

Introduction

The Altera® V-Series of reference designs deliver high-quality up, down, and cross conversion of standard definition (SD), high definition (HD) and 3 gigabits per second (Gbps) video streams in interlaced or progressive format. The reference designs are highly software and hardware configurable, enabling rapid system configuration and design. The designs have been developed targeting typical broadcast applications such as switcher, multi-viewer, converter, and video conferencing products.


The V-Series of reference designs is an evolution of the M-Series of video processing reference designs (M1–M5). The V-Series provides a roadmap of functionality for broadcast applications, as well as run-time capability and software configurability typically associated with ASSPs.

 For more information on the M-Series of reference designs, refer to the [Broadcast](#) page on the Altera website or contact the Altera Broadcast Business Unit.

The V-Series reference designs are working application templates that you can use directly or as a starting point to rapidly build further broadcast applications using a flexible, re-usable, and parameterizable video framework.

The configuration of video interfaces and control of video processing functions is conveniently implemented in software. This environment provides a very rapid development cycle for software control code changes, without the requirement for hardware recompilation.

All the hardware functions in the designs use standard, open interfaces and protocols to ease function re-use and system design. The Video and Image Processing Suite MegaCore functions use Avalon® Streaming (Avalon-ST) data interfaces and Avalon Memory-Mapped (Avalon-MM) control interfaces to simplify the connection of a chain of video functions and video system design. Video data is transmitted between the Video and Image Processing Suite functions using the Avalon-ST Video protocol, which eases building run-time controllable systems and handling error recovery.

 For a full description of how the Avalon-ST Video protocol and Avalon interfaces are implemented, refer to the [Interfaces](#) chapter in the *Video and Image Processing User Guide*. For more information about the Avalon-MM and Avalon-ST interfaces, refer to the [Avalon Interface Specifications](#).

The V-Series designs are delivered using the system level design tool SOPC Builder to facilitate system capture, integration, parameterization and system generation. SOPC Builder provides a library of parameterizable software and hardware IP modules, an abstracted view of the system interconnect, and automatically creates the bus arbitration logic connecting the individual components together to create an overall system.

 For more information on SOPC Builder, refer to the [SOPC Builder User Guide](#).

The designs use the Altera Video and Image Processing Suite MegaCore® functions library, the SDI MegaCore function, the DDR2 High Performance Memory Controller MegaCore function, the Nios® II processor, and supporting development tools.



For information about these MegaCore functions, refer to the *Video and Image Processing Suite User Guide*, *SDI MegaCore function User Guide*, *DDR and DDR2 SDRAM High-Performance Controller User Guide*, and the *Nios II Processor* handbooks on the Altera website.

Features

Table 1 lists the key features of the V2 reference design.

Table 1. Key Features of the V2 Reference Design

Feature	Description
Input	Two video input streams from a triple-rate serial digital interface SDI MegaCore function support SD-SDI, HD-SDI, or 3G-SDI progressive/interlaced inputs up to 1080p60 (such as: NTSC, PAL, 720p, 1080i, and 1080p). All frame rates up to 60Hz supported (including 30 Hz, 59.94 Hz, 60 Hz).
Output	Two video output streams from triple-rate serial digital interface SDI MegaCore functions support SD-SDI, HD-SDI, or 3G-SDI progressive/interlaced outputs up to 1080p60 (such as: NTSC, PAL, 720p, 1080i, and 1080p). All frame rates up to 60Hz supported (including 30 Hz, 59.94 Hz, 60 Hz).
Processing	Two fully independent streams are converted to their own output resolution and rate. Changes in input are auto-detected and the datapath updates itself, maintaining the requested output format. Format conversion functions include clipping, deinterlacing, scaling and frame rate conversion. Streams 1 and 2 both support: <ul style="list-style-type: none"> ■ High quality processing (motion adaptive deinterlacer, 6×6 tap scaler) ■ Voltage-controlled crystal oscillator (VCXO) used to synchronize input and output video clock rates (where applicable) to reduce judder ■ Performs frame rate conversion in the frame buffer function between input and output to support any input frame rate ■ Performs run-time controllable interlacing in the Interlacer MegaCore function to support interlaced output
Software	System initialization and configuration in software. Dynamic scaler coefficient generation and reload. Class application programming interface (API) provided to facilitate read/write access to the Video and Image Processing Suite register maps at run time.
System Tools	Rapid system capture and design with SOPC Builder, Quartus® II, and the Nios II development environments.
Design Framework	Highly parameterizable and modular hardware functions (15 Video and Image Processing Suite MegaCore functions, Nios II processor, open source system configuration software, video frame buffers, high performance memory controllers, peripherals, auto-generation of switch fabric). Standard Avalon-ST and Avalon-MM interfaces for rapid integration, and Avalon-ST Video protocol for video transmission between functions.
Device Support	Stratix II GX (2SGX90) Audio Video Development Kit
Device Utilization	93% logic, 40% memory implementation bits, 30% DSP blocks



