When the bit is set to logic '1', the gate is forced to inactive (ADC FIFO writes disabled), regardless of the trigger length specified (Trigger Length Register, Section 6.10.6), and the TRIGGER CAPTURE bit D05 is cleared in the ADC FIFO Status Register, Section 6.10.3.

6.10.2.8 HOLD MODE Bit D03

This bit enables a gate Hold after the trigger is received in Trigger mode (GATE/TRIG bit D02 = 1, below). When the bit is cleared to logic '0' (its default state), the selected gate is active (ADC FIFO writes enabled) for the specified trigger length after the trigger (specified using the Trigger Length Register, Section 6.10.6), and then goes inactive (ADC FIFO writes disabled). When the bit is set to logic '1', HOLD is enabled and the gate remains active (ADC FIFO writes enabled) after the trigger is received until the Trigger is cleared using the TRIG CLEAR bit D04, above.

Note that when HOLD is enabled, you must set the Trigger Length to any number greater than zero.

6.10.2.9 GATE/TRIG Bit D02

This bit selects Gate or Trigger mode for the gate selected by GATE SEL (bit D07) for enabling ADC FIFO writes. When this bit is cleared to logic '0' (its default state), Gate mode is selected. When this bit is set to logic '1', Trigger mode is selected, and you must set a trigger length using the Trigger Length Register, Section 6.10.6, or enable trigger Hold (HOLD MODE bit D03 above).

6.10.2.10 FIFO RESET Bit D01

This bit resets the ADC FIFO (1, 2, 3, or 4). When the bit is cleared to logic '0' (its default state), the FIFO is in run. When the bit is set to logic '1', the FIFO is in reset.

6.10.2.11 FIFO ENABLE Bit D00

This bit enables the ADC FIFO (1, 2, 3, or 4). When the bit is cleared to logic '0' (its default state), the FIFO is disabled. When the bit is set to logic '1', the FIFO is enabled.

6.10 ADC Registers (continued)

6.10.3 ADC FIFO Status Registers

There are four ADC FIFO Status Registers, one for each ADC channel, 1, 2, 3, and 4. The ADC FIFO Status Registers have several flag and status bits associated with data transfers from each ADC channel. The following table shows the bit layout of this register. The subsections following the table describe these bits.

Table 6−28: ADC FIFO Status Registers

ADC 1 FIFO: R/Clr @ BAR2+0x8148
ADC 2 FIFO: R/Clr @ BAR2+0x8150
ADC 3 FIFO: R/Clr @ BAR2+0x8158
ADC 4 FIFO: R/Clr @ BAR2+0x8160

	D15 − D12			D11	D10	D09	D08
Bit Name	Reserved			FULL FLAG	ALMOST FULL FLAG	ALMOST EMPTY FLAG	EMPTY FLAG
Function	Write with zeros, Mask when reading			**Read:** 0 = FIFO flag condition not active 1 = FIFO flag condition latched **Clear:** 1 = Clear latch			
	D07 − D06		D05 *	D04 *	D03 *	D02 *	D01 *	D00 *
Bit Name	Reserved		TRIGGER CAPTURE	FIFO WRITE ENABLED	FULL	ALMOST FULL	ALMOST EMPTY	EMPTY
Function	Write with zeros, Mask when reading		0 - Not done 1 - Done	0 - Not En 1 - Enabled	0 = FIFO flag condition not active 1 = FIFO flag condition active			

* These bits are Read Only
When reset, including power−up, the state of this register is unknown.
You should clear the register flag bits before using it, by writing '1's into bits D11 − D08.

6.10.3.1 FIFO Flag Bits — Bits D11 – D08

These four bits are latched read/clear bits associated with each FIFO flag condition. Note that when any status bit in this register (D03 – D00, Section 6.10.3.4) changes to '1', the corresponding flag bit will also be set to '1'. However, when a status bit changes from '1' to '0', the corresponding latched bit in this register does not clear, but remains at the logic '1' state.

Read: Logic '1' indicates that a FIFO flag has been active and not cleared, logic '0' indicates that the flag condition is not active.

Clear: Since these bits latch in response to a FIFO flag occur−rence, to detect subsequent FIFO flags, you must clear the bits in this register. To clear any bit in this register that is set to '1' you must write a '1' to that bit.

6.10 ADC Registers (continued)

6.10.3 ADC FIFO Status Registers (continued)

6.10.3.2 TRIGGER CAPTURE Bit D05

This bit indicates the trigger capture status of the data transfer (ADC FIFO 1, 2, 3, or 4, depending on the register being accessed). When read as logic '0' the trigger has not completed. When read as logic '1' the trigger length is completed.

6.10.3.3 FIFO WRITE ENABLED Bit D04

This bit indicates the write enable status of the FIFO (ADC FIFO 1, 2, 3, or 4, depending on the register being accessed). When read as logic '0' the FIFO is not write enabled. When read as logic '1' the FIFO is write enabled.

6.10.3.4 FIFO Status Bits Bits D03 – D00

Each status bit is a non–latched version of the associated FIFO flag condition. When read as logic '1', the associated FIFO flag condition is active (true). When read as logic '0', the associated FIFO flag condition is inactive.

Note that when any status bit changes to logic '1', the corresponding flag bit in this register (Section 6.10.3.1) will also be set to logic '1'.

Use the ADC FIFO Almost Full and Almost Empty Level Registers, Sections 6.10.8 and 6.10.9, to set the Almost Full and Almost Empty FIFO levels.

6.10 ADC Registers (continued)

6.10.4 ADC Post Trigger Delay Length Registers

There are eight ADC Post Trigger Delay Length Registers, two registers for each ADC data channel, 1, 2, 3, and 4. These registers specify the delay in storing data samples to the applicable ADC FIFO after the trigger is received. These registers are valid only when the associated ADC FIFO Control Register, Section 6.10.2, is set for POST trigger mode.

Each ADC channel's delay length is specified using two registers, one for the least significant 16 bits (LSB) and one for the most significant 16 bits (MSB). The following tables show the format of these registers.

Table 6–29: ADC Post Trigger Delay Length LSB Registers
ADC 1: R/W @ BAR2+0x8900
ADC 2: R/W @ BAR2+0x8910
ADC 3: R/W @ BAR2+0x8920
ADC 4: R/W @ BAR2+0x8930

	D15 – D00
Bit Name	D15 – D0
Function	Post Trigger Delay Length least significant bits (LSB)
All bits default to the logic '0' state at power on and reset	

Table 6–30: ADC Post Trigger Delay Length MSB Registers
ADC 1: R/W @ BAR2+0x8908
ADC 2: R/W @ BAR2+0x8918
ADC 3: R/W @ BAR2+0x8928
ADC 4: R/W @ BAR2+0x8938

	D15 – D00
Bit Name	D31– D16
Function	Post Trigger Delay Length most significant bits (MSB)
All bits default to the logic '0' state at power on and reset	

Post Trigger Delay Length is a 32–bit binary value, with the most significant bit (D31) in bit D15 of the MSB register. This value specifies the number of samples to delay, after the trigger is received, before writing A/D samples to the applicable ADC FIFO. See Section 4.7.4 for additional information on Post Triggering.

NOTE: When specifying a Post Trigger Delay Length for an ADC channel, be sure to write to BOTH registers for that ADC channel.

6.10 ADC Registers (continued)

6.10.5 ADC Pre Trigger Count Capture Registers

There are eight ADC Pre Trigger Count Capture Registers, two registers for each ADC data channel, 1, 2, 3, and 4. These read–only registers return the number of data samples stored in the applicable ADC FIFO before the trigger is received. These registers are valid only when the associated ADC FIFO Control Register, Section 6.10.2, is set for PRE trigger mode.

Each ADC channel's pre trigger count is returned using two registers, one for the least significant 16 bits (LSB) and one for the most significant 16 bits (MSB). The following tables show the format of these registers.

Table 6–31: ADC Pre Trigger Count Capture LSB Registers
ADC 1: R.O. @ BAR2+0x8940
ADC 2: R.O. @ BAR2+0x8950
ADC 3: R.O. @ BAR2+0x8960
ADC 4: R.O. @ BAR2+0x8970

	D15 – D00
Bit Name	D15 – D0
Function	Pre Trigger Count Capture least significant bits (LSB)
All bits default to the logic '0' state at power on and reset	

Table 6–32: ADC Pre Trigger Count Capture MSB Registers
ADC 1: R.O. @ BAR2+0x8948
ADC 2: R.O. @ BAR2+0x8958
ADC 3: R.O. @ BAR2+0x8968
ADC 4: R.O. @ BAR2+0x8978

	D15 – D14	D13	D12 – D00
Bit Name	Reserved	WRAP	D28 – D16
Function	Mask read	0 = No wrap 1 = Wrap	Pre Trigger Count Capture most significant bits (MSB)
All bits default to the logic '0' state at power on and reset			

The Pre Trigger Count Capture is a 29–bit binary value, with the most significant bit (D29) in bit D12 of the MSB register. This value indicates the actual number of A/D data samples stored into the applicable ADC FIFO before the trigger is received. When bit D13 of the MSB register reads as logic '1' the number of data samples stored has exceeded the 29–bit Pre Trigger Count Capture field and has wrapped around. See Section 4.7.4 for additional information on Pre Triggering.

NOTE: When reading a Pre Trigger Count for an ADC channel, be sure to read BOTH registers for that ADC channel.

6.10 ADC Registers (continued)

6.10.6 ADC Trigger Length Registers

There are eight ADC Trigger Length Registers, two for each ADC channel, 1, 2, 3, and 4. When Trigger mode is selected for an ADC FIFO gate (GATE/TRIG = 1, ADC FIFO Control Register, Section 6.10.2.9), these registers set the length that the gate is active (ADC FIFO writes enabled) after receipt of the trigger. These registers apply to data transfers from the LTC2255 to both the PCI bus ADC FIFO and the DDR Memory ADC FIFO.

ADC Trigger Length is specified using two registers, one for the least significant 16 bits (LSB) and one for the most significant 16 bits (MSB). The following tables show the contents of the Trigger Length Registers.

Table 6–33: ADC Trigger Length LSB Registers
ADC 1: R/W @ BAR2+0x8C10
ADC 2: R/W @ BAR2+0x8C48
ADC 3: R/W @ BAR2+0x8C80
ADC 4: R/W @ BAR2+0x8CB8

	D15 – D00
Bit Name	D15 – D0
Function	Trigger Length (N–1) least significant bits (LSB)
All bits default to the logic '0' state at power on and reset	

Table 6–34: ADC Trigger Length MSB Registers
ADC 1: R/W @ BAR2+0x8C18
ADC 2: R/W @ BAR2+0x8C50
ADC 3: R/W @ BAR2+0x8C88
ADC 4: R/W @ BAR2+0x8CC0

	D15 – D00
Bit Name	D31– D16
Function	Trigger Length (N–1) most significant bits (MSB)
All bits default to the logic '0' state at power on and reset	

Trigger length is a 32–bit binary value, with the most significant bit (D31) in bit D15 of the MSB register. This value specifies the length of the write gate after the trigger as the number of ADC FIFO writes, up to 65,535. To specify a trigger length of N FIFO writes, set this register to N–1.

See Section 4.7.4 for information on ADC Pre/Post Triggering.

NOTE: When specifying a trigger length, be sure to write to BOTH Trigger Length Registers for that ADC channel.

6.10 ADC Registers (continued)

6.10.7 ADC FIFO Interrupt Mask Registers

There are four ADC FIFO Interrupt Mask Registers, one for each ADC chan– nel, 1, 2, 3, and 4. The Interrupt Mask Registers contain enable bits for each interrupt condition defined. Each bit enables or disables the generation of a local bus interrupt (LINTix) to the PCI7142.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–35: ADC FIFO Interrupt Mask Registers
ADC 1 FIFO: R/W @ BAR2+0x8C20
ADC 2 FIFO: R/W @ BAR2+0x8C58
ADC 3 FIFO: R/W @ BAR2+0x8C90
ADC 4 FIFO: R/W @ BAR2+0x8CC8

	D15 – D06	D05	D04	D03	D02	D01	D00
Bit Name	Reserved	TRIGGER CAPTURE	Reserved	FULL	ALMOST FULL	ALMOST EMPTY	EMPTY
Function	Write with zeros, Mask when reading	0 = Disable Interrupt 1 = Enable Interrupt					
All bits default to the logic '0' state at power on and reset							

Each bit of this register enables or disables the generation of a local bus interrupt to the PCI7142, when interrupts are enabled for this register in the Application Interrupt Enable LINTx Registers, Section 6.9.4.

Setting the bit associated with a given interrupt to the logic '1' state enables the generation of a local bus interrupt when that interrupt condition occurs. When a bit is cleared to logic '0' (its default state), the local bus interrupt will not be generated by the associated interrupt condition.

The interrupt conditions are:

D05 – Trigger Capture, Trigger Length has been counted down to zero
D03 – FIFO is Full
D02 – FIFO is at the Almost Full level
D01 – FIFO is at the Almost Empty level
D00 – FIFO is Empty

6.10 ADC Registers (continued)

6.10.8 ADC FIFO Almost Empty Level Registers

There are four ADC FIFO Almost Empty Level Registers, one for each ADC channel, 1, 2, 3, and 4. This register sets the programmable boundary flag level of the Almost Empty flag for the associated FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–36: ADC FIFO Almost Empty Level Registers
ADC 1 FIFO: R/W @ BAR2+0x8C28
ADC 2 FIFO: R/W @ BAR2+0x8C60
ADC 3 FIFO: R/W @ BAR2+0x8C98
ADC 4 FIFO: R/W @ BAR2+0x8CD0

	D15 – D10	D09 – D00
Bit Name	Reserved	D9 – D0
Function	Write with zeros, Mask when reading	FIFO Almost Empty Level
All bits default to the logic '0' state at power on and reset		

The Almost Empty Level is a 10–bit binary value, with bit D09 the MSB. This value specifies the FIFO Almost Empty flag depth.

NOTE: All 7642 FIFOs are 64 bits wide. Thus, the FIFO level programmed into this register indicates the level of 64–bit words. For example, if you set this register to a level of 12, the FIFO sets a boundary flag when the data transfer reaches 24 32–bit words.

While the FIFO is being read, when the data is at or below the Almost Empty level, the FIFO sets the Almost Empty Flag.

- The FIFO Almost Empty Flag is routed to the associated ADC FIFO Status Register, Section 6.10.3.
- If the Almost Empty Flag is enabled by the associated ADC FIFO Interrupt Mask Register, Section 6.10.7, a local bus interrupt will be asserted to the PCI7142.

