1 Introduction
This note describes the functionality of the Output Data Looping firmware design for the XTC 2 TDC mezzanine card. It is intended to give instructions on the use of the design and how to interact with it. Contact the author of this document for further details on timing and how the logic is implemented.

2 Overview
The Output Data Looping design is intended to exercise the output path of the XTC 2 card, as well as provide an easy means of sending desired data patterns through the Transition cards to the Finder for testing and debugging purposes. The design consists of two FPGA firmware programs. Because it is necessary to provide some sort of configuration file to the Data FPGA, a simple design that sets the data bus outputs to high impedance to eliminate bus contention is used. The Data FPGA program serves no other purpose; all processing is done in the Kitchen Sink FPGA.

The Output Data Looping design contains a buffer that is 648 words deep by 32 bits wide. This provides enough space to store unique data for each 22-ns output clock cycle for an entire set of 36 beam crossings. Data is loaded into the buffer one byte at a time using the VME bus interface. An address counter starts at 0 on a bunch 0 crossing and increments every 22 ns except during the abort gaps. All necessary output synchronization signals—STROBE, WORD0, and output B0—are generated by the design and synchronized to the input B0 signal (there is some time, of course, between the input B0 pulse and the observance of the output B0 pulse). Two registers (0x00 and 0x04) are used to specify whether WORD0 occurs every 6 or 18 clock cycles, and whether the length of the output B0 pulse is 132 ns or 396 ns.

3 Register Space
Twelve registers compose the register space for this design. Table 1 gives a quick overview of these registers.
Table 1: This table summarizes the Output Data Looping 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 “Initial Contents” column specifies the values to which the registers are initialized.

<table>
<thead>
<tr>
<th>Address</th>
<th>R/W</th>
<th>Initial Contents</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>xE0-xFC (x38-x3F)</td>
<td>Var.</td>
<td>Var.</td>
<td>FPGA Prog CPLD Functions</td>
</tr>
<tr>
<td>x88-xDC (x22-x37)</td>
<td>---</td>
<td>---</td>
<td>UNUSED</td>
</tr>
<tr>
<td>x84 (x21)</td>
<td>R/W</td>
<td>x00</td>
<td>Transition Board Mode (Range: 0-3)</td>
</tr>
<tr>
<td>x7C-x80 (x1F-x20)</td>
<td>---</td>
<td>---</td>
<td>UNUSED</td>
</tr>
<tr>
<td>x78 (x1E)</td>
<td>W</td>
<td>x00</td>
<td>Transition Board FPGA Reprogram Signal (x00 = high, Else = low)</td>
</tr>
<tr>
<td>x74 (x1D)</td>
<td>---</td>
<td>---</td>
<td>UNUSED</td>
</tr>
<tr>
<td>x70 (x1C)</td>
<td>R</td>
<td>Ser. #</td>
<td>Bit 8 of the board serial number in bit 0 (lsb) of the register</td>
</tr>
<tr>
<td>x6C (x1B)</td>
<td>R</td>
<td>Ser. #</td>
<td>Bits 7:0 of the board serial number in bits 7:0 of the register</td>
</tr>
<tr>
<td>x24-x68 (x09-x1A)</td>
<td>---</td>
<td>---</td>
<td>UNUSED</td>
</tr>
<tr>
<td>x20 (x08)</td>
<td>W</td>
<td>---</td>
<td>Write byte to buffer (data = byte to be written)</td>
</tr>
<tr>
<td>x1C (x07)</td>
<td>R</td>
<td>---</td>
<td>Buffer byte</td>
</tr>
<tr>
<td>x18 (x06)</td>
<td>R/W</td>
<td>x00</td>
<td>Bits 11:8 of the memory address in bits 3:0 of the register</td>
</tr>
<tr>
<td>x14 (x05)</td>
<td>R/W</td>
<td>x00</td>
<td>Bits 7:0 of the memory address in bits 7:0 of the register</td>
</tr>
<tr>
<td>x10 (x04)</td>
<td>---</td>
<td>---</td>
<td>UNUSED</td>
</tr>
<tr>
<td>x0C (x03)</td>
<td>R</td>
<td>xF1</td>
<td>Board ID Register</td>
</tr>
<tr>
<td>x08 (x02)</td>
<td>R</td>
<td>Ver. #</td>
<td>Kitchen Sink FPGA Firmware Version Number</td>
</tr>
<tr>
<td>x04 (x01)</td>
<td>R/W</td>
<td>xFF</td>
<td>WRD0 frequency in 22-ns clock cycles (x00 = every 6, Else = every 18)</td>
</tr>
<tr>
<td>x00 (x00)</td>
<td>R/W</td>
<td>xFF</td>
<td>Output B0 length (x00 = 132ns, Else = 396ns)</td>
</tr>
</tbody>
</table>

Register Functionality:

0x00 – Output B0 Length
Valid Value Range: 0x00 – 0xFF (0 – 255)
Functionality: Read/Write
Description: The value written to this register determines the length of the output B0 pulse. A value of 0x00 will result in a short, 132-ns B0 pulse (used in the Michigan and UIUC 2-bin designs), while any other value will result in a longer, 396-ns B0 pulse (used in the UIUC 6-bin design).

0x04 – WORD0 Frequency
Valid Value Range: 0x00 – 0xFF (0 – 255)
Functionality: Read/Write
Description: The WORD0 pulse is synchronized with the output B0 pulse and has a duration of 22 ns. The value written to this register determines how often the pulse occurs. A value of 0x00 will result in a pulse every six 22-ns clock cycles (used in the 2-bin design), while any other value will result in a pulse every eighteen 22-ns clock cycles (used in the 6-bin design).
0x08 – Kitchen Sink FPGA Firmware Version Number
   **Valid Value Range:** N/A
   **Functionality:** Read-Only
   **Description:** The value stored in this register indicates the current version of the Kitchen Sink FPGA firmware.