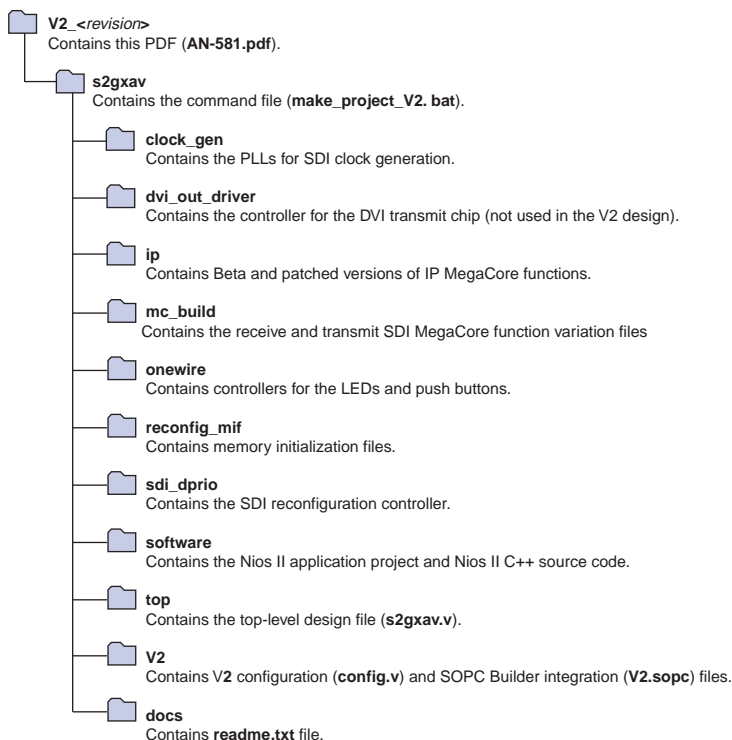
For information about the V1 reference design, refer to *AN 559: High Definition Video Reference Design (V1)*.

Installing the Reference Design

The V-Series reference designs are available as **.zip** files from the [Broadcast](#) pages on the Altera website.

[Figure 1](#) shows the directory structure for the V2 reference design files when they have been extracted from the **.zip** file.

Figure 1. Reference Design Directory Structure



The **V2** subdirectory contains the following two patch files:

- **userdocs_acs_patches_PC-Quartus_II-9.0_SP1-1.04.exe**
- **userdocs_acs_patches_Linux-Quartus_II-9.0_SP1-1.04.tar**

Install the version of ACS patch 1.04 that matches your operating system. This patch updates your Quartus II software to support the new BETA IP features required for this reference design.

The **s2gxav** subdirectory contains a command file **make_project_V2.bat**.

Run this command file to:

- Copy a SOPC Builder project file (**.sopc**) to the top level
- Copy the design configuration file (**config.v**) to the top level
- Generate the Quartus II project files (**.qpf**, and **.qsf**)

System Requirements

This section describes the hardware and software requirements to run the V2 reference design.

Hardware Requirements

The video monitoring reference design requires the following hardware components:

- Audio Video Development Kit Stratix II GX Edition including:
 - Stratix II GX video development board
 - Serial digital interface (SDI) inputs and outputs
 - DDR2 DIMM external memory
- A display or displays with SDI inputs (or DVI inputs with separate convertor boxes performing SDI to DVI conversion) supporting 1,920×1,080 resolution
- Two coaxial cables to connect the output BNC connectors SDI_OUT_P2 and/or SDI_OUT_P3 to the display(s)
- Two SDI sources providing progressive or interlaced output up to 1080p60 output.
- Two coaxial cables to connect the SDI sources to the BNC connectors SDI_IN0 and/or SDI_IN1 on the development board.

Software Requirements

Table 2 shows the operating systems and software tool versions that are supported by the V2 reference design.

Table 2. Operating Systems and Software Tool Versions supported by V2 Reference Design

Operating System	Altera Software Tools	Altera Development Kit
Windows XP or Linux	v9.0 SP1 with ACS patch 1.04 (Note 1)	Stratix II GX (2SGX 90) Audio Video

Note to Table 2:

- (1) The Altera software tools include the Quartus II software, SOPC Builder, Nios II EDS, and MegaCore IP Library (including the Video and Image Processing Suite).