6.10 ADC Registers (continued)

6.10.9 ADC FIFO Almost Full Level Registers

There are four ADC FIFO Almost Full Level Registers, one for each ADC channel, 1, 2, 3, and 4. This register sets the programmable boundary flag level of the Almost Full flag for the associated FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–37: ADC FIFO Almost Full Level Registers
ADC 1 FIFO: R/W @ BAR2+0x8C30
ADC 2 FIFO: R/W @ BAR2+0x8C68
ADC 3 FIFO: R/W @ BAR2+0x8CA0
ADC 4 FIFO: R/W @ BAR2+0x8CD8

	D15 – D10	D09 – D00
Bit Name	Reserved	D9 – D0
Function	Write with zeros, Mask when reading	FIFO Almost Full Level
All bits default to the logic '0' state at power on and reset		

The Almost Full Level is a 10–bit binary value, with bit D09 the MSB. This value specifies the FIFO Almost Full flag depth.

NOTE: All 7642 FIFOs are 64 bits wide. Thus, the FIFO level programmed into this register indicates the level of 64–bit words. For example, if you set this register to a level of 12, the FIFO sets a boundary flag when the data transfer reaches 24 32–bit words.

While the FIFO is being written to, when the data is at or above the Almost Full level, the FIFO sets the Almost Full Flag.

- The FIFO Almost Full Flag is routed to the associated ADC FIFO Status Register, Section 6.10.3.
- If the Almost Full Flag is enabled by the associated ADC FIFO Interrupt Mask Register, Section 6.10.7, a local bus interrupt will be asserted to the PCI7142.

6.10 ADC Registers (continued)

6.10.10 ADC FIFO Decimation Divide Registers

There are four ADC Decimation Divide Registers, one for each ADC channel, 1, 2, 3, and 4. This register sets the decimation rate of data samples written to the applicable ADC FIFO (1 or 2).

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–38: ADC FIFO Decimation Divide Register
ADC 1 FIFO: R/W @ BAR2+0x8C38
ADC 2 FIFO: R/W @ BAR2+0x8C70
ADC 3 FIFO: R/W @ BAR2+0x8CA8
ADC 4 FIFO: R/W @ BAR2+0x8CE0

	D15 – D12	D11 – D00
Bit Name	Reserved	B11 – B0
Function	Write with zeros, Mask when reading	12–bit rate divider 'N–1'
All bits default to the logic '0' state at power on and reset		

The rate divider is a 12–bit binary value, with bit D11 the MSB. To take every Nth sample, where N is any value between 1 and 4096, set this register to N–1.

6.11 DAC Registers

The following subsections describe the Signal FPGA registers that control digital to analog output processing, except for the DAC5686 (DAC5687 with Option 101). The DAC5686 contains internal programmable status and control registers. All reads and writes to the DAC5686 internal registers must use the DAC5686 Registers defined in Section 6.12.

NOTE: The following subsections refer only to the DAC5686. In all instances, the same descriptions apply to the DAC5687 with Option 101.

6.11.1 DAC Sync Bus Select Register

The DAC Sync Bus Select Register selects either Sync Bus A or Sync Bus B to be used as the clock and sync signal source for the digital–to–analog output processing, including the DAC5686 and the DAC FIFOs. (The clock for the DAC FIFOs is also controlled by the DAC CLK bit in the DAC Control/Status Register, Section 6.11.2.)

Refer to the Master Bus A/B Control Register, Section 6.8.1 or 6.8.4, to configure these signals for the selected Sync Bus.

The following table shows the contents of this register.

Table 6–39: DAC Sync Bus Select Register
R/W @ BAR2+0x8400

	D15 – D01	D00
Bit Name	Reserved	SEL
Function	Write with zeros, Mask when reading	0 = Bus A 1 = Bus B
Bit D00 defaults to the logic '1' state at power on and reset		

Setting this bit to logic '0' selects Sync Bus A, setting to '1' selects Sync Bus B.

6.11 DAC Registers (continued)

6.11.2 DAC Control/Status Register

The DAC Control/Status Register provides several control and status bits that control the interface to the DAC5686 and select the clock used to control writes to the DAC FIFO (illustrated in Figure 6–4 on the following page). The DAC5686 also contains internal status and control registers that are programmed using the DAC5686 Registers defined in Section 6.12.

The following table shows the contents of the DAC Control/Status Register. The subsections following the table describe the use of each bit.

Table 6−40: DAC Control/Status Register
R/W @ BAR2+0x8408

	D15 – D04	D03	D02	D01	D00 *
Bit Name	Reserved	DAC FIFO CLK	PLL VDD	DAC RESET	PLL LOCK
Function	Write with zeros	0 = PLLLOCK 1 = DAC CLK	0 = Disable 1 = Enable	0 = Run 1 = Reset	0 = Unlocked 1 = Locked

* This bit is Read Only
Bits D03 to D01 default to the logic '0' state at power on and reset.

6.11.2.1 DAC FIFO CLK — Bit D03

This bit selects the clock source for the DAC FIFO. When the bit is cleared to logic '0' (its default state), the clock output from the DAC5686 PLLLOCK pin is routed to the DAC FIFO. When set to logic '1', the DAC input clock (selected by the DAC Sync Bus Select Register, Section 6.11.1) is routed to the DAC FIFO.

Refer to the DCM Control Register, Section 6.4, for description of the Signal FPGA DCMs used for the DAC FIFO clock.

Table 6−41: DAC Control/Status Register States

DAC CLK (D03)	PLL VDD (D02)	PLL LOCK (D00)	Condition
0	0	invalid	DAC clock from PLLLOCK pin is selected. PLL is disabled.
0	1	valid	Invalid DAC CLK and PLL VDD selection.
1	0	invalid	DAC clock from PLLLOCK pin is bypassed. PLL is disabled. Used for DAC dual clock/full bypass modes.
1	1	valid	DAC clock from PLLLOCK pin is bypassed. PLL is enabled.

Refer to the DAC5686 data sheet (see Section 1.14) for description of PLL Mode operation.

6.11 DAC Registers (continued)

6.11.2 DAC Control/Status Register (continued)

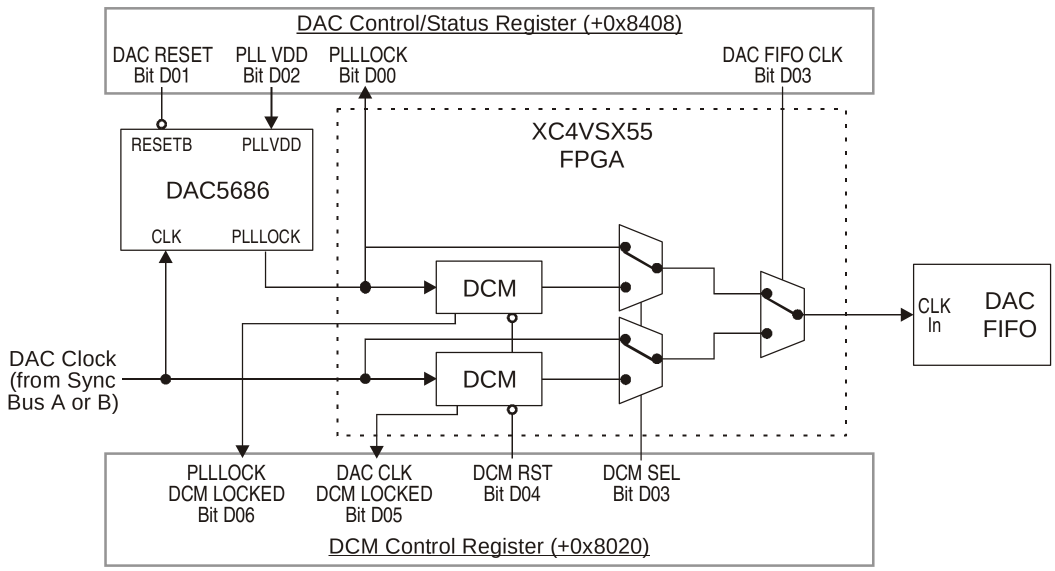


Figure 6–4: DAC FIFO Clock Logic

6.11.2.2 PLL VDD Bit D02

This bit enables the PLL supply voltage on the DAC5686. When the bit is cleared to logic '0' (its default state), the PLL is disabled on the DAC5686, and the PLLLOCK pin outputs the input rate clock. When set to logic '1', the PLL is enabled on the DAC5686, and the PLLLOCK pin indicates lock status (Section 6.11.2.4) and DAC CLK (bit D03) must be set to the Sync Bus clock.

6.11.2.3 DAC RESET Bit D01

This bit issues a reset to the DAC5686. When the bit is cleared to logic '0' (its default state), the DAC is in a normal run state. When the bit is set to logic '1', the DAC is in reset.

6.11.2.4 PLL LOCK Bit D00

This read–only bit is the status of the PLLLOCK pin on the DAC5686. The PLLLOCK pin output is only valid when PLL VDD is enabled (bit D03 set to logic '1'). When PLL is disabled, the PLLLOCK pin outputs the DAC input clock.

6.11 DAC Registers (continued)

6.11.3 DAC FIFO Control Register

The DAC FIFO Control Register contains bits that select the data packing modes and the source and characteristics of the gates used to control read–ing data samples from the PCI Interface DAC FIFO or from the DDR Mem–ory DAC FIFO for the DAC5686.

NOTE: The DAC FIFO register bit settings also control the same packing modes and operating characteristics of the DDR Memory DAC FIFO (see Figure 6–5 on page 179).

The following table shows the contents of the DAC FIFO Control Registers. The subsections following the table describe the bits in this register.

Table 6–42: DAC FIFO Control Register
R/W @ BAR2+0x8410

	D15 – D11					D10	D09	D08
Bit Name	Reserved					WORD SWAP	PACK MODE	
Function	Write with zeros, Mask when reading					0 = No Swap 1 = Swap	00 or 10 = Unpacked 01 = 8-bit Packed 11 = 16-bit Packed	
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	GATE SEL	Reserved	GATE CONTROL	TRIG CLEAR	HOLD MODE	GATE/TRIG	FIFO RESET	FIFO ENABLE
Function	0 = Gate A 1 = Gate B	Write 0s, Mask read	0 = Enable 1 = Disable	0 = None 1 = Force Gate Inactive	0 = None 1 = Hold	0 = Gate 1 = Trigger	0 = Run 1 = Reset	0 = Disable 1 = Enable
All bits default to the logic '0' state at power on and reset								

6.11 DAC Registers (continued)

6.11.3 DAC FIFO Control Register (continued)

6.11.3.1 WORD SWAP Bit D10

This bit sets the 16–bit word swapping mode for reading from the DAC FIFO. When the bit is cleared to logic '0' (its default state), the data is not swapped, and the first 16–bit data sample is in the lowest 16 bits of the 64–bit FIFO word. When the bit is set to logic '1', word swapping is enabled and each pair of 16–bit samples are swapped in the 64–bit FIFO word, as follows:

Bits D00:15 are swapped with bits D16:31,
Bits D32:47 are swapped with bits D48:63.

Refer to Section 7.2, FIFO Word Swap Formats, for additional information about the word swap modes.

6.11.3.2 PACK MODE Bits D09, D08

These two bits set the packing mode for transferring data from the DAC FIFO to the DAC5686 input channels. The settings for the two packing mode bits are:

00	Unpacked	Four 16–bit samples are read from each 64–bit word of the DAC FIFO, two samples for each DAC5686 input channel (inputs A and B) See **NOTE** below.
01	8–bit Packed	Eight 8–bit samples are read from each 64–bit word of the DAC FIFO for DAC5686 input channel B
10	Unpacked	(See Unpacked, code **00**, above)
11	16–bit Packed	Four 16–bit samples are read from each 64–bit word of the DAC FIFO for DAC5686 input channel B

Refer to Section 7.5, DAC Data Routing and Formats for additional information about these Pack Modes.

NOTE: The Model 7642 routes DAC5686 channel B output only to the front panel DAC OUT connector. However, both DAC input channels A and B may be used, depending on the packing mode selected and the operating mode of the DAC5686 (upconversion uses both inputs).

6.11 DAC Registers (continued)

6.11.3 DAC FIFO Control Register (continued)

6.11.3.3 GATE SEL — Bit D07

This bit selects the source of the gate for reads from the DAC FIFO. When the bit is cleared to logic '0' (its default state), Gate A is the source; when the bit is set to logic '1', Gate B is the source.

6.11.3.4 GATE CONTROL — Bit D05

This bit enables Gate operation on the DAC FIFO. When the bit is cleared to '0' (the default state), the FIFO is controlled by the Gate (gate control is Enabled). When the bit is set to '1', the FIFO is free running (gate control Disabled).

6.11.3.5 TRIG CLEAR — Bit D04

This bit forces the selected gate to the inactive state in Trigger mode (GATE/TRIG bit D02 = 1, below). When this bit is cleared to logic '0' (its default state), there is no effect on the gate. When the bit is set to logic '1', the gate is forced to inactive (DAC FIFO reads disabled), regardless of the trigger length specified (Trigger Length Register, Section 6.11.5), and the TRIGGER CAPTURE bit D05 is cleared in the DAC FIFO Status Register, Section 6.11.4.

6.11.3.6 HOLD MODE — Bit D03

This bit enables a gate Hold after the trigger is received in Trigger mode (GATE/TRIG bit D02 = 1, below). When the bit is cleared to logic '0' (its default state), the selected gate is active (DAC FIFO reads enabled) for the specified trigger length after the trigger (specified using the Trigger Length Register, Section 6.11.5), and then goes inactive (DAC FIFO reads disabled). When the bit is set to logic '1', HOLD is enabled and the gate remains active (DAC FIFO reads enabled) after the trigger is received until the Trigger is cleared using the TRIG CLEAR bit D04, above. Note that when HOLD is enabled, you must set the Trigger Length to any number greater than zero.

6.11.3.7 GATE/TRIG — Bit D02

This bit selects Gate or Trigger mode for the gate selected by GATE SEL (bit D07) for enabling DAC FIFO reads. When this bit is cleared to logic '0' (its default state), Gate mode is selected. When this bit is set to logic '1', Trigger mode is selected, and you must set a trigger length using the Trigger Length Register, Section 6.11.5, or enable trigger Hold (HOLD MODE bit D03 above).

6.11 DAC Registers (continued)

6.11.3 DAC FIFO Control Register (continued)

6.11.3.8 FIFO RESET Bit D01

This bit resets the DAC FIFO. When the bit is cleared to logic '0' (its default state), the FIFO is in run; when the bit is set to logic '1', the FIFO is in reset.

6.11.3.9 FIFO ENABLE Bit D00

This bit enables the DAC FIFO. When the bit is cleared to logic '0' (its default state), the FIFO is disabled; when the bit is set to logic '1', the FIFO is enabled.

6.11 DAC Registers (continued)

6.11.4 DAC FIFO Status Register

The DAC FIFO Status Register contains several flag and status bits associ–ated with data transfers from the DAC FIFO to the DAC5686.

The following table shows the bit layout of this register. The subsections following the table describe these bits.

