

NTSC Video to VGA Tutorial

By Patrick Kuczera

Partners: Shahzeb Asif & Andrew Zhong

Group 7

April. 5th, 2016

Introduction

This tutorial will demonstrate how to interface NTSC standard video from the ADV7181B TV decoder to the ADV7123 VGA port on the Altera DE2 development board.

Hardware Requirements

- Altera DE2-115 Development Board
- NTSC standard video camera with RCA video adapter
- Monitor with 4:3 aspect ratio and VGA adapter

Software Requirements

- Quartus II 32-Bit Version 12.1 SP 1

Assumptions

You are:

- Able to setup a quartus project as detailed within the introductory labs.
- Familiar with Qsys and are comfortable navigating its interface.
- Capable of wiring the top level vhdl/verilog file with qsys system components.

Provided Material

A demo quartus project is provided which was created following all the steps of this application note. This demo is obtained by downloading and unzipping *g7_video_app_notes.zip*. It includes the required Qsys system, top level VHDL, and DE2.qsf pin assignment file. This project is ready to be flashed to the Altera DE2-115 development board. Prior flashing the board, you must connect your camera and monitor via the RCA video and VGA ports, respectively. After flashing the board, powering on both camera and monitor, you should expect to see live stream video from the camera to monitor.

Procedure

The following is a list of components you must add to your Qsys system. Note that the component connections will be outlined at the end, as shown in Step 18.

1. Nios II Processor

	Nios II/e	Nios II/s	Nios II/f
Nios II Selector Guide	RISC 32-bit	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide	RISC 32-bit Instruction Cache Branch Prediction Hardware Multiply Hardware Divide Barrel Shifter Data Cache Dynamic Branch Prediction
Memory Usage (e.g. Stratix IV)	Two M9Ks (or equiv.)	Two M9Ks + cache	Three M9Ks + cache

Hardware Arithmetic Operation
Hardware multiplication type: Embedded Multipliers
 Hardware divide

Reset Vector
Reset vector memory: Onchip_Memory_s1
Reset vector offset: 0x00000000
Reset vector: 0x00084000

Exception Vector
Exception vector memory: Onchip_Memory_s1
Exception vector offset: 0x00000020
Exception vector: 0x00084020

2. On-Chip Memory (RAM or ROM)

On-Chip Memory (RAM or ROM)
altera_avalon_onchip_memory2

Memory type
Type: RAM (Writable)
 Dual-port access
 Single clock operation
Read During Write Mode: DONT_CARE
Block type: Auto

Size
Data width: 32
Total memory size: 16384 bytes
 Minimize memory block usage (may impact fmax)

Read latency
Slave s1 Latency: 1
Slave s2 Latency: 1

Memory initialization
 Initialize memory content
 Enable non-default initialization file
User created initialization file: Onchip_Memory
 Enable In-System Memory Content Editor feature
Instance ID: NONE

3. Clock Source

The screenshot shows the configuration window for the 'Clock Source' block. The title bar includes the MegaCore logo and the text 'Clock Source' and 'clock_source'. A 'Documentation' button is in the top right. The 'Block Diagram' tab is active, showing a block with two inputs: 'clk_in' and 'clk_in_reset', and two outputs: 'clk' and 'clk_reset'. The block is labeled 'clock_source'. The 'Parameters' tab is also visible, showing 'Clock frequency' set to 50000000 Hz, 'Clock frequency is known' checked, and 'Reset synchronous edges' set to 'None'.

4. Clock Signals for DE-series Board Peripherals

The screenshot shows the configuration window for the 'Clock Signals for DE-series Board Peripherals' block. The title bar includes the MegaCore logo and the text 'Clock Signals for DE-series Board Peripherals' and 'altera_up_clocks'. A 'Documentation' button is in the top right. The 'Block Diagram' tab is active, showing a block with two inputs: 'clk_in_primary' and 'clk_in_primary_reset', and three outputs: 'sys_clk', 'sys_clk_reset', and 'vga_clk'. The block is labeled 'altera_up_clocks'. The 'Configurations' tab is also visible, showing 'DE-Series Board' set to 'DE2'. The 'Optional Clocks' section has 'SDRAM' unchecked, 'Video (VGA and SMP Digital Camera)' checked, and 'Audio' unchecked. 'Audio Clock Frequency' is set to 12.288 MHz.

5. Audio and Video Config

The screenshot shows the 'Audio and Video Config' tool window for the 'altera_up_avalon_audio_and_video_config' component. The interface is divided into two main sections: 'Block Diagram' and 'Parameters'.

