MOSES

FPGA Design Guide

Justin Hogan

September 15, 2014

Contents

[Introduction 3](#_Toc402786182)

[Theory of Operation 3](#_Toc402786183)

[Buffering of Image Data 3](#_Toc402786184)

[FPGA Memory Map 4](#_Toc402786185)

[Input GPIO Bit Map 5](#_Toc402786186)

[DDR2\_FSM\_STATE Signal Value Enumeration 6](#_Toc402786187)

[Output GPIO Bit Map – output\_gpio\_register 7](#_Toc402786188)

[Output GPIO Bit Map – output\_ddr2\_addr\_register 7](#_Toc402786189)

[Output GPIO Bit Map – output\_ddr2\_ctrl\_register 8](#_Toc402786190)

[DDR2 Memory Map 9](#_Toc402786191)

[Operating the Camera Interface 9](#_Toc402786192)

[Debugging the Design 12](#_Toc402786193)

[Xilinx Internal Logic Analyzer 12](#_Toc402786194)

[Creating a Custom DDR2 Part 13](#_Toc402786195)

# Introduction

This document describes the functionality of the FPGA design implemented on the FreeForm/PCI-104 hardware. Included in this description are summaries of the hardware and software interfaces between the FPGA logic its surrounding systems. Of particular interest is capturing the way in which a host PC accesses the GPIO registers and the DDR2 memory.

The FreeForm/PCI-104 is a general-purpose FPGA board designed for inclusion in a PC/104 computer stack. These computer stacks generally feature a main computer board which accesses application-specific peripheral boards over a parallel bus. Originally this bus was an Industry Standard Architecture (ISA) bus. Advances in technology led to revisions of the PC/104 standard to include the availability of a PCI bus for accessing peripheral cards within the stack. The FreeForm/PCI-104 board conforms to the PCI-104 form factor, which enables access to the FPGA by the host processor via PCI bus accesses. These accesses are enabled by the PLX PCI 9056, which is a 32-bit, 66-MHz PCI Bus interface device.

The design described in this document is a system component of the Multi-Order Solar Extreme Ultraviolet Spectrograph (MOSES) sounding rocket payload. The design has four primary functions: buffer acquired camera data to local DDR2 memory, transfer acquired camera data from DDR2 to the host PC for storage via direct memory access (DMA), register general purpose output signals for payload command and control, and register general purpose input signals for payload command and control.

The MOSES instrument contains a CCD imaging system with a 16-bit parallel interface for data readout. Read-out is configured and performed by a system component referred to as the read-out electronics (ROE). Parallel data are synchronous to a 2-MHz clock. The data bus and clock signals are input to the FPGA, which aligns the data to the local 50-MHz clock, registers the data, and writes the data to local memory. After synchronization, the data are written to the local DDR2 memory for subsequent transfer to the host processor for storage and telemetry. There are a variety of general-purpose inputs and outputs that are implemented through FPGA logic. The input signals are used as event triggers to signal the host PC that action is required. The output signals are used to control system operation. The specifics of each input and output signal is detailed in a subsequent section.

# Theory of Operation

## Buffering of Image Data

Image data are input to the FPGA design via the GPIO header on the board edge. The data are processed as a 16-bit parallel interface with an accompanying 2-MHz pixel clock.

Clock Domain Synchronization

The data and pixel clock inputs are considered to be asynchronous signals necessitating synchronization with the 50-MHz system clock. The process of synchronization ensures that transitions on these input signals occur at appropriate times with respect to setup and hold times for registers. That is, synchronization ensures that the input data values are sampled when they are stable. A common technique for accomplishing this

Matching DDR2 Data Bus Width

# FPGA Memory Map

The local bus address bits 31:2 define the accessible memory space of the FPGA design. The following memory map is specific to these bits. In the host processor software, these values must be bit-shifted to the appropriate position in order to properly address the local bus peripherals. For example, if lb\_la(31:2) equals 0x01, the address value in software must be 0x01 << 2 = 0x04.

