XTC 2 Memory Space

Ryan Mokos

University of Illinois, Urbana-Champaign

Version 1.1 November 5, 2004

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.

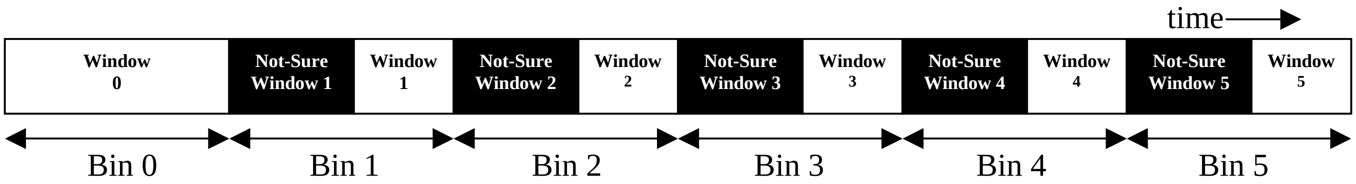

Figure 1: This diagram shows how all six time bins are laid out and illustrates the numbering scheme used in this document for normal windows and not-sure windows.

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 prototype boards). They connect to the FPGA-Programming CPLD (U23 on the XTC 2 prototype boards), and from there they are routed to the Kitchen Sink FPGA (U2 on the XTC 2 prototype boards).

The XTC 2 card contains 64 registers, most of which are located in the Kitchen Sink FPGA. Table 1 contains a summary 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 Ver.	Function
xFC	(x3F)	W	0	Erase Flash Memory (any data value may be written)
xF8	(x3E)	W	0	Reset Flash Memory Address (any data value may be written)
xF4	(x3D)	W	0	Write Next Byte to Flash Memory (data = byte to be written)
xF0	(x3C)	W	0	Flash Programming Complete (any data value may be written; resets flash memory address and configures FPGAs)
xEC	(x3B)	R	0	Read Next Byte in Flash Memory
xE8	(x3A)	R	0	Read Flash Memory Program/Erase Status (x00 = ready, xFF = busy)
xE4	(x39)	X	---	**DO NOT USE***
xE0	(x38)	X	---	**DO NOT USE***
x6C-xDC	(x1B-x37)	---	---	UNUSED
x68	x(1A)	R/W	1	Not-sure window 5 width (Range: x00-x0F) (x00 ~ ?ns, x?? ~ ?ns)
x64	x(19)	R/W	1	Not-sure window 4 width (Range: x00-x0F) (x00 ~ ?ns, x?? ~ ?ns)
x60	(x18)	R/W	1	Not-sure window 3 width (Range: x00-x0F) (x00 ~ ?ns, x?? ~ ?ns)
x5C	(x17)	R/W	1	Not-sure window 2 width (Range: x00-x0F) (x00 ~ ?ns, x?? ~ ?ns)
x58	(x16)	R/W	1	Not-sure window 1 width (Range: x00-x0F) (x00 ~ ?ns, x?? ~ ?ns)
x54	(x15)	R	1	Kitchen Sink FPGA Firmware Version Number***
x50	(x14)	R	1	Data FPGA Firmware Version Number***
x4C	(x13)	R/W	0	Buffer 3 Status (x00 = clear, xFF = full)
x48	(x12)	R/W	0	Buffer 2 Status (x00 = clear, xFF = full)
x44	(x11)	R/W	0	Buffer 1 Status (x00 = clear, xFF = full)
x40	(x10)	R/W	0	Buffer 0 Status (x00 = clear, xFF = full)
x3C	(x0F)	R	0	Buffer Byte
x38	(x0E)	W	0	Buffer Byte Address (Range: 0-71) (Automatically triggers read)
x34	(x0D)	W	0	Buffer Number (Range: 0-3)
x30	(x0C)	W	0	Level 1 Accept Delay Value (in CDF clock periods)
x2C	(x0B)	R/W	0	Clock 5 Delay (DELCLK[5]) (U13) **(1ns/step) (LE_L)
x28	(x0A)	R/W	0	Clock 4 Delay (DELCLK[4]) (U12) **(1ns/step) (LE_K)
x24	(x09)	R/W	0	Clock 3 Delay (DELCLK[3]) (U17) **(1ns/step) (LE_J)
x20	(x08)	R/W	0	Delayed Set/Clear Delay (U15) (1ns/step) (LE_I)
x1C	(x07)	R/W	0	Prompt Set/Clear Delay (U16) **(1ns/step) (LE_H)
x18	(x06)	R/W	0	Clock 2 Delay (DELCLK[2]) (U14) **(1ns/step) (LE_G)
x14	(x05)	R/W	0	Clock 1 Delay (DELCLK[1]) (U19) **(1ns/step) (LE_F)
x10	(x04)	R/W	0	CDFB0 Delay (U18) (1ns/step) (LE_E)
x0C	(x03)	R/W	0	PLL Output Delay (U10) (0.25ns/step) (LE_D)
x08	(x02)	R/W	0	PLL Input Delay (U11) (0.5ns/step) (LE_C)
x04	(x01)	R/W	0	CDFBC Delay (U20) (1ns/step) (LE_B)
x00	(x00)	R/W	0	Initial CDFCLK Delay (U21) **(1ns/step) (LE_A)

Table 1: This table summarizes the XTC 2 register space. The first address column 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. **All registers in this note are referenced using VME addresses unless otherwise specified.** The R/W column indicates whether the register is meant to be read, written, or both. The “First Ver.” column specifies the first FPGA firmware version/revision number in which each register is available.