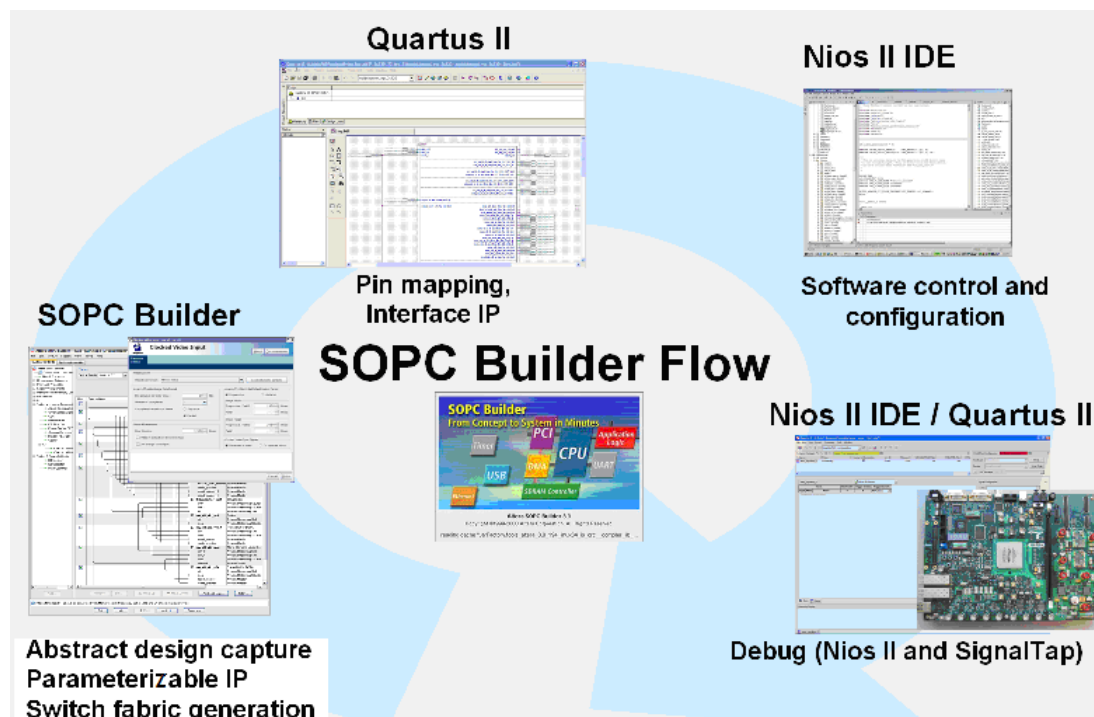


For more information on the Altera development kit software installation, refer to the documentation provided with the development kit.

Video Design Flow

The V-Series reference designs demonstrate a simple, yet highly parameterizable, design flow for rapid system development. **Figure 2** provides a high level view of the typical design flow within Altera's video design framework.

Figure 2. SOPC Builder Design Flow



The video design framework provides:

- Open interface and protocol standards to enable design re-use, and connection of custom IP functions with off-the-shelf IP including:
 - Data streaming interfaces and protocols for transmission of video data between IP functions in the framework (Avalon-ST Video protocol layers on the Avalon-ST interface)
 - Control interfaces (Avalon-MM master and slave interfaces)
 - Random access to external memory (Avalon-MM master and slave interfaces)
- System level tools and design methodology for rapid system construction, integration and re-design. SOPC Builder takes advantage of standard interfaces by presenting an abstracted view of the design, and generating an application specific switch fabric to construct the system.
- Parameterizable MegaCore IP functions that enable you to quickly construct complete video systems.
- Reference designs that demonstrate the capabilities of the video framework.
- Development kits to rapidly prototype the designs.

The reference designs described in this application note demonstrate each of these aspects of the framework. The video design tool flow is described in the following sections.

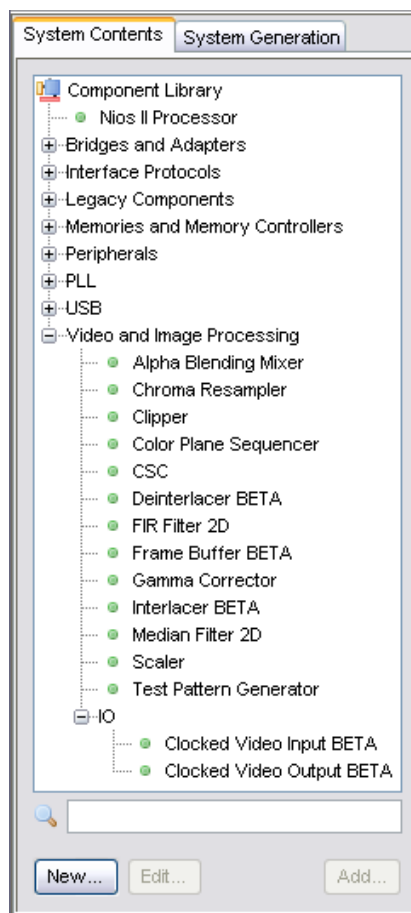
SOPC Builder

The SOPC Builder tool flow is the primary design flow for rapid video system development. Specifically, SOPC Builder simplifies the process of system design, including the data path, processor core and external memory integration. The tool enables you to capture the design at an abstract level with single point-to-point data connections rather than connecting individual data and control interface wires.

