XTC 2 Memory Space

Ryan Mokos

University of Illinois, Urbana-Champaign

Version 1.5 June 3, 2005

1 Introduction

This note is intended to be a stand-alone document describing, in detail, the memory/register space of the XTC 2 TDC mezzanine card. This information will be included later in a note detailing the functionality of the entire card.

Note: All 6-bin window number references in this note are based on the range 0 to 5 (as opposed to 1 to 6), and all not-sure window number references are based on the range 1 to 5. This is illustrated in Figure 1.

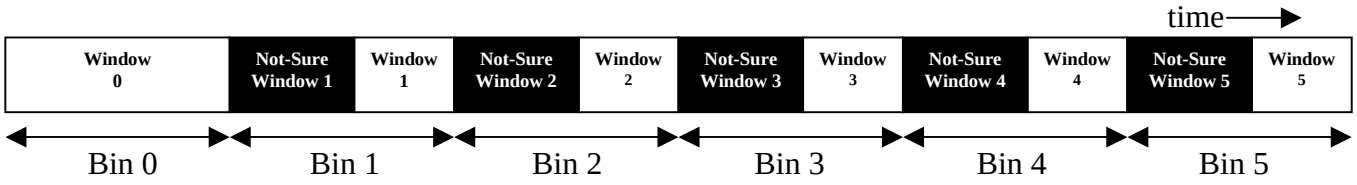

Figure 1: This diagram shows how all six time bins are laid out and illustrates the numbering scheme used in this document for normal windows and not-sure windows.

2 Overview

The VME bus consists of 13 signals, 8 of which are bidirectional data lines. These signals interface with the TDC through mezzanine connectors 2 and 3 (J1 and J2 on the XTC 2 PCBs). They connect to the FPGA-Programmer CPLD (U23 on the XTC 2 PCBs), and from there they are routed to the Kitchen Sink FPGA (U2 on the XTC 2 PCBs).

The XTC 2 card contains 64 registers, most of which are located in the Kitchen Sink FPGA. Tables 1 and 2 contain summaries of the entire register space. All of these registers are accessible through VME bus reads and writes. The address space is 0x02041Cyy, where yy represents all values between 0x00 and 0xFC that are divisible by 4. The lowest two bits of the yy byte are not passed to the XTC card, so the card sees address values 0x00 through 0x3F. When reading and writing data, only the most significant 8 bits of each 32-bit integer are used. As a result, bit shifting data values in the C code is necessary.

	Address		R/W	First Version			Initial Value		Function
	VME	XTC		2-Bin	6-Bin	CPLD	2-Bin	6-Bin	
CPLD Registers	xFC	(x3F)	W	---	---	0	N/A		Erase Flash Memory (any data value may be written)
	xF8	(x3E)	W	---	---	0	N/A		Reset Flash Memory Address (any data value may be written)
	xF4	(x3D)	W	---	---	0	N/A		Write Next Byte to Flash Memory (data = byte to be written)
	xF0	(x3C)	W	---	---	0	N/A		Flash Programming Complete (any data value may be written; resets flash memory address and configures FPGAs)
	xEC	(x3B)	R	---	---	0	mem(0)		Read Next Byte in Flash Memory
	xE8	(x3A)	R	---	---	0	xFF		Read FPGA Configuration & Flash Memory Program/Erase Status (7:4): FPGA Configuration (x0 = configured, xF = not configured) (3:0): Flash Status (x0 = ready, xF = busy)
	xE4	(x39)	W	---	---	1	N/A		Reset FPGA_Prog CPLD logic (any data value may be written)
	xE0	(x38)	R	---	---	1	Ver. #		FPGA_Prog CPLD Firmwave Version Number*
FPGA Registers	X84-xDC	(x21-x37)	---	---	---	---	---	---	UNUSED
	x80	(x20)	R/W	---	6	---	N/A	xFF	Not-Sure Window Enable (x00 = disable, Else = enable)
	x7C	(x1F)	---	---	---	---	---	---	UNUSED
	x78	(x1E)	W	1	3	---	x00	x00	Transition Board FPGA Reprogram Signal
	x74	(x1D)	R	1	6	---	x02	x06	Number of Time Bins in Design (= 2 or 6)
	x70	(x1C)	---	---	---	---	---	---	UNUSED
	x6C	(x1B)	---	---	---	---	---	---	UNUSED
	x68	(x1A)	R/W	---	1	---	N/A	x00	Not-Sure Window 5 Width (Range: x00-x0F) (x00 ~ ?ns, x0F ~ ?ns)
	x64	(x19)	R/W	---	1	---	N/A	x00	Not-Sure Window 4 Width (Range: x00-x0F) (x00 ~ ?ns, x0F ~ ?ns)
	x60	(x18)	R/W	---	1	---	N/A	x00	Not-Sure Window 3 Width (Range: x00-x0F) (x00 ~ ?ns, x0F ~ ?ns)
	x5C	(x17)	R/W	---	1	---	N/A	x00	Not-Sure Window 2 Width (Range: x00-x0F) (x00 ~ ?ns, x0F ~ ?ns)
	x58	(x16)	R/W	---	1	---	N/A	x00	Not-Sure Window 1 Width (Range: x00-x0F) (x00 ~ ?ns, x0F ~ ?ns)
	x54	(x15)	R	1	1	---	Ver. #	Ver. #	Kitchen Sink FPGA Firmware Version Number*
	x50	(x14)	R	1	1	---	Ver. #	Ver. #	Data FPGA Firmware Version Number*
	x4C	(x13)	R	0	0	---	x00	x00	Buffer 3 Status (x00 = clear, Else = full)
	x48	(x12)	R	0	0	---	x00	x00	Buffer 2 Status (x00 = clear, Else = full)
	x44	(x11)	R	0	0	---	x00	x00	Buffer 1 Status (x00 = clear, Else = full)
	x40	(x10)	R	0	0	---	x00	x00	Buffer 0 Status (x00 = clear, Else = full)
	x3C	(x0F)	R	0	0	---	x00	x00	Buffer Byte
	x38	(x0E)	W	0	0	---	x00	x00	Buffer Byte Address (Range: 0-72) (Automatically triggers read)
	x34	(x0D)	W	0	0	---	x00	x00	Buffer Number (Range: 0-3)
	x30	(x0C)	W	0	0	---	x03	x03	Level 1 Accept Delay Value (in CDFCLK periods)
	x2C	(x0B)	R/W	0	0	---	x00	x18	Clock 5 Delay (DELCLK[5]) (U13) **(1ns/step) (LE_L)
	x28	(x0A)	R/W	0	0	---	x00	x18	Clock 4 Delay (DELCLK[4]) (U12) **(1ns/step) (LE_K)
	x24	(x09)	R/W	0	0	---	x00	x18	Clock 3 Delay (DELCLK[3]) (U17) **(1ns/step) (LE_J)
	x20	(x08)	R/W	0	0	---	xFF	xD4	Delayed Set/Clear Delay (U15) (1ns/step) (LE_I)
	x1C	(x07)	R/W	0	0	---	x5E	x00	Prompt Set/Clear Delay (U16) **(1ns/step) (LE_H)
	x18	(x06)	R/W	0	0	---	x38	x18	Clock 2 Delay (DELCLK[2]) (U14) **(1ns/step) (LE_G)
	x14	(x05)	R/W	0	0	---	x31	x18	Clock 1 Delay (DELCLK[1]) (U19) **(1ns/step) (LE_F)
	x10	(x04)	R/W	0	0	---	x1E	x4E	CDFB0 Delay (U18) (1ns/step) (LE_E)
	x0C	(x03)	R/W	0	0	---	x34	x34	PLL Output Delay (U10) (0.25ns/step) (LE_D)
	x08	(x02)	R/W	0	0	---	xC8	xC8	PLL Input Delay (U11) (0.5ns/step) (LE_C)
	x04	(x01)	R/W	0	0	---	x1E	x4E	CDFBC Delay (U20) (1ns/step) (LE_B)
	x00	(x00)	R/W	0	0	---	x2E	x5E	Initial CDFCLK Delay (U21) **(1ns/step) (LE_A)

