

Road Runner and R3 Camera Link Models

Hardware Reference Manual

BitFlow, Inc.
300 Wildwood Ave
Woburn, MA 01801
USA
Tel: 781-932-2900
Fax: 781-933-9965
Email: support@bitflow.com
Web: www.bitflow.com
Revision E.0

© 2004 BitFlow, Inc. All Rights Reserved.

This document, in whole or in part, may not be copied, photocopied, reproduced, translated or reduced to any other electronic medium or machine readable form without the prior written consent of BitFlow, Inc.

BitFlow, Inc. makes no implicit warranty for the use of its products and assumes no responsibility for any errors that may appear in this document, nor does it make a commitment to update the information contained herein.

BitFlow, Inc. retains the right to make changes to these specifications at any time without notice.

All trademarks are properties of their respective holders.

Revision History:

Revision	Date	Comments
C.0	2001-06-01	First Printing (revision set to match other manuals)
C.1	2002-04-01	Minor changes
D.0	2002-12-01	Ported to structured format, minor changes
D.1	2004-08-23	Second printing, minor updates and reformatting

Table Of Contents

P - Preface

- Purpose R2/R3CL-P-1
 - Support Services R2/R3CL-P-1
 - Technical Support R2/R3CL-P-1
 - Sales Support R2/R3CL-P-1

1 - Introduction

- Overview R2/R3CL-1-1
- Description R2/R3CL-1-3

2 - Technical Description

- Road Runner CL Models R2/R3CL-2-1
 - Description R2/R3CL-2-1
 - Identification R2/R3CL-2-1
- R3 CL Models R2/R3CL-2-2
 - Description R2/R3CL-2-2
 - Identification R2/R3CL-2-2
- Camera Taps R2/R3CL-2-3
- Video FIFOs and MUXs R2/R3CL-2-4
- LUTs R2/R3CL-2-5
- CTABs and Camera Controls R2/R3CL-2-6
 - Horizontal Control Table R2/R3CL-2-6
 - The HCTAB Functions R2/R3CL-2-7
 - Vertical Control Table R2/R3CL-2-12
 - The VCTAB Functions R2/R3CL-2-13
- Camera Control Signals R2/R3CL-2-18
 - Definition R2/R3CL-2-18
- Host Access to VCTAB and HCTAB R2/R3CL-2-19
- The DMA Engine R2/R3CL-2-20
 - PCI (Destination) Address R2/R3CL-2-21
 - Local Bus (Source) Address R2/R3CL-2-21
 - Transfer Size R2/R3CL-2-22
 - Descriptor Pointer R2/R3CL-2-22
- Interrupts R2/R3CL-2-23
- The Acquisition Process R2/R3CL-2-25
 - Freeze R2/R3CL-2-25
 - Abort R2/R3CL-2-26
 - Snap R2/R3CL-2-26
 - Grab R2/R3CL-2-26
- Advanced Acquisition Modes R2/R3CL-2-28
 - Start-Stop Triggered Acquisition R2/R3CL-2-28

Continuous Acquisition R2/R3CL-2-28
Triggered Termination R2/R3CL-2-28

3 - Interfacing

Introduction R2/R3CL-3-1

4 - Power Requirements

Maximum Current Requirements from +5V Power Supply R2/R3CL-4-1

5 - Specifications

Introduction R2/R3CL-5-1
PCI bus compatibility R2/R3CL-5-2
Maximum Lines Per Frame R2/R3CL-5-3

6 - PCI Interface

Introduction R2/R3CL-6-1
Board Addressing R2/R3CL-6-2
 Example 1: R2/R3CL-6-2
 Example 2: R2/R3CL-6-3
Non-Register Memory R2/R3CL-6-4
 CTABS R2/R3CL-6-4
 LUTADDR R2/R3CL-6-4
 LUTDATA R2/R3CL-6-4
 QTABS R2/R3CL-6-4
 VFIFO R2/R3CL-6-4

7 - Register Specifications

Introduction R2/R3CL-7-1
 Bitfield definitions R2/R3CL-7-1
CON0 Register R2/R3CL-7-3
CON1 Register R2/R3CL-7-7
CON2 Register R2/R3CL-7-10
CON3 Register R2/R3CL-7-16
CON4 Register R2/R3CL-7-18
CON5 Register R2/R3CL-7-24
CON6 Register R2/R3CL-7-29
CON7 Register R2/R3CL-7-33
CON8 Register R2/R3CL-7-34
CON9 Register R2/R3CL-7-38
CON10 Register R2/R3CL-7-40
CON11 Register R2/R3CL-7-44

CON12 Register R2/R3CL-7-47

8 - Connector Pin-Outs

Description R2/R3CL-8-1

P1, Test Connector, 30-Pin R2/R3CL-8-2

P2, I/O Connector, 26-Pin R2/R3CL-8-3

P3, Camera Link Connector, 26-Pin R2/R3CL-8-4

Con5, The First PMC I/O Connector, 5-Pin R2/R3CL-8-5

Con6, The Second PMC I/O Connector, 6-Pin R2/R3CL-8-6

Table Of Contents

Preface

Chapter P

P.1 Purpose

This Hardware Reference Manual is intended for anyone installing or using the Road Runner Camera Link or R3 Camera Link frame grabbers. This manual only covers the camera link versions of these boards. There is a separate manual for the differential versions. The purpose of this manual is two-fold. First, this manual completely describes how the board works. Second, it is a reference manual describing in detail what all of the board's registers do.

P.1.1 Support Services

BitFlow, Inc. provides both sales and technical support for the Road Runner product.

P.1.2 Technical Support

Our web site is www.bitflow.com.

Technical support is available at 781-932-2900 from 9:00 AM to 6:00 PM Eastern Standard Time, Monday through Friday.

For technical support by email (support@bitflow.com) or by FAX (781-933-9965), please include the following:

- Product name
- Camera type and mode being used
- Software revision number
- Computer CPU type, PCI chipset, bus speed
- Operating system
- Example code (if applicable)

P.1.3 Sales Support

Contact your local BitFlow Sales Representative, Dealer, or Distributor for information about how BitFlow can help you solve your most demanding camera interfacing problems. Refer to the BitFlow, Inc. website (www.bitflow.com) for a list of North American and International sales representatives.

Introduction

Chapter 1

1.1 Overview

The Road Runner and the R3 are high-performance, PCI-based frame grabbers. They are targeted at Camera Link cameras. The DMA engine transfers data directly into system memory. The digital port can accept up to 24 bits of data. This data can correspond to any arrangement of camera output taps (e.g. two 8-bit taps, or two 12-bit taps). On board circuitry will reformat the data on-the-fly. Figure 1-1 shows a general block diagram of the Road Runner/R3 Camera Link. The main blocks include:

- The Camera Link block
- The Universal Asynchronous Receiver Transmitter (UART)
- Input multiplexers
- Video FIFOs
- LUTs
- CTABs and camera control block
- PCI9080 PCI bus interface and DMA engine

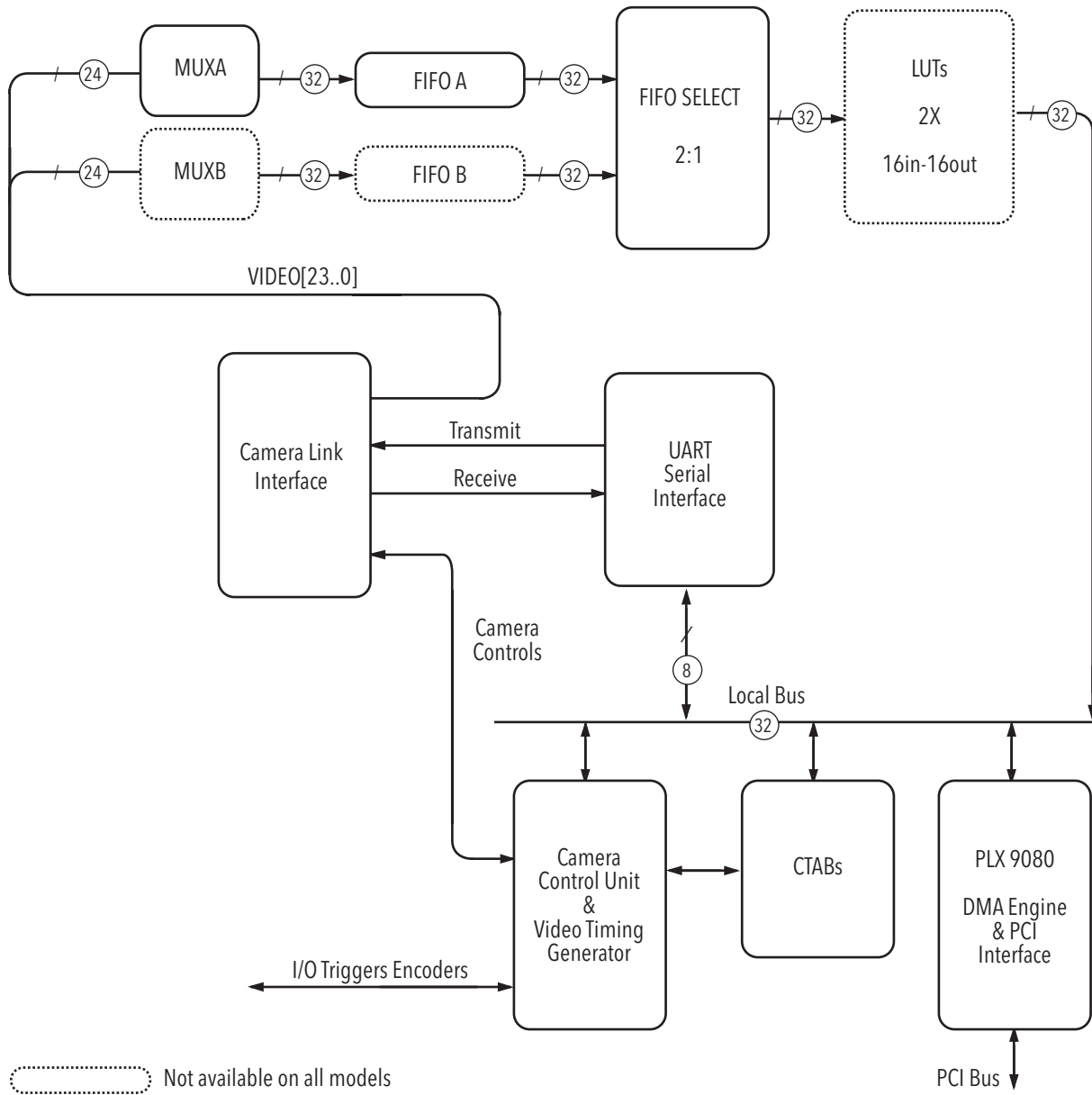


Figure 1-1 Road Runner/R3 Camera Link Block Diagram

1.2 Description

The Road Runner/R3 CL can receive up to three 8-bit taps through its Camera Link interface. The data will be fed into the MUXs, which will reformat the data before sending it to the FIFOs. At the output of the FIFOs there is a LUT, 16-in/16-out. The CTABs (Control Tables) will control the attached cameras. This block will also control I/O lines, triggers and encoders that are needed to communicate with external industrial systems.

The PCI bus interface is implemented with a PLX PCI9080 device. This chip has all the circuitry for performing DMA. The Descriptors Table, located in the host memory, is part of a DMA address generator.

On board there is a UART that will control the serial interface for camera control as defined by the Camera Link protocol.

Figure 1-2 shows the location of the connectors on the Road Runner/R3 CL, PCI version.

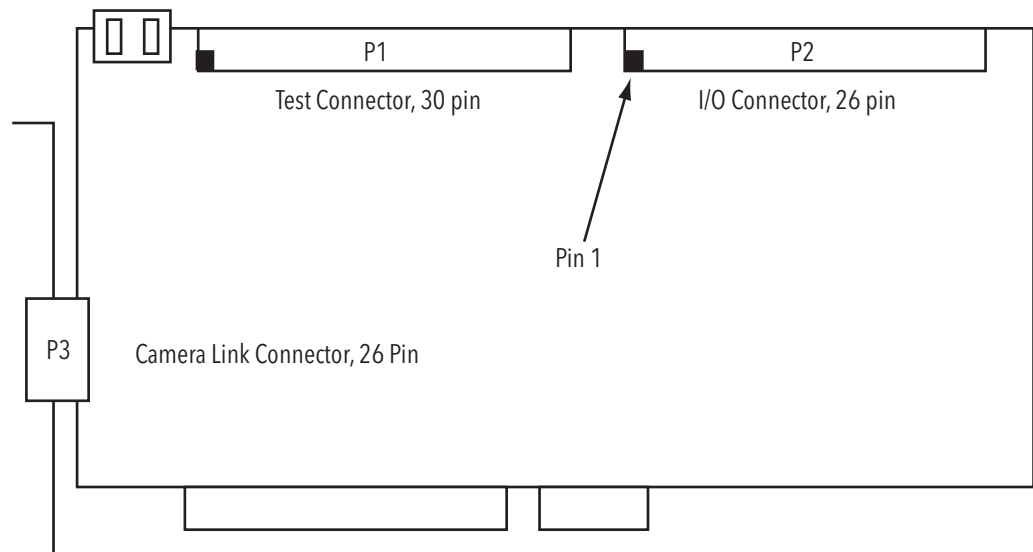


Figure 1-2 Road Runner/R3 CL PCI Layout

Connector P3 is the Camera Link connector. It has all the control signals to interface to a camera up to 24-bits wide. Those 24 bits are connected to taps T0, T1 and T2. Connector P3 comes out of the PC through a bracket.

Connector P1 is a test connector.

P2 is a connector that connects the I/O signals to a bracket that mounts on the PC bracket holder with an optional cable (CONN-RCL-IO). These I/O signals are related to industrial controls: triggers, encoders, and strobes.

In the upper left corner there are two mechanical switches that allow board identification in a system that has more than one Road Runner/R3 CL. The setting of those two switches can be read by the software, so that the software knows which board is in each slot.

Figure 1-3 shows the location of the connectors on the Road Runner/R3 CL, PMC version.

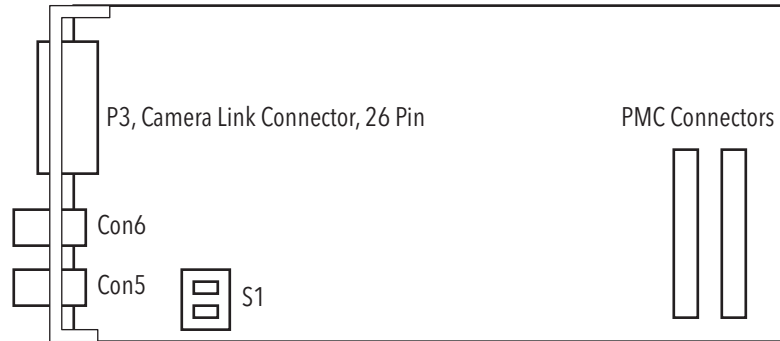


Figure 1-3 R3 CL PMC Layout

Connector P3 is the Camera Link connector. It has all the control signals to interface to a camera up to 24-bits wide. Those 24 bits are connected to taps T0, T1 and T2. Connector P3 comes out of the PC through a bracket.