The buffered camera data is stored to the DDR2 memory. The FreeForm/PCI-104 FPGA board contains two 8Meg x 16 x 4-banks DDR2 memory devices. These devices share the same address but thereby providing an 8Meg x 32 x 4-banks memory space to the FPGA. The data bus to the memory in the FPGA design is 32-bits. The total capacity of the DDR2 memory is 128MB. Each CCD captures a 1024x2048x16-bit array of data. Since there are four CCDs this results in a requirement to buffer 16,777,216 bytes. This is equivalent to 16MB. As the data is split between the two memory chips each chip is required to store 8MB. Each data location in the memory chips stores two bytes, so the FPGA design must address of 4MB memory space (0 to 4,194,303). In order to accomplish this, the design makes use of 22 of the available 31 address bits. The address values for buffered camera frame data will then range from 0x00000000 to 0x003FFFFF. This address range will be offset on the PCI9056 local bus by the value in the corresponding address space configuration registers. Per the “PCI 9056BA Data Book”, the PCI9056 has three Local Address spaces (Space 0, Space 1, and Expansion ROM). Only Space 0 and Space 1 are applicable to this discussion. Each address space is characterized by three registers, which must be initialized for proper operation. These registers are the Local Address Space Range Register (LAS*x*RR) (where *x* is the space number), the Local Address Space Local Base Address (LAS*x*BA), and the PCI Base Address Register (PCIBAR2, PCIBAR3).

For GPIO-associated functionality, the design uses PCIBAR2. The

# FPGA Memory Map

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ADDRESS** | **SOFTWARE ADDRESS** | **REGISTER NAME** | **REGISTER DESCRIPTION** | **DIRECTION** |
| 0x00 | 0x00 | INPUT\_GPIO | GPIO input register values | READ |
| 0x01 | 0x04 | INPUT\_GPIO\_INTERRUPT\_ENABLE | This enables interrupts on the rising edge of INPUT\_GPIO signals. | WRITE |
| 0x02 | 0x80 | INPUT\_GPIO\_INTERRUPT\_ACK | This clears interrupts on the corresponding bits in INPUT\_GPIO\_INTERRUPT\_REGISTER | WRITE |
| 0x03 | 0x0C | INPUT\_GPIO\_INTERRUPT | This register tells the process which input changed. Interrupts trigger only on transitions from LOW to HIGH on the input GPIO pins | READ |
| 0x04 | 0x10 | COUNTER\_PERIPHERAL | This register contains the current count value of a 32-bit, 1-Hz counter. This is simply used for input GPIO functionality testing, but may also be used to indicate how long the system has been running since a reset. | READ |
| 0x05 | 0x14 | OUTPUT\_GPIO | This register drives the output GPIO signals. Some of these signals are routed to external pins for command and control of payload components. Some are routed internally for control of the FPGA design. | WRITE |
| 0x06 | 0x18 | OUTPUT\_DDR2\_ADDR | This register contains the read address bits for the DDR2 memory. Currently only used for testing purposes. | WRITE |
| 0x07 | 0x1C | OUTPUT\_DDR2\_CTRL | This register provides signals to control the DDR2 read interface. Currently only used for testing purposed. | WRITE |
| 0x08 | 0x20 | DDR2\_READ\_EN | Asserting this address on the local bus interface activates the DDR2 interface for read operation. For DMA, the read address is generated internal to the FPGA therefore data must be read out sequentially. The DDR2 acts like a FIFO, the DMA continually reads data out of the FIFO. | WRITE |