Table 1: XTC 2 **Prototype Board** Register Space Summary

** The firmware version number registers were added after a couple versions had already been created. If zero is read from either of these registers, or if data can be written and then read back from either, the firmware predates the addition of these registers. The version numbers begin with 1 and should be changed for each firmware revision.*

*** XTC 2 prototype boards 2, 3, and 4 have an alternate (ALT) delay line chip configuration: the chips for all registers marked with a ** were replaced by 0.5ns/step chips. Doubling the values stored in these registers provides adequate compensation.*

	Address		R/W	First Version			Initial Value		Function
	VME	XTC		2-Bin	6-Bin	CPLD	2-Bin	6-Bin	
CPLD Registers	xFC	(x3F)	W	---	---	0	N/A		Erase Flash Memory (any data value may be written)
	xF8	(x3E)	W	---	---	0	N/A		Reset Flash Memory Address (any data value may be written)
	xF4	(x3D)	W	---	---	0	N/A		Write Next Byte to Flash Memory (data = byte to be written)
	xF0	(x3C)	W	---	---	0	N/A		Flash Programming Complete (any data value may be written; resets flash memory address and configures FPGAs)
	xEC	(x3B)	R	---	---	0	mem(0)		Read Next Byte in Flash Memory
	xE8	(x3A)	R	---	---	0	xFF		Read FPGA Configuration & Flash Memory Program/Erase Status (7:4): FPGA Configuration (x0 = configured, xF = not configured) (3:0): Flash Status (x0 = ready, xF = busy)
	xE4	(x39)	W	---	---	1	N/A		Reset FPGA_Prog CPLD logic (any data value may be written)
	xE0	(x38)	R	---	---	1	Ver. #		FPGA_Prog CPLD Firmwave Version Number*
FPGA Registers	x84-xDC	(x21-x37)	---	---	---	---	---	---	UNUSED
	x80	(x20)	R/W	---	16	---	N/A	xFF	Not-Sure Window Enable (x00 = disable, Else = enable)
	x7C	(x1F)	R/W	17	17	---	xFF	xFF	Shift_In_Bit Enable (x00 = disable [enable test mode], Else = enable)
	x78	(x1E)	W	16	16	---	x00	x00	Transition Board FPGA Reprogram Signal
	x74	(x1D)	R	16	16	---	x02	x06	Number of Time Bins in Design (= 2 or 6)
	x70	(x1C)	R	16	16	---	Ser. #	Ser. #	Board Serial Number (high serial number bit in the lsb of the register)
	x6C	(x1B)	R	16	16	---	Ser. #	Ser. #	Board Serial Number (low 8 bits)
	x68	(x1A)	R/W	---	16	---	N/A	x00	Not-Sure Window 5 Width (Range: x00-x0F) (x00 ~ ?ns, x0F ~ ?ns)
	x64	(x19)	R/W	---	16	---	N/A	x00	Not-Sure Window 4 Width (Range: x00-x0F) (x00 ~ ?ns, x0F ~ ?ns)
	x60	(x18)	R/W	---	16	---	N/A	x00	Not-Sure Window 3 Width (Range: x00-x0F) (x00 ~ ?ns, x0F ~ ?ns)
	x5C	(x17)	R/W	---	16	---	N/A	x00	Not-Sure Window 2 Width (Range: x00-x0F) (x00 ~ ?ns, x0F ~ ?ns)
	x58	(x16)	R/W	---	16	---	N/A	x00	Not-Sure Window 1 Width (Range: x00-x0F) (x00 ~ ?ns, x0F ~ ?ns)
	x54	(x15)	R	16	16	---	Ver. #	Ver. #	Kitchen Sink FPGA Firmware Version Number*
	x50	(x14)	R	16	16	---	Ver. #	Ver. #	Data FPGA Firmware Version Number*
	x4C	(x13)	R	16	16	---	x00	x00	Buffer 3 Status (x00 = clear, Else = full)
	x48	(x12)	R	16	16	---	x00	x00	Buffer 2 Status (x00 = clear, Else = full)
	x44	(x11)	R	16	16	---	x00	x00	Buffer 1 Status (x00 = clear, Else = full)
	x40	(x10)	R	16	16	---	x00	x00	Buffer 0 Status (x00 = clear, Else = full)
	x3C	(x0F)	R	16	16	---	x00	x00	Read Buffer Byte (Automatically triggers read of next address)
	x38	(x0E)	W	16	16	---	x00	x00	Buffer Byte Address (Range: 0-72) (Automatically triggers read)
	x34	(x0D)	W	16	16	---	x00	x00	Buffer Number (Range: 0-3)
	x30	(x0C)	W	16	16	---	x03	x03	Level 1 Accept Delay Value (in CDFCLK periods)
	x2C	(x0B)	R/W	16	16	---	x00	x30	Clock 5 Delay (DELCLK[5]) (U13) (0.5ns/step) (LE_L)
	x28	(x0A)	R/W	16	16	---	x00	x30	Clock 4 Delay (DELCLK[4]) (U12) (0.5ns/step) (LE_K)
	x24	(x09)	R/W	16	16	---	x00	x30	Clock 3 Delay (DELCLK[3]) (U17) (0.5ns/step) (LE_J)
	x20	(x08)	R/W	16	16	---	x7F	x6A	Delayed Set/Clear Delay (U15) **(2ns/step) (LE_I)
	x1C	(x07)	R/W	16	16	---	XBC	x00	Prompt Set/Clear Delay (U16) (0.5ns/step) (LE_H)
	x18	(x06)	R/W	16	16	---	x70	x30	Clock 2 Delay (DELCLK[2]) (U14) (0.5ns/step) (LE_G)
	x14	(x05)	R/W	16	16	---	x62	x30	Clock 1 Delay (DELCLK[1]) (U19) (0.5ns/step) (LE_F)
	x10	(x04)	R/W	16	16	---	x06	x0F	CDFB0 Delay (U18) **(5ns/step) (LE_E)
	x0C	(x03)	R/W	16	16	---	x34	x34	CHIP REMOVED [PLL Output Delay (U10) (0.25ns/step) (LE_D)]
	x08	(x02)	R/W	16	16	---	xC8	xC8	PLL Input Delay (U11) (0.5ns/step) (LE_C)
	x04	(x01)	R/W	16	16	---	x06	x0F	CDFBC Delay (U20) **(5ns/step) (LE_B)
	x00	(x00)	R/W	16	16	---	x5C	xBC	Initial CDFCLK Delay (U21) (0.5ns/step) (LE_A)

Table 2: XTC 2 **Production Board** Register Space Summary

** Firmware version numbers should be incremented for each firmware revision.*

***XTC 2 production boards 0, 132, 133, 144, 145, and 146 have an alternate (ALT) delay line chip configuration: chips for registers 0x04 and 0x10 were replaced by 0.5ns/step chips, and the chip for register 0x20 was replaced by a 1ns/step chip.*

Note: Version numbers 0-15 (0x00-0x0F) are reserved for prototype designs. Production board firmware design numbers begin with 16 (0x10).