Con5 and Con6 are reduced versions of the PCI board's P2 connector. These connectors are use to connect the I/O signals to to the board. These I/O signals are related to industrial controls: triggers, encoders, and strobes.

In the lower left corner there are two mechanical switches that allow board identification in a system that has more than one Road Runner/R3 CL. The setting of those two switches can be read by the software, so that the software knows which board is in each slot.

Technical Description

Chapter 2

2.1 Road Runner CL Models

2.1.1 Description

The Road Runner CL is available in four models:

- RCL-PCI-13
- RCL-PCI-13-L
- RCL-PCI-23
- RCL-PCI-23-L

Below is a description of the components that make up the model:

- RCL = Road Runner Camera Link
- PCI = PCI based model
- 1 (or 2) = Number of FIFOs installed
- 3 = Number of 8-bit input taps (always 3 for Camera Link)
- L = Optional Look Up Table (two banks 16-in/16-out).

2.1.2 Identification

The model and the installed options can be identified by the board's serial number. The serial number is written on the back of the board. The format is as follows:

#RCLP-F-R.R-SSSS-L

Where:

- #RCLP - Road Runner CL prefix
- F - Number of FIFOs
- R.R - Hardware revision
- SSSS - Serial Number
- L - Option 16-in/16-out LUT is mounted

The serial number for a model RCL-PCI-13, revision 2.4 hardware, would be:

#RCLP-1-2.4-4567

2.2 R3 CL Models

2.2.1 Description

The R3 CL is available in four models:

R3-PCI-CL13
R3-PCI-CL13-L
R3-PCI-CL23
R3-PCI-CL23-L

Below is a description of the components that make up the model:

R3 = R3 prefix
PCI = PCI based model
CL1 (or 2) = Number of FIFOs installed
3 = Number of 8-bit input taps (always 3 for Camera Link)
L = Optional Look Up Table (2x 16-in/16-out).

2.2.2 Identification

The model and the installed options can be identified by the board's serial number. The serial number is written on the back of the board. The format is as follows:

R3C-R.R-SSSS-L

Where:

R3C - R3 CL prefix
R.R - Hardware revision
SSSS - Serial Number
L - Option 16-in/16-out LUT is mounted

The serial number for a model R3-PCI-CL13, revision 2.4 hardware, would be:

R3C-2.4-4567

2.3 Camera Taps

The digital data on the three 8-bit taps is marked as T2, T1, T0. A camera with an 8-bit output will be connected to tap T0. A camera with 9-16 bit will connect on T1, T0. The MSB is on T1. A 24-bit camera will connect to T2, T1, T0.

2.4 Video FIFOs and MUXs

There are two channels, each one made up of a MUX and a FIFO. The channels are marked as A, B. Each channel accepts an 8/16/24-bit input from the three taps and packs it into 32 bits. The FIFO is a temporary storage device, for decoupling the camera input from the PCI bus. To get maximum efficiency, data over the PCI bus is always transferred as 32 bits.

The two channels are identical. Each MUX will select one or more taps from the T2,T1,T0 set and format it in a 32-bit word. The type of formatting will be done according to a 3-bit code associated with each channel, and according to the firmware downloaded into each MUX.

2.5 LUTs

At the output of the FIFOs there is an optional LUT. The FIFOs 16 LSB and the 16 MSB each pass through its own LUT. Each LUT is organized as two banks of 16-in/16-out.

The access to the LUTs from the host is done through a data port. The addressing is done through a 32-bit pointer, LUTADDR. Data is accessed through the 32-bit LUT-DATA port.

2.6 CTABs and Camera Controls

The circuitry for controlling the cameras has a programmable block that can generate a flexible set of horizontal and vertical signals.

The controls for non-standard cameras are implemented with two general purpose SRAMs: one for the horizontal axis called the Horizontal Control Table (HCTAB), the other for the vertical axis called the Vertical Control Table (VCTAB). The size of the HCTAB is 8Kx8. The size of the VCTAB is 32Kx8. In those tables are programmed the position and size of the various control signals. The HCTAB is addressed by a counter clocked by 1/4 of the horizontal pixel clock. The horizontal control signals' resolution is on a four pixel boundary. Both the HCTAB and the VCTAB are programmed from the host.

2.6.1 Horizontal Control Table

Figure 2-1 depicts the structure of the HCTAB. For clarity, the address and data path that allow the host to program the HCTAB are not shown.

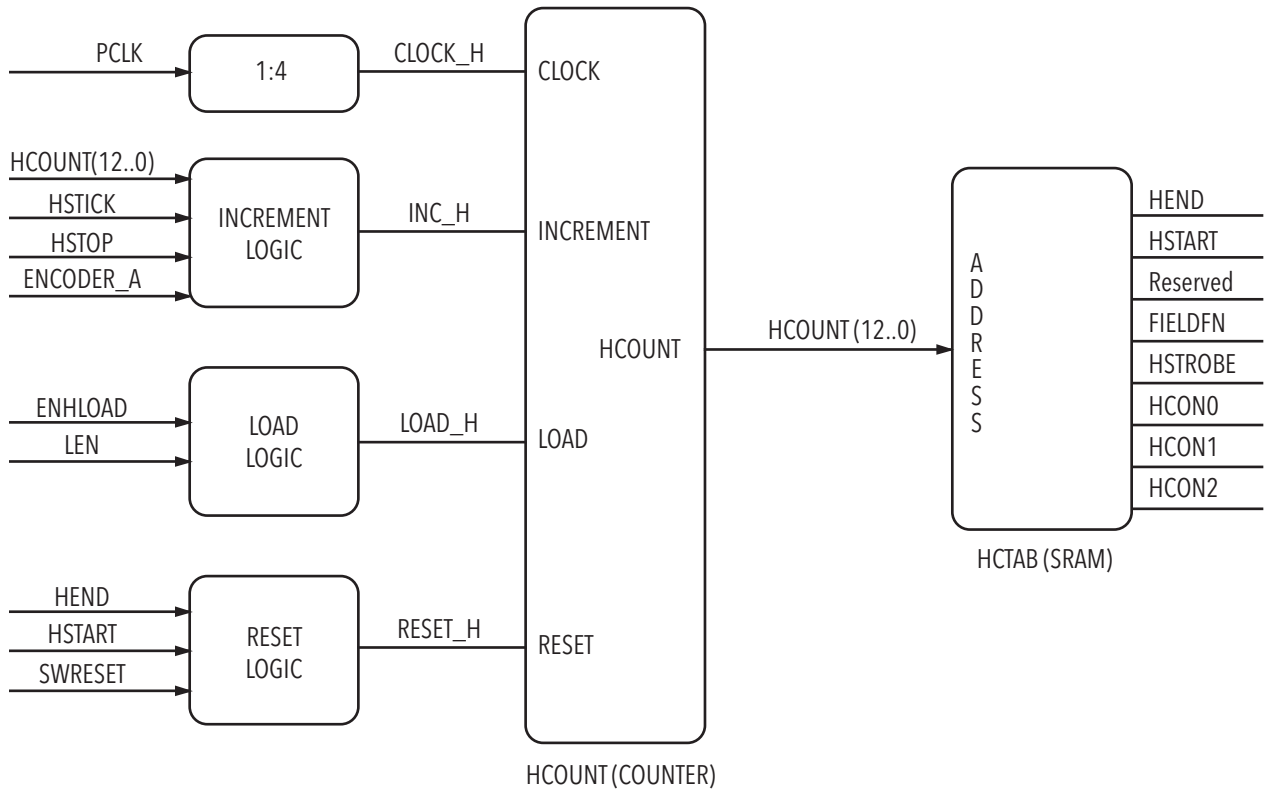


Figure 2-1 Horizontal Control

The Horizontal Control Table is made up of the following blocks:

- HCOUNT - a synchronous counter that can be incremented, loaded and reset. The clock that drives HCOUNT is a free running clock, PCLK/4, i.e., the pixel clock divided by four. HCOUNT is 13 bits wide and is connected to the address input of the HCTAB.
- Logic for generating INC_H - the increment control signal to the HCOUNT.
- Logic for generating LOAD_H - the load control signal to the HCOUNT. When LOAD_H is asserted, HCOUNT is loaded with the value of 2048.
- Logic for generating RESET_H - the reset control signal to the HCOUNT. When RESET_H is asserted, HCOUNT is reset to 0.
- HCTAB - a static memory (SRAM) that outputs eight HCTAB control signals. The address of this SRAM is driven by HCOUNT.
- Logic for generating CLOCK_H - the clock to the HCOUNT. This is a frequency divider. CLOCK_H is PCLK, the pixel clock divided by four.

Note: If RESET_H and LOAD_H are asserted simultaneously, RESET_H overrides.

As the HCOUNT counts, it scans the address of the HCTAB in ascending order. The output of the HCTAB depends on the data that has been written in the HCTAB by the host. If the HCOUNT is free running, it will cyclically scan all the HCTABs addresses. Any arbitrary cyclic waveform can be implemented by programming the HCTAB with the adequate data.

The LOAD_H and RESET_H will enable the synchronization between external events and the waveforms generated by the HCTAB. LOAD_H and RESET_H will force the HCOUNT to known values, 2048 and 0 respectively.

The INC_H signal will allow for stopping the counter from incrementing. In that case, the output of the HCTAB will be constant. While the HCOUNT is not incrementing, it can still be loaded or reset, see the logic below.

2.6.2 The HCTAB Functions

The functions assigned to the columns in the HCTAB are shown in Table 2-1.

Table 2-1 The HCTAB Functions

Bit Column	Function
D0	HEND
D1	HSTART
D2	Reserved
D3	FIELDFN

Table 2-1 The HCTAB Functions

Bit Column	Function
D4	HSTROBE
D5	HCON0
D6	HCON1
D7	HCON2

The HSTART and HEND define the horizontal acquisition window. The logic is shown in Table 2-2:

Table 2-2 Horizontal Acquisition Window

HSTART	HEND	Function
1	0	Start HWINDOW, the horizontal acquisition window.
0	1	End HWINDOW, the horizontal acquisition window.
1	1	End line; reset horizontal counter and increment vertical counter.

Also ends the HWINDOW, if still active.

Basically, HSTART and HEND tell us where the horizontal acquisition window starts and where it ends. Video will be acquired only while the HWINDOW is active. For a continuous acquisition window, we have to program only two entries in the HCTAB, regardless of the window's size.

FIELDFN is a signal that detects the Field Index for non-standard interlaced cameras. This signal is sampled by a vertical synchronization signal (FEN), and depending on the sampling point (inside the FIELDFN or outside the FIELDFN), the software should be able to determine the Field Index.

HSTROBE is ANDed with vertical strobe function to fire a strobe at a precise point in time. HCON0, HCON1 and HCON2 are general purpose horizontal functions. Some of them are combined with the vertical functions to generate CT0, CT1 and CT2 (the camera control functions).

INC_H

INC_H, the logic for incrementing HCOUNT.

There are only two instances when we want to inhibit the incrementing of HCTAB. The first instance is when HCOUNT reaches 0, "Stop at 0" case. The other instance is when HCOUNT reaches 2040, the "Stop at 2040" case.

Stop at 0

Usually, HCOUNT will reach 0 because of a RESET_H signal. After HCOUNT is reset, there are two programmable options:

HCOUNT keeps on counting.

HCOUNT stays at 0 until ENCODER_A is asserted.

The selection between option 1 and 2 is done by the HSTOP bit in CON5 as shown in Table 2-3..

Table 2-3 The HSTOP Function

HSTOP	Function
0	After reaching 000, keep counting.
1	After reaching 000, stop counting until ENCODER_A is asserted.

This operating mode is especially useful for synchronizing line scan cameras to external events. ENCODER_A is usually the output of an encoder or a part-in-place signal. Until this signal is asserted, the HCOUNT “waits” at address 000. After the ENCODER_A is asserted, HCOUNT starts counting, i.e., scanning the HCTAB in ascending order. At some address, usually in HCON1, we will program a sync signal to be asserted to the line scan camera. In response to this sync signal, the camera will give back a line, and it will assert LEN. The LEN will load address 2048 into HCOUNT. In the HCTAB, we will program the horizontal acquisition window after address 2048. At the end of the horizontal acquisition window the RESET_H will be asserted, which in turn will reset the HCOUNT. HCOUNT will “wait” at address 0 until ENCODER_A is asserted.

Stop at 2040

Using the previous example, assume that after we asserted the sync signal to the camera, we expect the camera to give us a line, i.e., assert LEN. While we expect the camera to assert LEN, HCOUNT is still being incremented. If it takes too long for the camera to respond, HCOUNT will eventually reach and pass beyond 2048. A horizontal acquisition window will be asserted even though the camera did not assert LEN. To avoid such a situation, just before address 2048, when HCOUNT reaches 2040, it will stop. It will stay at 2040 until LEN is asserted. Then, HCOUNT will be loaded with 2048.

The stop at 2040 will occur only if the HSTICK bit is asserted, see CON5. If HSTICK is not asserted, HCOUNT will be free running and it will not stop at 2040.

LOAD_H

LOAD_H, the logic of loading HCOUNT.

HCOUNT will be loaded with the value 2048 by the rising/falling edge of LEN, if ENHLOAD is asserted. LEN usually marks the start of valid data in a line. The horizontal acquisition window can be placed starting at address 2048.

ENHLOAD is a bit in CON5 that enables the LEN. There are cameras that do not assert LEN. In that case, the LEN differential receiver must be disabled, otherwise its behavior is unpredictable.

Operation on the rising/falling edge of LEN is selected by LENPOL, see CON3.

RESET_H

RESET_H the logic of reset HCOUNT.

HCOUNT can be reset from two sources:

- From within the HCTAB.
- From SWRESET.

In the first case, HCOUNT will be reset when HSTART AND HEND = 1. This will always happen at the end of the line.

In the second case, HCOUNT will be reset by SWRESET, a software reset that initializes the hardware. SWRESET is a bit in CON4.

Example:

Lets look at a simple example to clarify the concept of the HCTAB. Assume we want to program a free running horizontal window of 16 pixels active area. Just before the active area we want to fire a strobe with the function in HSTROBE. D0 (HEND) and D1 (HSTART) define the active horizontal acquisition window. They also define the RESET_H function. D4 is the strobe pulse.

Taking into account that the address counter is clocked by $\frac{1}{4}$ the pixel clock, the HCTAB memory map will be as shown in Table 2-4.

Table 2-4 HCTAB Example

HCTAB Address	HEND D0	HSTART D1	HSTROBE D4	Comments
0	0	0	0	You got here from address 8
1	0	0	0	
2	0	0	0	
3	0	0	1	Fire the strobe
4	0	1	0	Start Horizontal Acquisition Window
5	0	0	0	Acquire
6	0	0	0	Acquire
7	0	0	0	Acquire
8	1	1	0	Stop acquisition, assert RESET_H
9	0	0	0	

Now lets assume a line scan camera that does not supply any type of signal, but it has to be driven. We must supply a LEN type signal to the camera. In that case, the RESET_H signal will take care of generating a cyclical signal. The position of RESET_H relative to address 0 will determine the period of the horizontal scanning. If RESET_H is set at address X1, then the horizontal active window should be programmed between 0 and X1.