# Input GPIO Bit Map

|  |  |  |  |
| --- | --- | --- | --- |
| **BIT NUMBER** | **NAME** | **DESCRIPTION** | **INTERNAL/EXTERNAL** |
| 0 | UPLINK1\_LVL |  | E |
| 1 | UPLINK2\_LVL |  | E |
| 2 | UPLINK3\_LVL |  | E |
| 3 | UPLINK4\_LVL |  | E |
| 4 | UPLINK5\_LVL |  | E |
| 5 | UPLINK6\_LVL |  | E |
| 6 | UPLINK7\_LVL |  | E |
| 7 | UPLINK8\_LVL |  | E |
| 8 | UPLINK9\_LVL |  | E |
| 9 | TIMER1\_LVL |  | E |
| 10 | TIMER2\_LVL |  | E |
| 11 | TIMER3\_LVL |  | E |
| 12 | TIMER4\_LVL |  | E |
| 13 | TIMER5\_LVL |  | E |
| 14 | SS\_LVL |  | E |
| 15 | MOSES\_INPUT(reserved) |  | E |
| 16 | MOSES\_INPUT(reserved) |  | E |
| 17 | MOSES\_INPUT(reserved) |  | E |
| 18 | MOSES\_INPUT(reserved) |  | E |
| 19 | MOSES\_INPUT(reserved) |  | E |
| 20 | MOSES\_INPUT(reserved) |  | E |
| 21 | MOSES\_INPUT(reserved) |  | E |
| 22 | MOSES\_INPUT(reserved) |  | E |
| 23 | MOSES\_INPUT(reserved) |  | E |
| 24 | MOSES\_INPUT(reserved) |  | E |
| 25 | MOSES\_INPUT(reserved) |  | E |
| 26 | DDR2\_DATA\_ERROR\_FLAG | Active-high flag indicating invalid data read from DDR2 | I |
| 27 | DDR2\_FSM\_STATE(0) | DDR2 Data Manager finite-state machine status signal | I |
| 28 | DDR2\_FSM\_STATE(1) | DDR2 Data Manager finite-state machine status signal | I |
| 29 | DDR2\_FSM\_STATE(2) | DDR2 Data Manager finite-state machine status signal | I |
| 30 | DDR2\_FSM\_STATE(3) | DDR2 Data Manager finite-state machine status signal | I |
| 31 | FRAME\_READY\_SIGNAL | This flag triggers in interrupt signaling the VDX that one of the following has occurred:   1. A complete image set has been buffered to DDR2 2. A timer has expired indicating incomplete frame buffering has occurred. | I |

# DDR2\_FSM\_STATE Signal Value Enumeration

|  |  |  |  |
| --- | --- | --- | --- |
| **STATE VALUE** | **STATE NAME** | **DESCRIPTION** | **NEXT STATES** |
| 0 | IDLE | Data pipeline is reset, camera and memory interfaces are idle | BUFFER\_DATA0 |
| 1 | BUFFER\_DATA0 | Awaiting first 64-bits of data to arrive from camera interface. Transitions to next state when these bits are loaded into the DDR2 write-data FIFO. | BUFFER\_DATA1 |
| 2 | BUFFER\_DATA1 | Awaiting second 64-bits of data to arrive from camera interface. Transitions to next state when these bits are loaded into the DDR2 write-data FIFO. | BUFFER\_LOAD\_ADDR |
| 3 | BUFFER\_LOAD\_ADDR | Loads write address and command into the DDR2 address/command FIFO causing the first 128-bits of data to be written to memory in a Burst-4 transfer. | DMA\_INIT if frame is complete, otherwise BUFFER\_DATA0 to continue frame buffering. |
| 4 | DMA\_INIT | This state resets the memory address value and assigns control signals to the PCI local bus interface. | DMA\_IDLE |
| 5 | DMA\_IDLE | The FSM waits here for DMA activity to come over the PCI local bus. | DMA\_IDLE until PCI access is detected then DMA\_DDR\_FETCH. IDLE when the control signal is asserted. |
| 6 | DMA\_DDR\_FETCH | This state issues a read command to the DDR2 interface and increments the memory address so that it’s ready for the next read cycle. | DMA\_DATA\_WAIT |
| 7 | DMA\_DATA\_WAIT | The FSM waits in this state for the requested data to appear in the DDR2 read FIFO. | DMA\_DATA0 |
| 8 | DMA\_DATA0 | Places first 32-bits of read data on the local bus. | DMA\_DATA1 |
| 9 | DMA\_DATA1 | Places second 32-bits of read data on the local bus. | DMA\_DATA2 |
| 10 | DMA\_DATA2 | Places third 32-bits of read data on the local bus. | DMA\_DATA3 |
| 11 | DMA\_DATA3 | Places fourth 32-bits of read data on the local bus. | DMA\_IDLE if lb\_blastn (last byte signal) is asserted low otherwise DMA\_DDR\_FETCH to continue the DMA transfer. |
| 12 | *RESERVED* |  |  |
| 13 | *RESERVED* |  |  |
| 14 | *RESERVED* |  |  |
| 15 | *RESERVED* |  |  |