Block Diagram: A block labeled 'AV_Config' is shown with four input signals: 'clock_reset', 'clock_reset_reset', 'avalon_av_config_slave', and 'external_interface'. The block has three internal signals: 'clock', 'reset', and 'avalon'. A 'conduit' is also shown at the bottom of the block. The component name 'altera_up_avalon_audio_and_video_config' is displayed at the bottom right of the diagram.

Parameters:

- Components:** Audio/Video Device: On-Board Peripherals; DE Board: DE2; Auto Initialize Device(s)
- Auto Initialization Parameters for Audio:** Audio In Path: Line In to ADC; Audio Out - Enable DAC Output; Audio Out - Microphone Bypass; Audio Out - Line In Bypass; Data Format: Left Justified; Bit Length: 24; 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; Enable external exposure port

6. Video-In Decoder

The screenshot shows the 'Video-In Decoder' tool window for the 'altera_up_avalon_video_decoder' component. The interface is divided into two main sections: 'Block Diagram' and 'Parameters'.

Block Diagram: A block labeled 'Video_Decoder' is shown with three input signals: 'clock_reset', 'clock_reset_reset', and 'external_interface'. The block has three internal signals: 'clock', 'reset', and 'conduit'. An 'avalon_streaming' signal is shown as an output from the block. A purple arrow labeled 'avalon_decoder_source' points to the right from the block. The component name 'altera_up_avalon_video_decoder' is displayed at the bottom right of the diagram.

Parameters: Video-In Source: On-board Video In (NTSC or PAL)

7. Clipper

The screenshot shows the configuration window for the **Clipper** block (part number `altera_up_avalon_video_clipper`). The window is divided into two main sections: a **Block Diagram** on the left and a **Parameters** section on the right.

Block Diagram: A central block labeled `Video_Clipper` is shown. It has three input ports on the left: `clock_reset`, `clock_reset_reset`, and `avalon_clipper_sink`. It has two output ports on the right: `avalon_clipper_source` and `avalon_streaming`. The block is connected to a source block `avalon_clipper_source` and a sink block `avalon_streaming`. The diagram also shows the internal signals `clock`, `reset`, and `avalon_streaming` within the block.

Parameters:

- Incoming Frame Resolution:**
 - Width (# of pixels): 720
 - Height (# of lines): 244
- Reduce Frame Size:**
 - Columns to remove from the left side: 40
 - Columns to remove from the right side: 40
 - Rows to remove from the top: 2
 - Rows to remove from the bottom: 2
- 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 value for plane 1: 0x00000000
 - Added pixel value for plane 2: 0x00000000
 - Added pixel value for plane 3: 0x00000000
 - Added pixel value for plane 4: 0x00000000
- Pixel Format:**
 - Color Bits: 8
 - Color Planes: 2

8. Chroma Resampler

The screenshot shows the configuration window for the **Chroma Resampler** block (part number `altera_up_avalon_video_chroma_resampler`). The window is divided into two main sections: a **Block Diagram** on the left and a **Parameters** section on the right.

Block Diagram: A central block labeled `Video_Chroma_Resampler` is shown. It has three input ports on the left: `clock_reset`, `clock_reset_reset`, and `avalon_chroma_sink`. It has two output ports on the right: `avalon_chroma_source` and `avalon_streaming`. The block is connected to a source block `avalon_chroma_source` and a sink block `avalon_streaming`. The diagram also shows the internal signals `clock`, `reset`, and `avalon_streaming` within the block.

Parameters:

- Incoming Format:** YCrCb 422
- Outgoing Format:** YCrCb 444

9. Colour-Space Converter

The screenshot shows the configuration window for the **Colour-Space Converter** component. The title bar includes the MegaCore logo, the component name, and the identifier `altera_up_avalon_video_csc`, along with a **Documentation** button. The **Block Diagram** pane on the left shows a block named `Video_CSC` with three input signals: `clock_reset`, `clock_reset_reset`, and `avalon_csc_sink`. The block has two internal signals: `clock` and `reset`. It has one output signal, `avalon_streaming`, which is connected to an external `avalon_csc_source` signal. The component is identified as `altera_up_avalon_video_csc`. The **Mode** pane on the right shows the **Colour-Space Conversion** mode set to `444 YCrCb to 24-bit RGB`.