Some line scan cameras require a line start command after the previous line has been read out. We can program the HCTAB to stop incrementing after the line has been read. The ENCODER_A will restart the count (from address 000H). Program the line start command to the camera in the 0-2048 area. In response to the line start command, the camera will give back a LEN. Set LENPOL = 1 so that the loading occurs on the rising edge of LEN. After address 2048, program the position and size of the horizontal acquisition window by setting HSTART end HEND.

Table 2-5 shows the effect of LEN on HCOUNT for different settings of ENHLOAD and LENPOL.

Table 2-5 Effect of LEN on HCOUNT

LENPOL	ENHLOAD	Signal	HCOUNT @ Current CLOCK_H	HCOUNT @ Next CLOCK_H
0	0	Leading edge of LEN	HCOUNT	HCOUNT+1
0	1	Leading edge of LEN	HCOUNT	2048
1	0	Trailing edge of LEN	HCOUNT	HCOUNT+1
1	1	Trailing edge of LEN	HCOUNT	2048

Table 2-6 shows the effect of ENCODER_A and RESET_H on HCOUNT for different settings of HSTOP. The assumption is that the encoder polarity is positive, ENCPOL = 0. For ENCPOL = 1, substitute "rising edge" with "falling edge."

Table 2-6 The Effect of ENCODER_A on HCOUNT

HSTOP	Signal	HCOUNT @ Current CLOCK_H	HCOUNT @ Next CLOCK_H
0	None	HCOUNT	HCOUNT+1
1	None	0	0
1	Rising edge of ENCODER_A	0	1
1	Rising edge of ENCODER_A	HCOUNT > 0	HCOUNT+1
0	RESET_H	HCOUNT	0
1	RESET_H	HCOUNT	0

2.6.3 Vertical Control Table

Figure 2-2 depicts the structure of the Vertical Control Table (VCTAB). For clarity, the address and data path that allow the host to program the VCTAB are not shown.

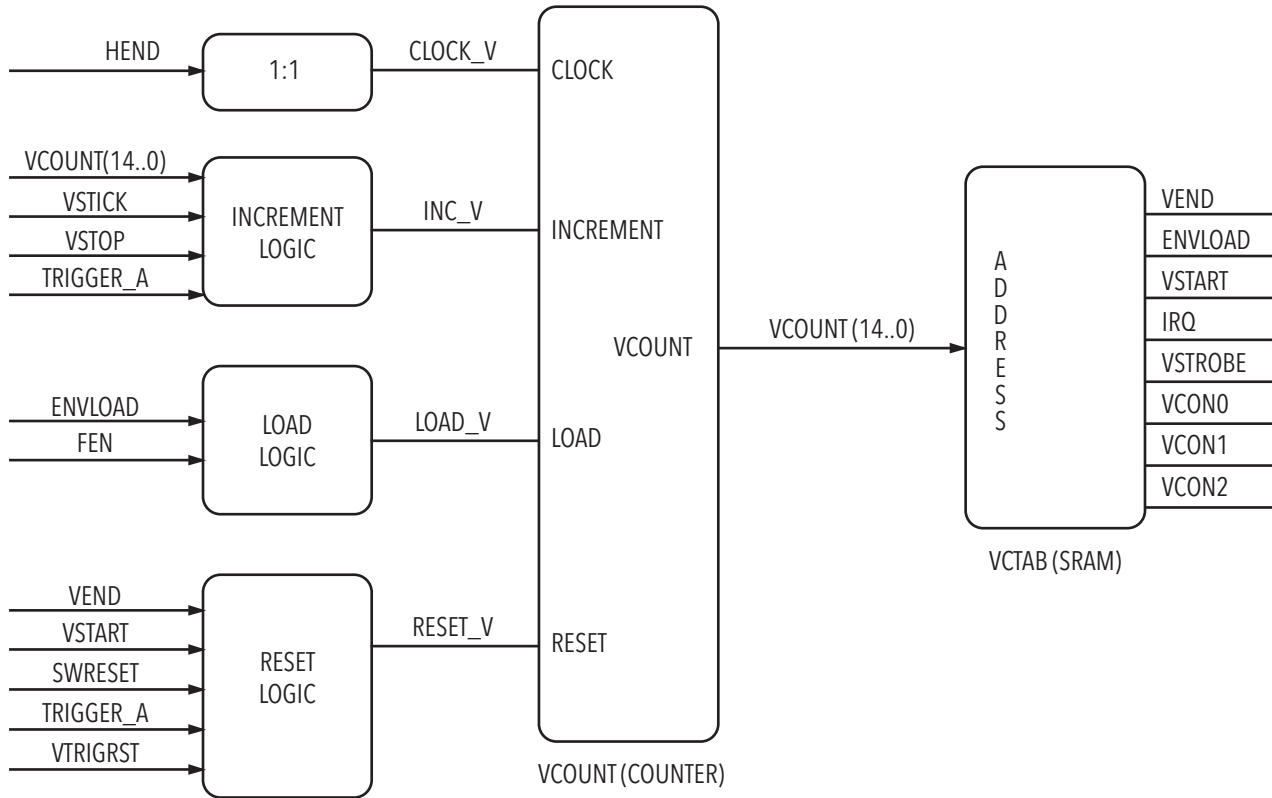


Figure 2-2 Vertical Control

The Vertical Control Table is made up of the following blocks:

VCOUNT - a synchronous counter that can be incremented, loaded and reset. The clock that drives VCOUNT is derived from the HCTAB, see below. VCOUNT is 15-bit wide and is connected to the address input of the VCTAB.

Logic for generating INC_V - the increment control signal to the VCOUNT.

Logic for generating LOAD_V - the load control signal to the VCOUNT. When LOAD_V is asserted, VCOUNT is loaded with the value of 4096.

Logic for generating RESET_V - the reset control signal to the VCOUNT. When RESET_V is asserted, VCOUNT is reset to 0.

Logic for generating CLOCK_V - the clock to the VCOUNT.

VCTAB - a static memory (SRAM) that outputs eight VCTAB control signals. The address of this SRAM is driven by VCOUNT.

If RESET_V and LOAD_V are asserted simultaneous, RESET_V overrides.

As the VCOUNT counts, it scans the address of the VCTAB in ascending order. The output of the VCTAB depends on the data that has been written in the VCTAB by the host. If the VCOUNT is free running, it will cyclically scan all the VCTAB's addresses. Any arbitrary cyclic waveform can be implemented by programming the VCTAB with the adequate data.

The LOAD_V and RESET_V will enable the synchronization between external events and the waveforms generated by the VCTAB. LOAD_V and RESET_V will force the VCOUNT to known values, 4096 and 0 respectively.

The INC_V signal will allow for stopping the counter from incrementing. In that case, the output of the VCTAB will be constant. While the VCOUNT is not incrementing, it can still be loaded or reset, see the logic below.

2.6.4 The VCTAB Functions

The functions assigned to the columns in the VCTAB are shown in Table 2-7.

Table 2-7 The VCTAB

Bit Column	Function
D0	VEND
D1	ENVLOAD
D2	VSTART
D3	IRQ
D4	VSTROBE
D5	VCON0
D6	VCON1
D7	VCON2

VSTART and VEND define the vertical acquisition window. The logic is shown in Table 2-8.

Table 2-8 The Vertical Acquisition Window

VSTART	VEND	Function
1	0	Start VWINDOW, the vertical acquisition window.
0	1	End VWINDOW, the vertical acquisition window.
1	1	End frame and reset VCOUNT. Also ends the VWINDOW, if still active.

Basically, VSTART and VEND tell us where the vertical acquisition window starts and where it ends. Video will be acquired only while the VWINDOW is active. For a continuous acquisition window, we have to program only two entries in the VCTAB, regardless of the window's size.

VWINDOW, the vertical acquisition window, does not have to be continuous. We can selectively acquire sections of a frame by properly manipulating VSTART and VEND. VWINDOW is used in conjunction with HWINDOW to control when video data is written to memory.

ENVLOAD enables the FEN to load the VCOUNT. The rationale behind the ENVLOAD column from the VCTAB is that some cameras might not give a FEN, but only two pulses: the start and end of FEN. With the ENVLOAD, we can mask out the unwanted one.

IRQ provides an interrupt to the bus, allowing an interrupt to occur at any point on the vertical axis.

VSTROBE, used in conjunction with HSTROBE, fires a strobe at a precise point in time.

VCON0, VCON1 and VCON2 are general purpose vertical functions, see usage below.

Note: If a VRESET pulse happens coincident with a LOAD, the LOAD is overriding.

CLOCK_V

CLOCK_V, the clocking of the VCOUNT.

CLOCK_V is generated by the HCTAB. Whenever HSTART AND HEND = 1, the VCOUNT will be incremented. This will always happen at the end of the line.

INC_V, the logic for incrementing VCOUNT.

There are only two instances when we want to inhibit the incrementing of VCTAB. The first instance is when VCOUNT reaches 000, the "Stop at 000" case. The other instance is when HCOUNT reaches 4088, the "Stop at 4088" case.

Stop at 0

Usually, VCOUNT will reach 0 because of a RESET_V signal. After VCOUNT is reset, there are two programmable options:

VCOUNT keeps on counting.

VCOUNT stays at 0 until TRIGGER_A is asserted.

The selection between option 1 and 2 is done by the VSTOP bit in CON5 as shown in Table 2-9.

Table 2-9 VSTOP

VSTOP	Function
0	After reaching 0, keep counting.
1	After reaching 0, stop counting until TRIGGER_A is asserted.

This operating mode is especially useful for synchronizing cameras to external events. TRIGGER_A is usually the output of a part-in-place signal. Until this signal is asserted, the VCOUNT waits at address 0. After the TRIGGER_A is asserted, VCOUNT starts counting, i.e., scanning the VCTAB in ascending order. At some address, usually in VCON0, we will program a sync signal to be asserted to the camera. In response to this sync signal, the camera will give back a frame, and it will assert FEN. The FEN will load address 4096 into VCOUNT. In the VCTAB, we will program the vertical acquisition window after address 4096. At the end of the vertical acquisition window the RESET_V will be asserted, which in turn will reset the VCOUNT. VCOUNT will wait at address 0 until TRIGGER_A is asserted.

Stop at 4088

Using the previous example, assume that after we asserted the sync signal to the camera, we expect the camera to give us a frame, i.e., assert FEN. While we expect the camera to assert FEN, VCOUNT is still being incremented. If it takes too long for the camera to respond, VCOUNT will eventually reach and pass beyond 4096. A vertical acquisition window will be asserted, even though the camera did not assert FEN. To avoid such a situation, just before address 4096, when VCOUNT reaches 4088, it will stop. It will stay at 4088 until FEN is asserted. Then, VCOUNT will be loaded with 4096.

The stop at 4088 will occur only if the ENVLOAD is asserted, see CON5. ENVLOAD will enable the operation of FEN. If ENVLOAD is not asserted, VCOUNT will be free running and it will not stop at 4088.

LOAD_V

LOAD_V, the logic of loading VCOUNT.

VCOUNT will be loaded with the value 4096 by the rising/falling edge of FEN, if ENVLOAD is asserted. FEN usually marks the start of a valid frame. The vertical acquisition window can be placed starting at address 4096.

ENVLOAD is a column in the VCTAB. There are cameras that do not assert FEN. In this case, the FEN differential receiver must be disabled, otherwise its behavior is unpredictable. Some other type of cameras assert only the start and stop of a frame. In this case, ENVLOAD can mask out the unwanted signals.

Operation on the rising/falling edge of FEN is selected by FENPOL, see CON3

RESET_V

RESET_V, the logic of reset VCOUNT.

VCOUNT can be reset from three sources:

- From within the VCTAB.
- From TRIGGER_A.
- From SWRESET.

In the first case, VCOUNT will be reset when VSTART AND VEND = 1. This will always happen at the end of the frame.

In the second case, VCOUNT can be reset by TRIGGER_A if VTRIGRST is asserted. VTRIGRST is a bit in CON5. It will enable the TRIGGER_A to reset the VCOUNT.

In the third case, VCOUNT will be reset by SWRESET, a software reset that initializes the hardware. SWRESET is a bit in CON4.

Note: Unpredictable behavior occurs for TRIGGER_A feature if both VSTOP and VTRIGRST are each set.

Table 2-10 shows the effect of FEN on VCOUNT for different settings of ENVLOAD and FENPOL.

Table 2-10 The Effect of FEN on VCOUNT

FENPOL	ENVLOAD	FEN	VCOUNT @ Current CLOCK_V	VCOUNT @ Next CLOCK_V
0	0	Leading edge of FEN	VCOUNT	VCOUNT + 1
0	1	Leading edge of FEN	VCOUNT	4096
1	0	Trailing edge of FEN	VCOUNT	VCOUNT + 1
1	1	Trailing edge of FEN	VCOUNT	4096

Table 2-11 shows the effect of TRIGGER_A and RESET_V on VCOUNT, for different settings of VSTOP. The table assumes TRIGPOL = 0. If TRIGPOL = 1, substitute "falling edge" instead of "rising edge" for TRIGGER_A.

Table 2-11 The Effect of TRIGGER_A and RESET_V on VCOUNT vs. VSTOP

VSTOP	Signal Asserted	VCOUNT @ Current CLOCK_V	VCOUNT @ Next CLOCK_Vn
0	None	VCOUNT	VCOUNT+1
0	Rising edge of TRIGGER_A	VCOUNT	VCOUNT+1
1	None	0	0
1	Rising edge of TRIGGER_A	0	1
1	Rising edge of TRIGGER_A	VCOUNT > 0	VCOUNT+1
1	RESET_V	VCOUNT	0
0	RESET_V	VCOUNT	0

Table 2-12 shows the effect of TRIGGER_A and RESET_V on VCOUNT, for different settings of VTRIGRST. The table assumes TRIGPOL = 0. If TRIGPOL = 1, substitute “falling edge” instead of “rising edge” for TRIGGER_A.

Table 2-12 The Effect of TRIGGER_A and RESET_V on VCOUNT vs. VTRIGRST

VTRIGRST	Signal Asserted	VCOUNT @ Current CLOCK_V	VCOUNT @ Next CLOCK_V
0	Rising edge of TRIGGER_A	VCOUNT	VCOUNT+1
0	RESET_V	VCOUNT	0
1	Rising edge of TRIGGER_A	VCOUNT	0
1	RESET_V	VCOUNT	0

2.7 Camera Control Signals

2.7.1 Definition

The CTABs control signals, CT0, CT1, CT2 and STROBE are functions of the horizontal and vertical CTABs. The equations are given in Table 2-13

Table 2-13 The CT Signals

Signal	Comments
$CT0 = VCON0 + (VCON1 * HCON0)$	Suggested for camera shutter control
$CT1 = VCON2 * HCON1$	Suggested for vertical sync control
$CT2 = HCON2$	Suggested for horizontal sync control
$STROBE = VSTROBE * HSTROBE$	Light strobe

Note: "*" is a logical AND. "+" is a logical OR

The Camera Link protocol defines four camera control signals, CC1, CC2, CC3 and CC4. These signals are general purpose and are not pre-defined by the specifications. Each camera manufacturer defines those signals according to their needs. To be able to accommodate all current and future needs, a general purpose allocation scheme has been designed to allocate the Road Runner/R3 CL's control signals, CT0, CT1, CT2 and STROBE, to the Camera Link general purpose controls, CC1, CC2, CC3 and CC4. For each CC, there are three control bits associated (see description of control register CON10). Table 2-14 shows the routing of the Road Runner/R3 CL control signals to the Camera Link's four CCs.