# Output GPIO Bit Map – output\_gpio\_register

This section details the output GPIO signals used in the FPGA design. These signals include both internally and externally used signals. This status is indicated in the table below. Internal signals are those which are used internal to the FPGA for the purpose of controlling operation of the FPGA logic. External signals are used for controlling elements of the spacecraft external to the FPGA board itself. The output GPIO register is a 32-bit wide, memory-mapped register accessible to the VDX processor via the PCI 9056 bridge. The

|  |  |  |  |
| --- | --- | --- | --- |
| **BIT NUMBER** | **NAME** | **DESCRIPTION** | **INTERNAL/EXTERNAL** |
| 0 | LATCH\_CLOCK |  | E |
| 1 | J31\_SWITCH |  | E |
| 2 | J30\_SWITCH |  | E |
| 3 | TCS4\_SWITCH |  | E |
| 4 | TCS2\_SWITCH |  | E |
| 5 | TCS3\_SWITCH |  | E |
| 6 | PMF\_SWITCH |  | E |
| 7 | TCS1\_SWITCH |  | E |
| 8 | SHUTTER\_SWITCH |  | E |
| 9 | ROE\_SWITCH |  | E |
| 10 | HA\_SWITCH |  | E |
| 11 | CC\_PWR |  | E |
| 12 | LED\_0 |  | E |
| 13 | LED\_1 |  | E |
| 14 | MOSES(reserved) |  |  |
| 15 | MOSES(reserved) |  |  |
| 16 | MOSES(reserved) |  |  |
| 17 | MOSES(reserved) |  |  |
| 18 | MOSES(reserved) |  |  |
| 19 | MOSES(reserved) |  |  |
| 20 | MOSES(reserved) |  |  |
| 21 | MOSES(reserved) |  |  |
| 22 | MOSES(reserved) |  |  |
| 23 | MOSES(reserved) |  |  |
| 24 | MOSES(reserved) |  |  |
| 25 | MOSES(reserved) |  |  |
| 26 | CAMERA\_LOGIC\_RESET | Active-low signal resets the DDR2/Camera Interface finite-state machine as well as the camera data input pipeline registers. | I |
| 27 | FRAME\_TRIGGER | Assertion of this bit initiates frame generation in the camera simulation | I |
| 28 | SOFTWARE\_SHUTDOWN | Assertion of this bit initiates the shutdown countdown sequence. | I |
| 29 | MOSES(reserved) |  |  |
| 30 | CAMERA\_MUX\_SEL | ‘1’ = ROE INTERFACE, ‘0’ = SIM | I |
| 31 | MOSES(reserved) |  |  |