All the connections in the SOPC Builder system use Avalon-ST and Avalon-MM interfaces. SOPC Builder enables you to rapidly build systems containing IP products that support the Avalon-ST Video protocol and Avalon-MM interfaces. The Video and Image Processing Suite MegaCore functions that support Avalon-ST Video protocol for data transmission can be connected together by the click of a button.

The MegaCore functions display in the SOPC Builder **System Contents** tab under the category **Video and Image Processing** (Figure 3 on page 6).

Figure 3. Video and Image Processing Functions in the SOPC Builder System Contents tab

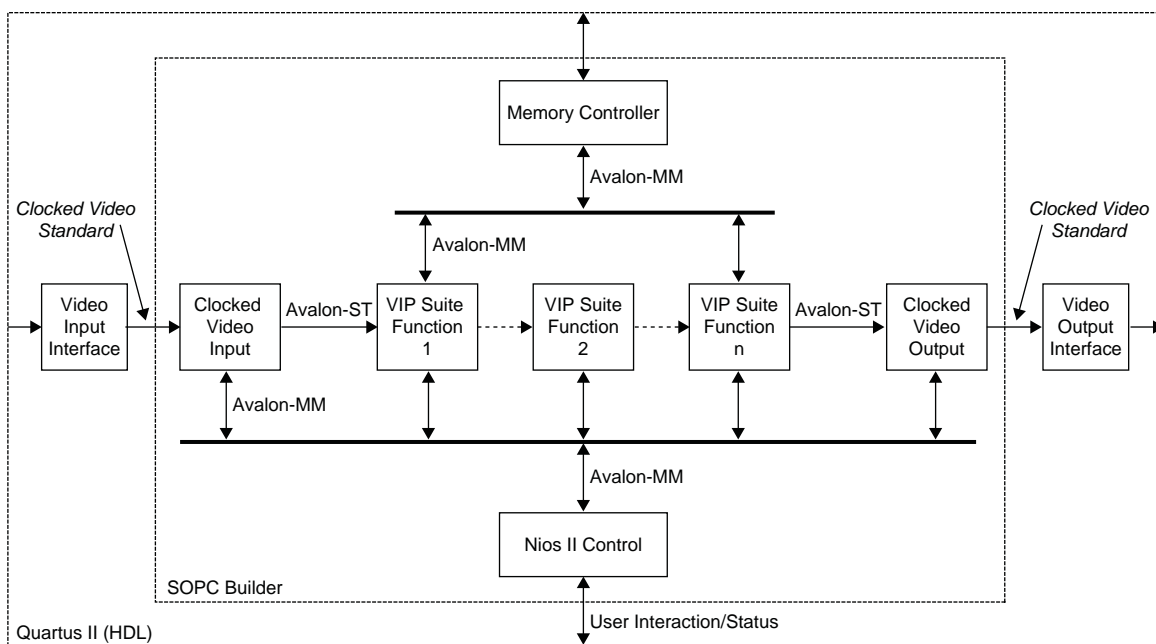


SOPC Builder automatically generates an interconnect switch fabric, including arbitration logic to connect together the memory mapped masters and slaves. A common example is where the system contains multiple Avalon-MM Masters, which need to buffer video data in an external memory using a single memory controller.

The video design flow in SOPC Builder (Figure 4 on page 7) has four main external connections:

- Video input from an external video interface. The connection to the external video interface is made using a parameterizable video conversion IP function. The IP function provides a bridge between a clocked video interface (such as a SDI MegaCore function) and the Avalon-ST flow controlled domain.
- Video output to an external video interface. The connection to the external video interface is made using a parameterizable video conversion IP function. The IP function provides a bridge between the Avalon-ST flow controlled domain and a clocked video interface (such as DVI).
- Connection to an external memory interface. This connection uses a DDR and DDR2 SDRAM High performance Controller MegaCore function. SOPC Builder generates the application specific switch fabric to arbitrate between multiple masters trying to access the controller.
- Connection to user controls and board components such as push buttons, LEDs, and quad seven-segment displays.

Figure 4. V-Series Video Design Flow Block Diagram



The Quartus II Software

The top-level system is described within the Quartus II software environment and the SOPC Builder system is integrated into the top level design. The Quartus II software environment is well suited for mapping external connections to the SOPC Builder system in the form of exported wires to video interface IP (such as SDI) and memory interfaces (such as DDR2) as well as making the appropriate pin assignments.

A wide range of tools are provided to facilitate timing closure and perform hardware compilation to generate an FPGA programming file.

The Quartus II software also provides a system-level debugging tool (SignalTap® II logic analyzer) that captures and displays real-time signal behavior so you can observe interactions between hardware and software in system designs.