Table 2-14 Controlling the CC Outputs

CCxCON	CCx
0 (000b)	CT0
1 (001b)	CT1
2 (010b)	CT2
3 (011b)	STROBE
4 (100b)	CLOCK_OUT
5 (101b)	0
6 (110b)	1
7 (111b)	Reserved

CLOCK_OUT is the clock supplied by the on board oscillator, see CON0.

2.8 Host Access to VCTAB and HCTAB

When looking from the host side, the two tables are organized as one memory 32Kx32 bits, where bits 0 to 7 are the HCTAB and bits 8 to 15 are the VCTAB. The host access looks like 32-bit wide memory on the local bus. The HCTAB, VCTAB are mapped in the lower half. Access should be as a 32-bit word, where the upper 16 bits are unused. In order to modify only one table, a read-modify-write cycle must be performed. The host can access the tables anytime, as they are memory mapped on the local bus. Whenever the circuit detects host access, it will automatically connect the tables to the local bus. For this reason, it is good practice to stop the tables from incrementing while programming. This can be done with the CTABHOLD bit. This bit disables clocking of both tables, but leaves the state unchanged. If the tables are programmed when they are running, unpredictable controls might be given to the camera.

2.9 The DMA Engine

Figure 2-3 shows the structure of the DMA engine. The DMA engine is designed around the PLX PCI9080 controller. The DMA will be done in Chaining Mode (see the PCI9080 manual).

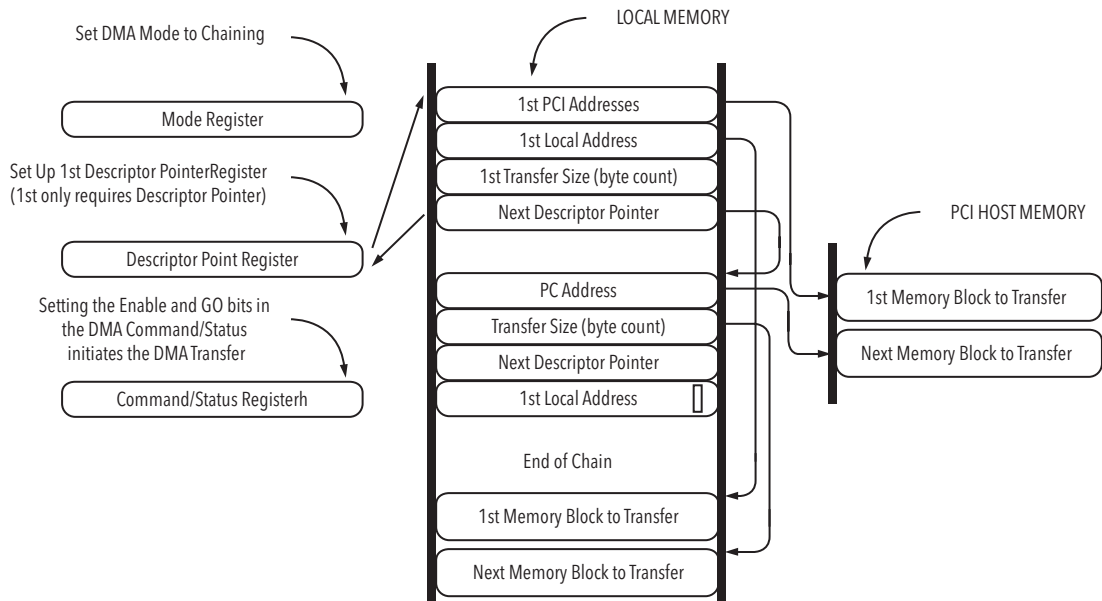


Figure 2-3 The Descriptor Table

The host sets up descriptor blocks in the Descriptor Table. This table is located in the host memory. Each descriptor block has four entries, each one 32 bits wide and consists of the following:

- PCI address
- Local address
- Transfer count and attributes
- Address of next descriptor block

To start a DMA transfer, the following conditions must be met:

- The Bus Master enable bit in the PCI9080 PCI configuration space is set.
- The Descriptor Table has been loaded with descriptors.
- The DMA channel enable bit has been set in the PCI9080 DMA command/status register.
- The address of the first descriptor has been written in the PCI9080 descriptor pointer register.
- The GO bit in the PCI9080's DMA command/status register has been set.

The PCI9080 loads the first descriptor block from the table into four internal registers and initiates the data transfer. The PCI9080 continues to load descriptor blocks and transfer data until it detects the End Of Chain bit set in the next descriptor pointer register.

The DMA process can be paused by de-asserting the DMA channel enable bit in the PCI9080 DMA command/status register.

The DMA process can be aborted by setting the abort in the PCI9080 DMA command/status register.

The PCI9080 can be programmed to assert a DMA interrupt in two cases:

- Completion of a block transfer.
- All blocks have been transferred.

The interrupt is asserted on the local bus. Control logic will pick it up and assert the interrupt on the PCI (see the Interrupts Section).

The following sections discuss the four entries that make up one descriptor block. For more information, refer to the PLX PCI9080 data book.

2.9.1 PCI (Destination) Address

This is a 32-bit address. It should be on a long word boundary as we always transfer 32-bit words. The address is loaded in the PCI9080 PCI address register. This register is R/W from the PCI

2.9.2 Local Bus (Source) Address

This is the address of the data on the local bus. For transfer of video data, it will be the address of the Video FIFO. In that case, a control bit can be set that will disable the incrementing of the local address during DMA transfers, i.e., all accesses on the local bus will be done to the same address. It is also possible to DMA other type of data between the Road Runner/R3 CL board and the PCI, like the CTABs and the Descriptor Tables.

The seven LSBs of the local address entry are not used by the PCI9080. Those bits will be used by external control logic (logic located outside the PCI9080 chip) as attributes. Those attributes will control specific hardware functions related to current transfer. Control logic will snoop the local bus and whenever the PCI9080 will read the local address from the Descriptor Table, it will latch the nine LS bits in the Attribute Register.

The following sections provide a description of the seven attribute bits.

EOX, End of Sequence

This bit denotes the end of sequence. If set, it tells the controller that the current descriptor is the last one in a transfer sequence. Used mainly for circular sequences, where the EOC bit is not set, see the next section.

BANK, BYPASS, LUT Controls

These two bits will control the LUTs during a DMA transfer. They select Bank 1, Bank 0, or bypass mode. When the board is in slave mode, these functions are controlled by the HBANK, HBYPASS (“H” stands for host access), see the registers description. For models that do not have the LUTs installed, those bits are don’t care.

MCHSEL

MCHSEL, controls the video FIFO that the DMA engine is reading from.

FLUSH

FLUSH, control bit for the local bus.

2.9.3 Transfer Size

This entry holds the size of the transfer in the 23 LSBs. Maximum transfer size per descriptor is 8 megabytes. PLX claims that zero size is legal. No transfer will take place, but the descriptor will be read in. Can be handy in controlling the events.

2.9.4 Descriptor Pointer

The 28 MSBs of this entry point to the address of the next descriptor block on the local bus. The four LSBs are control bits:

- Bit 0 Reserved
- Bit 1 End Of Chain (EOC)
- Bit 2 Interrupt after terminal count for this descriptor
- Bit 3 Direction of transfer

If the EOC bit is set, the PCI9080 will stop the DMA after transferring the current block. We can achieve a circular sequence, if the last descriptor points to the first one, and the EOC bit is not set. This is comes in handy if you want to have a continuous display, for example.

- Bit 2 will assert an interrupt at the completion of the current descriptor.
- Bit 3 will determine the direction of the DMA, 1 for local to PCI.

2.10 Interrupts

Interrupts to the PCI can come from two groups:

- Interrupts generated in the PCI9080 related to the DMA operation, INT_DMA.
- Interrupts related to the acquisition process.

Below is a description of each group:

Interrupts generated in the PCI9080 are related to the DMA operation. It can be from the end of a sequence or from the completion of a descriptor (see Interrupt Control/Status register of the PCI9080). In order to enable the DMA interrupts, the respective control bits must be set in the PCI9080 control registers. The DMA circuit in the PCI9080 cannot assert an interrupt on the PCI bus. It will assert the LINTO* (Local Interrupt Out) pin on the local bus. This signal will be routed to the local bus controller and ORed with INT0 to INT4. The output of this ORing will be connected to the LINT1* input of the PCI9080 chip. This pin will generate an interrupt on the PCI bus, if the PCI9080 PCI interrupt enable and the local interrupt enable bits are set (see the PCI9080 Interrupt Control/Status register).

The interrupts related to the acquisition process are listed below:

- INT[0], aka INT0_CTAB
- INT[1], aka INT1_FIFO
- INT[2], aka INT2_HW
- INT[3], aka INT3_TRIG
- INT[4], aka INT4_SER

Those bits are located in CON2.

INT0_CTAB is the interrupt programmed in the CTABs.

INT1_FIFO will be asserted if there is a Video FIFO overflow, i.e., the transfers over the PCI cannot keep up with the data rate from the camera.

INT2_HW will be set when there is a local bus time-out or a loss of sync from the camera.

INT3_TRIG will be generated by the active edge of the TRIGGERA signal.

INT4_SER is the interrupt from the on-board UART that communicates with the camera over the Camera Link.

Each interrupt source has its own interrupt enable bit in CON4. The host can read/write each one of the interrupt bits, except for the INT_SER. The INT_SER is the interrupt bit set by the UART. The setting/resetting of this bit is done by accessing the UART control register that deals with interrupts.

All the interrupts share one single line on the PCI bus. The SW will have to find out who asserted the interrupt and act accordingly.

With each interrupt, except for the INT_SER, there is an associated CMDWRITE code. This code will enable host access to the specific interrupt (see the CON4 description). This is needed to be able to reliably perform a read-modify-write operation.

2.11 The Acquisition Process

The Acquisition modes are controlled by the TRIGCON register as shown in Table 2-15.

Table 2-15 TRIGCON

TRIGCON	Function
0 (00b)	TRIGGERA active, B disabled
1 (01b)	Start on A, stop on B
2 (10b)	Continuous acquisition
3 (11b)	Triggered Termination

The process of acquiring data (writing) into the video FIFOs is independent of the process of reading data out of the video FIFOs. The readout is controlled by the DMA engine. The writing into the FIFOs is controlled by the acquisition command bits AQ.

Lets first analyze the writing into the video FIFOs. The trigger mode control bits will be TRIGCON = 0. For new modes, see the next section.

The definition of the acquisition command is shown in Table 2-16.

Table 2-16 AQ

AQ	Function
0 (00b)	FREEZE
1 (01b)	ABORT
2 (10b)	SNAP
3 (11b)	GRAB

In non-triggered acquisition (TRIGAQ = 0), the acquisition command will be latched at the start of the next active frame. In triggered acquisition, TRIGAQ = 1, the acquisition command will be latched at the beginning of the first frame after the trigger is asserted.

AQSTAT will mirror the type of acquisition in process. AQSTAT are actually the AQ bits latched at the start of the active video, after an acquisition command has been issued. They reflect the type of acquisition in progress. The following sections describe the four states.

2.11.1 Freeze

When a FREEZE is asserted, acquisition will stop at the end of the current frame. This is usually used to stop the grab in an orderly manner and to return the acquisition to state 00, that is, the idle state.

2.11.2 Abort

The ABORT command will stop acquisition immediately, unconditionally. Writing a 01 in the acquisition command will stop all acquisition activity. The acquisition command AQ and the AQSTAT bits will read back 00, as the acquisition is in the idle state.

2.11.3 Snap

This command will acquire a single frame (or frame/field in interlaced). After the frame has been acquired, the status will read back 00, the idle state.

After the AQCMD bits have been latched in the AQSTAT bits, at the start of the frame, they will be cleared. This way, the SW will know if a new SNAP/GRAB command has been issued while a SNAP is in progress.

A SNAP command can be issued while the Road Runner/R3 CL is acquiring in SNAP mode. This will make the Road Runner/R3 CL SNAP another frame. Issuing more than one SNAP command while acquiring in SNAP mode will have no effect.

A GRAB command can be issued while the Road Runner/R3 CL is acquiring in SNAP mode. After the SNAP is done, it will start a GRAB.

A FREEZE command while acquiring in SNAP mode will have no effect.

An ABORT command while snapping will immediately stop the acquisition.

2.11.4 Grab

This command will acquire continuous starting at the next active frame.

A SNAP command can be issued while the Road Runner/R3 CL is grabbing. The next frame will be a SNAP operation.

A GRAB command can be issued while in GRAB mode. You do that if you want to change the active channels, see below.

A FREEZE command will stop the acquisition after the current frame.

ABORT will stop acquisition unconditionally.

Figure 2-4 shows the behavior of the AQCMD and AQSTAT bits. The SNAP while displaying and acquire on trigger while displaying need more discussion.

Assume we have a live display and we want to SNAP a frame into system memory.

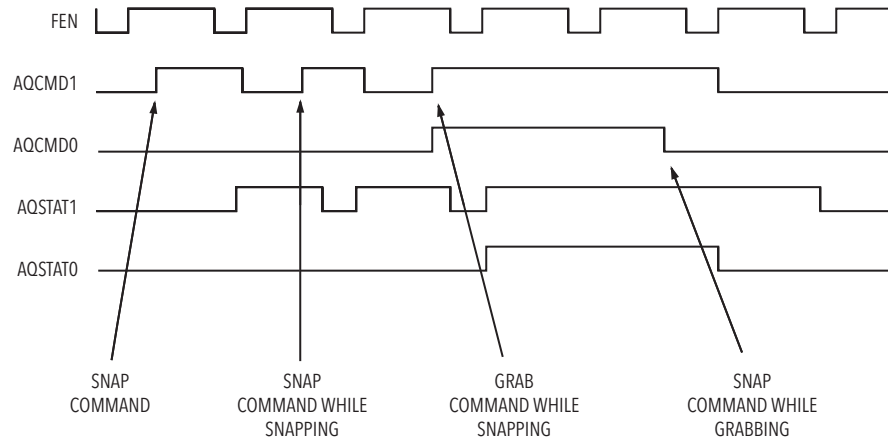


Figure 2-4 AQCMD and AQSTAT Bits Behavior

2.12 Advanced Acquisition Modes

Two new acquisition modes are available. These modes are set by the trigger control bits TRIGCON.

2.12.1 Start-Stop Triggered Acquisition

In this mode, the start of the acquisition will be initiated by TRIGGER_A. The end of the acquisition will be initiated by TRIGGER_B. In this mode, asserting TRIGGER_B is identical to a FREEZE command. Set CTABs, AQCMD, TRIGAQ, and descriptors for triggered GRAB.

Note: The Start-Stop mode can be done from external hardware, software, or a combination of both. SWTRIG_A,B have the same effect as TRIGGER_A,B.

2.12.2 Continuous Acquisition

In this mode, the Road Runner/R3 CL will acquire continuous acquisition. The CTABs, AQCMD, and TRIGAQ bits will not be involved in this type of acquisition. The acquisition will be controlled by TRIGGER_A only. Whenever TRIGGER_A is asserted HI, data will be acquired. The same effect can be achieved by using SWTRIG_A.