# Output GPIO Bit Map – output\_ddr2\_addr\_register

|  |  |  |  |
| --- | --- | --- | --- |
| **BIT NUMBER** | **NAME** | **DESCRIPTION** | **INTERNAL/EXTERNAL** |
| 0 | ADDR\_0 | DDR2 Read Address Bit | I |
| 1 | ADDR\_1 | DDR2 Read Address Bit | I |
| 2 | ADDR\_2 | DDR2 Read Address Bit | I |
| 3 | ADDR\_3 | DDR2 Read Address Bit | I |
| 4 | ADDR\_4 | DDR2 Read Address Bit | I |
| 5 | ADDR\_5 | DDR2 Read Address Bit | I |
| 6 | ADDR\_6 | DDR2 Read Address Bit | I |
| 7 | ADDR\_7 | DDR2 Read Address Bit | I |
| 8 | ADDR\_8 | DDR2 Read Address Bit | I |
| 9 | ADDR\_9 | DDR2 Read Address Bit | I |
| 10 | ADDR\_10 | DDR2 Read Address Bit | I |
| 11 | ADDR\_11 | DDR2 Read Address Bit | I |
| 12 | ADDR\_12 | DDR2 Read Address Bit | I |
| 13 | ADDR\_13 | DDR2 Read Address Bit | I |
| 14 | ADDR\_14 | DDR2 Read Address Bit | I |
| 15 | ADDR\_15 | DDR2 Read Address Bit | I |
| 16 | ADDR\_16 | DDR2 Read Address Bit | I |
| 17 | ADDR\_17 | DDR2 Read Address Bit | I |
| 18 | ADDR\_18 | DDR2 Read Address Bit | I |
| 19 | ADDR\_19 | DDR2 Read Address Bit | I |
| 20 | ADDR\_20 | DDR2 Read Address Bit | I |
| 21 | ADDR\_21 | DDR2 Read Address Bit | I |
| 22 | ADDR\_22 | DDR2 Read Address Bit | I |
| 23 | ADDR\_23 | DDR2 Read Address Bit | I |
| 24 | ADDR\_24 | DDR2 Read Address Bit | I |
| 25 | ADDR\_25 | DDR2 Read Address Bit | I |
| 26 | ADDR\_26 | DDR2 Read Address Bit | I |
| 27 | ADDR\_27 | DDR2 Read Address Bit | I |
| 28 | ADDR\_28 | DDR2 Read Address Bit | I |
| 29 | ADDR\_29 | DDR2 Read Address Bit | I |
| 30 | ADDR\_30 | DDR2 Read Address Bit | I |
| 31 | ADDR\_31 | DDR2 Read Address Bit | I |

# Output GPIO Bit Map – output\_ddr2\_ctrl\_register

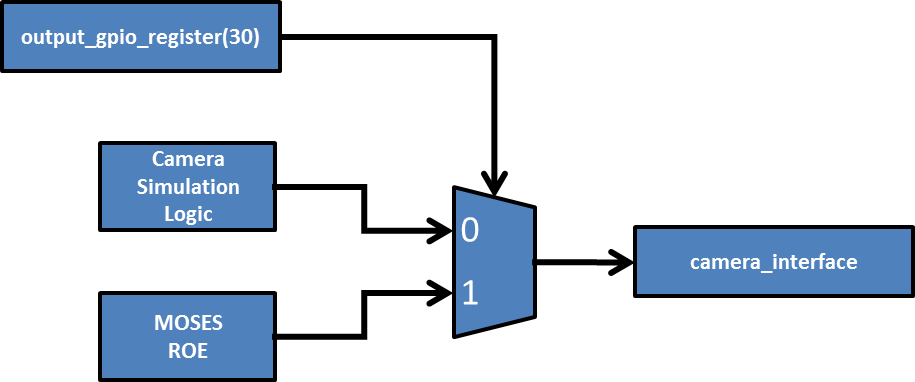
|  |  |  |  |
| --- | --- | --- | --- |
| **BIT NUMBER** | **NAME** | **DESCRIPTION** | **INTERNAL/EXTERNAL** |
| 0 | DDR2\_AF\_WREN | This pin drives the DDR2 address FIFO write-enable signal. | I |
| 1 |  | UNUSED | I |
| 2 |  | UNUSED | I |
| 3 |  | UNUSED | I |
| 4 |  | UNUSED | I |
| 5 |  | UNUSED | I |
| 6 |  | UNUSED | I |
| 7 |  | UNUSED | I |
| 8 |  | UNUSED | I |
| 9 |  | UNUSED | I |
| 10 |  | UNUSED | I |
| 11 |  | UNUSED | I |
| 12 |  | UNUSED | I |
| 13 |  | UNUSED | I |
| 14 |  | UNUSED | I |
| 15 |  | UNUSED | I |
| 16 |  | UNUSED | I |
| 17 |  | UNUSED | I |
| 18 |  | UNUSED | I |
| 19 |  | UNUSED | I |
| 20 |  | UNUSED | I |
| 21 |  | UNUSED | I |
| 22 |  | UNUSED | I |
| 23 |  | UNUSED | I |
| 24 |  | UNUSED | I |
| 25 |  | UNUSED | I |
| 26 |  | UNUSED | I |
| 27 |  | UNUSED | I |
| 28 |  | UNUSED | I |
| 29 |  | UNUSED | I |
| 30 |  | UNUSED | I |
| 31 |  | UNUSED | I |