For information on the Quartus II software, refer to the Quartus II Help.

Nios II IDE

The Nios II integrated development environment (IDE) is the primary software development tool for the Nios II family of embedded processors. All software development tasks can be accomplished within the Nios II IDE, including editing, building, and debugging programs.

The Nios II IDE provides a consistent development platform that works for all Nios II processor systems. Configuration of video interfaces and control of video processing functions can conveniently be implemented in software, which provides a very rapid development cycle for software control code changes, without the requirement for hardware re-compilation.

This environment provides you with all the standard software debug tools, including breakpoints, memory/variable/register windows, and single stepping. This flow is further enhanced by the inclusion of C++ software classes that provides a software API between the Nios II control code and the Video and Image Processing Suite MegaCore functions. The C++ classes contain member functions that provide easy control of the MegaCore functions and easy access to useful data flow information such as the number of frames that have been dropped or repeated by a frame buffer in the data path.



For information on the Nios II IDE software, refer to the Nios II Help.

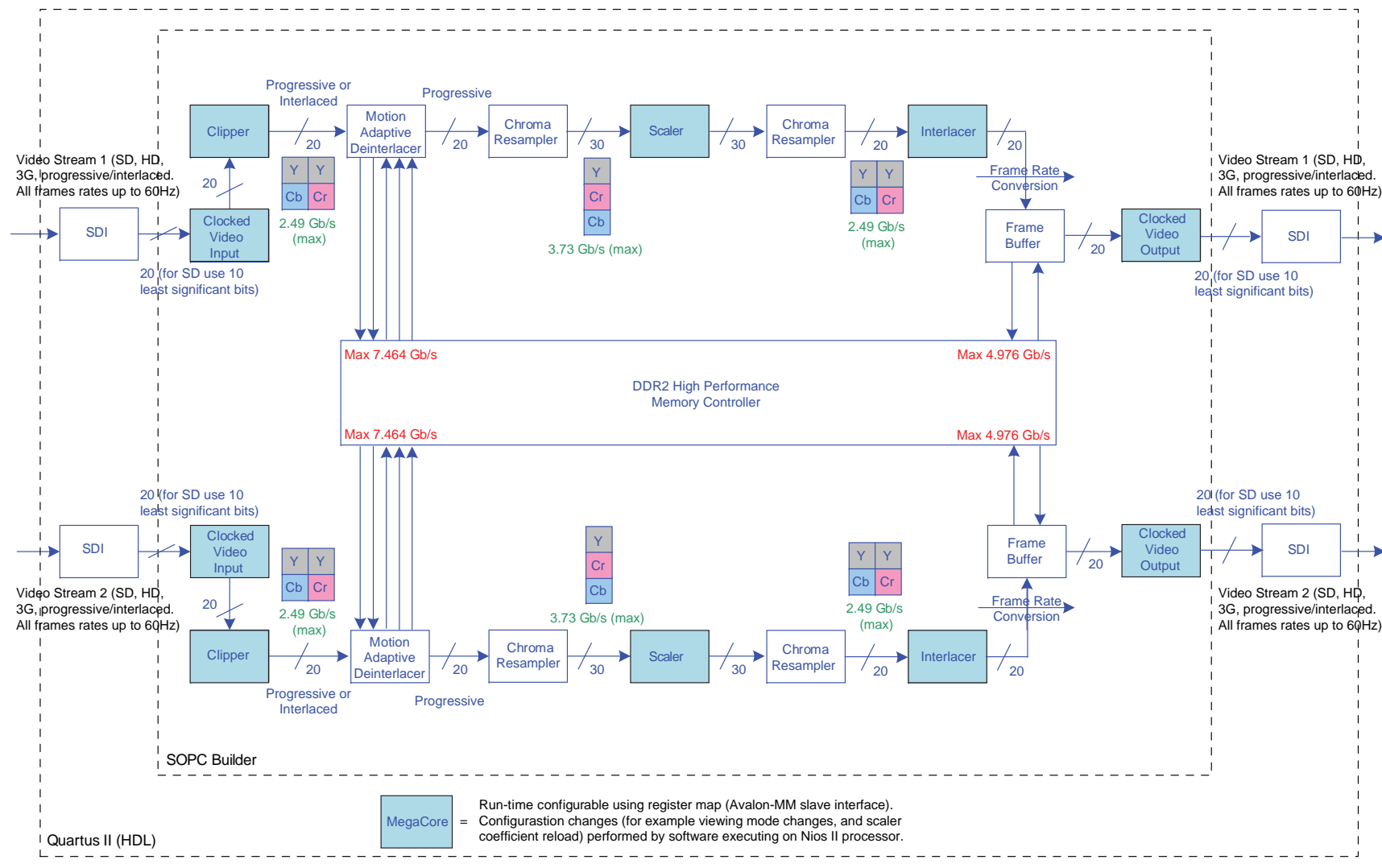
Functional Description

Figure 5 on page 9 shows a detailed block diagram of the V2 reference design.