An interrupt can be asserted at the start or the end of the acquisition, depending on bit CONTINTPOL in CON4.

2.12.3 Triggered Termination

This mode is used to terminate the acquisition immediately, without waiting for the end of the frame. It is slightly different than the assertion of the ABORT command. It will end the acquisition process and the DMA process in a graceful way. Acquisition can be started normally with a GRAB command, or triggered by TRIGGERA. The acquisition can be terminated by TRIGGERA or TRIGGERB, depending on the setting of bit TRIGASTOP in CON5.

If TRIGASTOP, termination by falling edge of TRIGGERA, TRIGGERA can still be used to start the acquisition on its rising edge. In this mode, the TRIGGERA has to be an active high level signal.

Interfacing

Chapter 3

3.1 Introduction

The trigger and the encoder input signals on the Road Runner/R3 CL board have been designed to interface to both TTL and RS422 signals. Figure 3-1 shows the electrical structure for the receiver.

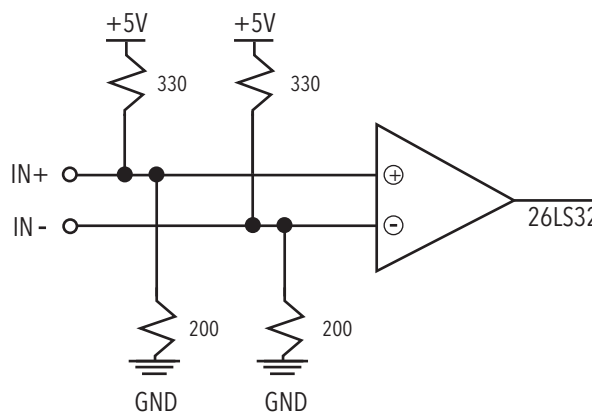


Figure 3-1 Input Circuit for the Encoder and Trigger Signals

Each differential input is biased at $\sim 2V$. For a RS422 input signal, connect to the IN+ and IN- inputs. For a TTL signal, connect the single ended input to IN+ (or IN-) and leave the other input unconnected. The unconnected input will be biased at $\sim 2V$, the TTL signal will swing between 0.8V and 2.4V. The ground of the TTL signal must be connected to the Road Runner/R3 CL's ground (pin 6 on the P5 connector, pin 5 on the CON-RUN-IO connector).

The minimum drive requirement for the TTL driver is 2 mA source @ 2.4V and 10 mA sink @ 0.4V. A 74F125 device or equivalent is adequate. Note that an open collector device will not do the job. You will have to add a pull-up resistor of about 100 Ohm.

The minimum pulse width on the trigger and the encoder inputs is two PCLK cycles. For a 20 Mhz PCLK that means 100 nanoseconds. For a 500 KHz PCLK that means 1uS.

Figure 3-2 illustrates the proper connection for a TTL Trigger or TTL-Encoder source to a Road Runner/R3 CL board.

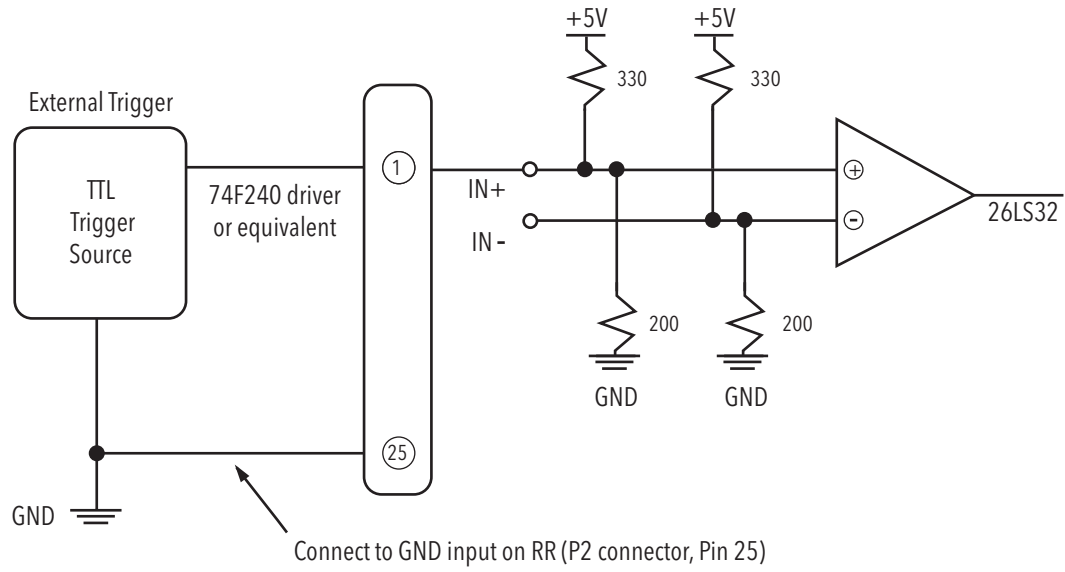


Figure 3-2 Connecting an External TTL Trigger or Encoder to a Road Runner/R3 CL Board

Power Requirements

Chapter 4

4.1 Maximum Current Requirements from +5V Power Supply

Table 4-1 shows the power requirements for the Road Runner/R3 Camera Link version.

Table 4-1 Power Requirements

Model	Maximum Current
RCL-PCI-13, R3-PCI-CL13	2 Amp. max
RCL-PCI-13-L, R3-PCI-CL13-L	2.1 Amp. max
RCL-PCI-23, R3-PCI-CL32	2.5 Amp. max.
RCL-PCI-23-L, R3-PCI-CL23-L	2.6 Amp. max

These numbers apply for all options. The Road Runner/R3 CL uses only the +5V power supply.

Specifications

Chapter 5

5.1 Introduction

This chapter describes the general specifications of the Road Runner and the R64. The numerical values for the specifications are listed in Table 5-1. If more information is available for a given specification there will be an entry in the column marked "Details".

Table 5-1 Road Runner/R3 Specifications

Specifications	Value	Units	Details
PCI Bus compatible frequencies	33	MHz	
PCI Bus compatible data widths	32	Bits	
PCI Bus compatible voltages	3.3 and 5	Volts	Section 5.2
Maximum Input Pixel Clock Frequency	66	MHz	
Minimum Input Pixel Clock Frequency	greater than 0	MHz	
Maximum Pixels Per Line (1 tap)	131,072 (128K)	Pixels	
Maximum Lines Per Frame	32,768 (32K)	Lines	Section 5.3
Minimum clocks between lines	16	Clocks	
Minimum lines between frames	0	Lines	
Minimum pixel clocks between frames	4	Clocks	
Minimum trigger pulse	1	Clock	
Minimum encoder pulse	2	Clock	
Current consumption (+5V)	2 to 2.6	Amps	Section 4.1
Temperature range	0 to 50	Degrees Celsius	
Humidity	25% to 80%		
Road Runner CL mechanical dimensions	6.6 x 4.2	Inches	
Road Runner CL mechanical dimensions	16.8 x 10.7	Centimeters	
R3 CL mechanical dimensions	6.6 x 4.2	Inches	
R3 CL mechanical dimensions	16.8 x 10.7	Centimeters	

5.2 PCI bus compatibility

The R3 will work in both 3.3V and 5V PCI buses. The Road Runner will only work in 5V PCI busses. Please note that this voltage specification is related only to the signalling level on the PCI bus, not the availability of a voltage on the connector. The different voltage PCI connectors have different keying. You cannot plug a 5V board into a 3.3V slot. The R3 is designed as a "universal" board and is key so that it will plug into both 3.3V and 5V slots. The Road Runner is key so that it will plug into only 5V slots.

5.3 Maximum Lines Per Frame

For area scan cameras, the maximum number of lines per frame 32,768. For line scan cameras, the board can be set up to continuously acquire lines, with no limit to the number of lines acquire continuous. In the mode, no lines are dropped. An interrupt can be sent to the host every N lines. N can be any number up to the number of lines the host is capable of storing (limited only by RAM in the computer).

PCI Interface

Chapter 6

6.1 Introduction

The registers, LUTs, CTABs and the PCI9080 registers will take up two 1 megabytes (Mb) address spaces. On the PCI side, those two 1 Mb address spaces will be located at the address assigned by the BIOS. On the local bus mapping registers will place the first 1 Mb at address 000h. The next 1 Mbyte will be consecutive to the first one. In the first address space are all the registers. The second address space is dedicated to the UART. Only address bits 0 to 20 will be needed to access all registers on the local bus.

All accesses to the board should be done as 32-bit long words.

The PCI90x0 registers are available on the local bus and on the PCI bus, except for the DMA register that are available on the local bus only. The PCI can access the DMA registers by doing an access to the local bus.

Note: The PCI90x0 takes up to two distinct address spaces on the PCI bus: one for its internal registers, and one for the local bus.

6.2 Board Addressing

The following is an address map of the local bus:

Table 6-1

Address (hex)	
0000 0000	PCI9080 chip,130x32
000A 0000	CTABS, 32Kx32
0018 0000	UART serial port registers
000C 0000	Video FIFO
000E 0004	Read LUT address
000E 0008	Read LUT data port
000E 000C	CON0, Control register 0
000E 0010	CON1
000E 0014	CON2
000E 0018	CON3
000E 001C	CON4
000E 0100	CON5
000E 0104	CON6
000E 0108	CON7
000E 010C	CON8
000E 0110	CON9
000E 0114	CON10
0008 0000	CON11
000E 0118	CON12

6.2.1 Example 1:

The range for PCI to local address will be loaded with the value FFE0 0000h from the EEPROM. This will tell the BIOS that the Road Runner needs 2 Mb of address space. The remap register will be loaded with the value 0000 0000h, which means that on the local bus, the 2 MB window will be mapped at address 0.

Lets assume the BIOS assigned the Road Runner a PCI base address of 1220 0000h.

Lets assume the PCI wants to access register CON3. On the local bus, CON3 is at address 000E 0018h. The host should assert on the PCI address 122E 0018h.

6.2.2 Example 2:

Now let's assume the host wants to access the DMA Command/Status register. This register can be accessed only from the local bus. The PCI can access this register by doing an access to the local bus. The register is at offset 128h from the PCI90x0 base address (chip select) on the local bus. On the local bus, the PCI90x0's base address (chip select) is mapped at address 0000 0000h. To access this register from the local bus, address 0000 0128h must be asserted on the local bus. For the host to access this register, it will have to assert on the PCI address 1220 0128h.

6.3 Non-Register Memory

This section contains descriptions of all of the memory on the Road Runner/R3 that is not used for control registers.

6.3.1 CTABS

HCTAB in bits 7-0, VCTAB in bits 15-8. Both LUTs accessed as one 32-bit word. Table size is 32K.

6.3.2 LUTADDR

This is a LUT address pointer. Addresses are for two lanes of 16in-16out. For the two lanes of 16in-16out, the LUTADDRR holds two 16-bit address pointers.

The bank pointed to is determined by the BANK bit.

6.3.3 LUTDATA

LUT data port, either 4x8 or 2x16, 1x32.

6.3.4 QTABS

Descriptor Tables, a 32Kx32 memory (Road Runner only).

6.3.5 VFIFO

This is the output of the Video FIFO. In DMA, the video channel selected will be determined by the MCHSEL in the attribute register.

In direct slave access, the channel the video channel selected will be determined by the SCHSEL bits in CON2. This mode is for testing purposes only.

Register Specifications

Chapter 7

7.1 Introduction

This section enumerates all of the bitfields in all of the registers on the Road Runner/R3-CL. All of the registers are 32 bits wide. These wide registers are named CON0, CON1, etc. Each registers is broken into one or more bitfields. Bitfields can be from one to 32 bits wide. Each bitfield controls a specific function on the board.

7.1.1 Bitfield definitions

What follows is an explanation of bitfield definition sections.

BITFIELD R/W, CON0[7..0

Bitfield explanation.

The definitions is broken into three sections.

Table 7-1

Section	Meaning
Bitfield name	This is the name of the bitfield. This name is use to program this bitfield from software or from within and camera configuration file. When programming bitfields from software using a Peek or Poke function, the bitfield is preceded with "REG_". For example the bitfield CFREQ is referred to in software as REG_CFREQ.
Bitfield details	This section describes how the bitfield is accessed. The first part describes the how the bits can be accessed. For example R/W means the register can be both read and write. See the next table for details. The second part is the wide register that the bitfield is located in. In the example above this bitfield is in CON0. The part in square brackets indicate which bits in this register the bitfield takes up. For example, [7..0] means the bitfield has 8 bits, and it occupies bit positions 0 to 7 in CON0.
Bitfield explanation	This section explains the purposed of the bitfield in detail. Usually meaning of every possible value of the bitfield is listed.

Table 7-2

Access	Meaning
R/W	Bitfield can be read and written.
RO	Bitfield can only be read. Writing to this bit has no effect.
WO	Bitfield can only be written. Reading from this bit will return meaningless values.

7.2 CON0 Register

Bit	Name
0	MUXA
1	MUXA
2	MUXA
3	Reserved
4	MUXB
5	MUXB
6	MUXB
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	SYNCSEL
13	SYNCSEL
14	SYNCSEL
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	CFREQ
20	CFREQ
21	CFREQ
22	XTL_7MHZ
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

MUXA

R/W, CON0[2..0]

These bits control the multiplexing of Channel A..

MUXA	Taps	Format	Function
0 (000b)	T0	1 → 4	8-bit pixels packed into 32-bit word
1 (001b)	T1,T0	2 → 4	16-bit pixels packed into 32-bit word
2 (010b)	T0,T1,T2	3 → 4	24-bit pixels transferred direct, T2 on LSByte, MSByte = 00h
3 (011b)	T2,T1,T0	3 → 4	24-bit pixels transferred direct, T0 on LSByte, MSByte = 00h
4 (100b)	T1,T0	2 → 4	10-bit pixels shifted by 2 and packed as 4 8-bit pixels
5 (101b)	T1,T0	2 → 4	12-bit pixels shifted by 4 and packed as 4 8-bit pixels
6 (110b)	T1,T0	2 → 4	14-bit pixels shifted by 6 and packed as 4 8-bit pixels
7 (111b)	Reserved code		

MUXB

R/W, CON0[6..4]

These bits control the multiplexing of Channel B.

MUXB	Taps	Format	Function
0 (000b)	T1	1 → 4	8-bit pixels packed into 32-bit word
1 (001b)	T2,T1	2 → 4	10-bit pixels shifted by 2 and packed as 4 8-bit pixels
2 (010b)	T2,T1	2 → 4	12-bit pixels shifted by 4 and packed as 4 8-bit pixels
3 (011b)	Reserved code		
4 (100b)	T2,T1	2 → 4	10-bit pixels packed into 32 bit word
5 (101b)	T2,T1	2 → 4	12-bit pixels packed into 32 bit word
6 (110b)	Reserved code		
7 (111b)	Reserved code		

SYNCSEL R/W, CON0[14..12]

These bits will select the source of the video and the control signals to the board from the Camera Link or from an on-board synthetic video generator. The latter is used for diagnostics and BIST (Built-In Self Test).