The first address column of Tables 1 and 2 contains the last two hex digits of the VME address. The second address column is the address seen by the XTC 2 card on the data lines. The R/W column indicates whether the register is read-only, write-only, or read/write. The "First Version" columns specify the first Kitchen Sink FPGA or FPGA_Prog CPLD firmware version/revision number in which each register is available. The "Initial Value" column indicates the value stored/available in each register immediately after power-up.

Note: All registers in this note are referenced using VME addresses unless otherwise specified.

3 Detailed Register Description

All registers except 0xE0 – 0xFC are located in the Kitchen Sink FPGA and cannot be read from or written to until the FPGA has been configured. The Kitchen Sink FPGA reprograms all of the delay line chips—as well as all other register values used by the internal FPGA logic (e.g., the not-sure window width registers)—only when data is written to one of those registers (0x00-0x30, 0x58-0x68, 0x78, 0x7C, and 0x80).

Registers 0xE0 – 0xFC are used by the FPGA-Programmer (FPGA_Prog) CPLD to read, erase, and write to the flash memory. These registers only require that this CPLD be configured.

Note: all register definitions given here are valid for the following firmware versions:

FPGA_Prog CPLD: 2
Output_Buffers CPLD: 1
2-Bin Prototype Kitchen Sink FPGA: 2 (2-Bin Data FPGA 0)
2-Bin Production Kitchen Sink FPGA: 18 (2-Bin Data FPGA 1)
6-Bin Prototype Kitchen Sink FPGA: 7 (6-Bin Data FPGA 3)
6-Bin Production Kitchen Sink FPGA: 18 (6-Bin Data FPGA 4)

3.1 Delay Line Chip Registers (0x00 – 0x2C)

Twelve registers, all of which can be read from and written to, store the 8-bit values used to program the delay line chips. Registers 0x00 through 0x20 have the same functionality as the corresponding registers in the Michigan XTC card and are the only delay line chip registers necessary for running the XTC 2 two-bin design. Registers 0x24 through 0x2C add the 3 extra window signals necessary for the 6-bin design. Before sending any data through the board, all registers necessary for the design to be used should be programmed first if their values are going to be different from the initial values listed in Tables 1 and 2. Figures 2 and 3 show which window edges are affected by each register.

All window signals are derived from CDFCLK, including DLSTCL, so delaying this signal will delay all others shown in these two figures. All of the window signals,

excluding DLSTCL, are dependent on one another, meaning that if one window edge is delayed, all subsequent window edges will be delayed by an equal amount. DLSTCL is only dependent on CDFCLK, so it remains unaffected by changes made to the window edges.

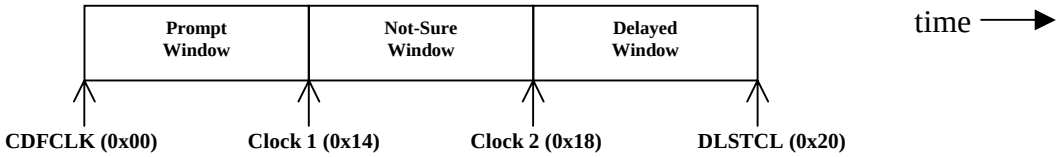

Figure 2: This figure shows which registers affect each window edge in the 2-bin design. The VME address for each register is given in hex next to the signal name.

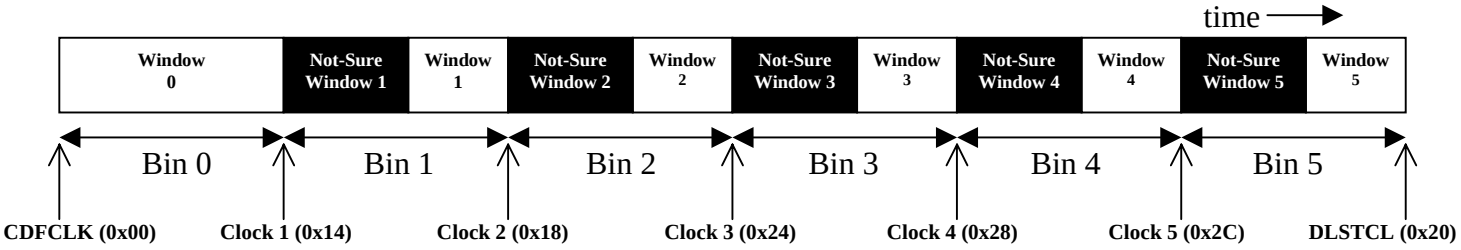

Figure 3: This figure shows which registers affect each window edge in the 6-bin design. The VME address for each register is given in hex next to the signal name.

All of the delay line chips have a typical step-zero delay of around 16.5ns. This means that setting the register value to 0x00 will delay the signal by 16.5ns. Incrementing the register value by one will then increase the signal delay by the step size (0.25ns, 0.5ns, 1ns, 2ns, or 5ns, depending on the chip). Thus, the total signal delay through any chip will be:

$$\textit{Total Signal Delay} = (\textit{Register Value}) * (\textit{Step Size}) + 16.5$$

Note: delay-line chip U10, corresponding to register 0x0C, was removed from all production boards because it was being used at frequencies beyond the specifications for the chip, which caused intermittent errors. A jumper was placed on each board to connect signals STR22 and CLK22 together. Delays were then added to the appropriate signals in the Output_Buffers CPLD and Kitchen_Sink FPGA to compensate for the removal of this chip.

Register Functionality:

0x00 – Initial CDFCLK Delay

Step Size:

Standard Prototype:	1ns	*Alternate Prototype:*	0.5ns
Standard Production:	0.5ns	*Alternate Production:*	0.5ns

Board Signal Affected: CDFCLK

Delay Chip Ref. Designator: U21

Latch Enable Signal: LE_A

Description: This initial delay for the CDFCLK signal is used to compensate for the time required for signals to reach the XTC 2 card from the detector.

0x04 – CDFBC Delay

Step Size:

Standard Prototype:	1ns	*Alternate Prototype:*	1ns
Standard Production:	5ns	*Alternate Production:*	0.5ns

Board Signal Affected: XTLBC

Delay Chip Ref. Designator: U20

Latch Enable Signal: LE_B

Description: This is used to delay the BC signal to synchronize it with the CDFCLK. If the CDFCLK delay is changed, the value in this register should probably be changed as well.

0x08 – PLL Input Delay

Step Size:

Standard Prototype:	0.5ns	*Alternate Prototype:*	0.5ns
Standard Production:	0.5ns	*Alternate Production:*	0.5ns

Board Signal Affected: PLLSYNC

Delay Chip Ref. Designator: U11

Latch Enable Signal: LE_C

Description: This is used for delaying the CDFCLK signal for the PLL, as well as creating the SYNCLR signal.