** Because of limited space in the FPGA Prog CPLD, not all address bits are used to decode the instructions. Performing reads or writes to addresses xE4 and xE0 may have unintended effects. It is also recommended that these addresses/registers not be used for any additional features.*

*** Due to a lack of 1ns/step delay line chips, XTC2 prototype boards #2, #3, and #4 (all labeled with the letters ALT) have chips that are 0.5ns/step. The part number on the chip indicates the step value: -50 = 0.5ns, -100 = 1ns. For each board, either 0.5ns/step chips were substituted for all locations marked with a **, or no substitutions were made. Doubling the values stored in these registers provides adequate compensation.*

**** 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.*

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. Every time that data is written to a Kitchen Sink register, 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).

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

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. 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 following 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.

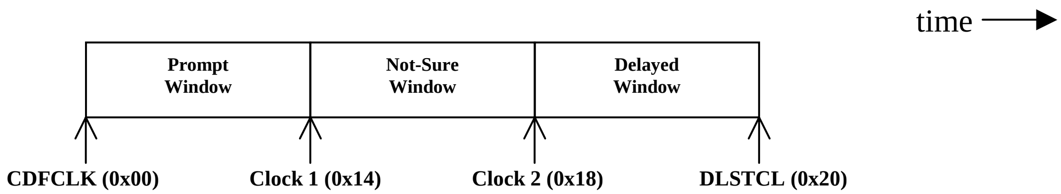

Figure 2: This figure shows which registers affect each window edge in the 2-bin design. The VME address for each register is given in hex next to the signal name.

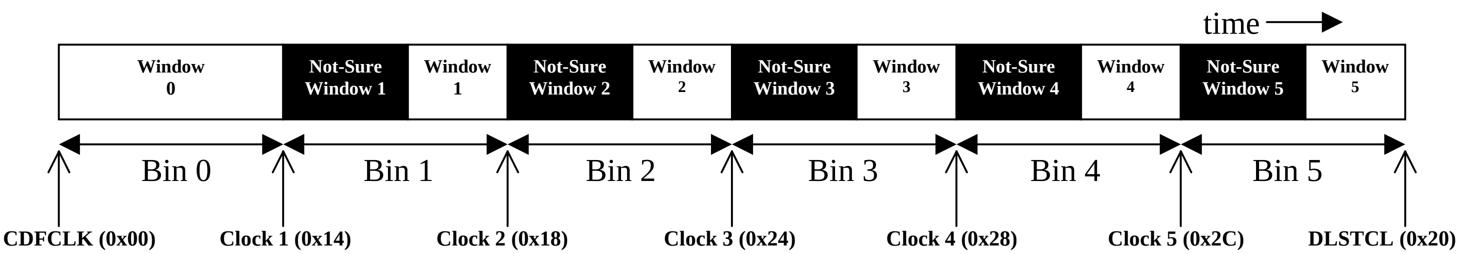

Figure 3: This figure shows which registers affect each window edge in the 6-bin design. The VME address for each register is given in hex next to the signal name.

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, or 1ns, depending on the chip). Thus, the total signal delay through any chip will be:

*Total Signal Delay = (Register Value) * (Step Size) + 16.5*

<u>Register Functionality</u>:

0x00 – Initial CDFCLK Delay

Step Size: 1ns (0.5ns on the ALT boards)

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: 1ns
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: 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

Step Size: 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.

0x10 – CDFB0 Delay

Step Size: 1ns
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: 1ns (0.5ns on the ALT boards)
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: 1ns (0.5ns on the ALT boards)
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: 1ns (0.5ns on the ALT boards)
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: 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: 1ns (0.5ns on the ALT boards)
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: 1ns (0.5ns on the ALT boards)
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: 1ns (0.5ns on the ALT boards)
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.

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 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.

<u>Register Functionality</u>:

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 was 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 – 0x47 (0 – 71)

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.

0x3C – 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.

0x40 – Buffer 0 Status

Valid Value Range: Only 0x00 and 0xFF (0 and 255)

Functionality: Read 0x00 or 0xFF; Write 0x00

Description: When read, this register indicates the current status of buffer 0. 0x00 indicates the buffer is empty; 0xFF indicates the buffer is full. Writing 0x00 will mark the buffer as free, allowing new data to be written to it when another level 1 accept event occurs.