SYNCSEL	Function
0 (000b)	Camera uses VALID signal on CL connector.
1 (001b)	Camera does not use VALID signal on CL connector.

Some codes are for testing. They will disconnect controls and data from the Camera Link and generate synthetic controls (LEN, FEN), data and clock. The clock will be set by the CFREQ bits.

SYNCSEL	Function
2 (010b)	Ramp
3 (011b)	000000h
4 (100b)	FFFFFFh
5 (101b)	AAAAAAh
6 (110b)	555555h
7 (111b)	ABCDEFh

CFREQ R/W, CON0[21..19]

These bits control the frequency of CLCKOUT, the internal clock generator. This clock can drive a camera. It is also used for the BIST.

CFREQ	Frequency
0 (000b)	0
1 (001b)	2.5 MHz
2 (010b)	5 MHz
3 (011b)	7.5 MHz
4 (100b)	10 MHz
5 (101b)	15 MHz
6 (110b)	20 MHz
7 (111b)	30 MHz

XTL_7MHZ

RO, CON0[22]

This bit indicates if the 7 MHz crystal is mounted on the board or not. This crystal is used as a clock input to the board's UART. The 7 MHz crystal allows for more accurate frequencies at higher baud rates.

XTL_7MHz	Function
0	Regular UART crystal
1	7 MHz UART crystal

7.3 CON1 Register

Bit	Name
0	AQCMD
1	AQCMD
2	AQSTAT
3	AQSTAT
4	FACTIVE
5	FCOUNT
6	FCOUNT
7	FCOUNT
8	PLXGAREV
9	PLXGAREV
10	PLXGAREV
11	PLXGAREV
12	PLXGAREV
13	PLXGAREV
14	PLXGAREV
15	PLXGAREV
16	EFA
17	AEFA
18	FULL_A
19	HFA
20	Reserved
21	EFB
22	AEFB
23	FULL_B
24	TOPA
25	TOPB
26	FI
27	FIRSTFI
28	Reserved
29	Reserved
30	Reserved
31	Reserved

AQCMD R/W, CON1[1..0]

The AQCMD Control Bits.

AQCMD	Command
0 (00b)	FREEZE
1 (01b)	ABORT
2 (10b)	SNAP (field/frame, depending on AQFRM bit)
3 (11b)	GRAB continuously

The acquisition command can be written any time by the host. After it has been written, it will be latched by the first start of the frame (end of vertical blank).

The AQCM bit will read back the last command written, until it gets latched. After it gets latched, the AQCMD will read 11b for GRAB, (00b) for all other cases. The ACMD bits will give you the state of the next frame.

AQSTAT RO, CON1[3..2]

The Acquisition Status bits give information about the status of the current frame.

AQSTAT	Status
0 (00b)	FREEZE
1 (01b)	Reserved
2 (10b)	SNAP (field/frame, depending on AQFRM bit)
3 (11b)	GRAB continuously

FACTIVE RO, CON1[4]

Active video bit.

FACTIVE	Meaning
0	Camera is in the vertical inactive area (Vertical Blank)
1	Camera in the vertical active area

FCOUNT RO, CON1[7..5]

Frame counter. This 3-bit counter is incremented every frame. The register is incremented and the end of the frame, when the VCTAB column VEND, goes high.

PLXGAREV RO, CON1[15..8]
Firmware revision

EFA, EFB RO, CON1[16], CON1[21]
Video FIFO's empty flags.

EFA, EFB	Meaning
0	FIFO not empty
1	FIFO empty

AEFA, AEFB RO, CON1[17], CON1[22]
FIFO's Almost Empty Flags.

AEFA, AEFB	Meaning
0	Eight or more than eight entries in FIFO
1	Less than eight entries in the FIFO

TOPA/B RO, CON1[24], CON1[25]
Diagnostics bits

FI RO, CON1[26]
Field index. Definition of Odd/Even depends on the camera.

FIRSTFI RO, CON1[27]
This bit is the F1 bit latched at the beginning of the acquisition. This information is needed so that the software can correctly display an interlaced image.

7.4 CON2 Register

Bit	Name
0	GPOUT0
1	GPOUT1
2	GPOUT2
3	GPOUT3
4	GPOUT4
5	GPOUT5
6	SW
7	SW
8	GPIN2
9	NORLUTS
10	ADVANCE
11	BURST
12	BURST
13	FRZFIFO
14	GPIN1
15	QTBSRC
16	GRNTCTL
17	INT_DMA
18	INT0_CTAB
19	INT1_FIFO
20	INT2_HW
21	INT3_TRIG
22	INT4_SER
23	LTEOX
24	LTMBANK
25	LTMBYPASS
26	SCHSEL
27	LTMCHSEL
28	Reserved
29	LTFLUSH
30	CONTRIGINT
31	CONTRIGINT

GPOUT0	R/W, CON2[0]	This bit controls the state of the GPOUT0 general purpose output. The signals are RS422.
GPOUT1	R/W, CON2[1]	This bit controls the state of the GPOUT1 general purpose output. The signals are RS422.
GPOUT2	R/W, CON2[2]	This bit controls the state of the GPOUT2 general purpose output. The signals are RS422.
GPOUT3	R/W, CON2[3]	This bit controls the state of the GPOUT3 general purpose output. The signals are RS422.
GPOUT4	R/W, CON2[4]	This bit controls the state of the GPOUT4 general purpose output. The signals are RS422.
GPOUT5	R/W, CON2[5]	This bit controls the state of the GPOUT5 general purpose output. The signals are RS422.
SW	RO, CON2[7..6]	Two on-board mechanical switches used to identify multiple boards in the same system.
GPIN2	RO, CON2[8]	General purpose TTL input bit from pin 24 of the I/O connector.

NORLUTS

R/W, CON2[9]

This bit will disable the Read LUTs and set them in BYPASS for SLAVE and DMA modes. Useful in testing.

NORLUTS	Meaning
0	Read LUTs mode set by BYPASS SLAVE/DMA
1	Read LUTs always in BYPASS mode

ADVANCE

R/W, CON2[10]

This bit will enable to advance the Video FIFO pointer with each SLAVE read.

ADVANCE	Meaning
0	Read video FIFO in SLAVE mode without advancing the read pointer
1	Advance video FIFO read pointer after each SLAVE mode read

BURST

R/W, CON2[12..11]

These bits control the length of the burst on the PCI bus. The control circuitry will start the DMA when there are at least eight entries in the video FIFO. For video sources that are slow, this can result in inefficient transfers over the PCI, i.e., many short bursts. The BURST can insert additional waiting time after the video FIFO accumulated eight entries. During this time the video FIFO will receive more data. This will result in fewer, but longer bursts.

BURST	Delay
0 (00b)	No delay
1 (01b)	128 clocks
2 (10b)	256 clocks
3 (11b)	512 clocks

FRZFIFO

R/W, CON2[13]

This bit controls the video FIFOs and the DMA in case of overflow. The bit will be set by the rising edge of INT1, which is the overflow interrupt. It can be reset by the host by writing a 0. For the host to be able to reset this bit, the CMDWRITE code should be (010b). This will enable a reliable R/M/W operation.

When this bit gets set, the DMA operation will be inhibited. This will guarantee that no corrupted data from the FIFOs will be read. As an overflow interrupt has been issued, the application can gracefully recover. It can reset the DMA and acquisition registers and then restart. This is very useful for display.

Note: If the INT_DMA is not enabled, the DMA will not be stopped.

FRZFIFO	Meaning
0	Nor overflow occurred
1	Overflow occurred

GPIN1

RO, CON2[14]

General purpose TTL input bit from pin 11 of the I/O connector.

QTBSRC

RO, CON2[15]

Always reads back 1 to notify software to use QTABs on the host.

GRNTCTL

R/W, CON2[16]

This bit controls the bus grant on the local bus. It is used to flush video FIFO on continuous acquisition.

GRNTCTL	Meaning
0	Grant bus to PCI90x0 if FIFO according to FLUSH bit
1	Grant bus to PCI90x0 if FIFO not empty

INT_DMA

RO, CON2[17]

This bit is the DMA interrupt bit from the PLX9080. It can be reset in the PLX chip (see the PLX manual).

INT_DMA	Meaning
0	No DMA interrupt
1	DMA interrupt asserted

INT0_CTAB

R/W, CON2[18]

This interrupt will be generated from the CTABs. Host and source can set the interrupt if the ENINT0_CTAB (interrupt enable bit) is set. Host cannot clear the interrupt if the interrupt enable bit is not set. For the host to access the INT_CTAB bit, one more condition must be fulfilled: the CMDWRITE code must be set to 1, see description of CON4.

INT1_FIFO

R/W, CON2[19]

This interrupt will be generated by a FIFO overflow. Host and source can set the interrupt if the ENINT1_FIFO (interrupt enable bit) is set. Host cannot clear the interrupt if the interrupt enable bit is not set. For the host to access the INT1_FIFO bit, one more condition must be fulfilled: the CMDWRITE code must be set to 2, see description of CON4.

INT2_HW

R/W, CON2[20]

This interrupt will be generated by a HW exception (local bus time-out or loss of sync). Host and source can set the interrupt if the ENINT2_HW (interrupt enable bit) is set. Host cannot clear the interrupt if the interrupt enable bit is not set. For the host to access the INT_HW bit, one more condition must be fulfilled: the CMDWRITE code must be set to 3, see description of CON4.

INT3_TRIG

R/W, CON2[21]

This interrupt will be generated by the active edge of TRIGGER_A (rising/falling edge if TRIGPOL = 0/1). Host and source can set the interrupt if the ENINT3_TRIG (interrupt enable bit) is set. Host cannot clear the interrupt if the interrupt enable bit is not set. For the host to access the INT3_TRIG bit, one more condition must be fulfilled: the CMDWRITE code must be set to 4, see description of CON4.

INT4_SER

R/W, CON2[22]

This interrupt will be generated when the interrupt line from the UART is asserted. This line is ANDed with ENINT4_SER bit, i.e. the UART interrupt will pass through only if ENINT4_SER is set. For resetting the UART interrupt, the host has to access the interrupt registers in the UART.

SCHSEL R/W, CON2[26]

This bit will control the channel select during direct slave access to the FIFO. When manually reading data out of the FIFO (not DMAing), set this bit to get the data from the channel that you are interested in. When DMAing, this bit is set automatically by the current quad.

SCHSEL	Channel Selected
0	Channel A
1	Channel B

**LTEOX,
LTMBANK,
LTMBYPASS,
LTMCHSEL,
LTFLUSH**

RO, CON2[23], CON2[24], CON2[25], CON2[27], CON2[29]

These bits are the current LaTched attributes. The control circuitry snoops the local bus and intercepts the attributes when the 90x0 reads the DMA local address. These attributes are used for the current DMA process. The "LT" attributes are the ones that have been intercepted and latched in the control circuit. This is useful for testing and debugging.

CONTRIGINT R/W, CON2[31..30]

These bits will control the generation of the interrupt from the TRIGGERA:

CONTRIGINT	Function
0 (00b)	Generate interrupt at start of TRIGGERA
1 (01b)	Generate interrupt at end of TRIGGERA
2 (10b)	Generate interrupt at start and end of TRIGGERA
3 (11b)	Reserved

Note: For TRIGPOL = 0/1, the start of the trigger is the rising/falling edge.

7.5 CON3 Register

Bit	Name
0	Reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	CLKDRVSEL
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

CLKDRVSEL R/W, CON3[10]

Selects clock output type.

CLKDRVSEL	Meaning
0	Clock output is RS422
1	Clock output is LVDS

7.6 CON4 Register

Bit	Name
0	ENINT0_CTAB
1	ENINT1_FIFO
2	ENINT2_HW
3	ENINT3_TRIG
4	ENINT4_SER
5	CMDWRITE
6	CMDWRITE
7	CMDWRITE
8	VF_MRST
9	VFRESET
10	SEN
11	FWSI
12	LD_FIFO
13	Reserved
14	Reserved
15	FORCEABORT
16	SWRESET
17	SWTRIGB
18	SWTRIGA
19	AQFLD
20	AQFLD
21	Reserved
22	AQFRM
23	RST_SER
24	OVF
25	Reserved
26	Reserved
27	Reserved
28	TRIGAQCMD
29	HBYPASS
30	HBANK
31	Reserved

**ENINT0_CTAB,
ENINT1_FIFO,
ENINT2_HW,
ENINT3_TRIG,
ENINT4_SER**

R/W, CON4[0], CON4[1], CON4[2], CON4[3], CON4[4]

Individual enable interrupt bits related to the acquisition process. The interrupts can be active (from its source or from the host), only if it has been enabled.

ENINT0_CTAB	Meaning
0	INT_CTAB disabled
1	INT_CTAB enabled

ENINT1_FIFO	Meaning
0	INT_FIFO disabled
1	INT_FIFO enabled

ENINT2_HW	Meaning
0	INT_HW disabled
1	INT_HW enabled

ENINT3_TRIG	Meaning
0	INT_TRIG disabled
1	INT_TRIG enabled

ENINT4_SER	Meaning
0	UART interrupt disabled
1	UART interrupt enabled

CMDWRITE

R/W, CON4[7..5]

These bits will enable the host to write to interrupt bits that can be modified by on-board circuitry. Disabling the host access to those bits will allow reliable read-modify-write operations to control bits residing in the same control register.

The AQCMD bits do not need a CMDWRITE protection feature because they are in a register by themselves.

The logic of the interrupt generated by the PCI9080 DMA is such that it does not need the features of the CMDWRITE bits. There is a dedicated bit for resetting the interrupt.

There is no need for CMDWRITE protection for the INT4_SER. This interrupt is generated in the UART.

The advantage of having the CMDWRITE logic is that the host cannot only reset the interrupt source, it can also set the interrupt source. This is a handy tool for debugging and simulating hardware interrupts from the software.

CMDWRITE	Host access enabled to
0 (000b)	None
1 (001b)	INT_CTAB
2 (010b)	INT_FIFO
3 (011b)	INT_HW
4 (100b)	INT_TRIG

VF_MRST

WO, CON4[8]

Video FIFO Master Reset.

This bit will always read back 0.

VF_MRST	Meaning
0	Assert Master Reset to all FIFOs
1	No effect

VFRESET

WO, CON4[9]

Video FIFO Partial Reset.

This bit will always read back 0.

VFRESET	Meaning
0	Assert partial reset to all FIFOs
1	No effect

SEN

R/W, CON4[10]

Reserved for FIFO programming

FWSI

R/W, CON4[11]

FIFO controls

LD_FIFO R/W, CON4[12]

FIFO controls

FORCEABORT R/W, CON4[15]

Used when resetting DMA engine when the FIFOs are empty.

SWRESET WO, CON4[16]

This is a SW reset to the board. It should be asserted each time the board is initialized. It resets the CTABs address counters and the timing generators.

This bit will always read back 0.

SWRESET	Meaning
0	No effect
1	Asserts SW reset

SWTRIGB WO, CON4[17]

This bit is to the software trigger B. It is always active HI and will be ORed with the TRIGGER_B. It does not have to be enabled by the TRIGAQ bit. Writing a 1 will unconditionally generate a trigger B to the board.

This bit will read back 0 for TRIGCON = 0,1,3. or TRIGCON = 2 it will be level sensitive and it will be treated as R/W.