0x0C – PLL Output Delay (Chip removed from production boards)

Step Size:

Standard Prototype:	0.25ns	*Alternate Prototype:*	0.25ns
Standard Production:	0.25ns	*Alternate Production:*	0.25ns

Board Signal Affected: STR22

Delay Chip Ref. Designator: U10

Latch Enable Signal: LE_D

Description: This is used to shift the 22-ns board clock around in time. U10 was removed from the production boards, but the register remains in the firmware.

0x10 – CDFB0 Delay

Step Size:

Standard Prototype:	1ns	*Alternate Prototype:*	1ns
Standard Production:	5ns	*Alternate Production:*	0.5ns

Board Signal Affected: XTLB0

Delay Chip Ref. Designator: U18

Latch Enable Signal: LE_E

Description: This is used to delay the B0 signal to synchronize it with the CDFCLK. If the CDFCLK delay is changed, the value in this register should probably be changed as well.

0x14 – Clock 1 Delay

Step Size:

Standard Prototype:	1ns	*Alternate Prototype:*	0.5ns
Standard Production:	0.5ns	*Alternate Production:*	0.5ns

Board Signal Affected: DELCLK[1]

Delay Chip Ref. Designator: U19

Latch Enable Signal: LE_F

Description: This is used to delay the CDFCLK signal (board signal DELCLK[0]) to create the trailing edge of window 0/leading edge of the not-sure part of window 1.

0x18 – Clock 2 Delay

Step Size:

Standard Prototype:	1ns	*Alternate Prototype:*	0.5ns
Standard Production:	0.5ns	*Alternate Production:*	0.5ns

Board Signal Affected: DELCLK[2]

Delay Chip Ref. Designator: U14

Latch Enable Signal: LE_G

Description: This is used to delay the DELCLK[1] signal to create the trailing edge of window 1/leading edge of the not-sure part of window 2.

0x1C – Prompt Set/Clear Delay (PRSTCL)

Step Size:

Standard Prototype:	1ns	*Alternate Prototype:*	0.5ns
Standard Production:	0.5ns	*Alternate Production:*	0.5ns

Board Signal Affected: PRSTCL

Delay Chip Ref. Designator: U16

Latch Enable Signal: LE_H

Description: This is used to delay the CDFCLK signal (board signal DELCLK[0]) to create a short (~10ns) pulse that stores the detected prompt bits in output flip-flops (“set” functionality) while clearing out the prompt bit detection flip-flops (“clear” functionality). This signal is not used in the 6-bin design.

0x20 – Delayed Set/Clear Delay (DLSTCL)

Step Size:

Standard Prototype:	1ns	*Alternate Prototype:*	1ns
Standard Production:	2ns	*Alternate Production:*	1ns

Board Signal Affected: DLSTCL

Delay Chip Ref. Designator: U15

Latch Enable Signal: LE_I

Description: This is used to delay the CDFCLK signal (board signal DELCLK[0]) to create a short (~10ns) pulse that, in the 2-bin design, stores the detected delayed bits in output flip-flops ("set" functionality) while clearing out the delayed bit detection flip-flops ("clear" functionality). It also marks the end of the delayed window. In the 6-bin design, DLSTCL marks the end of window 5.

0x24 – Clock 3 Delay

Step Size:

Standard Prototype:	1ns	*Alternate Prototype:*	0.5ns
Standard Production:	0.5ns	*Alternate Production:*	0.5ns

Board Signal Affected: DELCLK[3]

Delay Chip Ref. Designator: U17

Latch Enable Signal: LE_J

Description: This is used to delay the DELCLK[2] signal to create the trailing edge of window 2/leading edge of the not-sure part of window 3. This signal is not used in the 2-bin design.

0x28 – Clock 4 Delay

Step Size:

Standard Prototype:	1ns	*Alternate Prototype:*	0.5ns
Standard Production:	0.5ns	*Alternate Production:*	0.5ns

Board Signal Affected: DELCLK[4]

Delay Chip Ref. Designator: U12

Latch Enable Signal: LE_K

Description: This is used to delay the DELCLK[3] signal to create the trailing edge of window 3/leading edge of the not-sure part of window 4. This signal is not used in the 2-bin design.

0x2C – Clock 5 Delay

Step Size:

Standard Prototype:	1ns	*Alternate Prototype:*	0.5ns
Standard Production:	0.5ns	*Alternate Production:*	0.5ns

Board Signal Affected: DELCLK[5]

Delay Chip Ref. Designator: U13

Latch Enable Signal: LE_L

Description: This is used to delay the DELCLK[4] signal to create the trailing edge of window 4/leading edge of the not-sure part of window 5. This signal is not used in the 2-bin design.

3.2 Level 2 Buffer Registers (0x30 – 0x4C)

These registers are used for accessing the level 2 data stored in the 4 Kitchen Sink memory buffers when a level 1 accept is issued. The *Valid Value Range* fields listed below should be followed when writing to these registers. Unintended effects may occur if this is not done.

Note: the first byte (address 0) of each level 2 buffer contains a bunch count value in the range 0-158, which indicates the number of CDFCLK cycles that have elapsed since the bunch 0 crossing that occurred before the data stored in the buffer was written to the buffer. This count value is obtained by subtracting the length of the level 1 pipeline, stored in register 0x30, from the value of the counter when the level 1 accept is issued. For example, if the level 1 pipeline length is set to 10, and if a level 1 accept is issued 28 CDFCLKs after B0, the count value stored will be 28 – 10 = 18.

The procedure for reading an entire buffer containing all of the data from a single beam crossing (396ns) is as follows:

1. Check that the buffer is full by reading its buffer status register (0x40, 0x44, 0x48, or 0x4C; value should be 0xFF).
2. Write the buffer number to register 0x34.
3. Do either one of the following (the firmware still supports the old way):
   a. [Old way:] Do the following for each byte in the buffer (72 times total): write the address of the byte to be read to register 0x38, and then retrieve the byte by reading register 0x3C.
   b. [New way:] Write the address of the first byte to be read (0x01) to register 0x38, and then read register 0x3C for all bytes in the buffer (72 times). The byte address is automatically incremented after each register operation, so it is unnecessary to keep writing to register 0x38 after the first time.

Register Functionality:

0x30 – Level 1 Accept Delay Value (in CDFCLK periods)

Valid Value Range: 0x00 – 0xFF (0 – 255)
Functionality: Write (can be read as well)
Description: The value stored in this register indicates the number of CDFCLK periods between the generation of a set of accepted data and the level 1 accept signal that indicates that that data were accepted. So, if the value stored is 12, when the logic sees a level 1 accept event, it will store the data from 12 CDFCLK cycles ago in a buffer.

0x34 – Buffer Number

Valid Value Range: 0x00 – 0x03 (0 – 3)
Functionality: Write (can be read as well)
Description: The value stored in this register determines the buffer from which data is read when executing read commands via registers 0x38 and 0x3C.

0x38 – Buffer Byte Address

Valid Value Range: 0x00 – 0x48 (0 – 72)

Functionality: Write (can be read as well)

Description: The value stored in this register selects the byte to be read from the buffer indicated by register 0x34. Writing to this register will trigger an automatic read of the byte from the buffer and store the data in register 0x3C. The value in this register is then incremented automatically. See the note above in this section about the first byte of each level 2 buffer.