Table 6–43: DAC FIFO Status Register
R/Clr @ BAR2+0x8178

	D15 – D12		D11	D10	D09	D08
Bit Name	Reserved		FULL FLAG	ALMOST FULL FLAG	ALMOST EMPTY FLAG	EMPTY FLAG
Function	Write with zeros, Mask when reading		**Read:** 0 = FIFO flag condition not active 1 = FIFO flag condition latched **Clear:** 1 = Clear latch			
	D07 – D06	**D05 ***	**D04 ***	**D03 ***	**D02 ***	**D01 ***	**D00 ***
Bit Name	Reserved	TRIGGER CAPTURE	FIFO WRITE ENABLED	FULL	ALMOST FULL	ALMOST EMPTY	EMPTY
Function	Write with zeros, Mask when reading	0 - Not done 1 - Done	0 - Not En 1 - Enabled	0 = FIFO flag condition not active 1 = FIFO flag condition active			

* These bits are Read Only
When reset, including power–up, the state of this register is unknown.
You should clear the register flag bits before using it, by writing '1's into bits D11 – D08.

6.11.4.1 FIFO Flag Bits — Bits D11 – D08

These four bits are latched read/clear bits associated with each DAC FIFO flag condition. Note that when any status bit in this register (D03 – D00, Section 6.11.4.4) changes to '1', the corre–sponding flag bit will also be set to '1'. However, when a status bit changes from '1' to '0', the corresponding latched bit in this register does not clear, but remains at the logic '1' state.

Read: Logic '1' indicates that a FIFO flag has been active and not cleared, logic '0' indicates that the flag condition is not active.

Clear: Since these bits latch in response to a FIFO flag occur–rence, to detect subsequent FIFO flags, you must clear the bits in this register. To clear any bit in this register that is set to '1' you must write a '1' to that bit.

6.11 DAC Registers (continued)

6.11.4 DAC FIFO Status Register (continued)

6.11.4.2 TRIGGER CAPTURE Bit D05

This bit indicates the trigger capture status of the data transfer for the DAC FIFO. When read as logic '0' the trigger has not completed. When read as logic '1' the trigger length is completed.

6.11.4.3 FIFO WRITE ENABLED Bit D04

This bit indicates the write enable status of the DAC FIFO. When read as logic '0' the FIFO is not write enabled. When read as logic '1' the FIFO is write enabled.

6.11.4.4 FIFO Status Bits Bits D03 – D00

Each status bit is a non–latched version of the associated FIFO flag condition. When read as logic '1', the associated FIFO flag condition is active (true). When read as logic '0', the associated FIFO flag condition is inactive.

Note that when any status bit changes to '1' the corresponding flag bit in this register (Section 6.11.4.1) will also be set to '1'.

Use the DAC FIFO Almost Full and Almost Empty Level Registers, Sections 6.11.7 and 6.11.8, to set the Almost Full and Almost Empty FIFO levels.

6.11 DAC Registers (continued)

6.11.5 DAC Trigger Length Registers

There are two DAC Trigger Length Registers. The trigger length is specified using two registers, one for the least significant 16 bits (LSB) and one for the most significant 16 bits (MSB).

When Trigger mode is selected for a FIFO gate (GATE/TRIG = 1, DAC FIFO Control Register, Section 6.11.3.7), these registers set the length that the gate is active (DAC FIFO reads enabled) after receipt of the trigger.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–44: DAC Trigger Length LSB Register R/W @ BAR2+0x8418	
	D15 – D00
Bit Name	D15 – D0
Function	Trigger Length (N–1) least significant bits (LSB)
All bits default to the logic '0' state at power on and reset	

Table 6–45: DAC Trigger Length MSB Register R/W @ BAR2+0x8420	
	D15 – D00
Bit Name	D31– D16
Function	Trigger Length (N–1) most significant bits (MSB)
All bits default to the logic '0' state at power on and reset	

The trigger length is a 32–bit binary value, with the most significant bit (D31) in bit D15 of the MSB register. This value specifies the length of the read gate after the trigger as the number of FIFO reads, up to 65,535. To specify a trigger length of N FIFO reads, set this register to N–1.

See Section 4.7.4 for information on DAC Post Triggering.

NOTE: When specifying a trigger length, be sure to write to BOTH Trigger Length Registers.

6.11 DAC Registers (continued)

6.11.6 DAC FIFO Interrupt Mask Register

The DAC FIFO Interrupt Mask Register contains enable bits for each interrupt condition for the DAC FIFO. Each bit enables or disables the generation of a local bus interrupt (LINTix) to the PCI7142.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–46: DAC FIFO Interrupt Mask Register
R/W @ BAR2+0x8428

	D15 – D06	D05	D04	D03	D02	D01	D00
Bit Name	Reserved	TRIGGER CAPTURE	Reserved	FULL	ALMOST FULL	ALMOST EMPTY	EMPTY
Function	Write with zeros, Mask when reading	0 = Disable Interrupt 1 = Enable Interrupt					
All bits default to the logic '0' state at power on and reset							

Each bit of this register enables or disables the generation of a local bus interrupt to the PCI7142, when interrupts are enabled for this register in the Application Interrupt Enable LINTx Registers, Section 6.9.4.

Setting the bit associated with a given interrupt to the logic '1' state enables the generation of a local bus interrupt when that interrupt condition occurs. When a bit is cleared to logic '0' (its default state), the local bus interrupt will not be generated by the associated interrupt condition.

The interrupt conditions are:

D05 – Trigger Capture, Trigger Length has been counted down to zero
D03 – FIFO is Full
D02 – FIFO is at the Almost Full level
D01 – FIFO is at the Almost Empty level
D00 – FIFO is Empty

6.11 DAC Registers (continued)

6.11.7 DAC FIFO Almost Empty Level Register

The DAC FIFO Almost Empty Level Register sets the programmable boundary flag level of the Almost Empty flag for the associated DAC FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–47: DAC FIFO Almost Empty Level Register
R/W @ BAR2+0x8430

	D15 – D14	D13 – D00
Bit Name	Reserved	D13 – D0
Function	Write with zeros, Mask when reading	FIFO Almost Empty Level
All bits default to the logic '0' state at power on and reset		

The Almost Empty Level is a 14–bit binary value, with bit D13 the MSB. This value specifies the FIFO Almost Empty flag depth.

NOTE: All 7642 FIFOs are 64 bits wide. Thus, the FIFO level programmed into this register indicates the level of 64–bit words. For example, if you set this register to a level of 12, the FIFO sets a boundary flag when the data transfer reaches 24 32–bit words.

While the FIFO is being read, when the data is at or below the Almost Empty level, the FIFO sets the Almost Empty Flag.

- The FIFO Almost Empty Flag is routed to the associated DAC FIFO Status Register, Section 6.11.4.
- If the Almost Empty Flag is enabled by the associated DAC FIFO Interrupt Mask Register, Section 6.11.6, a local bus interrupt will be asserted to the PCI7142.

6.11 DAC Registers (continued)

6.11.8 DAC FIFO Almost Full Level Register

The DAC FIFO Almost Full Level Register sets the programmable boundary flag level of the Almost Full flag for the associated DAC FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–48: DAC FIFO Almost Full Level Register R/W @ BAR2+0x8438		
	D15 – D14	**D13 – D00**
Bit Name	Reserved	D13 – D0
Function	Write with zeros, Mask when reading	FIFO Almost Full Level
All bits default to the logic '0' state at power on and reset		

The Almost Full Level is a 14–bit binary value, with bit D13 the MSB. This value specifies the FIFO Almost Full flag depth.

NOTE: All 7642 FIFOs are 64 bits wide. Thus, the FIFO level programmed into this register indicates the level of 64–bit words. For example, if you set this register to a level of 12, the FIFO sets a boundary flag when the data transfer reaches 24 32–bit words.

While the FIFO is being written to, when the data is at or above the Almost Full level, the FIFO sets the Almost Full Flag.

- The FIFO Almost Full Flag is routed to the associated DAC FIFO Status Register, Section 6.11.4.

- If the Almost Full Flag is enabled by the associated DAC FIFO Interrupt Mask Register, Section 6.11.6, a local bus interrupt will be asserted to the PCI7142.

6.11 DAC Registers (continued)

6.11.9 DAC Post Trigger Delay Length Registers

There are two DAC Post Trigger Delay Length Registers. These registers specify the delay in reading data samples from the DAC FIFO, after the trigger.

This delay count is specified using two registers, one for the least significant 16 bits (LSB) and one for the most significant 16 bits (MSB). The following tables show the format of these registers.

Table 6–49: DAC Post Trigger Delay Length LSB Register
R/W @ BAR2+0x8440

	D15 – D00
Bit Name	D15 – D0
Function	Post Trigger Delay Length least significant bits (LSB)
All bits default to the logic '0' state at power on and reset	

Table 6–50: DAC Post Trigger Delay Length MSB Register
R/W @ BAR2+0x8448

	D15 – D00
Bit Name	D31– D16
Function	Post Trigger Delay Length most significant bits (MSB)
All bits default to the logic '0' state at power on and reset	

The Post Trigger Delay Length is a 32–bit binary value, with the most significant bit (D31) in bit D15 of the MSB register. This value specifies the number of samples to delay after the trigger before reading from the DAC FIFO.

See Section 4.7.4 for additional information on Post Triggering.

__NOTE:__ When specifying a Post Trigger Delay Length, be sure to write to BOTH Post Trigger Delay Length registers.

6.12 DAC5686 Registers

The DAC5686 (DAC5687 with Option 101) contains internal programmable status and control registers. All reads and writes to these DAC5686 internal registers are pro–grammed through the Model 7342 Signal FPGA. You can read and write data to the DAC5686 using the Model 7342 DAC5686 Registers defined in the Signal FPGA.

NOTE: The following subsections refer only to the DAC5686. In all instances, the same descriptions apply to the DAC5687 with Option 101.

The Signal FPGA defines two DAC5686 Registers, one for read accesses (DAC5686 Read Register) and one for write accesses (DAC5686 Write Register). The formats of these DAC Registers are provided in Sections 6.12.1 and 6.12.2, respectively, on the fol–lowing two pages.

Each DAC Write Register access to the DAC5686 internal registers consists of data and a register address offset value to steer the data to the correct internal register. There are 16 internal registers in the DAC5686 (32 registers in the DAC5687) that can be accessed in this manner. Refer to the Texas Instruments DAC5686 data sheet (see Section 1.14) for the offset values and descriptions of these registers.

Writing data to a DAC5686 internal register is a single step.

- Write to the DAC5686 Write Register (Section 6.12.2) with DATA DIRECTION = 0, REG OFFSET [4:0] = the offset of the DAC5686 internal register, DATA [7:0] = the data to be written, and the remaining bits = 0.

Reading a DAC5686 internal register consists of two steps.

1) First, write to the DAC5686 Write Register (Section 6.12.2) with DATA DIRECTION = 1, REG OFFSET [4:0] = the offset of the DAC5686 internal register, and the remaining bits = 0. This identifies, to the DAC5686, the address of the register to be accessed during the read operation.

2) Then, read the DAC5686 Read Register (Section 6.12.1). The value of the identified DAC5686 internal register is contained in DATA [7:0], where STATUS = 0 indicates a valid value.

6.12 DAC5686 Registers (continued)

6.12.1 DAC5686 Read Register

The following table shows the contents of the DAC5686 Read Register, and the paragraphs following this table provide descriptions of these bits.

Table 6–51: DAC5686 Read Register
R.O. @ BAR2+0x8470

	D15	D14	D13	D12	D11	D10	D09	D08
Bit Name	STATUS	Reserved						
Function	0 = Valid 1 = Busy	Mask when reading						
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	DATA 7	DATA 6	DATA 5	DATA 4	DATA 3	DATA 2	DATA 1	DATA 0
Function	Data read from the DAC register (when STATUS = 0)							
All bits default to the logic '0' state at power on and reset.								

6.12.1.1 STATUS — Bit D15

This bit indicates the read/write status of the DAC registers.

When you write to the DAC5686 Write Register, Section 6.12.2, this bit goes to logic '1' while the data is sent (serially). If you requested a Read (by setting Bit D15 = 1 in the DAC Write Regis–ter), this STATUS bit stays at '1' until the read from the DAC is complete.

When this STATUS bit reads logic '0', both registers are not busy, and if the DAC Write Register was set for a Read (Bit D15 = 1 in the DAC Write Register) the DATA bits contain valid data read from the DAC5686 internal register.

6.12.1.2 DATA — Bits D7 to D0

These bits contains the data read from the DAC5686 internal reg–ister selected by the DAC5686 Write Register, Section 6.12.2. This data is valid only when STATUS, bit D15, = 0.

__NOTE:__ Before you can access a DAC register, you must set the **sif4** bit in the **config_msb** DAC register, and specify a register offset to read, using the DAC Write Register.

6.12 DAC5686 Registers (continued)

6.12.2 DAC5686 Write Register

The following table shows the contents of the DAC5686 Write Register, and the paragraphs following this table provide descriptions of these bits.

Table 6−52: DAC5686 Write Register W.O. @ BAR2+0x8478								
	D15	**D14**	**D13**	**D12**	**D11**	**D10**	**D09**	**D08**
Bit Name	DATA DIRECTION	N1	N0	REG OFFSET 4	REG OFFSET 3	REG OFFSET 2	REG OFFSET 1	REG OFFSET 0
Function	0 = Write 1 = Read	Write with zeros		Offset of DAC register to write to				
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	DATA 7	DATA 6	DATA 5	DATA 4	DATA 3	DATA 2	DATA 1	DATA 0
Function	When DATA DIRECTION = 0, data to write to the DAC register at offset REG OFFSET[4..0]. When DATA DIRECTION = 1, these bits are do not care.							
All bits default to the logic '0' state at power on and reset.								

6.12.2.1 DATA DIRECTION — Bit D15

This bit indicates the read/write direction of the DATA [7:0] bits. Clear this bit to logic '0' (its default state) to indicate the DATA bits contain a valid value to write to the specified DAC5686 internal register. Set the bit to logic '1' to indicate the DATA bits are do not care, and the REG OFFSET is for a subsequent read by the DAC5686 Read Register, Section 6.12.2.

6.12.2.2 N1, N0 — Bits D14 to D13

These bits must be cleared to logic '0' (their default state).

6.12.2.3 REG OFFSET — Bits D12 to D8

Set these bits to the (address) offset of the DAC5686 internal register to write to. Refer to the Texas Instruments DAC5686 data sheet (see Section 1.14) for the offset values of these registers.

6.12.2.4 DATA — Bits D7 to D0

These bits contain data to write to the DAC5686 internal register. This data is valid only when DATA DIRECTION, bit D15, = 0.

6.13 DDR Memory Registers

The following subsections describe the Signal FPGA registers that control DDR Mem–ory operations. See Figure 6–5 on the next page for illustration of the DDR Memory Control logic. See Section 4.6 for additional information on DDR Memory Operation.