SWTRIGA WO, CON4[18]

This bit is the software trigger A. It is always active HI, and will be ORed with the external TRIGGER_A. The SWTRIGA does not have to be enabled by the TRIGAQ bit. Writing a 1 will unconditionally generate a trigger A to the board.

This bit will always read back 0 if TRIGCON = 0, 1, 3.

This bit has an additional mode of operation, continuous acquisition, for TRIGCON = 2. In this mode SWTRIGA is level sensitive. Data will be acquired as long as SWTRIGA is asserted. When SWTRIGA is latched, it will read back the last value written to. For the continuous acquisition case, TRIGGERB will also be treated as "level sensitive."

AQFLD R/W, CON4[20..19]

Controls acquisition for interlaced cameras.

AQFLD	Acquire
0 (00b)	Start acquire on next field
1 (01b)	Start acquire on odd field
2 (10b)	Start acquire on even field
3 (11b)	Reserved

AQFRM R/W, CON4[22]

Acquire one or two fields.

AQFRM	Meaning
0	Acquire two fields
1	Acquire one field

RST_SER R/W, CON4[23]

Writing a one to this bitfield will reset the UART.

OVF RO, CON4[24]

This is the video FIFO's overflow bit. This bit is not latched. If the overflow condition goes away, the OVF bit will be de-asserted.

OVF	Meaning
0	No overflow in any of the FIFO channels
1	Overflow in one or more of the FIFO channels.

TRIGAQCMD R/W, CON4[28]

Set for triggered acquisition mode. After the rising/falling edge of the selected trigger, the current acquisition command will be executed. TRIGCON select the trigger, TRIGPOL will select the polarity.

TRIGAQCMD	Meaning
0	Normal acquisition mode (non-triggered)
1	Triggered acquisition mode

HBYPASS R/W, CON4[29]

This bit will enable to bypass the read LUT when reading the FIFO in Direct Slave mode.

HBYPASS	Meaning
0	Data from FIFOs will pass through the LUT
1	Data from FIFOs will bypass the LUT

HBANK R/W, CON4[30]

This bit will select the LUT bank, when the LUTs are accessed from the host. In master mode, the bank is controlled by the BANK in the attribute register.

HBANK	Meaning
0	Select bank 0
1	Select bank 1

7.7 CON5 Register

Bit	Name
0	CTABHOLD
1	HSTOP
2	VSTOP
3	TRIGCON
4	TRIGCON
5	ENCDRCON
6	ENCDRCON
7	ENHLOAD
8	VTRIGRST
9	AACTIVE
10	BACTIVE
11	TRIG_POL
12	ENC_POL
13	LEN_SIZE
14	TRIGASTOP
15	ENXTRIG
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

CTABHOLD R/W, CON5[0]

This bit will freeze the outputs of the CTABs. When the CTABs are frozen, both the horizontal and the vertical counter will not increment or respond to any external signals, and the CT0, CT1, CT2 and STROBE outputs will be held at their current levels.

CTABHOLD	Meaning
0	CTABs run
1	CTABs frozen, output of CTABs fixed

HSTOP R/W, CON5[1]

This bit will determine if the HCTAB address counter will free run. If HSTOP = 1, the HCOUNT will be reset by the HRESET signal and stay at 0000h until an ENCODER signal is asserted.

HSTOP	Meaning
0	HCOUNT is free running
1	After HCOUNT is reset, wait for an ENCODER to continue

VSTOP R/W, CON5[2]

This bit will determine if the VCTAB address counter will free run. If VSTOP is 1, the VCOUNT will be reset by the reset signal (VSTART=VEND=1) and stay at 0 until a TRIGGER comes in. If VSTOP is 0, the VCOUNT will be free running.

VSTOP	Meaning
0	VCOUNT is free running
1	After VCOUNT is reset, wait for TRIGGER to continue

TRIGCON R/W, CON5[4..3]

These bits control the operation of the two triggers, A and B.

TRIGCON	Function
0 (00b)	TRIGGERA active, B disabled
1 (01b)	Start on A, stop on B
2 (10b)	Continuous acquisition
3 (11b)	Triggered Termination

Triggered Termination (start stop triggering mode), TRIGCON = 11b

This mode is used to terminate the acquisition immediately, without waiting for the end of the frame. It is slightly different than the assertion of the ABORT command. It will end the acquisition process and the DMA process in a graceful way. Acquisition can be started normally with a GRAB command, or triggered by TRIGGERA. The acquisition can be terminated by TRIGGERA or TRIGGERB, depending on the setting of bit TRIGASTOP.

In Triggered Termination when the termination trigger is asserted, the VCOUNT will be reset to 0. The CTABs should be programmed to issue an interrupt. This interrupt should be programmed to abort the DMA process. It will not abort the acquisition process (in contrast to the ABORT command).

ENCDRCON

R/W, CON5[6..5]

These bits control the operation of the two encoders, A and B.

ENCDRCON	Function
0 (00b)	Encoder A active, B disabled
1 (01b)	TBD
2 (10b)	TBD
3 (11b)	TBD

ENHLOAD

R/W, CON5[7]

This bit will determine if the HCTAB address counter, HCOUNT, should be loaded by LEN. Useful for cameras that do not give back LEN.

ENHLOAD	Meaning
0	HCOUNT is free running
1	LEN will load the HCOUNT

VTRIGRST

R/W, CON5[8]

This bit will enable the trigger to reset the VCTAB address counter. Do not use this mode when VSTOP = 1 or TRIGAQCMD = 1 since these modes give the board conflicting instructions.

VTRIGRST	Meaning
0	Normal mode
1	VCTAB counter reset when trigger asserts

**AACTIVE,
BACTIVE,**

R/W, CON5[9], CON5[10]

These bits mark the active video channels. Video will be clocked only in the FIFOs that are active.

AACTIVE, BACTIVE	Meaning
0	Channel inactive, no video will be clocked in that FIFO
1	Channel active, video will be clocked in that FIFO

TRIGPOL

R/W, CON5[11]

Polarity of TRIGGERA input.

TRIGPOL	Meaning
0	Trigger asserted on rising edge
1	Trigger asserted on falling edge

ENCPOL

R/W, CON5[12]

Polarity of ENCODERA input.

ENCPOL	Meaning
0	Encoder asserted on rising edge
1	Encoder asserted on falling edge

LEN_SIZE

R/W, CON5[13]

Enable LEN is size mode. In this mode, the HAW is dictated only by LEN. Every pixel is acquired when LEN is asserted, the CTABs are ignored.

LEN_SIZE	Meaning
0	HAS controlled by CTABs
1	HAW controlled by LEN

TRIGASTOP R/W, CON5[14]

This bit controls how acquisition is terminated in when TRIGCON = 3, triggered termination mode.

TRIGASTOP	Meaning
0	Terminate acquisition assertion of TRIGGERB
1	Terminate acquisition with falling edge of TRIGGERA

In the second mode, termination by falling edge of TRIGGERA, TRIGGERA can still be used to start the acquisition on its rising edge. In this mode, the TRIGGERA has to be an active high level signal.

The difference between the two modes is that with TRIGASTOP = 1, the start and the end of the acquisition process can be controlled by a single wire, TRIGGERA.

ENXTRIG R/W, CON5[15]

This bit will enable the external hardware TRIGGERA and B. The purpose of this signal is to be able to shut off these triggers. The external triggers come in through differential receivers. If no signal is connected, the inputs are floating and the output of the receiver is unpredictable.

ENXTRIG	Meaning
0	External TRIGGERA and TRIGGERB disabled
1	External TRIGGERA and TRIGGERB enabled

7.8 CON6 Register

Bit	Name
0	LENPOL
1	FENPOL
2	TRIM
3	TRIM
4	TRIM
5	TGAQW
6	TGAQW
7	FISEL
8	RD_TRIGA
9	RD_TRIGB
10	RD_ENCA
11	RD_ENCB
12	reserved
13	HSTICK
14	RANDOM_FEN
15	BY16
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

LENPOL

R/W, CON6[0]

Polarity of LEN input.

LENPOL	Meaning
0	LEN asserted on rising edge
1	LEN asserted on falling edge

FENPOL

R/W, CON6[1]

Polarity of FEN input.

FENPOL	Meaning
0	FEN asserted on rising edge
1	FEN asserted on falling edge

TRIM

R/W, CON6[4..2]

The horizontal CTABs and the control functions work off PCLK/4. If the data from the camera is misaligned relative to LEN by one, two, or three pixels, we will not be able to correct that with the CTABs. This will have adverse effect especially for multi-tap cameras. TRIM will delay LEN relative to the data by one to seven pixels.

TRIM	Delay
0 (000b)	No delay
1 (001b)	One pixel
2 (010b)	Two pixels
3 (011b)	Three pixels
4 (100b)	Four pixels
5 (101b)	Five pixels
6 (110b)	Six pixels
7 (111b)	Seven pixels

TGAQW

R/W, CON6[6..5]

These bits define the width of the data send to the FIFOs

Note: The special MUXA firmware must be used for 24 bits packed mode.

TGAQW	Width
0 (00b)	8 bits
1 (01b)	16 bits
2 (10b)	32 bits
3 (11b)	24 bits packed

FISEL

R/W, CON6[7]

Selects between the FI supplied by the camera or the FI detected by the CTABs.

FISEL	Meaning
0	Select FI from camera
1	Select FI from CTAB detector

RD_TRIGA

RO, CON6[8]

This bit reflects the value of the TRIGGERA input on the I/O connector.

RD_TRIGB

RO, CON6[9]

This bit reflects the value of the TRIGGERB input on the I/O connector.

RD_ENCA

RO, CON6[10]

This bit reflects the value of the ENCODERA input on the I/O connector.

RD_ENCB

RO, CON6[11]

This bit reflects the value of the ENCODERB input on the I/O connector.

HSTICK R/W, CON6[13]

This bit controls the behavior of the Horizontal Control Table. If the horizontal counter is clocked up to 2040 before LEN is asserted, it will stop, if this bit is set. The horizontal counter will remain at 2040 until LEN is asserted.

HSTICK	Meaning
0	Normal operation
1	Horizontal counter sticks at 2040 until LEN is asserted

RANDOM_FEN R/W, CON6[14]

Controls how horizontal counter (HCOUNT) and HAW is reset when FEN asserts. The purpose is to synchronize the horizontal timing to a know state when FEN asserts.

RANDOM_FEN	Meaning
0	Normal operation, assertion of FEN does not effect HCOUNT
1	When FEN asserts, HCOUNT is reset to 0000h and HAW is reset.

BY16 R/W, CON6[15]

Controls how the horizontal counter (HCOUNT) is incremented.

BY16	Meaning
0	HCOUNT is incremented every four PCLKs
1	HCOUNT is incremented every sixteen PCLKs

7.9 CON7 Register

Bit	Name
0	Reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

7.10 CON8 Register

Bit	Name
0	CATBREV
1	CTABREV
2	CTABREV
3	CTABREV
4	CTABREV
5	CTABREV
6	CTABREV
7	CTABREV
8	FEN_IS_TRIG
9	LAL
10	TRIG_FREE
11	ENC_FILTER
12	Reserved
13	Reserved
14	FEN_SIZE
15	VSTICK
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

CTABREV RO, CON8[7..0]

Firmware revision.

FEN_IS_TRIG R/W, CON8[8]

Setting this register routes the signal coming into the board's FEN input (on the CL connector) to the trigger circuit.

FEN_IS_TRIG	Meaning
0	TRIGGER input controls trigger circuit
1	FEN input controls trigger circuit

LAL R/W, CON8[9]

Controls what data is read from the VCOUNT register.

LAL	Meaning
0	Read instantaneous VCOUNT value
1	On triggered termination, VCOUNT will hold the value of last acquired line, before the VCOUNT was reset to 0. Value is held until the end of the next frame.

TRIG_FREE R/W, CON8[10]

The TRIG_FREE bit controls the triggering for triggered termination.

TRIG_FREE = 0

In this mode, the acquisition will start with TRIGGERA. When the VCOUNT reaches the end of the buffer, it will be reset. The VCOUNT will stay at 0000h until another trigger is asserted.

The acquisition in this mode can be terminated with the falling edge of TRIGGERA or TRIGGERB.

The data buffer can not be overwritten if there is no new TRIGGERA.

TRIG_FREE = 1 (a.k.a. Luggage Mode)

In this mode, the acquisition will start with TRIGGERA. When VCOUNT reaches the end of the buffer, it will be reset. VCOUNT will not stick at 000, it will keep on counting. New data will overwrite the old data in the buffer. The acquisition will stop only on TRIGGERB or falling edge of TRIGGERA, or FREEZE command.

TRIG_FREE	Meaning
0	Trigger edge mode, when trigger asserts one frame is acquired
1	Trigger level mode, frames continuously acquired while trigger is asserted

ENC_FILTER R/W, CON8[11]

Enable a noise filter on the encoder input.

ENC_FILTER	Meaning
0	Encoder input not filtered
1	Encoder input filtered

FEN_SIZE R/W, CON8[14]

When this bit is set, the FEN signal, and only the FEN signal, is used by the board to determine the VAW. The method of acquisition differs from the standard in that the CTABs are not used to determine which lines to acquire and which lines to ignore. The CTABs are used only to control the CT outputs. The only caveat is that the VAW will be the exact size of the FEN. In other words, the board will acquire all lines when FEN is asserted. This is contrasted when normal acquisition where any window of lines can be acquired by programming the CTABs. The primary purpose for this mode is to overlap acquisition of the frame and exposure control of the camera from the board.

VSTICK	Meaning
0	Normal acquisition, CTABs control VAW
1	Overlapping acquisition, FEN controls VAW

VSTICK R/W, CON8[15]

This register controls what happens with the vertical control table counter (VCOUNT) reaches 4088. Normally the counter jumps to 4096 when the FEN is asserted. However, under certain circumstances (e.g., long exposure periods) there may be more

than 4096 lines before FEN is asserted. In these cases, set this bit so that the counter will stick indefinitely at 4088 until FEN is asserted. Write a 1 to SWRESET will also unstick the counter, but in this case the counter will jump to 0.

VSTICK	Meaning
0	VCOUNT runs normally
1	VCOUNT sticks at 4088 (ff8h) until FEN asserts

7.11 CON9 Register

Bit	Name
0	TRIG_DELAY
1	TRIG_DELAY
2	TRIG_DELAY
3	TRIG_DELAY
4	TRIG_DELAY
5	TRIG_DELAY
6	TRIG_DELAY
7	TRIG_DELAY
8	TRIG_DELAY
9	TRIG_DELAY
10	EN_TRIG_DELAY
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

TRIG_DELAY R/W, CON9[9..0]

These bits determine by how many lines to delay the external TRIGGERA. When the delay is enabled, EN_TRIG_DELAY = 1, the total delay is given by $8 \times N + 3$, where N is the number programmed in TRIGGER_DELAY.

Note: If TRIGGER_DELAY = 0, there will be a delay of three lines.

EN_TRIG_DELAY R/W, CON9[10]

This bit will enable the delay line for the TRIGGERA.