Video Input

Video is input to the Stratix II GX 90 device using two triple-rate SDI MegaCore functions, supporting data in SD-SDI, HD-SDI or 3G-SDI standard format. Both video streams, with active picture data in Y'CbCr 4:2:2 format and associated embedded synchronization signals, are input from the SDI functions to the Clocked Video Input MegaCore functions.

Figure 5. V2 Reference Design Block Diagram



The Clocked Video Input MegaCore function converts from a notionally clocked video format (such as BT656 or DVI) to the flow-controlled Avalon-ST Video protocol. All Video and Image Processing Suite MegaCore functions (except the Line Buffer Compiler) transmit data using the Avalon-ST Video protocol, which allows an increased level of productivity through design flexibility and re-use.

The Clocked Video Input MegaCore function strips the incoming clocked video of horizontal and vertical blanking, leaving active picture data. Using this data with horizontal and vertical synchronization information, the MegaCore function creates the necessary Avalon-ST Video control and active picture packets.

With default parameterization, no conversion is done to the active picture data; the color plane information remains the same as in the clocked video format. However, in the V-Series designs, the Clocked Video Input MegaCore function is configured to accept SD-SDI SMPTE 259M (sequential 10-bit data—the least significant 10 bits of the 20-bit wide data received from the SDI MegaCore function), and output data in Avalon-ST Video (two 10-bit color planes in parallel). Thus the Clocked Video Input MegaCore function supports SD-SDI, HD-SDI and 3G-SDI data input at run-time.

The Clocked Video Input MegaCore function also provides clock-crossing capabilities that allow video formats running at different frequencies to enter the system. It also provides greater design flexibility by enabling the video processing system clock to be decoupled from the video input pixel clock. In the V-Series designs, the pixel clock and the system clocks are separate, both with a clock frequency of 148.5 MHz.

Video Processing

The V2 reference design processes two SD-SDI, HD-SDI or 3G-SDI progressive or interlaced video streams.

The SDI video streams are formatted as either SD-SDI SMPTE 259M (Y'CbCr 4:2:2 10-bit wide data, alternating Cb Cr), HD-SDI (SMPTE 292M) or 3G-SDI SMPTE 424M (Y'CbCr 4:2:2 20-bit wide data, Y' and CbCr in parallel). The Clocked Video Input MegaCore function produces data in the format Y'CbCr 4:2:2, 10 bits per color plane, 2 colors in parallel for SD-SDI, HD-SDI, and 3G-SDI data.

The first processing function in the data path clips a rectangular region parameterized at run time using the control interface. For example, when NTSC video is input, a region of 720 pixels by 240 lines from each field, offset three lines from the top of the field is clipped. In this case, the Clipper MegaCore function outputs fields with an equal number of odd and even lines, for further processing by the Deinterlacer MegaCore function.

The Deinterlacer MegaCore function is configured to accept both progressive and interlaced video in Y'CbCr 4:2:2 format. When the input video is in interlaced format, the motion adaptive algorithm produces a video stream in progressive format. When the input video is in progressive format, the data is output unprocessed. The data is also buffered in external memory when in this mode.

The progressive video stream is converted to 4:4:4 format by a Chroma Resampler MegaCore function so that it can be scaled by the polyphase algorithm of the parameterizable Scaler MegaCore function (with six horizontal and six vertical taps). The Scaler MegaCore function is configured with an Avalon-MM slave control interface to allow run-time specification of the scaler output resolution. In addition, when the scaler ratio changes, software running on the Nios II processor calculates and reloads suitable coefficients for improved picture quality. After scaling the video stream is converted back to 4:2:2 format by another Chroma Resampler.

Before outputting the video stream, the video data is buffered in external memory by the Frame Buffer MegaCore function. Frame rate conversion occurs at this point if the input and output frame rates do not match.

Video Output

A video stream of progressive or interlaced Y'CbCr 4:2:2 data is output from the SDI_OUT_P2 and SDI_OUT_P3 ports on the Stratix II GX audio video development board.

The Clocked Video Output MegaCore function converts from the Avalon-ST Video protocol to clocked video formats (such as BT656 and DVI). It formats Avalon-ST Video into clocked video by inserting horizontal and vertical blanking and generating horizontal and vertical sync information using the Avalon-ST Video control and active picture packets. No conversion is done to the active picture data, the color plane information remains the same as in the Avalon-ST Video format.

The Clocked Video Output MegaCore function also provides clock-crossing capabilities to provide greater design flexibility by enabling the video processing system clock to be decoupled from the video output pixel clock.