0x3C – Read Buffer Byte

Valid Value Range: N/A

Functionality: Read-Only

Description: When an address value is written to register 0x38, the byte is read from the buffer indicated by register 0x34 and stored in this register. The byte address, stored in register 0x38, is automatically incremented each time this register is read.

0x40 – Buffer 0 Status

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Write 0x00 – 0xFF, Read 0x00 or 0xFF

Description: When read, this register indicates the current status of buffer 0. 0x00 indicates the buffer is empty; any other value indicates the buffer is full. Data can now be written to buffer 0 regardless of whether this register indicates the buffer is full.

0x44 – Buffer 1 Status

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Write 0x00 – 0xFF, Read 0x00 or 0xFF

Description: When read, this register indicates the current status of buffer 1. 0x00 indicates the buffer is empty; any other value indicates the buffer is full. Data can now be written to buffer 1 regardless of whether this register indicates the buffer is full.

0x48 – Buffer 2 Status

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Write 0x00 – 0xFF, Read 0x00 or 0xFF

Description: When read, this register indicates the current status of buffer 2. 0x00 indicates the buffer is empty; any other value indicates the buffer is full. Data can now be written to buffer 2 regardless of whether this register indicates the buffer is full.

0x4C – Buffer 3 Status

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Write 0x00 – 0xFF, Read 0x00 or 0xFF

Description: When read, this register indicates the current status of buffer 3. 0x00 indicates the buffer is empty; any other value indicates the buffer is full. Data can now be written to buffer 3 regardless of whether this register indicates the buffer is full.

3.3 Firmware Version Number Registers (0x50, 0x54, 0xE0)

The firmware version numbers of both FPGA designs and the FPGA_Prog CPLD design are stored in these read-only registers. 0x50 and 0x54 are located in the Kitchen Sink, while 0xE0 is implemented in the FPGA_Prog CPLD. The version numbers begin with 1 and should be incremented each time the firmware is revised. Note that these registers were undefined in the first few firmware versions that were created. As a result, if these registers return a value of 0x00 when read (except for the 2-bin Data FPGA design, which has not been revised), or if the value stored in the register can be modified (meaning any value can be written and then successfully read back), the firmware version predates the addition of these registers. Also note that if a change is made only to the Data FPGA firmware, the Data FPGA version number must be incremented, requiring that the Kitchen Sink FPGA firmware be resynthesized as well (since both FPGA version registers are implemented in the Kitchen Sink).

Current Design Number Definitions:

2-Bin Data FPGA

Version 0x00	• First version to work
Version 0x01	• Added a counter to the Ouput_Selector component that sets all data outputs to high impedance if shift_in_bit is not high for at least 1 out of every 64 shift_clk cycles (it should be high 1 in every 16 shift_clk cycles). This was added to provide a way of setting all the data outputs to high impedance using a Kitchen Sink register: just stop sending the shift_in_bit pulse. • Added the maxskew attribute to the shift_clk signal

6-Bin Data FPGA

Version 0x00	• First version to work

Version 0x01	• Added not-sure windows (using New_Channel component instead of Channel)
	  • Added port ldel(3)
	  • Switched to using a gray counting scheme for the window signals (ldel)
	  • Added not-sure flip-flops and logic to the Channel component

Version 0x02	• Switched to using 6 clear signals instead of 3, as well as dlstcl
	  • Removed ports clr01, clr23, clr45
	  • Added port clk_out
	  • Added port dlstcl
	  • Added internal signal dlstcl_delayed
	  • Modified Channel logic appropriately
	• Switched back to using the Channel component instead of the New_Channel component

Version 0x03	• Added not-sure window enable functionality
	  • Added port ns_enable
	  • Modified Channel logic appropriately
	• Optimized locations of Channel components to minimize channel-to-channel propagation delay differences

Version 0x04	• Added a counter to the Ouput_Selector component that sets all data outputs to high impedance if shift_in_bit is not high for at least 1 out of every 64 shift_clk cycles (it should be high 1 in every 16 shift_clk cycles). This was added to provide a way of setting all the data outputs to high impedance using a Kitchen Sink register: just stop sending the shift_in_bit pulse. • Added the maxskew attribute to the shift_clk signal

2-Bin Kitchen Sink FPGA

Version 0x00	• First version to work • Use with v. 0x00 of the 2-Bin Data FPGA (prototype board)

Version 0x01	• Added keep_hierarchy attributes to all files • Added ports str22, l1b0, l1b1, and fpga_reprog_n • Added L1 accept address lines • Added firmware version registers (VME 0x50, 0x54) • Added register for storing the number of time bins (VME 0x74) • Added register for the Transition Board FPGA Reprogram signal (VME 0x78) • Changed use of cdfl1a to active-high • Added flip-flop for detecting edges of cdfl1a • Added logic for storing the L1 accept address • Revamped logic for the buffer status registers and the state machine that prohibits L1 accepts for filled L2 buffers • Modified the clear signal definition to allow writing to all L2 buffers regardless of their status • Changed the name of the 22-ns clock for the Data_Align component from clk22 to str22 • Set up clk132 (clk132_int) to pass through the Sync component, where it is delayed by an appropriate number of clk22 clock cycles (becomes clk132_dly), before going into the CDFL1A_Logic component. This properly aligns the 132-ns clock with the time at which data should be stored in the L2 buffers. • Modified Reg_Prog_Logic and TDC_Bus_Logic so the registers are only programmed when necessary rather than on all writes • Modified TDC_Bus_Logic so the registers are automatically programmed after being initialized • Added VME register initialization values • Added a buffer to delay the str22 clock signal in the CDFL1A_Logic component to eliminate the flip-flop input setup time errors on last_clk132 in Data_Align • Use with v. 0x00 of 2-Bin Data FPGA (prototype board)

Version 0x02	• Modified the L2 buffers to include the bunch count number, which indicates how many CDFCLK cycles after B0 the recorded data occurred. This value is stored in location 0 of each buffer, which means that the data in each buffer is now in locations 1-72 instead of 0-71.
	  • Modified Buffers component
	  • Modified CDFL1A_Logic component
	  • Modified Buffer_Read_Logic component
	• Use with v. 0x00 of 2-Bin Data FPGA (prototype board)

Version 0x10 *(Created from Version 0x02)*	• Added board serial number registers (VME 0x6C, 0x70) • Added port board_serial_num • Modified TDC_Bus_Logic appropriately • Changed pin assignment of port le_c from P139 to P125 • Modified the initial register contents in Reg_Init for the standard production boards • Use with v. 0x00 of 2-Bin Data FPGA (production board)

Version 0x11	• Modified the functionality of registers 0x38 and 0x3C: after each buffer read operation, the buffer address is incremented
	  • Modified Buffer_Read_Logic component
	• Modified the Data_Align component in the CDFL1A_Logic so that the 32 data bits that come directly from the data port are delayed by a few nanoseconds before being sent to the BRAM
	• Modified the Sync component to allow enabling and disabling of the shift_in_bit signal
	  • Added shift_in_bit enable register (VME 0x7C)
	  • Changed the initial value of register 0x7C in Reg_Init from 0x00 to 0xFF
	  • Changed the definition of prog_regs_clk in TDC_Bus_Logic so the registers in Reg_Prog_Logic are reprogrammed on a write to register 0x7C
	• Added a process to generate an output test pattern (alternating ones and zeros every output wrd0 period) when shift_in_bit is disabled (shift_in_bit_enable = '0')
	  • Changed direction of the data port from 'in' to 'inout'
	• Updated the values in Reg_Init to reflect the delay line chip changes in the standard production boards
	• Use with v. 0x01 of 2-Bin Data FPGA (production board)