0x0C – Kitchen Sink FPGA Firmware Version Number
   **Valid Value Range:** N/A
   **Functionality:** Read-Only
   **Description:** When read, the contents of this register alternate between 0xF1 and 0xDD.

0x14 – Memory Address (1)
   **Valid Value Range:** 0x00 – 0xFF (0 – 255)
   **Functionality:** Read/Write
   **Description:** This register contains the low 8 bits (7:0) of the memory buffer address. Changing the value of this register will result in an automatic read of the new address, the data from which can be read from register 0x1C.

0x18 – Memory Address (2)
   **Valid Value Range:** 0x00 – 0x0F (0 – 15)
   **Functionality:** Read/Write
   **Description:** This register contains the high 4 bits (11:8) of the memory buffer address. These bits are stored in the four least-significant bits of the register (3:0). Any values written to the upper 4 bits of this register will be ignored. Changing the value of this register will result in an automatic read of the new address, the data from which can be read from register 0x1C.

0x1C – Buffer Byte
   **Valid Value Range:** N/A
   **Functionality:** Read-Only
   **Description:** When an address value is written to registers 0x14 and 0x18, the byte is read from the memory buffer and stored in this register.

0x20 – Byte to Write
   **Valid Value Range:** 0x00 – 0xFF (0 – 255)
   **Functionality:** Write-Only
   **Description:** The byte written to this register will be written to the memory buffer at the address specified by registers 0x14 and 0x18.
0x6C – Board Serial Number (1)

Valid Value Range: N/A
Functionality: Read-Only
Description: This register contains the low 8 bits (7:0) of the board serial number, and is only valid for XTC 2 production boards (v2.1).

0x70 – Board Serial Number (2)

Valid Value Range: N/A
Functionality: Read-Only
Description: This register contains the high bit (8) of the board serial number, and is only valid for XTC 2 production boards (v2.1). This bit is stored in the least-significant bit of the register. Bits 7:1 of the register are all zeros.

0x78 – Transition Board FPGA Reprogram

Valid Value Range: 0x00 – 0xFF (0 – 255)
Functionality: Read/Write
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.

0x84 – Transition Board Mode

Valid Value Range: 0x00 – 0x03 (0 – 3)
Functionality: Read/Write
Description: This register is used to set the Transition Board operating mode. The values of the two least-significant bits of this register are placed on the two reserved lines that go from the XTC 2 to the Transition Board (PCB signals RESERVED_1 and RESERVED_0).

4 Operation

Looping through the data in the memory buffer begins immediately after FPGA configuration, and is synchronized with the input BC, B0, and CDFCLK signals. The user only needs to choose the WORD0 and B0 pulse settings using registers 0x00 and 0x04, and load the desired data into the memory buffer.

The procedure for loading data into the buffer is as follows:

1. Write the 12 bits of the address to registers 0x14 and 0x18.
2. Write the data byte to register 0x20.
3. Repeat steps 1 and 2 until all desired data has been written. The default value for all data bytes is 0x00.

See the code in data_loop.c under the “Write pattern to buffers” comment for an example of how to store data in the first 54 words of the memory buffer.
The procedure for reading back the data from the buffer for verification is as follows:
1. Write the 12 bits of the address to registers 0x14 and 0x18.
2. Read the data byte from register 0x1C.
3. Compare the value read back with what was written.
4. Repeat steps 1-3 until all desired data has been verified.

Figure 1 shows a diagram of the memory buffer, along with the addressing scheme used. Bytes may be written in any order. Words 0-215 (each 32 bits in length) are transmitted in succession every 22 ns, and represent the first train of 12 bunch crossings. Data from word 215 is held during the abort gap. Then words 216-431 are transmitted, corresponding to the second train, and word 431 is held during the abort gap. Finally, words 432-647 are transmitted, with word 647 being held during the abort gap. Word 0 is then sent at the beginning of the next bunch 0 crossing.

<table>
<thead>
<tr>
<th>Bits</th>
<th>Word #</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>31:24</td>
<td>0 1 2 3</td>
<td>215 216</td>
<td>431 432</td>
<td>647 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>23:16</td>
<td>3 7 11 15</td>
<td>863 867</td>
<td>1727 1731</td>
<td>2591 9</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>15:8</td>
<td>2 6 10 14</td>
<td>862 866</td>
<td>1726 1730</td>
<td>2590 7</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7:0</td>
<td>1 5 9 13</td>
<td>861 865</td>
<td>1725 1729</td>
<td>2589 6</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0 4 8 12</td>
<td>860 864</td>
<td>1724 1728</td>
<td>2588 5</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

5 Revisions

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Modifications</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/17/2005</td>
<td>1.0</td>
<td>Original document</td>
</tr>
<tr>
<td>5/9/2005</td>
<td>1.1</td>
<td>-Modified initial contents of register 0x08 in Table 1</td>
</tr>
<tr>
<td>9/20/2005</td>
<td>1.2</td>
<td>-Updated Table 1: added registers 0x78 and 0x84; noted that the register range 0xE0-0xFC is already used by the FPGA_Prog CPLD -Updated Section 3: added register definitions for registers 0x78 and 0x84</td>
</tr>
</tbody>
</table>
12/2/2005  1.3

-Updated Table 1: moved registers 0x0C and 0x10 to registers 0x6C and 0x70; redefined register 0x0C and left register 0x10 unused
-Updated Section 3: moved the register definitions for registers 0x0C and 0x10 to 0x6C and 0x70; redefined register 0x0C; removed register 0x10; updated the section text to reflect these changes

For questions, comments, or problems, contact:

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