# DDR2 Memory Map

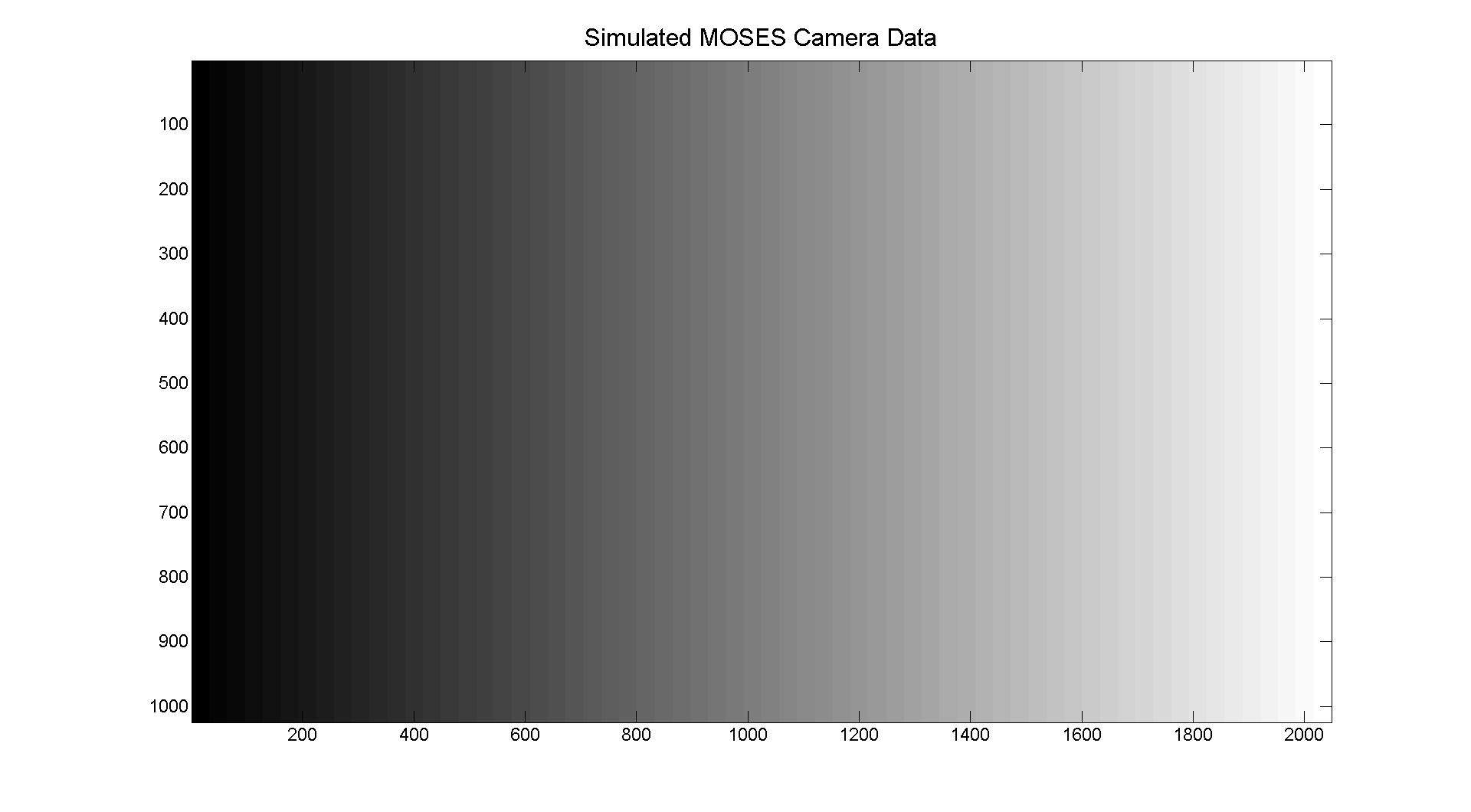
# Operating the Camera Interface

The camera interface logic uses handshaking and status monitoring to initiate exposures and mediate the transfer of data from the FPGA system to processor memory via DMA over the PCI bus. There exists a logic component which controls the interface to the DDR2 memory. This logic is a finite-state machine the default behavior of which is to accept data from the camera interface and write it to the DDR2 memory. This state is referred to as the “buffer” state. Upon detection of a complete frame transfer to memory, the logic then transitions to a in which it responds to DDR2 read requests on the PCI local bus. This state is referred to as the “DMA” state. Upon completion of a DMA transfer the logic returns to buffer mode and awaits activity on the camera interface. Completion of a DMA transfer is signaled by assertion of output\_gpio\_register(29) by the host PC. Detection of activity on the camera interface before a frame transfer is complete results in a transition to an error state to indicate that a frame was missed. If the handshaking between the host PC software and the FPGA hardware is done properly, such a scenario *should* never occur.