Version 0x12	• Changed the NUMBER_OF_EXTRA_BUFFERS in STR22_Delay of the CDFL1A_Logic component from 0 to 5 to compensate for the loss of the DS1023S-25 delay line chip on the XTC 2 board, which delayed clk22 by ~8ns, creating str22. The extra delay added should be between about 3.5ns and 10ns. • Use with v. 0x01 of 2-Bin Data FPGA (production board)

6-Bin Kitchen Sink FPGA

Version 0x00	• First version to work • Use with v. 0x00 of 6-Bin Data FPGA (prototype board)

Version 0x01	• Added not-sure windows
	  • Added port ldel(3)
	  • Switched to using a gray counting scheme for the window signals (ldel)
	• Use with v. 0x01 of 6-Bin Data FPGA (prototype board)

Version 0x02	• Added register for simulating L1 accepts (VME 0x6C) • Changed use of cdfl1a to active-high • Use with v. 0x01 of 6-Bin Data FPGA (prototype board)
Version 0x03	• Added L1 accept address lines • Removed register for simulating L1 accepts from v. 0x02 • Added flip-flop for detecting edges of cdfl1a • Added logic for storing the L1 accept address • Revamped logic for the buffer status registers and the state machine that prohibits L1 accepts for filled L2 buffers • Changed the name of the 22-ns clock for the Data_Align component from clk22 to str22 • Added register 0x78 (Transition Board FPGA Reprogram signal) • Use with v. 0x01 of 6-Bin Data FPGA (prototype board)
Version 0x04	• Removed ports clr01, clr23, and clr45 • Added clk_out signal • Modified the clear signal definition to allow writing to all L2 buffers regardless of their status • Use with v. 0x02 of 6-Bin Data FPGA (prototype board)
Version 0x05	• Set up clk132 (clk132_int) to pass through the Sync component, where it is delayed by an appropriate number of clk22 clock cycles (becomes clk132_dly), before going into the CDFL1A_Logic component. This properly aligns the 132-ns clock with the time at which data should be stored in the L2 buffers. • Use with v. 0x02 of 6-Bin Data FPGA (prototype board)
Version 0x06	• Added register for the number of time bins in the design (VME 0x74) • Added not-sure window enable register (VME 0x80) • Modified reg_data(7) to be used for both register programming and the not-sure window enable • Modified Reg_Prog_Logic and TDC_Bus_Logic so the registers are only programmed when necessary rather than on all writes • Modified TDC_Bus_Logic so the registers are automatically programmed after being initialized • Added VME register initialization values • Added a buffer to delay the str22 clock signal in the CDFL1A_Logic component to eliminate the flip-flop input setup time errors on last_clk132 in Data_Align • Replaced the asynchronous generation of b0del with a shift register implementation that also triples the width of the signal • Modified not-sure window widths so their range is closer to 5-25 ns • Use with v. 0x03 of 6-Bin Data FPGA (prototype board)

Version 0x07	• Modified the L2 buffers to include the bunch count number, which indicates how many CDFCLK cycles after B0 the recorded data occurred. This value is stored in location 0 of each buffer, which means that the data in each buffer is now in locations 1-72 instead of 0-71.
	  • Modified Buffers component
	  • Modified CDFL1A_Logic component
	  • Modified Buffer_Read_Logic component
	• Use with v. 0x03 of 6-Bin Data FPGA (prototype board)

Version 0x10 *(Created from Version 0x06)*	• Added board serial number registers (VME 0x6C, 0x70)
	• Added port board_serial_num
	• Modified TDC_Bus_Logic component
	• Changed pin assignment of port le_c from P139 to P125
	• Modified the L2 buffers to include the bunch count number, which indicates how many CDFCLK cycles after B0 the recorded data occurred. This value is stored in location 0 of each buffer, which means that the data in each buffer is now in locations 1-72 instead of 0-71.
	  • Modified Buffers component
	  • Modified CDFL1A_Logic component
	  • Modified Buffer_Read_Logic component
	• Modified the initial register contents in Reg_Init for the standard production boards
	• Use with v. 0x03 of 6-Bin Data FPGA (production board)

Version 0x11	• Modified the functionality of registers 0x38 and 0x3C: after each buffer read operation, the buffer address is incremented • Modified Buffer_Read_Logic component • Modified the Data_Align component in the CDFL1A_Logic so that the 32 data bits that come directly from the data port are delayed by a few nanoseconds before being sent to the BRAM • Modified the Sync component to allow enabling and disabling of the shift_in_bit signal • Added shift_in_bit enable register (VME 0x7C) • Changed the initial value of register 0x7C in Reg_Init from 0x00 to 0xFF • Changed the definition of prog_regs_clk in TDC_Bus_Logic so the registers in Reg_Prog_Logic are reprogrammed on a write to register 0x7C • Added a process to generate an output test pattern (alternating ones and zeros every output wrd0 period) when shift_in_bit is disabled (shift_in_bit_enable = '0') • Changed direction of the data port from 'in' to 'inout' • Updated the values in Reg_Init to reflect the delay line chip changes in the standard production boards • Use with v. 0x04 of 6-Bin Data FPGA (production board)
Version 0x12	• Changed the NUMBER_OF_EXTRA_BUFFERS in STR22_Delay of the CDFL1A_Logic component from 0 to 5 to compensate for the loss of the DS1023S-25 delay line chip on the XTC 2 board, which delayed clk22 by ~8ns, creating str22. The extra delay added should be between about 3.5ns and 10ns. • Use with v. 0x04 of 6-Bin Data FPGA (production board)

FPGA_Prog CPLD

Version 0x00	• First version to work
Version 0x01	• Modified Flash_Interface so the state machine starts in PROG_DONE rather than DO_NOTHING, which configures the FPGAs on power-up • Modified Flash_Interface so that the FPGA program signal is only asserted on PROG_COMPLETE_1 (and no longer on PROG_COMPLETE_2 as well) • Added the max_addr signal that makes the Flash_Interface state machine break out of the configuration sequence once the address counter has counted too high (prevents board from locking up) • Modified register 0xE8 (VME) to include the FPGA configuration status • Added logic to reset the Reset_Counter and hence the chip logic using signal reset_reset when a write is performed to register 0xE4 (VME) • Added a CPLD firmware version number register (VME 0xE0)

Version 0x02	• Manually encoded FLASH_STATE_TYPE in the Flash_Interface so that only 1 bit changes between WAIT_FOR_CMD and the 4 states to which WAIT_FOR_CMD leads • Manually encoded COMMAND_TYPE in the Flash_Interface (the same gray-count sequence that is automatically generated by Leonardo is used) • Added a register for flash_io(6) (internal signal flash_io_6) to prevent the main Flash_Interface state machine from jumping states if flash_io(6) changes near a rising edge of clk264 (this eliminated unintentional erase commands from being issued) • Replaced all instances of flash_io(6) with flash_io_6 in Flash_Interface_Proc in the Flash_Interface • Added Inc_Proc to the Flash_Interface to make the int_inc signal synchronous • Added Reset_Proc to the Flash_Interface to make the int_reset signal synchronous • Added Programming_Proc to the Flash_Interface to make the programming signal synchronous

Output_Buffers CPLD (Note: version number not available in registers)

Version 0x00	• First version to work
Version 0x01	• Added a 1-buffer delay to the str22 signal to compensate for the removal of U10

Register Functionality:

0x50 – Data FPGA Firmware Version Number

Functionality: Read-Only (can write to early versions)

Description: The value stored in this register indicates the current version of the Data FPGA firmware.