The video frame is described using the mode registers that are accessed using the Avalon-MM control port on the Clocked Video Output MegaCore function. If you turn off **Use control port** in the MegaWizard interface for the Clocked Video Output MegaCore function, then the output video format always has the format specified in the MegaWizard interface.

The Clocked Video Output MegaCore function can be configured to support between 1 to 14 different modes and each mode has a bank of registers that describe the output frame. When the Clocked Video Output MegaCore function receives a new control packet on the Avalon-ST Video input, it searches the mode registers for a mode that is valid and has a field width and height that matches the width and height in the control packet. Both the register VidModeMatch and the signal vid_mode_match show the currently selected mode. Once found, the Clocked Video Output MegaCore function restarts the video output with those format settings. If a matching mode is not found, the video output format is unchanged and a restart does not occur.

The clocked video output function is configured with two video modes, progressive 1280x720 pixels and interlaced 1920x1080 pixels.

In addition, the Clocked Video Output MegaCore function provides a queue where pixels can wait when the video output is in blanking and does not need pixel data. If this FIFO ever becomes full, then the flow controlled interface indicates that the clocked video output is not ready for data and earlier parts of the pipe are stalled.

IP Configurations

This section provides information on the configuration of the IP components in the V2 reference design.

Input SDI MegaCore Function

The SDI MegaCore function is configured as a triple-rate receiver serial digital interface. Two instances are required, one for each source. The SDI input clock frequency is 148.5/148.35 MHz for 3Gb/s (3G-SDI) or 74.25/74.175 MHz for 1.5Gb/s (HD-SDI) video inputs, or 27.0 MHz for SD-SDI video inputs. In this design, a clock frequency of 148.5 MHz is used which allows use of SD-SDI, HD-SDI and 3G-SDI input clocks.



For more information about the SDI MegaCore function, refer to the *SDI MegaCore Function User Guide*.

Clocked Video Input (BETA) MegaCore Function

Two instances of the Clocked Video Input MegaCore function are required, one for each video stream. Both MegaCore functions have the following configurations:

- Input of 20-bit wide video (two 10-bit color planes in parallel, Y' with alternating Cb and Cr), and syncs embedded in the video to support HD-SDI or 3G-SDI (SMPTE 292M/424M).
- The least significant 10 bits of 20-bit wide input data (two 10-bit color planes in sequence, Y' with alternating Cb and Cr), and syncs embedded in the video stream to support SD-SDI. This feature is enabled by turning on **Allow color planes in sequence input** in the **Parameter Settings** GUI. The MegaCore function re-sequences the color planes to output the data as two 10-bit color planes in parallel.
- Output of two 10-bit color planes in parallel transmitted as Avalon-ST Video protocol.
- FIFO depth of one line of the maximum video line length supported (1920 pixels). The data is 20 bits wide.
- Run-time control is enabled by turning on **Use control port**. This provides an Avalon-MM slave interface, with interrupt support to indicate resolution changes. Status information, such as the image resolution, stability of video and FIFO fill level are exposed through the register map to the software control code.
- Outputs timing signals that allow the phase frequency detector (PFD) and Clocked Video Output MegaCore function to lock the output video stream to the timing of the input video stream. (BETA functionality.)



For more information about the Clocked Video Input MegaCore function, refer to the *Video and Image Processing Suite User Guide*.

Clipper MegaCore Function

Two instances of the Clipper MegaCore function are required, one for each video stream. Both MegaCore functions are configured as follows:

- Support up to a maximum resolution of 1920×1080 pixels.


- A **Rectangle** clipping method to specify the width and height of the output video stream.
- Run-time control of the offset, width and height of the clipped rectangular region. This is enabled by turning on **Include Avalon-MM Interface** in the **Parameter Settings** GUI.
- Input of two 10-bit color planes in parallel, Y' with alternating Cb and Cr.
- Capable of clipping progressive (frames) or interlaced (fields) video.



For more information about the Clipper MegaCore function, refer to the [Video and Image Processing Suite User Guide](#).

Deinterlacer (BETA) MegaCore Function

Two instances of the Deinterlacer MegaCore function are required. Both of these MegaCore functions have the following configurations:

- High quality motion adaptive deinterlacing algorithm.
 - Supports up to a maximum resolution of 1920×1080 pixels.
 - Input of two 10-bit color planes in parallel (Y'CbCr 4:2:2). This feature is enabled by turning on **4:2:2 support for motion adaptive algorithm** in the **Parameter Settings** GUI. (BETA functionality).
 - Support for interlaced video input up to a resolution of 1080i60.
 - Support for progressive video input up to a resolution of 1080p60. This feature is enabled by turning on **Passthrough mode** in the **Parameter Settings** GUI.
 - Double buffering is used to store the video input in external memory.
 - Separate clock domains to decouple the Avalon-MM master interfaces (that transfer data to the external memory controller) from the data processing function.
-  Higher memory bandwidth can be achieved using the built in clock-crossing support compared to using a separate clock crossing bridge.
- The Avalon-MM master port widths (for the five bursting read and write masters) are set to 256 bits. The masters are configured to read/write large bursts of data for each bus transaction (Burst target = 64).