6.13.1 DDR Memory Control Register

The DDR Memory Control register contains bits that select the source and destination and control the operation of all DDR Memory data transfers.

The following table shows the contents of the DDR Memory Control Regis–ter. The subsections following the table provide descriptions of the bits in this register.

Table 6–53: DDR Memory Control Register
R/W @ BAR2+0x8800

	D15	D14	D13	D12	D11	D10	D09	D08
Bit Name	Reserved	BANK 1 PACK	Reserved	BANK 0 PACK	Reserved	DAC SOURCE	Reserved	Reserved
Function	Write 0s, Mask read	0 = ADC 3 1 = ADC 3 & 4	Write 0, Mask read	0 = ADC 1 1 = ADC 1 & 2	Write 0s, Mask read	0 = DAC FIFO 1 = Memory	Write 0s, Mask read	Write 0s, Mask read
	D07	**D06**	**D05**	**D04**	**D03**	**D02**	**D01**	**D00**
Bit Name	Reserved	BANK 2 ENABLE	BANK 1 ENABLE	BANK 0 ENABLE	Reserved	BANK R/W	BANK SELECT	
Function	Write 0s, Mask read	0 = Disable 1 = Enable	0 = Disable 1 = Enable	0 = Disable 1 = Enable	Write 0s, Mask read	0 = Read 1 = Write	See Section 6.13.1.5	
All bits default to the logic '0' state at power on and reset								

6.13.1.1 BANK x PACK — Bits D14 & D12

Each of these bits enables packing data from two DDR Memory ADC FIFOs (ADC1 & 2, or ADC3 & 4, depending on the bit accessed) to one of the DDR Memory banks. When cleared to logic '0' (the default state) only one ADC channel is routed to the associated DDR Memory Bank (ADC1 FIFO to DDR BANK 0, or ADC3 FIFO to DDR BANK 1). When set to logic '1' two ADC channels are packed into the associated DDR Memory (ADC1 & 2 FIFOs to BANK 0, or ADC3 & 4 FIFOs to BANK 1).

See Section 7.4, ADC FIFO to DDR Memory Routing, for addi–tional information about data packing from the ADC FIFOs to DDR Memory.

6.13 DDR Memory Registers (continued)

6.13.1 DDR Memory Control Register (continued)

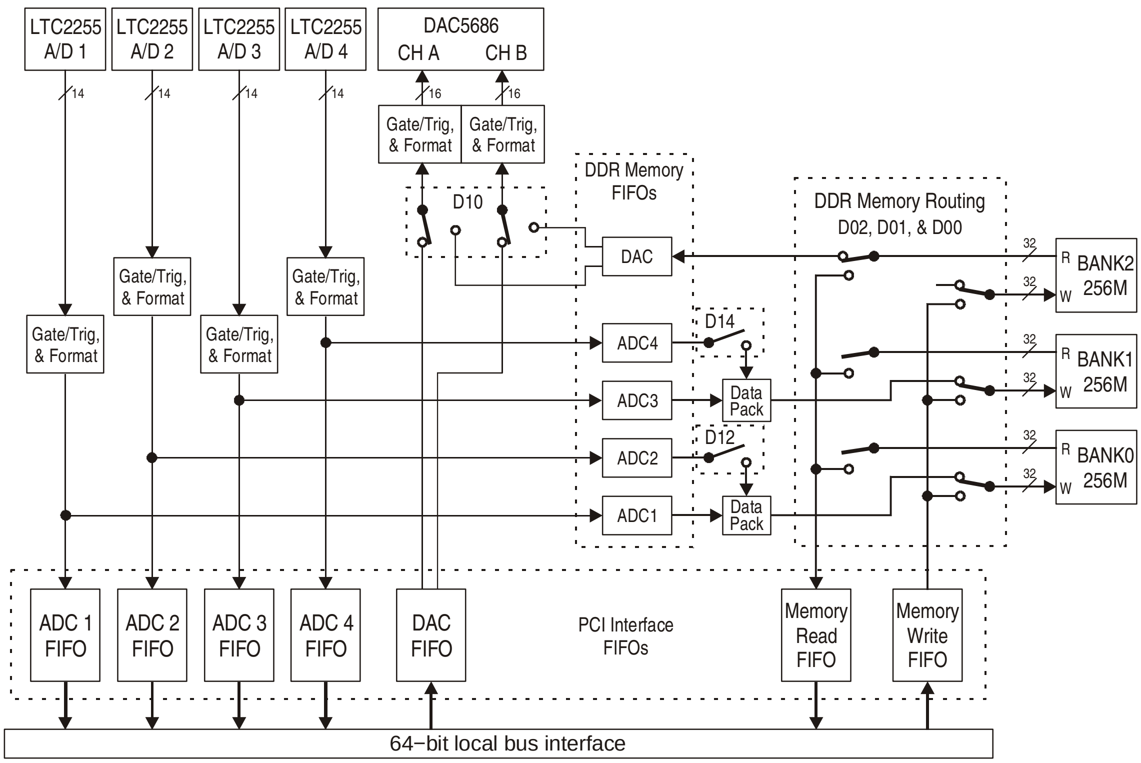


Figure 6−5: DDR Memory Control Logic
(DDR Memory Control Register bits indicated by Dxx − default position shown)

Note	**Refer to Section 6.10.2 for setup and control of the DDR Memory ADC FIFOs.** **Refer to Section 6.11.3 for setup and control of the DDR Memory DAC FIFOs.**

6.13.1.2 DAC SOURCE Bit D10

This bit selects the source for the DAC5686 input channels. When cleared to logic '0' (the default state) the PCI interface DAC FIFO is routed to the DAC5686 input channels. When set to logic '1' the DDR Memory Bank (DAC FIFO) is routed to the DAC5686 input channels.

6.13 DDR Memory Registers (continued)

6.13.1 DDR Memory Control Register (continued)

6.13.1.3 BANK x ENABLE — Bits D06, D05, D04

Each of these bits enables operation of the respective memory bank (0, 1, or 2 depending on the bit accessed). When cleared to logic '0' (the default state) the bank is Disabled. When set to logic '1' the bank is Enabled and the data transfer specified for that memory bank is initiated, starting with a memory bank read, fol–lowed by a memory bank write.

NOTE: Before you change the DDR Memory Bank settings in bits D02, D01, or D00, you must first Disable the associated memory bank using the applicable BANK ENABLE bit. After making such a change, then Enable that memory bank.

NOTE: The ADC FIFO register settings, Section 6.10.2, control the packing modes and operating characteristics of the DDR Memory ADC FIFOs (see Figure 6–5). Thus, if you enable a DDR Memory Bank for transfer from an ADC channel, you must also 'Enable' the associated ADC FIFO, Section 6.10.2.

6.13.1.4 BANK R/W — Bit D02

This bit selects the direction of data transfer, using the Memory Read or Memory Write FIFO, for the DDR Memory bank selected in the BANK SELECT bits, D01:D00, below. When cleared to logic '0' (the default state) the transfer is a read from the selected memory bank using the Memory Read PCI interface FIFO. When set to logic '1' the transfer is a write to the selected memory bank using the Memory Write PCI interface FIFO.

See Table 6–54 on the next page for the possible DDR Memory bank data routing paths.

6.13 DDR Memory Registers (continued)

6.13.1 DDR Memory Control Register (continued)

6.13.1.5 BANK SELECT Bits D01, D00

These two bits select the DDR Memory bank for the Memory Read or Memory Write FIFO data transfer. The settings for these two bits are:

00 none (default)
01 BANK 0
10 BANK 1
11 BANK 2

The direction of the data transfer and the FIFO used are specified using the BANK R/W bit, D02, above. Note that when you access a selected memory bank with the Memory Read or Memory Write FIFO you can only perform one memory bank transfer at a time, either read or write, and you cannot access that memory bank from any other device.

Data routing through the DDR Memory banks is as follows:

Table 6–54: DDR Memory Bank Data Paths

D02	D01	D00	Memory Bank Writes *	Memory Bank Reads
0	0	0	ADC1/ADC2 to BANK 0 ADC3/ADC4 to BANK 1	BANK 2 to DAC
0	0	1	ADC3/ADC4 to BANK 1	BANK 0 to Mem. Read FIFO BANK 2 to DAC
0	1	0	ADC1/ADC2 to BANK 0	BANK 1 to Mem. Read FIFO BANK 2 to DAC
0	1	1	ADC1/ADC2 to BANK 0 ADC3/ADC4 to BANK 1	BANK 2 to Mem. Read FIFO
1	0	0	ADC1/ADC2 to BANK 0 ADC3/ADC4 to BANK 1	BANK 2 to DAC
1	0	1	Mem. Write FIFO to BANK 0 ADC3/ADC4 to BANK 1	BANK 2 to DAC
1	1	0	ADC1/ADC2 to BANK 0 Mem. Write FIFO to BANK 1	BANK 2 to DAC
1	1	1	ADC1/ADC2 to BANK 0 ADC3/ADC4 to BANK 1 Mem. Write FIFO to BANK 2	none

DDR Memory Bank writes (packing) from ADC2 and ADC4 to the associated DDR BANK must be enabled using bits D14 and D12. See Section 6.13.1.1.

6.13 DDR Memory Registers (continued)

6.13.2 DDR Memory Depth Registers

There are six DDR Memory Depth Registers, two registers for each memory bank, 0, 1, and 2. These registers specify the amount of 32–bit words in each memory bank, in 8–word increments, to be used for data input and output.

Each memory bank's depth is specified using two registers, one for the least significant 16 bits (LSB) and one for the most significant 16 bits (MSB). The following tables show the contents of these registers.

Table 6−55: DDR Memory Depth LSB Registers
Bank 0: R/W @ BAR2+0x8808
Bank 1: R/W @ BAR2+0x8818
Bank 2: R/W @ BAR2+0x8828

	D15 − D03	D02 − D00
Bit Name	D15 − D3	D02 − D0
Function	Memory Bank Depth least significant bits (LSB)	Must write zeros
All bits default to the logic '0' state at power on and reset		

Table 6−56: DDR Memory Depth MSB Registers
Bank 0: R/W @ BAR2+0x8810
Bank 1: R/W @ BAR2+0x8820
Bank 2: R/W @ BAR2+0x8830

	D15 − D00
Bit Name	D31− D16
Function	Memory Bank Depth most significant bits (MSB)
All bits default to the logic '0' state at power on and reset	

The Memory Bank Depth is a 32–bit binary value, with the most significant bit (D31) in bit D15 of the MSB register. This value specifies the number of 32–bit words, in multiples of eight words, up to a maximum of 67,108,864 (**0x0400 0000**) words.

A Memory Bank Depth value of 0 (**0x0**) is not valid.

NOTE: When specifying a Memory Bank Depth for a memory bank, be sure to write to BOTH registers for that bank.

The use of these registers is described on the following page.

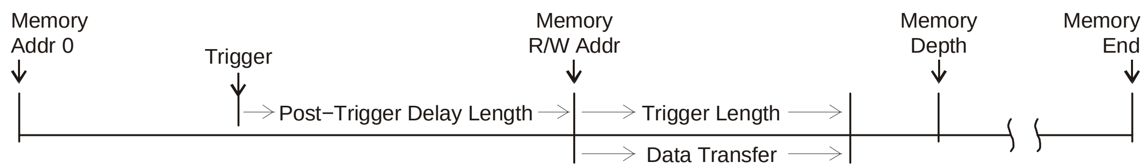
6.13 DDR Memory Registers (continued)

6.13.2 DDR Memory Depth Registers (continued)

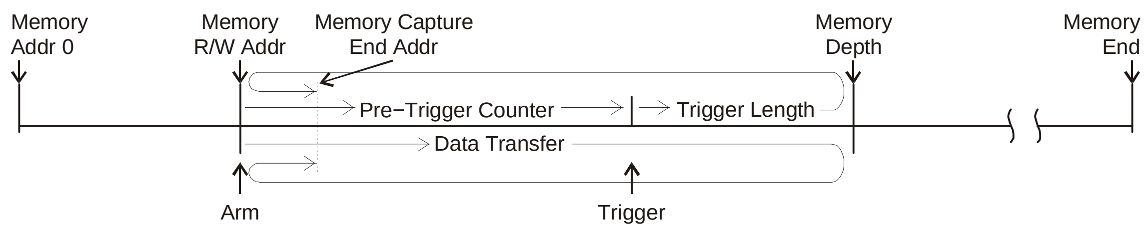
Note that the Memory Bank R/W Address and Depth must be large enough to accommodate the data transfer specified by the Trigger Length.

There are two types of triggering provided for ADC and DAC FIFOs, Pre–Trigger and Post–Trigger. These triggering modes, described in Section 4.7.4, can be used as follows with DDR Memory transfers.

- ❑ Post–Triggering (ADC or DAC channels) – With Post–Triggering, the data transfer starts at the end of the Post–Trigger delay length specified by the Post Trigger Delay Length Register, Section 6.10.4 or 6.11.9, and continues until the end of the Trigger Length specified by the associated Trigger Length Register, Section 6.10.6 or 6.11.5.



- ❑ Pre–Triggering (ADC channels to Memory only) – In Pre–Triggering, data storage starts when the ADC FIFO is ARMed, and continues, wrapping around at the Memory Bank Depth, until the end of the Trigger Length specified by the ADC Trigger Length Register, Section 6.10.6. Use the Memory Capture End Address, Section 6.13.4, to determine the address of the last data sample stored. Use the ADC Pre Trigger Count Capture Register, Section 6.10.5, to determine the actual number of A/D data samples stored in the ADC FIFO before receipt of the trigger. (Note that the pre–trigger data storage may wrap around multiple times prior to receipt of the trigger—only one wrap around is illustrated below).



6.13 DDR Memory Registers (continued)

6.13.3 DDR Memory R/W Address Registers

There are six DDR Memory R/W Address Registers, two registers for each memory bank, 0, 1, and 2. These registers specify the starting address of the data to be read from or written to the respective memory bank.

Each memory bank's address is specified using two registers, one for the least significant 16 bits (LSB) and one for the most significant 16 bits (MSB). The following tables show the format of these registers.

Table 6–57: DDR Memory R/W Address LSB Registers
Bank 0: R/W @ BAR2+0x8838
Bank 1: R/W @ BAR2+0x8848
Bank 2: R/W @ BAR2+0x8858

	D15 – D00
Bit Name	D15 – D0
Function	Memory Bank Address least significant bits (LSB)
All bits default to the logic '0' state at power on and reset	

Table 6–58: DDR Memory R/W Address MSB Registers
Bank 0: R/W @ BAR2+0x8840
Bank 1: R/W @ BAR2+0x8850
Bank 2: R/W @ BAR2+0x8860

	D15 – D00
Bit Name	D31– D16
Function	Memory Bank Address most significant bits (MSB)
All bits default to the logic '0' state at power on and reset	

The Memory Bank Address is a 32–bit binary value, with the most significant bit (D31) in bit D15 of the MSB register. For Memory Write transfers, the associated bank registers specify the address to start writing data to the memory bank. For Memory Read transfers, these registers specify the address to start reading data from the memory bank.