The camera interface includes logic designed to simulate data arriving from the ROE. The simulated camera logic is multiplexed with the input signals arriving from the ROE. The camera source is selected via a memory-mapped output GPIO bit (output\_gpio\_register(30)). Setting this select bit passes the MOSES ROE data to the camera interface logic. Clearing this select bit passes the internally simulated camera data to the camera interface logic.

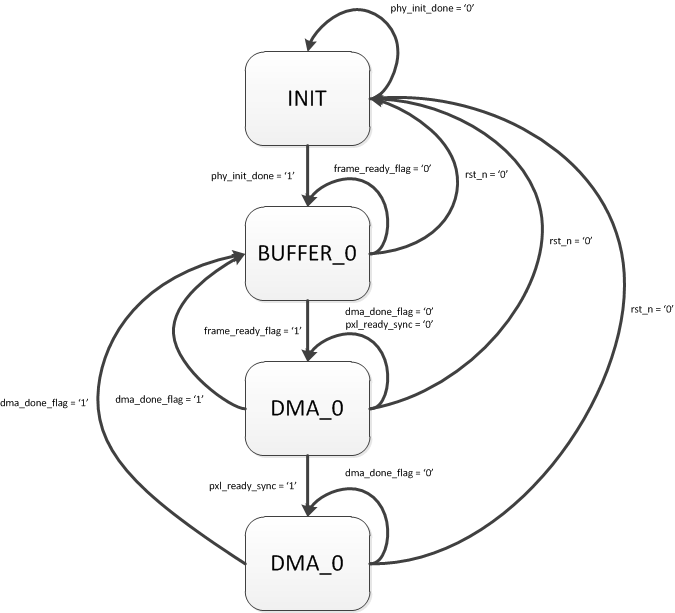


The camera interface includes 16-bit parallel data along with a 2-MHz pixel clock. The two most significant bits are used as a frame identifier indicating from which CCD on the instrument the data originated, and by extension, to what spectral band the data belong. There are a total of four frames of dimension 1024x2048x16. There are three data frames and one noise frame. All frames are buffered and retained for processing. To simulate camera data two counters are used. The first counter is a two bit counter which generates the CCD identification bits. The second counter is a 14-bit data counter which increments across columns from 0 to 2047, repeating for each row. An image representing the expected data from the simulated camera interface is shown below. Each CCD frame contains identical data.



