

Terasic D5M Camera Sensor

ECE492 – Winter 2014 - Group 9

Brent Erickson

Mavis Chan

Sydney Bitner

Introduction

Terasic provides a 5MP camera sensor that is compatible with the DE2 board. There is supplied University IP that helps to interface with and configure the camera. This tutorial will demonstrate the hardware setup required in order to write Camera data to SRAM.

QSys

Required Components

- Nios II Processor
- Audio and Video Config
- Video-In Decoder
- Bayer Pattern Resampler
- Clipper
- Scaler
- DMA Controller
- SRAM/SSRAM Controller

Usage

Adding the above components and setting the given configuration values will output a 320x240 8-bit color image.

Audio and Video Config

The screenshot displays the 'Audio and Video Config' window for a camera. The window is titled 'Audio and Video Config - camera' and contains the following elements:

- Block Diagram:** Shows a 'camera' block connected to 'altera_up_avalon_audio_and_video_config'. The connections are labeled: 'clock' (clock_reset), 'reset' (clock_reset_reset), 'avalon' (avalon_av_config_slave), and 'conduit' (external_interface). There is a 'Show signals' checkbox.
- Components:** Audio/Video Device: 5 Megapixel Camera (TRDB_D5M), DE Board: DE2, and a checked 'Auto Initialize Device(s)' option.
- Auto Initialization Parameters for Audio:** Audio In Path: Microphone to ADC, Audio Out - Enable DAC Output (checked), Audio Out - Microphone Bypass (unchecked), Audio Out - Line In Bypass (unchecked), Data Format: Left Justified, Bit Length: 24, and Sampling Rate: 48 kHz.
- Auto Initialization Parameters for Video:** Video Source Format: NTSC.
- Auto Initialization Parameters for 5 Megapixel Camera (TRDB_D5M):** Resolution: 2592 x 1944, and an unchecked 'Enable external exposure port' option.

Buttons for 'Cancel' and 'Finish' are located at the bottom right of the window.

Video-In Decoder

The screenshot shows the configuration window for the Video-In Decoder. The title bar reads "Video-In Decoder - D5M_Decoder". The window is divided into three main sections:

- Block Diagram:** Shows a block named "D5M_Decoder" with three input ports: "clock_reset" (clock), "clock_reset_reset" (reset), and "external_interface" (conduit). The block is connected to "avalon_decoder_source" and "altera_up_avalon_video_decoder".
- Parameters:** The "Video-In Source" is set to "SMP Daughtercard (THDB_D5M)".
- Info:** A message box states: "Info: D5M_Decoder: Video In Stream: Format: 2592 x 1944 (default) with Colour: 8 (bits) x 1 (planes) (RGB Bayer Pattern)".

Buttons for "Cancel" and "Finish" are located at the bottom right.

Bayer Pattern Resampler

The screenshot shows the configuration window for the Bayer Pattern Resampler. The title bar reads "Bayer Pattern Resampler - video_bayer_resampler_0". The window is divided into three main sections:

- Block Diagram:** Shows a block named "video_bayer_resampler_0" with three input ports: "clock_reset" (clock), "clock_reset_reset" (reset), and "avalon_bayer_sink" (avalon_streaming). The block is connected to "avalon_bayer_source" and "altera_up_avalon_video_bayer_resampler".
- Parameters:** The "Video Source" is set to "SMP daughtercard (THDB_D5M)". Under the "Input Video Format" section, the "Use default video format" checkbox is checked. The "Width (# of pixels)" is set to 2592 and the "Height (# of lines)" is set to 1944.
- Info:** A message box states: "Info: video_bayer_resampler_0: Change in Resolution: 2592 x 1944 -> 1296 x 972".

Buttons for "Cancel" and "Finish" are located at the bottom right.

Clipper

The screenshot shows the Altera Clipper configuration window for the component `video_clipper_0`. The window is titled "Clipper - video_clipper_0" and includes a "Documentation" button in the top right corner. The main interface is divided into two primary sections: a "Block Diagram" on the left and a configuration panel on the right.

Block Diagram: This section shows a block diagram of the `video_clipper_0` component. It has three input signals: `clock_reset`, `clock_reset_reset`, and `avalon_clipper_sink`. The component is connected to an `avalon_streaming` block, which in turn connects to an `avalon_clipper_source` block. The diagram is labeled with `video_clipper_0` and `altera_up_avalon_video_clipper`.