Note that this address must be within the Memory Bank Depth specified for the associated memory bank using the DDR Memory Depth Registers, Section 6.13.2.

NOTE: When specifying a Memory Bank Address for a memory bank, be sure to write to BOTH registers for that bank.

6.13 DDR Memory Registers (continued)

6.13.4 DDR Memory Capture End Address Registers

There are six DDR Memory Capture End Address Registers, two registers for each memory bank, 0, 1, and 2. Each read–only register reports the memory address of the last data sample stored in the applicable DDR Memory bank after the completion of the write transfer from an ADC channel or Memory Write PCI FIFO to that memory bank.

Each memory bank's end address is reported using two registers, one for the least significant 16 bits (LSB) and one for the most significant 16 bits (MSB). The following tables show the format of these registers.

Table 6–59: DDR Memory Capture End Address LSB Registers
Bank 0: R.O. @ BAR2+0x8868
Bank 1: R.O. @ BAR2+0x8878
Bank 2: R.O. @ BAR2+0x8888

	D15 – D00
Bit Name	D15 – D0
Function	DDR Memory End Address least significant bits (LSB)
All bits default to the logic '0' state at power on and reset	

Table 6–60: DDR Memory Capture End Address MSB Registers
Bank 0: R.O. @ BAR2+0x8870
Bank 1: R.O. @ BAR2+0x8880
Bank 2: R.O. @ BAR2+0x8890

	D15 – D00
Bit Name	D31– D16
Function	DDR Memory End Address most significant bits (MSB)
All bits default to the logic '0' state at power on and reset	

The DDR Memory End Address is a 32–bit binary value, with the most significant bit (D31) in bit D15 of the MSB register. This value specifies the memory bank address of the last data sample that was written to that memory bank from the selected resource (ADC channel or Memory Write FIFO). The data path is defined by the DDR Memory Control Register bits D00 – D02, Section 6.13.1.

6.13 DDR Memory Registers (continued)

6.13.5 DDR Memory Trigger Address Registers

There are four DDR Memory Trigger Registers, two registers for each memory bank, 0 and 1. Each read–only register reports the starting memory address in the DDR Memory Bank at receipt of the trigger for a transfer to the Memory Read PCI interface FIFO.

These registers are valid only for data transfers from DDR Memory Bank 0 or 1 to the Memory Read FIFO.

This address is reported using two registers, one for the least significant 16 bits (LSB) and one for the most significant 16 bits (MSB). The following tables show the format of these registers.

Table 6–61: DDR Memory Trigger Address LSB Registers
Bank 0: R.O. @ BAR2+0x8898
Bank 1: R.O. @ BAR2+0x88A8

	D15 – D00
Bit Name	D15 – D0
Function	DDR Memory Trigger Address least significant bits (LSB)
All bits default to the logic '0' state at power on and reset	

Table 6–62: DDR Memory Trigger Address MSB Registers
Bank 0: R.O. @ BAR2+0x88A0
Bank 1: R.O. @ BAR2+0x88B0

	D15 – D00
Bit Name	D31– D16
Function	DDR Memory Trigger Address most significant bits (MSB)
All bits default to the logic '0' state at power on and reset	

The DDR Memory Trigger Address is a 32–bit binary value, with the most significant bit (D31) in bit D15 of the MSB register. This value specifies the starting memory bank address at receipt of the trigger for a transfer from this memory bank to the Memory Read PCI interface FIFO.

6.13 DDR Memory Registers (continued)

6.13.6 DDR Memory FIFO Control Registers

There are two FIFO Control Registers for the DDR Memory, one each for the DDR Memory Read and DDR Memory Write FIFOs. Each register contains bits that control the DDR Memory FIFO.

The following table shows the contents of these registers. The subsections following the table provide descriptions of the bits in this register.

Table 6−63: DDR Memory FIFO Control Registers
DDR Memory Read FIFO: R/W @ BAR2+0x8D58
DDR Memory Write FIFO: R/W @ BAR2+0x8D88

	D15 – D02	D01	D00
Bit Name	Reserved	FIFO RESET	FIFO ENABLE
Function	Write with zeros, Mask when reading	0 = Run 1 = Reset	0 = Disable 1 = Enable

All bits default to the logic '0' state at power on and reset

6.13.6.1 FIFO RESET — Bit D01

This bit resets the DDR Memory Read or DDR Memory Write FIFO. When the bit is cleared to logic '0' (its default state), the FIFO is in run; when the bit is set to logic '1', the FIFO is in reset.

6.13.6.2 FIFO ENABLE — Bit D00

This bit enables the DDR Memory Read or DDR Memory Write FIFO. When the bit is cleared to logic '0' (its default state), the FIFO is disabled; when the bit is set to logic '1', the FIFO is enabled.

6.13 DDR Memory Registers (continued)

6.13.7 DDR Memory FIFO Status Registers

There are two DDR Memory FIFO Status Registers, one each for the DDR Memory Read and DDR Memory Write FIFOs. Each register has several flag and status bits associated with the respective DDR Memory FIFO.

The following table shows the bit layout of this register. The subsections following the table provide descriptions of these bits.

Table 6–64: DDR Memory FIFO Status Registers
DDR Memory Read FIFO: R/Clr @ BAR2+0x8188
DDR Memory Write FIFO: R/Clr @ BAR2+0x8190

	D15 – D12	D11	D10	D09	D08
Bit Name	Reserved	FULL FLAG	ALMOST FULL FLAG	ALMOST EMPTY FLAG	EMPTY FLAG
Function	Write with zeros, Mask when reading	**Read:** 0 = FIFO flag condition not active 1 = FIFO flag condition latched **Clear:** 1 = Clear latch			
	D07 – D04	**D03 ***	**D02 ***	**D01 ***	**D00 ***
Bit Name	Reserved	FULL	ALMOST FULL	ALMOST EMPTY	EMPTY
Function	Write with zeros, Mask when reading	0 = FIFO flag condition not active 1 = FIFO flag condition active			

* These bits are Read Only
When reset, including power−up, the state of this register is unknown.
You should clear the register flag bits before using it, by writing ‘1’s into bits D11 − D08.

6.13.7.1 FIFO Flag Bits — Bits D11 – D08

These four bits are latched read/clear bits associated with each DAC FIFO flag condition. Note that when any status bit in this register (D03 – D00, Section 6.13.7.2) changes to '1', the corre–sponding flag bit will also be set to '1'. However, when a status bit changes from '1' to '0', the corresponding latched bit in this register does not clear, but remains at the logic '1' state.

<u>**Read**</u>: Logic '1' indicates that a FIFO flag has been active and not cleared, logic '0' indicates that the flag condition is not active.

<u>**Clear**</u>: Since these bits latch in response to a FIFO flag occur–rence, to detect subsequent FIFO flags, you must clear the bits in this register. To clear any bit in this register that is set to '1' you must write a '1' to that bit.

6.13 DDR Memory Registers (continued)

6.13.7 DDR Memory FIFO Status Registers (continued)

6.13.7.2 FIFO Status Bits — Bits D03 – D00

Each status bit is a non–latched version of the associated FIFO flag condition. When read as logic '1', the associated FIFO flag condition is active (true). When read as logic '0', the associated FIFO flag condition is inactive.

Note that when any status bit changes to logic '1', the corre–sponding flag bit in this register (Section 6.13.7.1) will also be set to logic '1'.

The Almost Full and Almost Empty FIFO levels are set using the DDR Memory FIFO Almost Full and Almost Empty Level Regis–ters, Sections 6.13.9 and 6.13.10.

6.13 DDR Memory Registers (continued)

6.13.8 DDR Memory FIFO Interrupt Mask Registers

There are two FIFO Interrupt Mask Registers for the DDR Memory, one each for the DDR Memory Read and DDR Memory Write FIFOs. The FIFO Interrupt Mask Registers contain enable bits for each FIFO interrupt condition defined. Each bit enables or disables the generation of a local bus interrupt (LINTix) to the PCI7142.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–65: DDR Memory FIFO Interrupt Mask Registers
DDR Memory Read FIFO: R/W @ BAR2+0x8D70
DDR Memory Write FIFO: R/W @ BAR2+0x8DA0

	D15 – D04	D03	D02	D01	D00
Bit Name	Reserved	FULL	ALMOST FULL	ALMOST EMPTY	EMPTY
Function	Write with zeros, Mask when reading	0 = Disable Interrupt 1 = Enable Interrupt			
All bits default to the logic '0' state at power on and reset					

Each bit of this register enables or disables the generation of a local bus interrupt to the PCI7142, when interrupts are enabled for this register in the Application Interrupt Enable LINTx Registers, Section 6.9.4.

Setting the bit associated with a given interrupt to the logic '1' state enables the generation of a local bus interrupt when that interrupt condition occurs. When a bit is cleared to logic '0' (its default state), the local bus interrupt will not be generated by the associated interrupt condition.

The four flag conditions are:

D03 – FIFO Full
D02 – FIFO Almost Full
D01 – FIFO Almost Empty
D00 – FIFO Empty

6.13 DDR Memory Registers (continued)

6.13.9 DDR Memory FIFO Almost Empty Level Registers

There are two FIFO Almost Empty Level Registers for the DDR Memory, one each for the DDR Memory Read and DDR Memory Write FIFOs. This register sets the programmable boundary flag level of the Almost Empty flag for the associated FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6−66: DDR Memory FIFO Almost Empty Level Registers
DDR Memory Read FIFO: R/W @ BAR2+0x8D78
DDR Memory Write FIFO: R/W @ BAR2+0x8DA8

	D15 − D10	D09 − D00
Bit Name	Reserved	D9 − D0
Function	Write with zeros, Mask when reading	FIFO Almost Empty Level
All bits default to the logic '0' state at power on and reset		

The Almost Empty Level is a 10−bit binary value, with bit D09 the MSB. This value specifies the FIFO Almost Empty flag depth.

NOTE: All 7642 FIFOs are 64 bits wide. Thus, the FIFO level programmed into this register indicates the level of 64−bit words. For example, if you set this register to a level of 12, the FIFO sets a boundary flag when the data transfer reaches 24 32−bit words.

While the FIFO is being read, when the data is at or below the Almost Empty level, the FIFO sets the Almost Empty Flag.

- The Almost Empty Flag is routed to the associated DDR Memory FIFO Status Register, Section 6.13.7.
- If the Almost Empty Flag is enabled by the associated DDR Memory FIFO Interrupt Mask Register, Section 6.13.8, a local bus interrupt will be asserted to the PCI7142.

6.13 DDR Memory Registers (continued)

6.13.10 DDR Memory FIFO Almost Full Level Registers

There are two FIFO Almost Full Level Registers for the DDR Memory, one each for the DDR Memory Read and DDR Memory Write FIFOs. This register sets the programmable boundary flag level of the Almost Full flag for the associated FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–67: DDR Memory FIFO Almost Full Level Registers
DDR Memory Read FIFO: R/W @ BAR2+0x8D80
DDR Memory Write FIFO: R/W @ BAR2+0x8DB0

	D15 – D10	D09 – D00
Bit Name	Reserved	D9 – D0
Function	Write with zeros, Mask when reading	FIFO Almost Full Level
All bits default to the logic '0' state at power on and reset		

The Almost Full Level is a 10–bit binary value, with bit D09 the MSB. This value specifies the FIFO Almost Full flag depth.

NOTE: All 7642 FIFOs are 64 bits wide. Thus, the FIFO level programmed into this register indicates the level of 64–bit words. For example, if you set this register to a level of 12, the FIFO sets a boundary flag when the data transfer reaches 24 32–bit words.

While the FIFO is being written to, when the data is at or above the Almost Full level, the FIFO sets the Almost Full Flag.

- The FIFO Almost Full Flag is routed to the associated DDR Memory FIFO Status Register, Section 6.13.7.

- If the Almost Full Flag is enabled by the associated DDR Memory FIFO Interrupt Mask Register, Section 6.13.8, a local bus interrupt will be asserted to the PCI7142.

6.14 User FIFO Registers

The following subsections describe the Signal FPGA registers that control the User Out and User In FIFOs. These User FIFOs are not connected to any board resources in the default 7642 configuration. The user may route FPGA signals through these FIFOs with custom FPGA logic using the available Pentek GateFlow® FPGA Design Kit, Model 4953 Option 142 (see Section 1.12).

6.14.1 User FIFO Control Registers

There are two User FIFO Control Registers, one each for the User Out and User In FIFOs. Each register contains bits that control reading or writing data samples from or to the FIFO.

The following table shows the contents of the User FIFO Control Registers. The subsections following the table describe these bits.

Table 6–68: User FIFO Control Registers
User Out FIFO: R/W @ BAR2+0x8E08
User In FIFO: R/W @ BAR2+0x8E28

	D15 – D08		
Bit Name	Reserved		
Function	Write with zeros, Mask when reading		
	D07	**D01**	**D00**
Bit Name	Reserved	FIFO RESET	FIFO ENABLE
Function	Write with zeros, Mask when reading	0 = Run 1 = Reset	0 = Disable 1 = Enable
All bits default to the logic '0' state at power on and reset			

6.14.1.1 FIFO RESET — Bit D01

This bit resets the User FIFO (Out or In, depending on the register being accessed). When the bit is cleared to logic '0' (its default state), the FIFO is in run; when the bit is set to logic '1', the FIFO is in reset.

6.14.1.2 FIFO ENABLE — Bit D00

This bit enables the User FIFO (Out or In). When the bit is cleared to logic '0' (its default state), the FIFO is disabled; when the bit is set to logic '1', the FIFO is enabled.

6.14 User FIFO Registers (continued)

6.14.2 User FIFO Status Registers

There are two User FIFO Status Registers, one each for the User Out and User In FIFOs. Each register contains several flag and status bits associated with data transfers from each User FIFO.

The following table shows the bit layout of this register. The subsections following the table describe these bits.