0x54 – Kitchen Sink FPGA Firmware Version Number

Functionality: Read-Only (can write to early versions)

Description: The value stored in this register indicates the current version of the Kitchen Sink FPGA firmware.

0xE0 – FPGA_Prog CPLD Firmware Version Number

Functionality: Read-Only (use was forbidden in earlier versions)

Description: The value stored in this register indicates the current version of the FPGA_Prog CPLD firmware.

3.4 Not-Sure Window Width Registers (0x58 – 0x68)

These registers are used by the Kitchen Sink logic to determine the width of the not-sure windows. Similar to delay line chips, a value of 0x00 represents the step-zero, or

minimum, width value. Only the values listed in the *Valid Value Range* field should be used. Figure 4 shows which register affects each window edge, and Table 3 lists the different window widths, obtained from simulation, for each valid register value.

Note: these registers are undefined in Kitchen Sink firmware version 0.

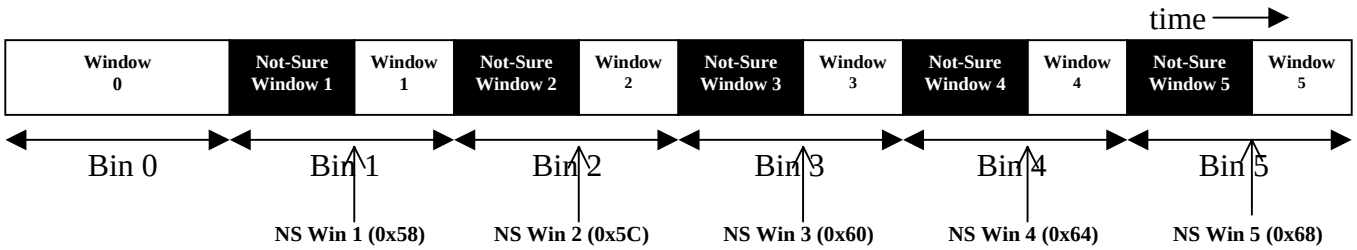

Figure 4: This figure shows which registers affect each not-sure window edge in the 6-bin design. The VME address for each register is given in hex next to the signal name.

Register Value	NS Win 1 Width (ns)	NS Win 2 Width (ns)	NS Win 3 Width (ns)	NS Win 4 Width (ns)	NS Win 5 Width (ns)
0x00	12.3	11.5	13.0	12.5	10.6
0x01	13.5	12.9	14.5	14.0	11.9
0x02	14.5	13.8	15.2	14.6	12.8
0x03	15.4	14.9	16.4	15.8	14.1
0x04	16.5	16.0	16.9	16.8	14.6
0x05	18.2	16.5	18.7	18.2	15.6
0x06	18.8	17.9	19.6	19.0	16.8
0x07	20.1	19.4	21.0	20.0	18.1
0x08	20.8	19.8	22.0	20.4	19.5
0x09	22.4	20.9	23.5	22.0	20.9
0x0A	23.2	21.7	24.3	22.9	21.8
0x0B	24.0	23.1	25.4	23.8	23.2
0x0C	24.7	24.0	25.9	24.4	23.7
0x0D	26.8	25.2	28.0	26.5	25.7
0x0E	27.4	25.5	28.7	27.1	26.4
0x0F	28.4	27.4	30.2	28.7	27.6

Table 3: This table lists the not-sure window widths for valid values of the not-sure window width registers. The times were taken from post place-and-route simulations done on the Kitchen Sink FPGA version 1 firmware.

Register Functionality:

0x58 – Not-Sure Window 1 Width

Valid Value Range: 0x00 – 0x0F (0 – 15)

Functionality: Read/Write

Description: This register value determines the number of extra buffers used in delaying the leading edge of window 1, which effectively determines the width of not-sure window 1.

0x5C – Not-Sure Window 2 Width

Valid Value Range: 0x00 – 0x0F (0 – 15)

Functionality: Read/Write

Description: This register value determines the number of extra buffers used in delaying the leading edge of window 2, which effectively determines the width of not-sure window 2.

0x60 – Not-Sure Window 3 Width

Valid Value Range: 0x00 – 0x0F (0 – 15)

Functionality: Read/Write

Description: This register value determines the number of extra buffers used in delaying the leading edge of window 3, which effectively determines the width of not-sure window 3.

0x64 – Not-Sure Window 4 Width

Valid Value Range: 0x00 – 0x0F (0 – 15)

Functionality: Read/Write

Description: This register value determines the number of extra buffers used in delaying the leading edge of window 4, which effectively determines the width of not-sure window 4.

0x68 – Not-Sure Window 5 Width

Valid Value Range: 0x00 – 0x0F (0 – 15)

Functionality: Read/Write

Description: This register value determines the number of extra buffers used in delaying the leading edge of window 5, which effectively determines the width of not-sure window 5.

3.5 Miscellaneous Registers (0x6C – 0x80)

These registers perform various functions, including providing access to the board serial number, the number of time bins in the design, enabling/disabling shift_in_bit and the not-sure windows, and reprogramming the Transition board FPGA.

Note: some of these registers (namely 0x6C, 0x70, and 0x7C) are not implemented in the prototype designs. While they are free read/write registers, to maintain as much compatibility between the prototype and production boards as possible, they should not be used for other functions.

Register Functionality:

0x6C – Board Serial Number (Production Boards Only)

Functionality: Read-Only

Description: The low 8 bits of the 9-bit board serial number are stored in this register.

0x70 – Board Serial Number (Production Boards Only)

Functionality: Read-Only

Description: The most significant bit of the 9-bit board serial number is stored in the least significant bit of this register. The other 7 bits of this register are set to 0.

0x74 – Number of Time Bins

Valid Value Range: Only 0x02 and 0x06 (0 and 6)

Functionality: Read-Only 0x02 and 0x06

Description: This register is set to either 2 or 6, indicating the number of time bins in the design.

0x78 – Transition Board FPGA Reprogram

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Write 0x00 – 0xFF (can read 0x00 or 0xFF)

Description: This register is used to reprogram the FPGA on the Transition Board. Write 0xFF (or any value greater than 0x00) followed by 0x00 to this register to begin the FPGA reconfiguration process. The register must be equal to 0xFF (or any value greater than 0x00) for at least 300ns. Since VME writes take significantly more time than this, 0x00 may be written immediately after 0xFF.