Configuration Panel: This panel contains several sections for configuring the clipper's behavior:

- Incoming Frame Resolution:** Width (# of pixels): 1296, Height (# of lines): 972.
- Reduce Frame Size:** Columns to remove from the left side: 8, Columns to remove from the right side: 8, Rows to remove from the top: 6, Rows to remove from the bottom: 6.
- Enlarge Frame Size:** Columns to add to the left side: 0, Columns to add to the right side: 0, Rows to add to the top: 0, Rows to add to the bottom: 0. Added pixel values for planes 1, 2, 3, and 4 are all set to `0x00000000`.
- Pixel Format:** Color Bits: 8, Color Planes: 3.

At the bottom of the window, an information message states: "Info: video_clipper_0: Change in Resolution: 1296 x 972 -> 1280 x 960". The window concludes with "Cancel" and "Finish" buttons.

Scaler

The screenshot shows the Altera Scaler configuration window for the component `video_scaler_0`. The window is titled "Scaler - video_scaler_0" and includes a "Documentation" button in the top right corner. The main interface is divided into two primary sections: a "Block Diagram" on the left and a configuration panel on the right.

Block Diagram: This section shows a block diagram of the `video_scaler_0` component. It has three input signals: `clock_reset`, `clock_reset_reset`, and `avalon_scaler_sink`. The component is connected to an `avalon_streaming` block, which in turn connects to an `avalon_scaler_source` block. The diagram is labeled with `video_scaler_0` and `altera_up_avalon_video_scaler`.

Configuration Panel: This panel contains several sections for configuring the scaler's behavior:

- Scaling Parameters:** Width Scaling Factor: 0.25, Height Scaling Factor: 0.25.
- Incoming Frame Resolution:** Width (# of pixels): 1280, Height (# of lines): 960.
- Pixel Format:** Color Bits: 8, Color Planes: 3.

At the bottom of the window, an information message states: "Info: video_scaler_0: Change in Resolution: 1280 x 960 -> 320 x 240". The window concludes with "Cancel" and "Finish" buttons.

DMA Controller

The screenshot shows the 'DMA Controller - video_dma_controller_0' configuration window. It is divided into two main sections: 'Block Diagram' and 'Configurations'.

Block Diagram: A block diagram of the 'altera_up_avalon_video_dma_controller' component. It shows four input signals on the left: 'clock_reset', 'clock_reset_reset', 'avalon_dma_sink', and 'avalon_dma_control_slave'. These connect to the 'clock', 'reset', 'avalon_streaming', and 'avalon' ports of the 'video_dma_controller_0' block. The 'avalon' port is connected to an 'avalon_dma_master' block.

Configurations:

- Addressing Parameters:** Addressing Mode: Consecutive; Default Buffer Starting Address: 0x00000000; Default Back Buffer Start Address: 0x00000000.
- Frame Resolution:** Width (# of pixels): 320; Height (# of lines): 240.
- Pixel Format:** Color Bits: 8; Color Planes: 3.
- Mode:** DMA Direction: From Stream to Memory.

Buttons for 'Cancel' and 'Finish' are located at the bottom right.

SRAM/SSRAM Controller

The screenshot shows the 'SRAM/SSRAM Controller - sram_0' configuration window. It is divided into two main sections: 'Block Diagram' and 'Configurations'.

Block Diagram: A block diagram of the 'altera_up_avalon_sram' component. It shows four input signals on the left: 'clock_reset', 'clock_reset_reset', 'external_interface', and 'avalon_sram_slave'. These connect to the 'clock', 'reset', 'conduit', and 'avalon' ports of the 'sram_0' block.

Configurations:

- DE-Series Board:** DE2.
- Use as a pixel buffer for video out.

Buttons for 'Cancel' and 'Finish' are located at the bottom right.