<table>
<tr><th colspan="7">Table 6–69: User FIFO Status Registers
User In FIFO: R/Clr @ BAR2+0x8168
User Out FIFO: R/Clr @ BAR2+0x8170</th></tr>
<tr><td></td><td colspan="2">D15 – D12</td><td>D11</td><td>D10</td><td>D09</td><td>D08</td></tr>
<tr><td>Bit Name</td><td colspan="2">Reserved</td><td>FULL FLAG</td><td>ALMOST FULL FLAG</td><td>ALMOST EMPTY FLAG</td><td>EMPTY FLAG</td></tr>
<tr><td>Function</td><td colspan="2">Write with zeros,
Mask when reading</td><td colspan="4">Read: 0 = FIFO flag condition not active
1 = FIFO flag condition latched
Clear: 1 = Clear latch</td></tr>
<tr><td></td><td>D07 – D05</td><td>D04 *</td><td>D03 *</td><td>D02 *</td><td>D01 *</td><td>D00 *</td></tr>
<tr><td>Bit Name</td><td>Reserved</td><td>FIFO WRITE ENABLED</td><td>FULL</td><td>ALMOST FULL</td><td>ALMOST EMPTY</td><td>EMPTY</td></tr>
<tr><td>Function</td><td>Write with zeros,
Mask when reading</td><td>0 - Not En
1 - Enabled</td><td colspan="4">0 = FIFO flag condition not active
1 = FIFO flag condition active</td></tr>
<tr><td colspan="7">* These bits are Read Only
When reset, including power–up, the state of this register is unknown.
You should clear the register flag bits before using it, by writing ‘1’s into bits D11 – D08.</td></tr>
</table>

6.14.2.1 FIFO Flag Bits — Bits D11 – D08

These four bits are latched read/clear bits associated with each User FIFO flag condition. Note that when any status bit in this register (D03 – D00, Section 6.14.2.3) changes to '1', the corresponding flag bit will also be set to '1'. However, when a status bit changes from '1' to '0', the corresponding latched bit in this register does not clear, but remains at the logic '1' state.

Read: Logic '1' indicates that a FIFO flag has been active and not cleared, logic '0' indicates that the flag condition is not active.

Clear: Since these bits latch in response to a FIFO flag occurrence, to detect subsequent FIFO flags, you must clear the bits in this register. To clear any bit in this register that is set to '1' you must write a '1' to that bit.

6.14 User FIFO Registers (continued)

6.14.2 User FIFO Status Registers (continued)

6.14.2.2 FIFO WRITE ENABLED Bit D04

This bit indicates the write enable status of the User FIFO (Out or In, depending on the register being accessed). When read as logic '0' the FIFO is not write enabled. When read as logic '1' the FIFO is write enabled.

6.14.2.3 FIFO Status Bits Bits D03 – D00

Each status bit is a non–latched version of the associated FIFO flag condition. When read as logic '1', the associated FIFO flag condition is active (true). When read as logic '0', the associated FIFO flag condition is inactive. Note that when any status bit changes to logic '1', the corresponding flag bit in this register (Section 6.14.2.1) will also be set to logic '1'.

Use the User FIFO Almost Full and Almost Empty Level Regis– ters, Sections 6.14.4 and 6.14.5, to set the Almost Full and Almost Empty FIFO levels.

6.14 User FIFO Registers (continued)

6.14.3 User FIFO Interrupt Mask Registers

There are two User FIFO Interrupt Mask Registers, one each for the User Out and User In FIFOs. The User FIFO Interrupt Mask Registers contain enable bits for FIFO interrupt conditions. Each bit enables or disables the generation of a local bus interrupt (LINTix) to the PCI7142.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6−70: User FIFO Interrupt Mask Registers
User Out FIFO: R/W @ BAR2+0x8E10
User In FIFO: R/W @ BAR2+0x8E30

	D15 – D06	D05	D04	D03	D02	D01	D00
Bit Name	Reserved			FULL	ALMOST FULL	ALMOST EMPTY	EMPTY
Function	Write with zeros, Mask when reading			0 = Disable Interrupt 1 = Enable Interrupt			
All bits default to the logic '0' state at power on and reset							

Each bit of this register enables or disables the generation of a local bus interrupt to the PCI7142, when interrupts are enabled for this register in the Application Interrupt Enable LINTx Registers, Section 6.9.4.

Setting the bit associated with a given interrupt to the logic '1' state enables the generation of a local bus interrupt when that interrupt condition occurs. When a bit is cleared to logic '0' (its default state), the local bus interrupt will not be generated by the associated interrupt condition.

The User FIFO interrupt conditions are:

D03 – FIFO is Full
D02 – FIFO is at the Almost Full level
D01 – FIFO is at the Almost Empty level
D00 – FIFO is Empty

6.14 User FIFO Registers (continued)

6.14.4 User FIFO Almost Empty Level Registers

There are two User FIFO Almost Empty Level Registers, one each for the User Out and User In FIFOs. This register sets the programmable boundary flag level of the Almost Empty flag for the associated FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–71: User FIFO Almost Empty Level Registers
User Out FIFO: R/W @ BAR2+0x8E18
User In FIFO: R/W @ BAR2+0x8E38

	D15 – D10	D09 – D00
Bit Name	Reserved	D9 – D0
Function	Write with zeros, Mask when reading	FIFO Almost Empty Level
All bits default to the logic '0' state at power on and reset		

The Almost Empty Level is a 10–bit binary value, with bit D09 the MSB. This value specifies the FIFO Almost Empty flag depth.

NOTE: All 7642 FIFOs are 64 bits wide. Thus, the FIFO level programmed into this register indicates the level of 64–bit words. For example, if you set this register to a level of 12, the FIFO sets a boundary flag when the data transfer reaches 24 32–bit words.

While the FIFO is being read, when the data is at or below the Almost Empty level, the FIFO sets the Almost Empty Flag.

- The FIFO Almost Empty Flag is routed to the associated User FIFO Status Register, Section 6.14.2.
- If the Almost Empty Flag is enabled by the associated User FIFO Interrupt Mask Register, Section 6.14.3, a local bus interrupt will be asserted to the PCI7142.

6.14 User FIFO Registers (continued)

6.14.5 User FIFO Almost Full Level Registers

There are two User FIFO Almost Full Level Registers, one each for the User Out and User In FIFOs. This register sets the programmable boundary flag level of the Almost Full flag for the associated FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–72: User FIFO Almost Full Level Registers
User Out FIFO: R/W @ BAR2+0x8E20
User In FIFO: R/W @ BAR2+0x8E40

	D15 – D10	D09 – D00
Bit Name	Reserved	D9 – D0
Function	Write with zeros, Mask when reading	FIFO Almost Full Level
All bits default to the logic '0' state at power on and reset		

The Almost Full Level is a 10–bit binary value, with bit D09 the MSB. This value specifies the FIFO Almost Full flag depth.

<u>**NOTE:**</u> All 7642 FIFOs are 64 bits wide. Thus, the FIFO level programmed into this register indicates the level of 64–bit words. For example, if you set this register to a level of 12, the FIFO sets a boundary flag when the data transfer reaches 24 32–bit words.

While the FIFO is being written to, when the data is at or above the Almost Full level, the FIFO sets the Almost Full Flag.

- The FIFO Almost Full Flag is routed to the associated User FIFO Status Register, Section 6.14.2.
- If the Almost Full Flag is enabled by the associated User FIFO Interrupt Mask Register, Section 6.14.3, a local bus interrupt will be asserted to the PCI7142.

6.15 Test FIFO Registers

The following subsections describe the Signal FPGA registers that control the Test FIFO operations.

6.15.1 Test FIFO Control Register

There is one FIFO Control Register for the Test FIFO. This register contains bits that are used to control the FIFO operation.

The following table shows the contents of this register. The subsections following the table provide descriptions of the bits in this register.

Table 6–73: Test FIFO Control Register
R/W @ BAR2+0x8D38

	D15 – D02	D01	D00
Bit Name	Reserved	RESET	ENABLE
Function	Write with zeros, Mask when reading	0 = Run 1 = Reset	0 = Disable 1 = Enable
All bits default to the logic '0' state at power on and reset			

6.15.1.1 RESET Bit D01

This bit resets the Test FIFO. When the bit is cleared to logic '0' (its default state), the FIFO is in run; when the bit is set to logic '1', the FIFO is in reset.

6.15.1.2 ENABLE Bit D00

This bit enables the Test FIFO. When the bit is cleared to logic '0' (its default state), the FIFO is disabled; when the bit is set to logic '1', the FIFO is enabled.

6.15 Test FIFO Registers (continued)

6.15.2 Test FIFO Status Register

The Test FIFO Status Register contains several flag and status bits associated with the Test FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–74: Test FIFO Status Register
R/Clr @ BAR2+0x8198

	D15 – D12	D11	D10	D09	D08
Bit Name	Reserved	FULL FLAG	ALMOST FULL FLAG	ALMOST EMPTY FLAG	EMPTY FLAG
Function	Write with zeros, Mask when reading	**Read:** 0 = FIFO flag condition not active 1 = FIFO flag condition latched **Clear:** 1 = Clear latch			
	D07 – D04	**D03 ***	**D02 ***	**D01 ***	**D00 ***
Bit Name	Reserved	FULL	ALMOST FULL	ALMOST EMPTY	EMPTY
Function	Write with zeros, Mask when reading	0 = FIFO flag condition not active 1 = FIFO flag condition active			

* These bits are Read Only
When reset, including power–up, the state of this register is unknown.
You should clear the register flag bits before using it, by writing ‘1’s into bits D11 – D08.

6.15.2.1 FIFO Flag Bits — Bits D11 – D08

These four bits are latched read/clear bits associated with each Test FIFO flag condition. Note that when any status bit in this register (D03 – D00, Section 6.15.2.2) changes to '1', the corre–sponding flag bit will also be set to '1'. However, when a status bit changes from '1' to '0', the corresponding latched bit in this register does not clear, but remains at the logic '1' state.

Read: Logic '1' indicates that a FIFO flag has been active and not cleared, logic '0' indicates that the flag condition is not active.

Clear: Since these bits latch in response to a FIFO flag occur–rence, to detect subsequent FIFO flags, you must clear the bits in this register. To clear any bit in this register that is set to '1' you must write a '1' to that bit.

6.15 Test FIFO Registers (continued)

6.15.2 Test FIFO Status Register (continued)

6.15.2.2 FIFO Status Bits — Bits D03 – D00

Each status bit is a non–latched version of the associated FIFO flag condition. When read as logic '1', the associated FIFO flag condition is active (true). When read as logic '0', the associated FIFO flag condition is inactive.

Note that when any status bit changes to logic '1', the corresponding flag bit in this register (Section 6.15.2.1) will also be set to logic '1'.

Use the Test FIFO Almost Full and Test FIFO Almost Empty Level Registers, Sections 6.15.5 and 6.15.4, to set the Almost Full and Almost Empty FIFO levels.

6.15 Test FIFO Registers (continued)

6.15.3 Test FIFO Interrupt Mask Register

There is one Test FIFO Interrupt Mask Register. The FIFO Interrupt Mask Register contains enable bits for each FIFO interrupt condition defined. Each bit enables or disables the generation of a local bus interrupt (LINTix) to the PCI7142.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6–75: Test FIFO Interrupt Mask Register
R/W @ BAR2+0x8D40

	D15 – D04	D03	D02	D01	D00
Bit Name	Reserved	FULL	ALMOST FULL	ALMOST EMPTY	EMPTY
Function	Write with zeros, Mask when reading	0 = Disable Interrupt 1 = Enable Interrupt			
All bits default to the logic '0' state at power on and reset					

Each bit of this register enables or disables the generation of a local bus interrupt to the PCI7142, when interrupts are enabled for this register in the Application Interrupt Enable LINTx Registers, Section 6.9.4.

Setting the bit associated with a given interrupt to the logic '1' state enables the generation of a local bus interrupt when that interrupt condition occurs. When a bit is cleared to logic '0' (its default state), the local bus interrupt will not be generated by the associated interrupt condition.

6.15 Test FIFO Registers (continued)

6.15.4 Test FIFO Almost Empty Level Register

There is one Test FIFO Almost Empty Level Register. This register sets the programmable boundary flag level of the Almost Empty flag for the FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6−76: Test FIFO Almost Empty Level Register
R/W @ BAR2+0x8D48

	D15 − D10	D09 − D00
Bit Name	Reserved	D9 − D0
Function	Write with zeros, Mask when reading	FIFO Almost Empty Level
All bits default to the logic '0' state at power on and reset		

The Almost Empty Level is a 10−bit binary value, with bit D09 the MSB. This value specifies the FIFO Almost Empty flag depth.

NOTE: All 7642 FIFOs are 64 bits wide. Thus, the FIFO level programmed into this register indicates the level of 64−bit words. For example, if you set this register to a level of 12, the FIFO sets a boundary flag when the data transfer reaches 24 32−bit words.

While the FIFO is being read, when the data is at or below the Almost Empty level, the FIFO sets the Almost Empty Flag.

- The Almost Empty Flag is routed to the Test FIFO Status Register, Section 6.15.2.
- If the Almost Empty Flag is enabled by the Test FIFO Interrupt Mask Register, Section 6.15.3, a local bus interrupt will be asserted to the PCI7142.

6.15 Test FIFO Registers (continued)

6.15.5 Test FIFO Almost Full Level Register

There is one Test FIFO Almost Full Level Register. This register sets the programmable boundary flag level of the Almost Full flag for the associated FIFO.

The following table shows the bit layout of this register. The paragraphs following the table provide descriptions of these bits.

Table 6−77: Test FIFO Almost Full Level Register R/W @ BAR2+0x8D50		
	D15 − D10	**D09 − D00**
Bit Name	Reserved	D9 − D0
Function	Write with zeros, Mask when reading	FIFO Almost Full Level
All bits default to the logic '0' state at power on and reset		

The Almost Full Level is a 10−bit binary value, with bit D09 the MSB. This value specifies the FIFO Almost Full flag depth.

<u>**NOTE:**</u> All 7642 FIFOs are 64 bits wide. Thus, the FIFO level programmed into this register indicates the level of 64−bit words. For example, if you set this register to a level of 12, the FIFO sets a boundary flag when the data transfer reaches 24 32−bit words.

While the FIFO is being written to, when the data is at or above the Almost Full level, the FIFO sets the Almost Full Flag.

- The FIFO Almost Full Flag is routed to the Test FIFO Status Register, Section 6.15.2.
- If the Almost Full Flag is enabled by the Test FIFO Interrupt Mask Register, Section 6.15.3, a local bus interrupt will be asserted to the PCI7142.

Chapter 7: Data Routing and Formatting

7.1 Overview

This chapter describes the ADC input and DAC output data routing and formatting.

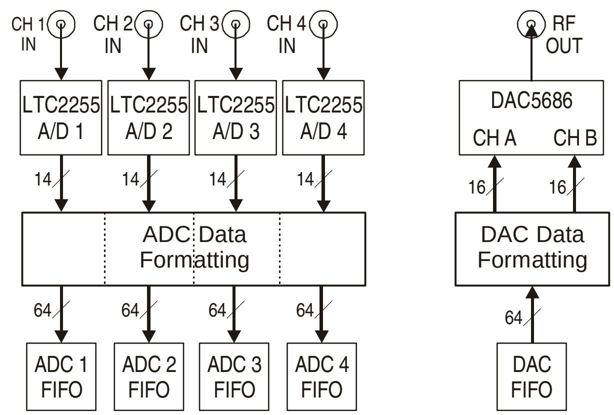


Figure 7−1: Data Routing Block Diagram

NOTE: All FIFOs on the Model 7642 are 64 bits wide. Thus, the FIFO input/output data formats presented in this chapter are shown as 64−bit formats.