0x44 – Buffer 1 Status

Valid Value Range: Only 0x00 and 0xFF (0 and 255)

Functionality: Read 0x00 or 0xFF; Write 0x00

Description: When read, this register indicates the current status of buffer 1. 0x00 indicates the buffer is empty; 0xFF indicates the buffer is full. Writing 0x00 will mark the buffer as free, allowing new data to be written to it when another level 1 accept event occurs.

0x48 – Buffer 2 Status

Valid Value Range: Only 0x00 and 0xFF (0 and 255)

Functionality: Read 0x00 or 0xFF; Write 0x00

Description: When read, this register indicates the current status of buffer 2. 0x00 indicates the buffer is empty; 0xFF indicates the buffer is full. Writing 0x00 will mark the buffer as free, allowing new data to be written to it when another level 1 accept event occurs.

0x4C – Buffer 3 Status

Valid Value Range: Only 0x00 and 0xFF (0 and 255)

Functionality: Read 0x00 or 0xFF; Write 0x00

Description: When read, this register indicates the current status of buffer 3. 0x00 indicates the buffer is empty; 0xFF indicates the buffer is full. Writing 0x00 will mark the buffer as free, allowing new data to be written to it when another level 1 accept event occurs.

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

The firmware version number of the two FPGA designs is stored in these read-only Kitchen Sink registers. 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, 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 registers are implemented in the Kitchen Sink).

<u>Current Design Number Definitions</u>:

2-Bin Data FPGA 0: Working 2-bin design

6-Bin Data FPGA 0: Working 6-bin design

6-Bin Data FPGA 1: Untested 6-bin design
Not-sure windows added

2-Bin Kitchen Sink FPGA 0: Working Kitchen Sink design used with the 2-Bin Data FPGA 0 design
Data buffers untested

6-Bin Kitchen Sink FPGA 0: Working Kitchen Sink design used with the 6-Bin Data FPGA 0 design
Data buffers untested

2-Bin Kitchen Sink FPGA 1: Untested Kitchen Sink design used with the 2-Bin Data FPGA 1 design
Logic for not-sure windows included
Data buffers untested

6-Bin Kitchen Sink FPGA 1: Untested Kitchen Sink design used with the 6-Bin Data FPGA 1 design
Logic for not-sure windows included
Data buffers untested

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.

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 window edges are affected by each register, and Table 2 lists the different window widths, obtained from simulation, for each valid register value.

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

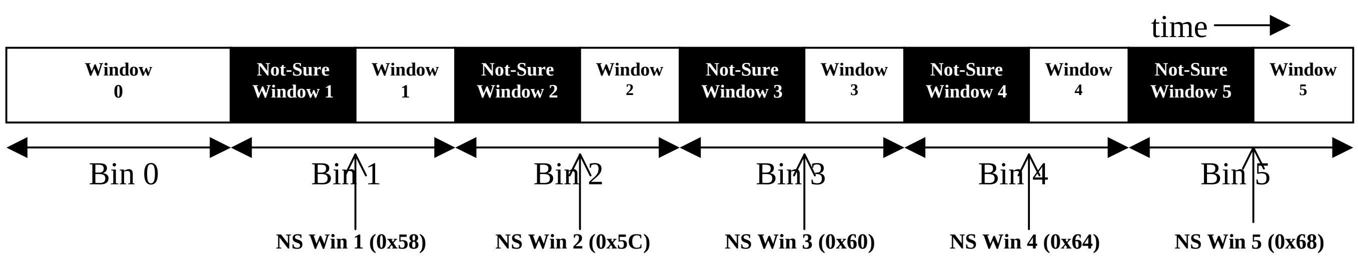

Figure 4: This figure shows which registers affect each not-sure window edge in the 6-bin design. The VME address for each register is given in hex next to the signal name.

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

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)
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 2: 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 Free Registers (0x6C – 0xDC)

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

3.6 Flash Memory Registers (0xE0 – 0xFC)

These registers, implemented in the FPGA-Programming 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 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 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-Programming CPLD that programming is complete by writing any data value to register 0xF0. This will begin configuration of the FPGAs.
8. Poll register 0xE8 until FPGA configuration is complete.

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

1. Write any data value to register 0xF0.

Register Functionality:

0xE0 – DO NOT USE THIS REGISTER

Valid Value Range: None
Functionality: None
Description: This register should neither be read nor written to, as doing so may have unintended effects.

0xE4 – DO NOT USE THIS REGISTER

Valid Value Range: None
Functionality: None
Description: This register should neither be read nor written to, as doing so may have unintended effects.

0xE8 – Read 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. 0x00 means the flash is ready for the next operation; 0xFF 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-Programming 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.

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-Programming 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-Programming CPLD to erase the flash memory.

4 Revisions

Date	Version	Modifications
10/27/2004	1.0	Original document
11/5/2004	1.1	Updated the "Current Design Number Definitions" in the Firmware Version Number Registers section (3.3)