10. Scaler

The screenshot shows the configuration window for the **Scaler** component. The title bar includes the MegaCore logo, the component name, and the identifier `altera_up_avalon_video_scaler`, along with a **Documentation** button. The **Block Diagram** pane on the left shows a block named `Video_Scaler` with three input signals: `clock_reset`, `clock_reset_reset`, and `avalon_scaler_sink`. The block has two internal signals: `clock` and `reset`. It has one output signal, `avalon_streaming`, which is connected to an external `avalon_scaler_source` signal. The component is identified as `altera_up_avalon_video_scaler`. The **Scaling Parameters** pane on the right shows `Width Scaling Factor` set to `0.5` and `Height Scaling Factor` set to `1`. The **Incoming Frame Resolution** pane shows `Width (# of pixels)` set to `640` and `Height (# of lines)` set to `240`. The **Pixel Format** pane shows `Color Bits` set to `8` and `Color Planes` set to `3`.

11. DMA Controller

The screenshot shows the configuration window for the **DMA Controller** component. The title bar includes the MegaCore logo, the component name, and the identifier `altera_up_avalon_video_dma_controller`, along with a **Documentation** button. The **Block Diagram** pane on the left shows a block named `Video_DMA` with four input signals: `clock_reset`, `clock_reset_reset`, `avalon_dma_sink`, and `avalon_dma_control_slave`. The block has three internal signals: `clock`, `reset`, and `avalon`. It has one output signal, `avalon`, which is connected to an external `avalon_dma_master` signal. The component is identified as `altera_up_avalon_video_dma_controller`. The **Addressing Parameters** pane on the right shows `Addressing Mode` set to `X-Y`, `Default Buffer Starting Address` set to `0x00000000`, and `Default Back Buffer Start Address` set to `0x00000000`. The **Frame Resolution** pane shows `Width (# of pixels)` set to `320` and `Height (# of lines)` set to `240`. The **Pixel Format** pane shows `Color Bits` set to `8` and `Color Planes` set to `3`. The **Mode** pane shows `DMA Direction` set to `From Stream to Memory`.

12. SRAM/SSRAM Controller

The screenshot shows the configuration window for the SRAM/SSRAM Controller. The title bar includes the MegaCore logo, the component name "SRAM/SSRAM Controller", the part number "altera_up_avalon_sram", and a "Documentation" button. The "Block Diagram" tab is active, showing a block named "Pixel_Buffer" with four input signals: "clock_reset", "clock_reset_reset", "external_interface", and "avalon_sram_slave". These signals are connected to internal ports "clock", "reset", "conduit", and "avalon" respectively. The "Configurations" tab is also visible, showing "DE-Series Board" set to "DE2" and a checked option "Use as a pixel buffer for video out".

13. Pixel Buffer DMA Controller

The screenshot shows the configuration window for the Pixel Buffer DMA Controller. The title bar includes the MegaCore logo, the component name "Pixel Buffer DMA Controller", the part number "altera_up_avalon_video_pixel_buffer_dma", and a "Documentation" button. The "Block Diagram" tab is active, showing a block named "Pixel_DMA" with three input signals: "clock_reset", "clock_reset_reset", and "avalon_control_slave". These are connected to internal ports "clock", "reset", and "avalon". Two output signals, "avalon_pixel_dma_master" and "avalon_pixel_source", are connected to the "avalon" and "avalon_streaming" ports respectively. The "Addressing Parameters" tab is active, showing "Addressing Mode" set to "X-Y", "Default Buffer Start Address" and "Default Back Buffer Start Address" both set to "0x00000000". The "Frame Resolution" tab shows "Width (# of pixels)" set to "320" and "Height (# of lines)" set to "240". The "Pixel Format" tab shows "Color Space" set to "24-bit RGB".

14. RGB Resampler

The screenshot shows the configuration window for the RGB Resampler. The title bar includes the MegaCore logo, the component name "RGB Resampler", the part number "altera_up_avalon_video_rgb_resampler", and a "Documentation" button. The "Block Diagram" tab is active, showing a block named "Pixel_RGB_Resampler" with three input signals: "clock_reset", "clock_reset_reset", and "avalon_rgb_sink". These are connected to internal ports "clock", "reset", and "avalon_streaming". One output signal, "avalon_rgb_source", is connected to the "avalon_streaming" port. The "Parameters" tab is active, showing "Incoming Format" set to "24-bit RGB", "Outgoing Format" set to "30-bit RGB", and "Alpha Value for Output" set to "1023".

15. Scaler

The screenshot displays the configuration window for the **Scaler** block (`altera_up_avalon_video_scaler`). The interface is divided into two main sections: a **Block Diagram** and a **Properties** panel.