EN_TRIG_DELAY	Meaning
0	Delay disabled
1	Delay enabled

7.12 CON10 Register

Bit	Name
0	CC1_CON
1	CC1_CON
2	CC1_CON
3	CC2_CON
4	CC2_CON
5	CC2_CON
6	CC3_CON
7	CC3_CON
8	CC3_CON
9	CC4_CON
10	CC4_CON
11	CC4_CON
12	HTRIM
13	HTRIM
14	HTRIM
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

CC1_CON R/W, CON10[2..0]

These bits control the functionality of camera control bit CC1 on the Camera Link connector.

CC1_CON	CC1 Output
0 (000b)	CT0
1 (001b)	CT1
2 (010b)	CT2
3 (011b)	STROBE
4 (100b)	CLOCK
5 (101b)	0
6 (110b)	1
7 (111b)	reserved

CC2_CON R/W, CON10[5..3]

These bits control the functionality of camera control bit CC2 on the Camera Link connector.

CC2_CON	CC2 Output
0 (000b)	CT0
1 (001b)	CT1
2 (010b)	CT2
3 (011b)	STROBE
4 (100b)	CLOCK
5 (101b)	0
6 (110b)	1
7 (111b)	reserved

CC3_CON

R/W, CON10[8..6]

These bits control the functionality of camera control bit CC3 on the Camera Link connector.

CC3_CON	CC3 Output
0 (000b)	CT0
1 (001b)	CT1
2 (010b)	CT2
3 (011b)	STROBE
4 (100b)	CLOCK
5 (101b)	0
6 (110b)	1
7 (111b)	reserved

CC4_CON

R/W, CON10[11..9]

These bits control the functionality of camera control bit CC4 on the Camera Link connector.

CC4_CON	CC4 Output
0 (000b)	CT0
1 (001b)	CT1
2 (010b)	CT2
3 (011b)	STROBE
4 (100b)	CLOCK
5 (101b)	0
6 (110b)	1
7 (111b)	reserved

HTRIM

R/W, CON10[14..12]

These bits control the delay between LEN asserting and the actual start of the horizontal active region. This adjustment is only needed under very special circumstances

HTRIM	Delay (in pixel clocks)
0 (000b)	0
1 (001b)	1
2 (010b)	2
3 (011b)	3
4 (100b)	4
5 (101b)	5
6 (110b)	6
7 (111b)	7

7.13 CON11 Register

Bit	Name
0	Reserved
1	Reserved
2	Reserved
3	Reserved
4	Reserved
5	Reserved
6	Reserved
7	Reserved
8	Reserved
9	Reserved
10	Reserved
11	Reserved
12	Reserved
13	Reserved
14	Reserved
15	Reserved
16	CFGDATA
17	CFGSTATUS
18	CFGGEN
19	CFGDDNL
20	CFGCLOCK
21	Reserved
22	OCCNFIG
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

Note: These bits are used to download the firmware in the gate arrays. These should not be used in any user application.

CFGDATA WO, CON11[16]

This bit is the serial configuration data written to the gate arrays.

On read back this bit will always read 0.

CFGDATA	Meaning
0	Data presented to gate arrays will be 0
1	Data presented to gate arrays will be 1

CFGSTATUS RO, CON11[17]

This bit is the STATUS bit of the first Altera device.

CFGGEN RO, CON11[18]

This bit is the CONFIG bit, an open collector. Through this bit you can read the gate arrays CONFIG bit. Asserting this bit will be done through the OCCNFIG bit (see below).

CFGGEN	Meaning
0	Gate arrays are reset
1	Gate arrays active

CFGDONE RO, CON11[19]

This read-only bit is the DONE bit of the last gate array in the configuration chain.

CFGDONE	Meaning
0	Configuration completed
1	Configuration in process

CFGLOCK WO, CON11[20]

This is the configuration clock. Writing a 1 to this bit will generate one single clock pulse to the gate arrays.

On read back this bit will always read 0.

CFGLOCK	Meaning
0	Clock will not be asserted
1	Clock will be asserted

OCCNFIG

R/W, CON11[22]

This bit controls the open collector feature of the CONFIG bit.

OCCNFIG	Meaning
0	CONFIG pin is input mode only, i.e., read-only
1	CONFIG pin is in output mode, asserted 0

7.14 CON12 Register

Bit	Name
0	VCOUNT
1	VCOUNT
2	VCOUNT
3	VCOUNT
4	VCOUNT
5	VCOUNT
6	VCOUNT
7	VCOUNT
8	VCOUNT
9	VCOUNT
10	VCOUNT
11	VCOUNT
12	VCOUNT
13	VCOUNT
14	VCOUNT
15	Reserved
16	Reserved
17	Reserved
18	Reserved
19	Reserved
20	Reserved
21	Reserved
22	Reserved
23	Reserved
24	Reserved
25	Reserved
26	Reserved
27	Reserved
28	Reserved
29	Reserved
30	Reserved
31	Reserved

VCOUNT

RO, CON12[14..0]

This register indicates the current value of the vertical control table counter.

Connector Pin-Outs

Chapter 8

8.1 Description

There are three connectors on the Road Runner/R3 CL, PCI version:

P1, test connector, 30-pin (not populated on production models)

P2, the service connector, 26-pin

P3, the Camera Link connector, 26-pin

There are also three, slightly different, connectors on the Road Runner/R3 CL, PMC version:

P3, the Camera Link connector, 26-pin

CON5, the first I/O connector

CON6, the second I/O connector

The Camera Link connector is mounted on the bracket.

The service connector is mounted on the top of the board. It has general purpose I/O and control signals that are not directly related to cameras. These signals usually do not have to be on the same cable that connects to a camera. A flat cable will allow to bring out those signals to 25 pin DSUB installed on a PCI bracket.

On the service connector, all the differential inputs are LVDS (RS644). They can be driven by either RS422 or LVDS drivers.

All the differential drivers are RS422. They can drive RS422 or LVDS receivers.

Con5 and Con6 are reduced version of the P2 connector on the PCI board.

8.2 P1, Test Connector, 30-Pin

Table 8-1 shows the pin out for the P1 connector.

Table 8-1 P1 Test Connector

Pin	Signal
1	GND
2	CLK-CL
3	GND
4	LEN
5	FEN
6	VALID
7	DATA_0
8	SPARE
9	CC1
10	CC2
11	CC3
12	CC4
13	GND
14	TRIGGERA+
15	TRIGGERA-
16	TIGGERA_OUT
17	ENCODERA+
18	ENCODERA-
19	ENCODERA_OUT
20	GND
21	RD_FIFO
22	WR_FIFO
23	TEST_PLX
24	TEST_CTAB
25	NC
26	GND
27	SOUT-RS232
28	GND
29	SIN-RS232
30	GND

8.3 P2, I/O Connector, 26-Pin

Table 8-2 shows the pin out for the P2 connector.

Table 8-2 P2, the I/O Connector

Pin	I/O	Signal	Comment
1	In	TRIGGERA+	LVDS/TTL
2	In	TRIGGERA-	LVDS/TTL
3	In	TRIGGERB+	LVDS/TTL
4	In	TRIGGERB-	LVDS/TTL
5	In	ENCODERA+	LVDS/TTL
6	In	ENCODERA-	LVDS/TTL
7	In	ENCODERB+	LVDS/TTL
8	In	ENCODERB-	LVDS/TTL
9	Out	CC4+	RS422
10	Out	CC4-	RS422
11	In	GPIN1	TTL
12	Out	GPOUT0+	RS422
13	Out	GPOUT0-	RS422
14	Out	GPOUT1+	RS422
15	Out	GPOUT1-	RS422
16	Out	GPOUT2+	RS422
17	Out	GPOUT2-	RS422
18	Out	GPOUT3+	RS422
19	Out	GPOUT3-	RS422
20	Out	GPOUT4+	RS422
21	Out	GPOUT4-	RS422
22	Out	GPOUT5+	RS422
23	Out	GPOUT5-	RS422
24	In	GPIN2	TTL
25	Out	GND	
26	Out	GND-	

8.4 P3, Camera Link Connector, 26-Pin

Table 8-3 shows the pin out for the P3 connector.

Table 8-3 P3 Camera Link Connector

Pin	I/O	Signal
1		GND
2	Out	CC4-
3	Out	CC3+
4	Out	CC2-
5	Out	CC1+
6	In	SER_RCV+
7	Out	SER_XMT-
8	In	DATA3+
9	In	CLK+
10	In	DATA2+
11	In	DATA1+
12	In	DATA0-
13		GND
14		GND
15	Out	CC4+
16	Out	CC3-
17	Out	CC2+
18	Out	CC1-
19	In	SER_RCV-
20	Out	SER_XMT+
21	In	DATA3-
22	In	CLK-
23	In	DATA2-
24	In	DATA1-
25	In	DATA0-
26		GND

8.5 Con5, The First PMC I/O Connector, 5-Pin

Table 8-4 shows the pin out for the Con5 connector.

Table 8-4 The Con5 Connector

Pin	I/O	Signal	Comment
1		GND	TTL
2	In	GPIN1	TTL
3	In	GPIN0	TTL
4	Out	GPOUT0	TTL
B	Out	GPOUT1	TTL

8.6 Con6, The Second PMC I/O Connector, 6-Pin

Table 8-5 shows the pin out for the Con6 connector.

Table 8-5 The Con6 Connector

Pin	I/O	Signal	Comment
1	In	ENCODER-	LVDS
2	Out	STROBE+	RS422
3	Out	STROBE-	RS422
4	In	TRIGGER+	LVDS
5	In	TRIGGER-	LVDS
6	In	ENCODER+	LVDS

A

AACTIVE R2/R3CL-7-27
 ADVANCE R2/R3CL-7-12
 AEFA,AEFB R2/R3CL-7-9
 AQCMD R2/R3CL-7-8
 AQFLD R2/R3CL-7-22
 AQFRM R2/R3CL-7-22
 AQSTAT R2/R3CL-7-8

B

BACTIVE R2/R3CL-7-27
 BITFIELDNAME R2/R3CL-7-1
 BURST R2/R3CL-7-12
 BY16 R2/R3CL-7-32

C

CC1_CON R2/R3CL-7-41
 CC2_CON R2/R3CL-7-41
 CC3_CON R2/R3CL-7-42
 CC4_CON R2/R3CL-7-42
 CFGCLOCK R2/R3CL-7-45
 CFGDATA R2/R3CL-7-45
 CFGDONE R2/R3CL-7-45
 CFGGEN R2/R3CL-7-45
 CFGSTATUS R2/R3CL-7-45
 CFREQ R2/R3CL-7-5
 CLKDRVSEL R2/R3CL-7-17
 CMDWRITE R2/R3CL-7-19
 Con5 R2/R3CL-8-5
 Con6 R2/R3CL-8-6
 CONTRIGINT R2/R3CL-7-15
 CTABHOLD R2/R3CL-7-25
 CTABREV R2/R3CL-7-35
 CTABS R2/R3CL-6-4

E

EFA,EFB R2/R3CL-7-9
 EN_TRIG_DELAY R2/R3CL-7-39
 ENC_FILTER R2/R3CL-7-36
 ENCDRCON R2/R3CL-7-26
 ENCPOL R2/R3CL-7-27
 ENHLOAD R2/R3CL-7-26
 ENINT₀_CTAB R2/R3CL-7-19

ENINT₁_FIFO R2/R3CL-7-19
 ENINT₂_HW R2/R3CL-7-19
 ENINT₃_TRIG R2/R3CL-7-19
 ENINT₄_SER R2/R3CL-7-19
 ENXTRIG R2/R3CL-7-28

F

FACTIVE R2/R3CL-7-8
 FCOUNT R2/R3CL-7-8
 FEN_IS_TRIG R2/R3CL-7-35
 FEN_SIZE R2/R3CL-7-36
 FENPOL R2/R3CL-7-30
 FI R2/R3CL-7-9
 FIRSTFI R2/R3CL-7-9
 FISEL R2/R3CL-7-31
 FORCEABORT R2/R3CL-7-21
 FRZFIFO R2/R3CL-7-12
 FWSI R2/R3CL-7-20

G

GPIN₁ R2/R3CL-7-13
 GPIN₂ R2/R3CL-7-11
 GPOUT₀ R2/R3CL-7-11
 GPOUT₁ R2/R3CL-7-11
 GPOUT₂ R2/R3CL-7-11
 GPOUT₃ R2/R3CL-7-11
 GPOUT₄ R2/R3CL-7-11
 GPOUT₅ R2/R3CL-7-11
 GRNTCTL R2/R3CL-7-13

H

HBANK R2/R3CL-7-23
 HBYPASS R2/R3CL-7-23
 HSTICK R2/R3CL-7-32
 HSTOP R2/R3CL-7-25
 HTRIM R2/R3CL-7-43

I

INT_DMA R2/R3CL-7-13
 INT₀_CTAB R2/R3CL-7-14
 INT₁_FIFO R2/R3CL-7-14
 INT₂_HW R2/R3CL-7-14
 INT₃_TRIG R2/R3CL-7-14

INT4_SER R2/R3CL-7-14

L

LAL R2/R3CL-7-35
LD_FIFO R2/R3CL-7-21
LEN_SIZE R2/R3CL-7-27
LENPOL R2/R3CL-7-30
LTEXO R2/R3CL-7-15
LTF LUSH R2/R3CL-7-15
LTMBANK R2/R3CL-7-15
LTMBYPASS R2/R3CL-7-15
LTMCHSEL R2/R3CL-7-15
LUTADDR R2/R3CL-6-4
LUTDATA R2/R3CL-6-4

M

MUXA R2/R3CL-7-4
MUXB R2/R3CL-7-4

N

NORLUTS R2/R3CL-7-12

O

OCCNFIG R2/R3CL-7-46
OVF R2/R3CL-7-22

P

P1 R2/R3CL-8-2
P2 R2/R3CL-8-3
P3 R2/R3CL-8-4
PLXGAREV R2/R3CL-7-9
PMC Version R2/R3CL-1-4

Q

QTABS R2/R3CL-6-4
QTBSRC R2/R3CL-7-13

R

RANDOM_FEN R2/R3CL-7-32
RD_ENCA R2/R3CL-7-31
RD_ENCB R2/R3CL-7-31
RD_TRIGA R2/R3CL-7-31
RD_TRIGB R2/R3CL-7-31
RST_SER R2/R3CL-7-22

S

SCHSEL R2/R3CL-7-15
SEN R2/R3CL-7-20
Specifications R2/R3CL-5-1
SW R2/R3CL-7-11
SWRESET R2/R3CL-7-21
SWTRIGA R2/R3CL-7-21
SWTRIGB R2/R3CL-7-21
SYNCSEL R2/R3CL-7-5

T

TGAQW R2/R3CL-7-30
TOPA,TOPB R2/R3CL-7-9
TRIG_DELAY R2/R3CL-7-39
TRIG_FREE R2/R3CL-7-35
TRIGAQCMD R2/R3CL-7-23
TRIGASTOP R2/R3CL-7-28
TRIGCON R2/R3CL-7-25
TRIGPOL R2/R3CL-7-27
TRIM R2/R3CL-7-30

V

VCOUNT R2/R3CL-7-48
VF_MRST R2/R3CL-7-20
VFRESET R2/R3CL-7-20
VSTICK R2/R3CL-7-36
VSTOP R2/R3CL-7-25
VTRIGRST R2/R3CL-7-26

X

XTL_7MHZ R2/R3CL-7-6