Full Qsys Machine with interconnects

Use	Connections	Name	Description	Export
<input checked="" type="checkbox"/>		<input type="checkbox"/> clk_0	Clock Source	clk
		clk_in	Clock Input	reset
		clk_in_reset	Reset Input	<i>Double-click to</i>
		clk	Clock Output	<i>Double-click to</i>
		clk_reset	Reset Output	<i>Double-click to</i>
<input checked="" type="checkbox"/>		<input type="checkbox"/> clk_1	Clock Source	clk_0
		clk_in	Clock Input	<i>Double-click to</i>
		clk_in_reset	Reset Input	<i>Double-click to</i>
		clk	Clock Output	<i>Double-click to</i>
		clk_reset	Reset Output	<i>Double-click to</i>
<input checked="" type="checkbox"/>		<input type="checkbox"/> altpll_0	Avalon ALTPLL	<i>Double-click to</i>
		<input type="checkbox"/> inclk_interface	Clock Input	<i>Double-click to</i>
		inclk_interface_reset	Reset Input	<i>Double-click to</i>
		pll_slave	Avalon Memory Mapped Slave	<i>Double-click to</i>
		c0	Clock Output	altpll_0_c0
	c1	Clock Output	<i>Double-click to</i>	
	c2	Clock Output	altpll_0_c2	
	areset_conduit	Conduit	<i>Double-click to</i>	
	locked_conduit	Conduit	<i>Double-click to</i>	
	phasedone_conduit	Conduit	<i>Double-click to</i>	
<input checked="" type="checkbox"/>	<input type="checkbox"/> altpll_1	Avalon ALTPLL	<i>Double-click to</i>	
	inclk_interface	Clock Input	<i>Double-click to</i>	
	inclk_interface_reset	Reset Input	<i>Double-click to</i>	
	pll_slave	Avalon Memory Mapped Slave	<i>Double-click to</i>	
	c0	Clock Output	altpll_1_c0	
	areset_conduit	Conduit	<i>Double-click to</i>	
	locked_conduit	Conduit	<i>Double-click to</i>	
	phasedone_conduit	Conduit	<i>Double-click to</i>	
<input checked="" type="checkbox"/>	<input type="checkbox"/> nios2_qsys_0	Nios II Processor	<i>Double-click to</i>	
	clk	Clock Input	<i>Double-click to</i>	
	reset_n	Reset Input	<i>Double-click to</i>	
	data_master	Avalon Memory Mapped Master	<i>Double-click to</i>	
	instruction_master	Avalon Memory Mapped Master	<i>Double-click to</i>	
	jtag_debug_module_re...	Reset Output	<i>Double-click to</i>	
	jtag_debug_module	Avalon Memory Mapped Slave	<i>Double-click to</i>	
	custom_instruction_m...	Custom Instruction Master	<i>Double-click to</i>	
<input checked="" type="checkbox"/>	<input type="checkbox"/> sysid_qsys_0	System ID Peripheral	<i>Double-click to</i>	
	clk	Clock Input	<i>Double-click to</i>	
	reset	Reset Input	<i>Double-click to</i>	
	control_slave	Avalon Memory Mapped Slave	<i>Double-click to</i>	
<input checked="" type="checkbox"/>	<input type="checkbox"/> timer_0	Interval Timer	<i>Double-click to</i>	
	clk	Clock Input	<i>Double-click to</i>	
	reset	Reset Input	<i>Double-click to</i>	
	s1	Avalon Memory Mapped Slave	<i>Double-click to</i>	
<input checked="" type="checkbox"/>	<input type="checkbox"/> camera	Audio and Video Config	<i>Double-click to</i>	
	clock_reset	Clock Input	<i>Double-click to</i>	
	clock_reset_reset	Reset Input	<i>Double-click to</i>	
	avalon_av_config_slave	Avalon Memory Mapped Slave	<i>Double-click to</i>	
	external_interface	Conduit	camera_external_...	
<input checked="" type="checkbox"/>	<input type="checkbox"/> D5M_Decoder	Video-In Decoder	<i>Double-click to</i>	
	clock_reset	Clock Input	<i>Double-click to</i>	
	clock_reset_reset	Reset Input	<i>Double-click to</i>	
	avalon_decoder_source	Avalon Streaming Source	<i>Double-click to</i>	
	external_interface	Conduit	d5m_decoder_ex...	
<input checked="" type="checkbox"/>	<input type="checkbox"/> video_bayer_resamp...	Bayer Pattern Resampler	<i>Double-click to</i>	
	clock_reset	Clock Input	<i>Double-click to</i>	
	clock_reset_reset	Reset Input	<i>Double-click to</i>	
	avalon_bayer_sink	Avalon Streaming Sink	<i>Double-click to</i>	
	avalon_bayer_source	Avalon Streaming Source	<i>Double-click to</i>	
<input checked="" type="checkbox"/>	<input type="checkbox"/> video_clipper_0	Clipper	<i>Double-click to</i>	
	clock_reset	Clock Input	<i>Double-click to</i>	
	clock_reset_reset	Reset Input	<i>Double-click to</i>	
	avalon_clipper_sink	Avalon Streaming Sink	<i>Double-click to</i>	
	avalon_clipper_source	Avalon Streaming Source	<i>Double-click to</i>	
<input checked="" type="checkbox"/>	<input type="checkbox"/> video_scaler_0	Scaler	<i>Double-click to</i>	
	clock_reset	Clock Input	<i>Double-click to</i>	
	clock_reset_reset	Reset Input	<i>Double-click to</i>	
	avalon_scaler_sink	Avalon Streaming Sink	<i>Double-click to</i>	
	avalon_scaler_source	Avalon Streaming Source	<i>Double-click to</i>	
<input checked="" type="checkbox"/>	<input type="checkbox"/> video_dma_controlle...	DMA Controller	<i>Double-click to</i>	
	clock_reset	Clock Input	<i>Double-click to</i>	
	clock_reset_reset	Reset Input	<i>Double-click to</i>	
	avalon_dma_sink	Avalon Streaming Sink	<i>Double-click to</i>	
	avalon_dma_control_s...	Avalon Memory Mapped Slave	<i>Double-click to</i>	
	avalon_dma_master	Avalon Memory Mapped Master	<i>Double-click to</i>	
<input checked="" type="checkbox"/>	<input type="checkbox"/> sram_0	SRAM/SSRAM Controller	<i>Double-click to</i>	
	clock_reset	Clock Input	<i>Double-click to</i>	
	clock_reset_reset	Reset Input	<i>Double-click to</i>	
	external_interface	Conduit	sram_0_external_...	
	avalon_sram_slave	Avalon Memory Mapped Slave	<i>Double-click to</i>	