0x7C – Shift In Bit Enable/Test Mode Disable

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Read/Write

Description: This register is used to enable (any value but 0x00) and disable (0x00) the shift_in_bit signal that pulses once out of every 16 shift_clk (22-ns period) cycles. The shift_in_bit signal is used by the Data FPGA firmware in conjunction with shift_clk to determine when each set of multiplexed output data is sent. Disabling the shift_in_bit signal results in stopping all data transmission from the Data FPGA. In addition, the Data FPGA contains a counter that sets all data outputs to high impedance if a shift_in_bit pulse has not been observed for 64 shift_clk cycles. This is done to allow the Kitchen Sink FPGA to enter a test mode safely, without bus contention, in which it transmits alternating ones and zeros on the data lines that are inverted every wrd0 period (396 ns). This means that "101010…1010" is transmitted first, followed by "010101…0101." So, to summarize all of this: writing 0x00 to this register will cause the XTC 2 to stop normal data transmission and enter the test mode; writing any other value to this register will cause the XTC 2 to exit the test mode and process/transmit data as normal. The default value for this register on power-up is 0xFF (normal operation).

0x80 – Not-Sure Window Enable (6-Bin Designs Only)

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Read/Write

Description: This register is used to enable (any value but 0x00) and disable (0x00) the not-sure window logic in the 6-bin designs. It does not exist in the 2-bin designs. Note that disabling the not-sure windows does not actually change the detection mechanism. 11 flip-flops are still used—one for each window. This register controls the logic after detection: if disabled, any hits in the not-sure part of each time bin are counted as hits in that bin, regardless of whether hits were registered in the previous time bin.

3.6 Free Registers (0x84 – 0xDC)

These read/write registers are currently undefined and may be used for additional functionality.

3.7 Flash Memory Registers (0xE4 – 0xFC)

These registers, implemented in the FPGA-Programmer CPLD, are used to program the flash memory with FPGA configuration files. The address used for read and write operations is supplied by a counter in the CPLD to which the user does *not* have direct access. Reads and writes automatically increment the counter, and writing to register 0xF8 resets the counter to 0.

The procedure for programming the flash is as follows:

1. Erase the flash memory by writing any data value to register 0xFC.
2. Poll register 0xE8, bits 3:0 until the erasure is complete.
3. Reset the flash memory address by writing any data value to register 0xF8.
4. Do the following for all data bytes, one at a time: write the byte to register 0xF4, and then poll register 0xE8, bits 3:0 until the write operation has finished. It is helpful to add a statement that ensures that register 0xE8 is polled a minimum of 4 times for each byte that is written. Not having this added delay seemed to cause some addresses to be skipped on one of the prototype boards.
5. Reset the flash memory address by writing any data value to register 0xF8.
6. Verify successfully programming by do the following for each data byte: read the byte back from memory by reading from register 0xEC, and then compare the read back value with the byte that should have been written.
7. If all bytes were successfully written, instruct the FPGA-Programmer CPLD that programming is complete by writing any data value to register 0xF0. This will begin configuration of the FPGAs.
8. Poll register 0xE8, bits 7:4 until FPGA configuration is complete.

The procedure for instructing the CPLD to configure the FPGAs is as follows:

1. Write any data value to register 0xF0.

Register Functionality:

0xE4 – FPGA_Prog CPLD Reset

Valid Value Range: 0x00 – 0xFF (0 – 255)
Functionality: Write-Only
Description: Writing to this register (the data value does not matter) resets the FPGA-Programmer CPLD logic.

0xE8 – FPGA Configuration and Flash Program/Erase Status

Valid Value Range: N/A
Functionality: Read-Only
Description: Once a program, erase, or FPGA configuration operation has begun (by writing to registers 0xF4, 0xFC, or 0xF0), this register can be polled to determine when the operation is complete. The high nibble (bits 7:4) gives the FPGA status: 0x0 means both FPGAs are configured; 0xF means at least one of the FPGAs is not configured. The low nibble (bits 3:0) indicates the flash memory status: 0x0 means the flash is ready for the next operation; 0xF means the flash is busy.

0xEC –Read Next Byte in Flash Memory

Valid Value Range: N/A
Functionality: Read-Only
Description: Reading from this register retrieves the byte at the current flash memory address and then automatically increments the address counter.

0xF0 – Flash Programming Complete

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Write-Only

Description: Writing to this register (the data value does not matter) instructs the FPGA-Programmer CPLD that flash memory programming is complete. The CPLD resets the flash address counter to 0 and then configures the FPGAs.

0xF4 – Write Next Byte to Flash Memory

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Write-Only

Description: The data value written to this register is written to the flash memory at the address specified by the address counter. The address counter is then incremented automatically.

0xF8 – Reset Flash Memory Address

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Write-Only

Description: Writing to this register (the data value does not matter) instructs the FPGA-Programmer CPLD to reset the flash address counter to 0.

0xFC –Erase Flash Memory

Valid Value Range: 0x00 – 0xFF (0 – 255)

Functionality: Read/Write

Description: Writing to this register (the data value does not matter) instructs the FPGA-Programmer CPLD to erase the flash memory.

4 Revisions

Date	Version	Modifications
10/27/2004	1.0	-Original document
11/5/2004	1.1	-Updated section 3.3: added more design number definitions
3/16/2005	1.2	-Updated entire document: replaced all instances of "FPGA-Programming CPLD" with "FPGA-Programmer CPLD" -Updated Table 1: added registers; expanded the "First Version" column; added the "Initial Value" columns; changed the buffer status registers to read-only -Added Table 2 for the production design -Updated section 3: the KS FPGA only reprograms the delay line chips and other registers when certain registers are written to; added note on firmware versions for which the register definitions are valid -Updated section 3.1: added reference to the initial register values in Tables 1 and 2; expanded "step size" definitions -Updated section 3.2: modified registers 0x40-0x4C; added note about the first byte added to the beginning of each level 2 buffer; updated the valid value range of register 0x38 -Updated section 3.3: added register 0xE0; updated and added more design number definitions -Updated section 3.4: renumbered Table 2 to Table 3; marked the values in Table 3 as old -Updated register ranges in the title of sections 3.5 and 3.6 -Updated section 3.6: removed register 0xE0 from this section; modified registers 0xE4 and 0xE8; modified the flash programming procedure to account for the update to 0xE8 -Renumbered sections 3.5 and 3.6 to 3.6 and 3.7 -Added section 3.5: Miscellaneous Registers
4/7/2005	1.3	-Updated Table 1: changed register 0xE0 -Updated Table 2: changed registers 0x04, 0x10, 0x1C, 0x20, 0x3C, 0x7C, and 0xE0; changed the ** note for Table 2 -Updated Section 3: updated the list of registers for which the KS FPGA reprograms the delay line chips; updated firmware version numbers -Updated Section 3.1: updated the "Standard Production" delay line chip step values -Updated Section 3.2: modified the buffer read procedure listing; updated the descriptions of registers 0x38 and 0x3C -Updated Section 3.3: updated and added more design number definitions -Updated Section 3.5: updated the register range in the title; added register 0x7C -Updated Section 3.6: updated the register range in the title

5/9/2005	1.4	-Updated Table 2: grayed out register 0x0C because chip U10 was removed from all production boards -Updated Section 3: updated firmware version numbers -Updated Section 3.1: added note about the removal of U10; updated the description of register 0x0C -Updated Section 3.3: added more design number definitions; added a table for the Output_Buffers CPLD versions
6/3/2005	1.5	-Updated Section 3: updated firmware version numbers -Updated Section 3.3: added more design number definitions

For questions, comments, or problems, contact:

Ryan Mokos
University of Illinois, Urbana-Champaign
mokos@uiuc.edu
Office phone: 217-244-8397