Analog Data Inputs

The Model 7642 has four analog RF input channels from front panel MMCX connectors. Each input channel includes an LTC2255 A/D converter that provides data directly to the Signal FPGA. The outputs of the four A/D converters are delivered, in parallel, to the Signal FPGA through four separate 14−bit paths. The Signal FPGA routes the LTC2255 digital output data directly to four ADC FIFOs.

See Section 7.3 for description of the ADC data input routing formats.

Analog Data Outputs

The Model 7642 has one Texas Instruments 16−bit DAC5686 that is capable of operating in D/A only or quadrature modulation modes. The D/A has built−in interpolation filters settable to 2x, 4x, 8x, and 16x. The Signal FPGA delivers 8−bit or 16−bit FIFO data to the DAC5686 where the analog output is sent to a 1−to−1 transformer and then passed to the front panel MMCX connector.

See Section 7.5 for description of the DAC data output routing formats.

7.2 FIFO Word Swap Formats

The data formats for the ADC and DAC FIFOS allow you to use a WORD SWAP fea– ture with the FIFO data. When the WORD SWAP control bit is set to logic '1' in the associated FIFO Control Register (ADC FIFO Control Register, Section 6.10.2 or DAC FIFO Control Register, Section 6.11.3), word swapping is enabled and each pair of 16– bit samples are swapped in the 64–bit FIFO word, as follows:

Bits D00:15 are swapped with bits D16:31,
Bits D32:47 are swapped with bits D48:63.

The data is swapped as illustrated below.

Without WORD SWAP				
Bit	**D63 – D48**	**D47 – D32**	**D31 – D16**	**D15 – D0**
Value	$d[15:0]_d$	$d[15:0]_c$	$d[15:0]_b$	$d[15:0]_a$

With WORD SWAP				
Bit	**D63 – D48**	**D47 – D32**	**D31 – D16**	**D15 – D0**
Value	$d[15:0]_c$	$d[15:0]_d$	$d[15:0]_a$	$d[15:0]_b$

This data WORD SWAP can be applied to any of the FIFO formats in the following sec– tions: Section 7.3 ADC data routing formats and Section 7.5 DAC output data routing formats.

7.3 ADC Data Routing and Formats

Each input LTC2255 A/D converter channel provides 14–bit digital data directly to the ADC FIFOs (A/D 1 data to ADC 1 FIFO, A/D 2 to ADC 2 FIFO, etc.). The ADC output can be written to a FIFO at a programmed decimation rate, and may be unpacked or packed as set by the associated ADC FIFO Control Register, Section 6.10.2.4.

The following subsections describe these data formats—these formats apply to both the PCI Interface and DDR Memory ADC FIFOs. WORD SWAP, Section 7.2, can be applied to any of these ADC FIFO formats.

7.3.1 ADC FIFO, Unpacked

Each ADC FIFO receives the raw 14–bit data directly from its associated LTC2255 A/D (A/D 1 data to ADC 1 FIFO, A/D 2 data to ADC 2 FIFO, etc.). Writing to the FIFO is controlled by the selected gate. The data is written to the FIFO at a programmed decimation rate N, where N is 1 to 4096, only when the gate is enabled.

When PACK MODE (bit D08 in the ADC FIFO Control Register) is cleared to 0, data is unpacked with two consecutive A/D samples duplicated in each 64–bit FIFO word. Each A/D sample is left justified in the 14 most signifi–cant bits out of 16 in each 16–bit segment of the 64–bit FIFO data word.

- The first sample, received at time ‘t’, is placed into bits D15 to D2. The same sample is duplicated in bits D31 to D18.
- The next sample, received at time ‘t+1’, is placed into bits D47 to D34. The same sample is duplicated in bits D63 to D50.

The ADC FIFO data is in the following format:

Table 7–1: Data Format – ADC FIFO, Unpacked

Bit	D63 – D50	D49 D48	D47 – D34	D33 D32	D31 – D18	D17 D16	D15 – D2	D1 D0
Value	14–bit A/D sample[13:0] at time (t+1)	00	14–bit A/D sample[13:0] at time (t+1)	00	14–bit A/D sample[13:0] at time (t)	00	14–bit A/D sample[13:0] at time (t)	00

With WORD SWAP enabled the data format is the same.

7.3 ADC Data Routing and Formats (continued)

7.3.2 ADC FIFO, Time Packed

Each ADC FIFO receives the raw 14–bit data directly from its associated LTC2255 A/D (A/D 1 data to ADC 1 FIFO, A/D 2 to ADC 2 FIFO, etc.). The data is written to the FIFO at a programmed decimation rate N (1 to 4096), only when the selected gate is enabled.

When PACK MODE (bit D08 in the ADC FIFO Control Register) is set to 1, data is packed with four A/D samples in each 64–bit FIFO word.

Without WORD SWAP (disabled), data is packed with four consecutive A/D samples in each 64–bit FIFO word. Each A/D sample is left justified in the 14 most significant bits out of 16 in each of the 16–bit segments of the 64–bit FIFO data word.

- The first sample, received at time 't', is placed into bits D15 to D2.
- The next sample, received at time 't+1', is placed into bits D31 to D18.
- The next sample, received at time 't+2', is placed into bits D47 to D34.
- The next sample, received at time 't+3', is placed into bits D63 to D50.

The ADC FIFO data is packed in the following format:

Table 7–2: Data Format – ADC FIFO, Time Packed

Bit	D63 – D50	D49 D48	D47 – D34	D33 D32	D31 – D18	D17 D16	D15 – D2	D1 D0
Value	14–bit A/D sample[13:0] at time (t+3)	00	14–bit A/D sample[13:0] at time (t+2)	00	14–bit A/D sample[13:0] at time (t+1)	00	14–bit A/D sample[13:0] at time (t)	00

With WORD SWAP enabled, each consecutive pair of samples (e.g., t and t+1) is swapped as follows:

Table 7–3: Data Format – ADC FIFO, Time Packed, Word Swap

Bit	D63 – D50	D49 D48	D47 – D34	D33 D32	D31 – D18	D17 D16	D15 – D2	D1 D0
Value	14–bit A/D sample[13:0] at time (t+2)	00	14–bit A/D sample[13:0] at time (t+3)	00	14–bit A/D sample[13:0] at time (t)	00	14–bit A/D sample[13:0] at time (t+1)	00

7.4 ADC FIFO to DDR Memory Routing

Each input A/D converter channel also provides 14–bit raw digital data directly to the DDR Memory ADC FIFOs in parallel with the PCI Interface ADC FIFOs. The DDR Memory ADC FIFO data formats are the same as shown in Section 7.3, ADC Data Routing and Formats. The DDR Memory ADC FIFOs can transfer these samples to DDR Memory Banks 0 or 1 using the DDR Memory Control Register, Section 6.13.1.

NOTE: Packing must be enabled (PACK MODE set to 'Pack') by the associated ADC FIFO Control Register, Section 6.10.2.4, for any data transfers to the DDR Memories.

The BANK x PACK bits of the DDR Memory Control Register, Section 6.13.1.1, can set up transfers from a single ADC FIFO to a DDR Memory Bank (ADC1 FIFO to DDR BANK 0, or ADC3 FIFO to DDR BANK 1), or as packed data from two ADC FIFOs to each DDR Memory Bank (ADC1 & 2 FIFOs to BANK 0, or ADC3 & 4 FIFOs to BANK 1). The following subsections describe the DDR Memory data formats for each of these data transfers.

WORD SWAP, Section 7.2, can be applied to any of the ADC FIFO transfers. WORD SWAP does not alter the data transferred from the DDR Memory ADC FIFOs to DDR Memory. WORD SWAP affects only the data path from the LTC2255 A/D converter to the associated ADC FIFO. Thus, if WORD SWAP is enabled, the data is swapped when written from the A/D into the ADC FIFO, as illustrated in Section 7.3.2, before any transfer to DDR Memory. This WORD SWAP does affect the resulting DDR memory data format as shown in the following subsections.

NOTE: Although the DDR Memory data transfers are in 32–bit words, since the only user access to Bank 0 or Bank 1 is using the PCI Interface Memory Read FIFO the following DDR Memory formats are presented as 64–bit formats in the following sections.

7.4 ADC FIFO to DDR Memory Routing (continued)

7.4.1 Channel Unpacked

When only one ADC FIFO channel is routed to its associated DDR Memory Bank (the BANK x PACK bits of the DDR Memory Control Register are cleared, Section 6.13.1.1), successive 16–bit words from the DDR Memory ADC FIFO are written to each 32–bit word of the respective DDR Memory Bank (ADC1 FIFO to BANK 0, or ADC3 FIFO to BANK 1).

Without WORD SWAP the resulting DDR Memory data is in the following format (for each 64–bit word accessed through the Memory Read FIFO):

Table 7–4: Data Format – DDR Memory, Channel Unpacked

BANK 0								
Bit	**D63 – D50**	**D49 D48**	**D47 – D34**	**D33 D32**	**D31 – D18**	**D17 D16**	**D15 – D2**	**D1 D0**
Value	14–bit A/D 1 sample[13:0] at time (t+3)	00	14–bit A/D 1 sample[13:0] at time (t+2)	00	14–bit A/D 1 sample[13:0] at time (t+1)	00	14–bit A/D 1 sample[13:0] at time (t)	00
BANK 1								
Bit	**D63 – D50**	**D49 D48**	**D47 – D34**	**D33 D32**	**D31 – D18**	**D17 D16**	**D15 – D2**	**D1 D0**
Value	14–bit A/D 3 sample[13:0] at time (t+3)	00	14–bit A/D 3 sample[13:0] at time (t+2)	00	14–bit A/D 3 sample[13:0] at time (t+1)	00	14–bit A/D 3 sample[13:0] at time (t)	00

With WORD SWAP consecutive samples (e.g., t and t+1) are swapped:

Table 7–5: Data Format – DDR Memory, Channel Unpacked, Word Swap

BANK 0								
Bit	**D63 – D50**	**D49 D48**	**D47 – D34**	**D33 D32**	**D31 – D18**	**D17 D16**	**D15 – D2**	**D1 D0**
Value	14–bit A/D 1 sample[13:0] at time (t+2)	00	14–bit A/D 1 sample[13:0] at time (t+3)	00	14–bit A/D 1 sample[13:0] at time (t)	00	14–bit A/D 1 sample[13:0] at time (t+1)	00
BANK 1								
Bit	**D63 – D50**	**D49 D48**	**D47 – D34**	**D33 D32**	**D31 – D18**	**D17 D16**	**D15 – D2**	**D1 D0**
Value	14–bit A/D 3 sample[13:0] at time (t+2)	00	14–bit A/D 3 sample[13:0] at time (t+3)	00	14–bit A/D 3 sample[13:0] at time (t)	00	14–bit A/D 3 sample[13:0] at time (t+1)	00

7.4 ADC FIFO to DDR Memory Routing (continued)

7.4.2 Channel Packed

When the ADC2 and/or ADC4 FIFO channels are enabled by the BANK x PACK bits of the DDR Memory Control Register, Section 6.13.1.1, channel packing interleaves successive 16–bit words from each of the two associated DDR Memory ADC FIFOs to each 32–bit word of the respective DDR Memory Bank (ADC1 and 2 FIFOs to BANK 0, or ADC3 and 4 FIFOs to BANK 1).

Without WORD SWAP the resulting DDR Memory data is in the following format (for each 64–bit word accessed through the Memory Read FIFO):

Table 7–6: Data Format – DDR Memory, Channel Packed

BANK 0								
Bit	**D63 – D50**	**D49 D48**	**D47 – D34**	**D33 D32**	**D31 – D18**	**D17 D16**	**D15 – D2**	**D1 D0**
Value	14–bit A/D 2 sample[13:0] at time (t+1)	00	14–bit A/D 1 sample[13:0] at time (t+1)	00	14–bit A/D 2 sample[13:0] at time (t)	00	14–bit A/D 1 sample[13:0] at time (t)	00
BANK 1								
Bit	**D63 – D50**	**D49 D48**	**D47 – D34**	**D33 D32**	**D31 – D18**	**D17 D16**	**D15 – D2**	**D1 D0**
Value	14–bit A/D 4 sample[13:0] at time (t+1)	00	14–bit A/D 3 sample[13:0] at time (t+1)	00	14–bit A/D 4 sample[13:0] at time (t)	00	14–bit A/D 3 sample[13:0] at time (t)	00

With WORD SWAP consecutive samples (e.g., t and t+1) are swapped:

Table 7–7: Data Format – DDR Memory, Channel Packed, Word Swap

BANK 0								
Bit	**D63 – D50**	**D49 D48**	**D47 – D34**	**D33 D32**	**D31 – D18**	**D17 D16**	**D15 – D2**	**D1 D0**
Value	14–bit A/D 2 sample[13:0] at time (t)	00	14–bit A/D 1 sample[13:0] at time (t)	00	14–bit A/D 2 sample[13:0] at time (t+1)	00	14–bit A/D 1 sample[13:0] at time (t+1)	00
BANK 1								
Bit	**D63 – D50**	**D49 D48**	**D47 – D34**	**D33 D32**	**D31 – D18**	**D17 D16**	**D15 – D2**	**D1 D0**
Value	14–bit A/D 4 sample[13:0] at time (t)	00	14–bit A/D 3 sample[13:0] at time (t)	00	14–bit A/D 4 sample[13:0] at time (t+1)	00	14–bit A/D 3 sample[13:0] at time (t)	00

7.5 DAC Data Routing and Formats

The Signal FPGA receives digital data from the DAC FIFOs, and then presents the digital data to the DAC5686's dual 16–bit interfaces (DAC input channels A and B). There are three packing modes of DAC FIFO data: Unpacked, 16–bit Packed, and 8–bit Packed. To choose the packing mode, set the Pack Mode bits in the DAC FIFO Control Register, Section 6.11.3. WORD SWAP, Section 7.2, can be applied to any of these modes.

The following subsections describe these data formats.

7.5.1 Unpacked

In Unpacked mode, the FPGA presents the DAC5686 with 16–bit digital samples from the DAC FIFO for both DAC input channels (see page 213 for illustration of the routing of the data samples to each DAC5686 channel).

Each 64–bit FIFO data word contains two consecutive 16–bit data samples for each DAC channel.

Without WORD SWAP (disabled), data is unpacked with data for DAC channel B in the upper 16 bits of each 32–bit half, and data for DAC channel A in the lower 16 bits of each 32–bit half of the 64–bit word.

- The first sample, for time ‘t’, is placed into the lowest 32 bits of the 64–bit word, with the channel B sample in bits D31 to D16 and the channel A sample in bits D15 to D0.
- The next sample, for time ‘t+1’, is placed into the highest 32 bits of the 64–bit word, with the channel B sample in bits D63 to D48 and the channel A sample in bits D47 to D32.