Top Level

PIN Mapping Changes

Note: We assume that the camera is connected to GPIO_1

Remap the following pins currently assigned to GPIO_1 to CCD_DATA. This is necessary to access the block of data pins required for decoding. The pins are “backwards” in relation to traditional pin mapping and are not a contiguous set of pins. Re-mapping allows us to access them as if they were.

CCD_DATA[11]	Location	PIN_K26
CCD_DATA[10]	Location	PIN_M23
CCD_DATA[9]	Location	PIN_M19
CCD_DATA[8]	Location	PIN_M20
CCD_DATA[7]	Location	PIN_N20
CCD_DATA[6]	Location	PIN_M21
CCD_DATA[5]	Location	PIN_M24
CCD_DATA[4]	Location	PIN_M25
CCD_DATA[3]	Location	PIN_N24
CCD_DATA[2]	Location	PIN_P24
CCD_DATA[1]	Location	PIN_R25
CCD_DATA[0]	Location	PIN_R24

VHDL Changes

Entity

The newly added PINs need to be added to the toplevel entity:

```
entity BioLock is
  port (
    -- Reset and Clock
    KEY          : in std_logic_vector(0 downto 0);
    CLOCK_50     : in std_logic;
    CLOCK_27    : in std_logic;

    -- Off Chip
    GPIO_1       : inout std_logic_vector(35 downto 0);
    CCD_DATA     : in std_logic_vector(11 downto 0);
    ...
    ...
```

Nios Component

The camera components from the qsys machine also need to be added to the nios_system component:

--Camera

```
camera_external_interface_SDAT : inout std_logic := 'X';    -- SDAT
camera_external_interface_SCLK : out  std_logic;    -- SCLK
d5m_decoder_external_interface_PIXEL_CLK  : in  std_logic := 'X';    -- PIXEL_CLK
d5m_decoder_external_interface_LINE_VALID : in  std_logic := 'X';    -- LINE_VALID
d5m_decoder_external_interface_FRAME_VALID : in  std_logic := 'X';    -- FRAME_VALID
d5m_decoder_external_interface_PIXEL_DATA : in  std_logic_vector(11 downto 0) := (others => 'X');
-- PIXEL_DATA
```

Port Map

The following need to be added to the process for the camera reset and clock pins

```
GPIO_1(17) <= '1'; --reset
GPIO_1(16) <= CLOCK_50;
```

The following need to be added to the port map for the nios system:

```
camera_external_interface_SDAT => GPIO_1(23), -- camera_external_interface.SDAT
camera_external_interface_SCLK => GPIO_1(24), -- .SCLK
d5m_decoder_external_interface_PIXEL_CLK  => GPIO_1(0), -- .PIXEL_CLK
d5m_decoder_external_interface_LINE_VALID => GPIO_1(21), -- .LINE_VALID
d5m_decoder_external_interface_FRAME_VALID => GPIO_1(22), -- .FRAME_VALID
d5m_decoder_external_interface_PIXEL_DATA => CCD_DATA,
```

Other Info

At this point, data is being written directly into SRAM. There are 320x240x4 bytes used and can be read directly from SRAM by the nios processor. Each set of 4 bytes corresponds to a single pixel (R, G, B, NULL). This format correlates to the BMP format.

Readings

The following readings proved useful when researching how to get the D5M module working:

- http://www.terasic.com.tw/attachment/archive/282/TRDB_D5M_UserGuide.pdf
- <http://www.alteraforum.com/forum/showthread.php?t=29018>
- <http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&CategoryNo=39&No=281&PartNo=2>
- <https://www.youtube.com/watch?v=dPq7ROJiM90> (for proof that it actually worked for someone else)