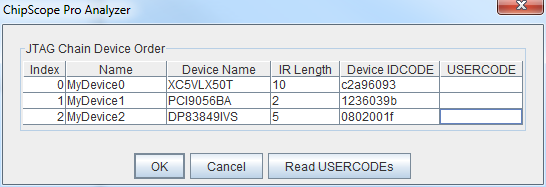
For testing purposes, the camera simulation interface is selected to feed the camera interface. The simulation logic idly awaits a frame trigger signal from the host PC. The generation of a frame is initiated by asserting **output\_gpio\_register(27)**. Once this signal is asserted, the counters activate along with the 2-MHz clock generation logic to provide data to the system. The camera interface detects each rising edge of the 2-MHz clock and latches the data accordingly. The camera interface shifts in two data words at a time to present a 32-bit data vector to the DDR2 control logic. Once two data words are present in the shift register, the camera interface latches the value to a 32-bit register and asserts a flag to indicate to the DDR2 control logic that valid data is available on the memory write bus.

The DDR2 control logic

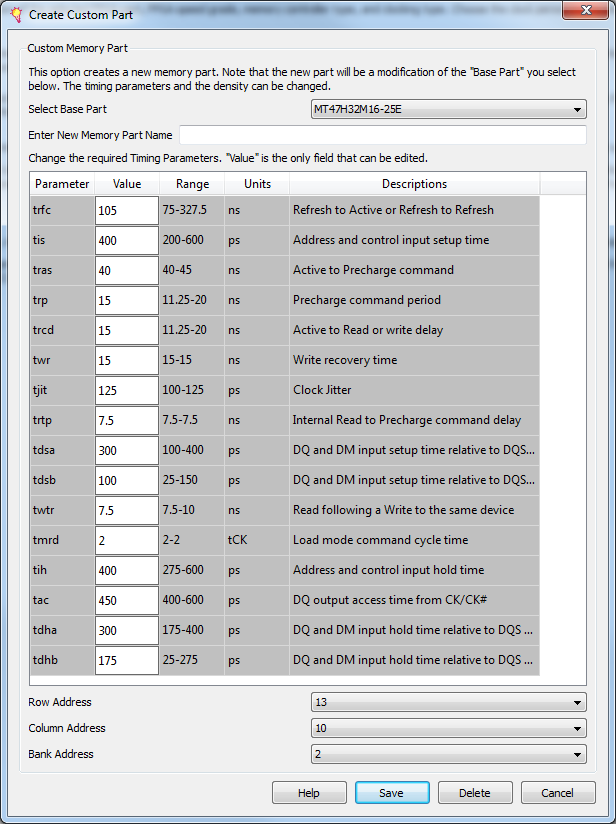


# Debugging the Design

## Xilinx Internal Logic Analyzer



# Creating a Custom DDR2 Part



Requires modification post-generation to remove global clock buffers from input clock path (when the core is generated to include internal PLL resources rather than generating clk0, clk90 and clk\_div externally).

The following file modifications were made to ./ipcore\_dir/<*ddr2\_core\_directory*>/user\_design/rtl/ddr2\_infrastructure.vhd

Additionally, in order for the changes to the generated \*.vhd files to take effect, the generated core (\*.xco) must be removed from the project and replaced with all the \*.vhd files in the ./user\_design/rtl/ directory. In this way the MIG-generated files are used as a starting point for implementing the desired functionality and custom changes can be made in order for the design to be synthesized and implemented properly.

1. In response to the following errors in the translate process:
   1. NgdBuild:770 - IBUFG 'SINGLE\_ENDED\_CLKS\_INST.SYS\_CLK\_INST' and BUFG 'ddr2\_clk200\_BUFG' on net 'ddr2\_clk0' are lined up in series. Buffers of the same direction cannot be placed in series.
   2. Comment out the SYS\_CLK\_INST instance of IBUFG and route the input clock signal sys\_clk directly into the PLL.