The DAC FIFO data is in the following format:

Table 7–8: Data Format – DAC FIFO, Unpacked

Bit	D63 – D48	D47 – D32	D31 – D16	D15 – D0
Value	data[15:0] for DAC channel B at time (t+1)	data[15:0] for DAC channel A at time (t+1)	data[15:0] for DAC channel B at time (t)	data[15:0] for DAC channel A at time (t)

With WORD SWAP enabled, channel A and B data are swapped as follows:

Table 7–9: Data Format – DAC FIFO, Unpacked, Word Swap

Bit	D63 – D48	D47 – D32	D31 – D16	D15 – D0
Value	data[15:0] for DAC channel A at time (t+1)	data[15:0] for DAC channel B at time (t+1)	data[15:0] for DAC channel A at time (t)	data[15:0] for DAC channel B at time (t)

7.5 DAC Data Routing and Formats (continued)

7.5.1 Unpacked (continued)

The data routing ***without*** WORD SWAP is illustrated below.

DAC5686 Input Channels		
DAC Bit	**DA[15 :0]**	**DB[15:0]**
Word 1	$d[15:0]A_{(t)}$	$d[15:0]B_{(t)}$
Word 2	$d[15:0]A_{(t+1)}$	$d[15:0]B_{(t+1)}$
Word 3	$d[15:0]A_{(t+2)}$	$d[15:0]B_{(t+2)}$
Word 4	$d[15:0]A_{(t+3)}$	$d[15:0]B_{(t+3)}$
Word n	...	...

Bit	**D63:48**	**D47:32**	**D31:D16**	**D15:D0**
Word 1	$d[15:0]B_{(t+1)}$	$d[15:0]A_{(t+1)}$	$d[15:0]B_{(t)}$	$d[15:0]A_{(t)}$
Word 2	$d[15:0]B_{(t+3)}$	$d[15:0]A_{(t+3)}$	$d[15:0]B_{(t+2)}$	$d[15:0]A_{(t+2)}$
Word n	...	...	...	...
	DAC FIFO			

The data routing ***with*** WORD SWAP is illustrated below.

DAC5686 Input Channels		
DAC Bit	**DA[15 :0]**	**DB[15:0]**
Word 1	$d[15:0]B_{(t)}$	$d[15:0]A_{(t)}$
Word 2	$d[15:0]B_{(t+1)}$	$d[15:0]A_{(t+1)}$
Word 3	$d[15:0]B_{(t+2)}$	$d[15:0]A_{(t+2)}$
Word 4	$d[15:0]B_{(t+3)}$	$d[15:0]A_{(t+3)}$
Word n	...	...

Bit	**D63:48**	**D47:32**	**D31:D16**	**D15:D0**
Word 1	$d[15:0]A_{(t+1)}$	$d[15:0]B_{(t+1)}$	$d[15:0]A_{(t)}$	$d[15:0]B_{(t)}$
Word 2	$d[15:0]A_{(t+3)}$	$d[15:0]B_{(t+3)}$	$d[15:0]A_{(t+2)}$	$d[15:0]B_{(t+2)}$
Word n	...	...	...	...
	DAC FIFO			

7.5 DAC Data Routing and Formats (continued)

7.5.2 Packed, 16–bit

In 16–bit Packed mode, the FPGA presents the DAC5686 with 16–bit digital samples from the DAC FIFO for DAC input channel B only (see page 215 for illustration of the routing of the data samples to the DAC5686).

Without WORD SWAP (disabled), each 64–bit FIFO data word contains four consecutive 16–bit samples for DAC channel B. The samples are time–packed with the least significant 16 bits of the FIFO word containing the first sample.

- The first sample, for time ‘t’, is placed into bits D15 to D0.
- The next sample, for time ‘t+1’, is placed into bits D31 to D16.
- The next sample, for time ‘t+2’, is placed into bits D47 to D32.
- The next sample, for time ‘t+3’, is placed into bits D63 to D48.

The data is packed into the DAC FIFO in the following format:

Table 7–10: Data Format – DAC FIFO, 16–bit Packed

Bit	D63 – D48	D47 – D32	D31 – D16	D15 – D0
Value	data[15:0] for DAC channel B at time (t+3)	data[15:0] for DAC channel B at time (t+2)	data[15:0] for DAC channel B at time (t+1)	data[15:0] for DAC channel B at time (t)

With WORD SWAP enabled, each consecutive pair of data (e.g., t and t+1) is swapped as follows:

Table 7–11: Data Format – DAC FIFO, 16–bit Packed, Word Swap

Bit	D63 – D48	D47 – D32	D31 – D16	D15 – D0
Value	data[15:0] for DAC channel B at time (t+2)	data[15:0] for DAC channel B at time (t+3)	data[15:0] for DAC channel B at time (t)	data[15:0] for DAC channel B at time (t+1)

7.5 DAC Data Routing and Formats (continued)

7.5.2 Packed, 16–bit (continued)

The data routing ***without*** WORD SWAP is illustrated below.

DAC5686 Input Channels		
DAC Bit	**DA**	**DB[15:0]**
Word 1	–	$d[15:0]_{(t)}$
Word 2	–	$d[15:0]_{(t+1)}$
Word 3	–	$d[15:0]_{(t+2)}$
Word 4	–	$d[15:0]_{(t+3)}$
Word n	–	...

Bit	D63:48	D47:32	D31:D16	D15:D0
Word 1	$d[15:0]_{(t+3)}$	$d[15:0]_{(t+2)}$	$d[15:0]_{(t+1)}$	$d[15:0]_{(t)}$
Word 2	$d[15:0]_{(t+7)}$	$d[15:0]_{(t+6)}$	$d[15:0]_{(t+5)}$	$d[15:0]_{(t+4)}$
Word n	...	...	...	...
	DAC FIFO			

The data routing ***with*** WORD SWAP is illustrated below.

DAC5686 Input Channels		
DAC Bit	**DA**	**DB[15:0]**
Word 1	–	$d[15:0]_{(t+1)}$
Word 2	–	$d[15:0]_{(t)}$
Word 3	–	$d[15:0]_{(t+3)}$
Word 4	–	$d[15:0]_{(t+2)}$
Word n	–	...

Bit	D63:48	D47:32	D31:D16	D15:D0
Word 1	$d[15:0]_{(t+2)}$	$d[15:0]_{(t+3)}$	$d[15:0]_{(t)}$	$d[15:0]_{(t+1)}$
Word 2	$d[15:0]_{(t+6)}$	$d[15:0]_{(t+7)}$	$d[15:0]_{(t+4)}$	$d[15:0]_{(t+5)}$
Word n	...	...	...	...
	DAC FIFO			

7.5 DAC Data Routing and Formats (continued)

7.5.3 Packed, 8–bit

In 8–bit Packed mode, the FPGA presents the DAC5686 with 8–bit digital samples from the DAC FIFO for DAC input channel B only (see page 217 for illustration of the routing of the data samples to the DAC5686).

Without WORD SWAP (disabled), each 64–bit FIFO data word contains eight consecutive 8–bit samples for DAC channel B. The samples are time–packed with the least significant eight bits of the FIFO word containing the first sample.

- The first sample, for time ‘t’, is placed into bits D9 to D0.
- The second sample, time ‘t+1’, is placed into bits D15 to D8.
- The third sample, time ‘t+2’, is placed into bits D23 to D16.

.
.
.

- The eighth sample, for time ‘t+7’, is placed into bits D63 to D56.

The data is packed into the DAC FIFO in the following format:

Table 7–12: Data Format – DAC FIFO, 8–bit Packed

Bit	D63:56	D55:48	D47:D40	D39:D32	D31:24	D2316	D15:D8	D7:D0
Value	data[7:0] for DAC channel B at time (t+7)	data[7:0] for DAC channel B at time (t+6)	data[7:0] for DAC channel B at time (t+5)	data[7:0] for DAC channel B at time (t+4)	data[7:0] for DAC channel B at time (t+3)	data[7:0] for DAC channel B at time (t+2)	data[7:0] for DAC channel B at time (t+1)	data[7:0] for DAC channel B at time (t)

With WORD SWAP enabled, consecutive sets of data (e.g., t plus t+1 and t+2 plus t+3) are swapped as follows:

Table 7–13: Data Format – DAC FIFO, 8–bit Packed, Word Swap

Bit	D63:56	D55:48	D47:D40	D39:D32	D31:24	D2316	D15:D8	D7:D0
Value	data[7:0] for DAC channel B at time (t+5)	data[7:0] for DAC channel B at time (t+4)	data[7:0] for DAC channel B at time (t+7)	data[7:0] for DAC channel B at time (t+6)	data[7:0] for DAC channel B at time (t+1)	data[7:0] for DAC channel B at time (t)	data[7:0] for DAC channel B at time (t+3)	data[7:0] for DAC channel B at time (t+2)

7.5 DAC Data Routing and Formats (continued)

7.5.3 Packed, 8–bit (continued)

The data routing ***without*** WORD SWAP is illustrated below.

DAC5686 Input Channels			
DAC Bit	**DA**	**DB[15:8]**	**DB[7:0]**
Word 1	–	$d[7:0]_{(t)}$	0
Word 2	–	$d[7:0]_{(t+1)}$	0
Word 3	–	$d[7:0]_{(t+2)}$	0
Word 4	–	$d[7:0]_{(t+3)}$	0
Word n	–	...	0

Bit	**D63:56**	**D55:48**	**D47:D40**	**D39:D32**	**D31:24**	**D2316**	**D15:D8**	**D7:D0**
Word 1	$d[7:0]_{(t+7)}$	$d[7:0]_{(t+6)}$	$d[7:0]_{(t+5)}$	$d[7:0]_{(t+4)}$	$d[7:0]_{(t+3)}$	$d[7:0]_{(t+2)}$	$d[7:0]_{(t+1)}$	$d[7:0]_{(t)}$
Word 2	$d[7:0]_{(t+15)}$	$d[7:0]_{(t+14)}$	$d[7:0]_{(t+13)}$	$d[7:0]_{(t+12)}$	$d[7:0]_{(t+11)}$	$d[7:0]_{(t+10)}$	$d[7:0]_{(t+9)}$	$d[7:0]_{(t+8)}$
Word n	...	...	...	...	...	...	...	...
	DAC FIFO							

The data routing ***with*** WORD SWAP is illustrated below.

DAC5686 Input Channels			
DAC Bit	**DA**	**DB[15:8]**	**DB[7:0]**
Word 1	–	$d[7:0]_{(t+2)}$	0
Word 2	–	$d[7:0]_{(t+3)}$	0
Word 3	–	$d[7:0]_{(t)}$	0
Word 4	–	$d[7:0]_{(t+1)}$	0
Word n	–	...	0

Bit	**D63:56**	**D55:48**	**D47:D40**	**D39:D32**	**D31:24**	**D2316**	**D15:D8**	**D7:D0**
Word 1	$d[7:0]_{(t+5)}$	$d[7:0]_{(t+4)}$	$d[7:0]_{(t+7)}$	$d[7:0]_{(t+6)}$	$d[7:0]_{(t+1)}$	$d[7:0]_{(t)}$	$d[7:0]_{(t+3)}$	$d[7:0]_{(t+2)}$
Word 2	$d[7:0]_{(t+13)}$	$d[7:0]_{(t+12)}$	$d[7:0]_{(t+15)}$	$d[7:0]_{(t+14)}$	$d[7:0]_{(t+9)}$	$d[7:0]_{(t+8)}$	$d[7:0]_{(t+11)}$	$d[7:0]_{(t+10)}$
Word n	...	...	...	...	...	...	...	...
	DAC FIFO							

This page is intentionally blank

Appendix A: PCI Configuration Space Registers

A.1 Introduction

This Appendix shows the pre–defined header region of PCI Configuration Space.

A.2 PCI Configuration Space Registers

The PCI Configuration Space consists of fields that uniquely identify the device and allow the device to be generically controlled. The first 16 Bytes (four words) are defined the same for all types of PCI devices. The remaining words can have different layouts depending on the base function that the device supports. The PCI interface on the Model 7642 uses Header Type 00h, which has the layout shown in the table below. This table lists the PCI Configuration Space registers, their functions, and their base address in Configuration Space.

Table A–1: PCI Configuration Space Header (Type 00h)

<table>
<tr><th>Address</th><th colspan="4">Register Function</th></tr>
<tr><td>0x00</td><td colspan="2">Device ID</td><td colspan="2">Vendor ID</td></tr>
<tr><td>0x04</td><td colspan="2">Status</td><td colspan="2">Command</td></tr>
<tr><td>0x08</td><td colspan="3">Class Code</td><td>Revision ID</td></tr>
<tr><td>0x0C</td><td>BIST</td><td>Header Type</td><td>Latency Timer</td><td>Cache Line Size</td></tr>
<tr><td>0x10</td><td colspan="4">Base Address Register 0 (BAR0)</td></tr>
<tr><td>0x14</td><td colspan="4">Base Address Register 1 (BAR1)</td></tr>
<tr><td>0x18</td><td colspan="4">Base Address Register 2 (BAR2)</td></tr>
<tr><td>0x1C</td><td colspan="4">Base Address Register 3 (BAR3)</td></tr>
<tr><td>0x20</td><td colspan="4">Base Address Register 4 (BAR4)</td></tr>
<tr><td>0x24</td><td colspan="4">Base Address Register 5 (BAR5)</td></tr>
<tr><td>0x28</td><td colspan="4">Cardbus CIS Pointer</td></tr>
<tr><td>0x2C</td><td colspan="2">Subsystem ID</td><td colspan="2">Subsystem Vendor ID</td></tr>
<tr><td>0x30</td><td colspan="4">Expansion ROM Base Address</td></tr>
<tr><td>0x34</td><td colspan="4">Reserved</td></tr>
<tr><td>0x38</td><td colspan="4">Reserved</td></tr>
<tr><td>0x3C</td><td>Max_Lat</td><td>Min_Gnt</td><td>Interrupt Pin</td><td>Interrupt Line</td></tr>
</table>

The Base Address Registers defined for the Model 7642 are shown on the next page.

A.2 PCI Configuration Space Registers (continued)

In the Model 7642, the first four Base Address Registers of PCI Configuration Space are used to configure the memory maps of board resources and registers. The following table shows the use of these registers in the 7642.

Table A–2: PCI Base Address Registers

Address	Register	Model 7642 Use
`0x10`	Base Address Register 0 (BAR0)	Memory accesses to PCI7140 Local, Runtime, and DMA registers
`0x14`	Base Address Register 1 (BAR1)	Memory accesses to 7642 FIFOs
`0x18`	Base Address Register 2 (BAR2)	Memory accesses to 7642 FPGA Global, Clock/Sync/ Gate, Interrupt,D/A, A/D, Registers
`0x1C`	Base Address Register 3 (BAR3)	not used on 7642