Block Diagram: Shows the `Pixel_Scaler` block with the following connections:

- `clock_reset` (input) to `clock` (input)
- `clock_reset_reset` (input) to `reset` (input)
- `avalon_scaler_sink` (input) to `avalon_streaming` (input)
- `avalon_streaming` (output) to `avalon_streaming` (output)
- `avalon_streaming` (output) to `avalon_scaler_source` (output)

Properties Panel:

- Scaling Parameters:**
 - Width Scaling Factor: 2
 - Height Scaling Factor: 2
- Incoming Frame Resolution:**
 - Width (# of pixels): 320
 - Height (# of lines): 240
- Pixel Format:**
 - Color Bits: 10
 - Color Planes: 3

16. Dual-Clock FIFO

The screenshot displays the configuration window for the **Dual-Clock FIFO** block (`altera_up_avalon_video_dual_clock_buffer`). The interface is divided into two main sections: a **Block Diagram** and a **Properties** panel.

Block Diagram: Shows the `Dual_Clock_FIFO` block with the following connections:

- `clock_stream_in` (input) to `clock` (input)
- `clock_stream_in_reset` (input) to `reset` (input)
- `clock_stream_out` (input) to `clock` (input)
- `clock_stream_out_reset` (input) to `reset` (input)
- `avalon_dc_buffer_sink` (input) to `avalon_streaming` (input)
- `avalon_streaming` (output) to `avalon_streaming` (output)
- `avalon_streaming` (output) to `avalon_dc_buffer_source` (output)

Properties Panel:

- Pixel Format:**
 - Color Bits: 10
 - Color Planes: 3

17. VGA Controller

The screenshot displays the configuration window for the **VGA Controller** block (`altera_up_avalon_video_vga_controller`). The interface is divided into two main sections: a **Block Diagram** and a **Properties** panel.

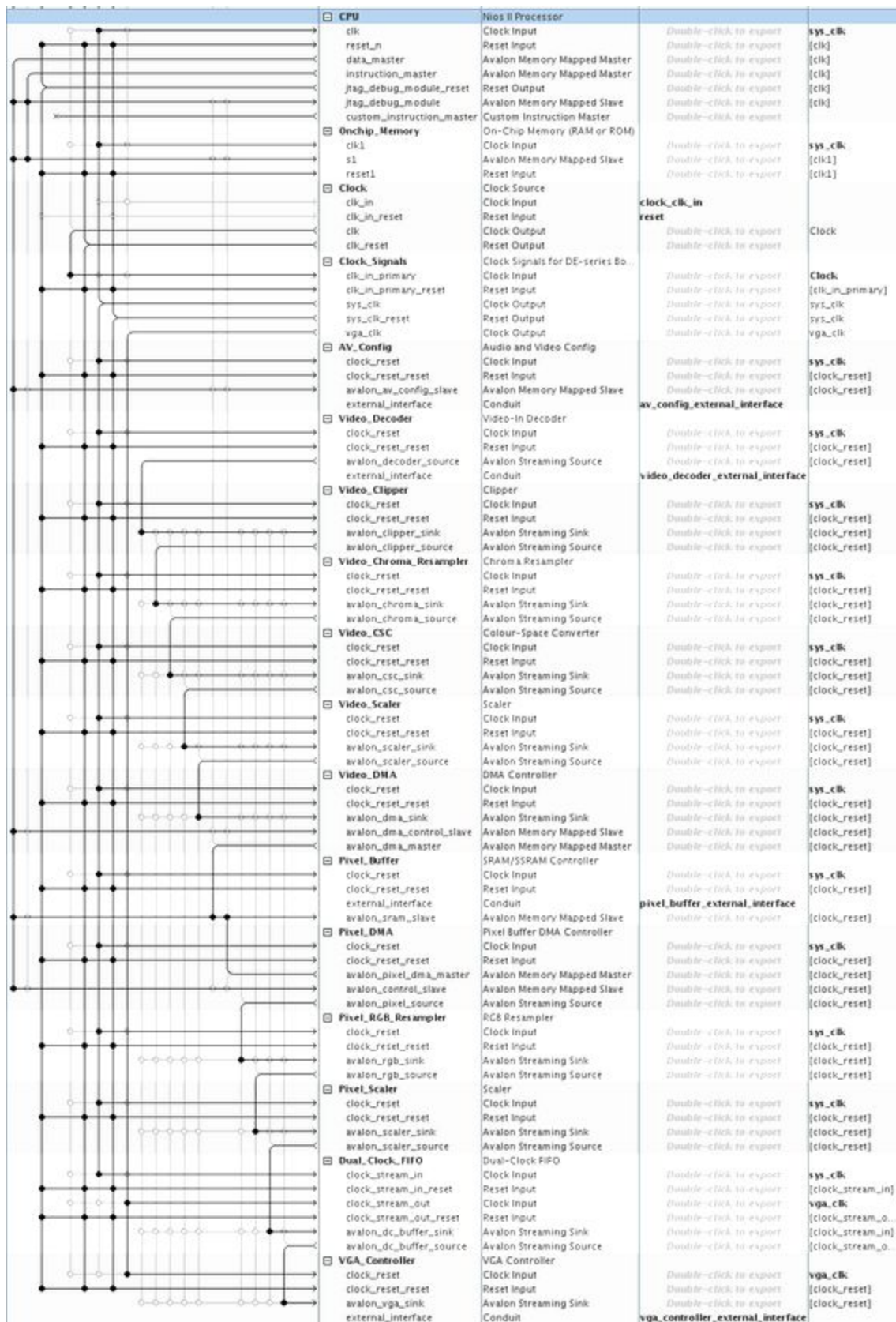
Block Diagram: Shows the `VGA_Controller` block with the following connections:

- `clock_reset` (input) to `clock` (input)
- `clock_reset_reset` (input) to `reset` (input)
- `avalon_vga_sink` (input) to `avalon_streaming` (input)
- `external_interface` (input) to `conduit` (input)

Properties Panel:

- Device and Mode:**
 - DE-Series Board: DE2
 - Video Out Device: VGA Connector

18. Component Connections



After adding all components as listed above, refresh memory addresses by System->Assign Base Addresses. Generate the system and copy the VHDL example from Qsys to your top level file.

Your top level VHDL should be wired similarly(names may differ):

```

u0 : component niosII_system
  port map (
    reset_reset_n           => KEY(0),
    clock_clk_in_clk        => CLOCK_50,

    pixel_buffer_external_interface_DQ           => SRAM_DQ,
    pixel_buffer_external_interface_ADDR         => SRAM_ADDR,
    pixel_buffer_external_interface_LB_N        => SRAM_LB_N,
    pixel_buffer_external_interface_UB_N        => SRAM_UB_N,
    pixel_buffer_external_interface_CE_N        => SRAM_CE_N,
    pixel_buffer_external_interface_OE_N        => SRAM_OE_N,
    pixel_buffer_external_interface_WE_N        => SRAM_WE_N,

    vga_controller_external_interface_CLK        => VGA_CLK,
    vga_controller_external_interface_HS         => VGA_HS,
    vga_controller_external_interface_VS         => VGA_VS,
    vga_controller_external_interface_BLANK      => VGA_BLANK,
    vga_controller_external_interface_SYNC       => VGA_SYNC,
    vga_controller_external_interface_R          => VGA_R,
    vga_controller_external_interface_G          => VGA_G,
    vga_controller_external_interface_B          => VGA_B,

    av_config_external_interface_SDAT           => I2C_SDAT,
    av_config_external_interface_SCLK           => I2C_SCLK,

    video_decoder_external_interface_TD_CLK27   => TD_CLK27,
    video_decoder_external_interface_TD_DATA    => TD_DATA,
    video_decoder_external_interface_TD_HS      => TD_HS,
    video_decoder_external_interface_TD_VS      => TD_VS,
    video_decoder_external_interface_TD_RESET   => TD_RESET
  );

```

Next, make sure to update your pin assignments by setting TD_CLK27 to PIN_C16 in the DE2.qsf file. A proper video output will not be achievable without this.

```

444 set_location_assignment PIN_V24 -to GPIO_1[32]
445 set_location_assignment PIN_V23 -to GPIO_1[33]
446 set_location_assignment PIN_W25 -to GPIO_1[34]
447 set_location_assignment PIN_W23 -to GPIO_1[35]
448 set_location_assignment PIN_C16 -to TD_CLK27
449
450 set_global_assignment -name PARTITION METRIC TYPE SOURCE -section id Top

```

Finally, compile your project and flash it to the DE2 board. After connecting your monitor and camera via VGA and RCA adapters respectively, you should obtain streamed 640x480 30-bit RGB video.

References

[1]	Altera Corporation. (August 2012) "Video IP Cores" [Online] Last accessed: Feb 28, 2016 ftp://ftp.altera.com/up/pub/Altera_Material/12.1/University_Program_IP_Cores/Audio_Video/Video.pdf
[2]	Gerry Finlay. (2013, March 27) "NTSC to VGA using Altera University Program IP Cores" [Online] Last Accessed: Feb 19, 2015 https://www.ualberta.ca/~delliott/local/ece492/appnotes/2013w/G15_Video_Out/G15_SOPC_Video