For more information about the Deinterlacer MegaCore function, refer to the [Video and Image Processing Suite User Guide](#).

Chroma Resampler MegaCore Function

Four instances of the Chroma Resampler MegaCore Function are required (two instances in each video stream).

The Chroma Resampler MegaCore functions before the Scaler has the following configuration:

- Input of two 10-bit color planes in parallel (Y' with alternating Cb and Cr), 4:2:2.
- Output of three 10-bit color planes in parallel, 4:4:4.
- Luma adaptive algorithm for horizontal resampling.

The Chroma Resampler MegaCore functions after the Scaler are configured as follows:

- Input of two 10-bit color planes in parallel, 4:4:4.
- Output of three 10-bit color planes in parallel (Y' with alternating Cb and Cr), 4:2:2.
- Luma adaptive algorithm for horizontal resampling.



For more information about the Chroma Resampler MegaCore function, refer to the *Video and Image Processing Suite User Guide*.

Scaler MegaCore Function

Two instances of the Scaler MegaCore function are required. Both of these MegaCore functions have the following configuration:

- Polyphase scaling algorithm with 6 horizontal and 6 vertical taps, and Lanczos-2 filtering function.
- Input and output of two 10-bit color planes in parallel (Y'CbCr).
- Run-time control of the output image size up to a maximum resolution of 1920×1080 pixels using an Avalon-MM Slave interface.



Software is provided to change the output resolution depending on the viewing mode or output resolution.

- Run-time coefficient calculation and reloading.



The Scaler automatically configures itself to process the next input video frame by decoding the Avalon-ST Video control packet. When an input or output resolution change occurs, the ratio between the input frame and output frame changes. To retain a high quality scaled image, it is necessary to change the coefficients used in the scaling algorithm. The coefficient loading is performed through the Avalon-MM slave interface. This feature is enabled by turning on **Load coefficient data at runtime** in the **Parameter Settings** GUI.



For more information about the Scaler MegaCore function, refer to the *Video and Image Processing Suite User Guide*.

Interlacer (BETA) MegaCore Function

Two instances of the Interlacer MegaCore function are required. Both of these MegaCore functions have the following configuration:

- Line discarding algorithm.
- Run-time control of interlaced output (when disabled progressive frames are passed through unchanged).
- Passthrough of interlaced input unchanged.
- Bufferless mode (no external memory required).

Frame Buffer (BETA) MegaCore Function

Two instances of the Frame Buffer MegaCore function are required. Both instances have the following configuration:

- Input and output of two 10-bit color planes in parallel (Y'CbCr, 4:2:2).
- 256-bit wide Avalon-MM read and write masters (for 64-bit DDR2 Memory interface).
- Frame dropping and frame repeating are enabled to perform rate conversion when the input and output frame rates don't match.



Dropping and repeating frames at multiple points in a video pipeline may potentially result in a juddering effect, with multiple consecutive frames being dropped/repeated.

- The Avalon-MM master port widths (for the read and write masters) are set to 256 bits. The masters are configured to read/write large bursts of data for each bus transaction (Burst target = 64).
- Support for Interlaced video. (BETA functionality.)



For more information about the Frame Buffer MegaCore function, refer to the [Video and Image Processing Suite User Guide](#).

DDR2 SDRAM High Performance Controller MegaCore Function

This MegaCore function is the controller for the external DDR2 SDRAM and is required as a buffer for the two Deinterlacer and the two Frame Buffers. The DDR2 SDRAM High Performance Controller has the following configuration:

- Micron MT9HF6472AY-5EB38 (72-bit, 512MByte, 533MT/s, CL4, 266.7 MHz).
- Memory set up in Half-Rate mode (266.7 MHz internally, 133.35 MHz externally).
- Data width set to use 64 bits with a local data interface of 256 bits (due to half rate mode).



For more information about the DDR2 SDRAM High Performance Controller MegaCore Function, refer to the [DDR and DDR2 SDRAM High Performance Controller User Guide](#).

Nios II Processor

The Nios II Processor is configured as a Nios II/e core, clocked at 62.5 MHz. The processor's Avalon-MM data master can access the register map of many of the components in the V2 system, including a number of the Video and Image Processing Suite MegaCore functions and system peripherals.

[Figure 6 on page 16](#) shows the data master connections in the SOPC Builder patch panel view for the V2 reference design.

The Nios II processor executes the control software for the system as described in [“Nios II Software” on page 30](#), providing a highly configurable software design flow.

Figure 6. Data Master Connections in the SOPC Builder Patch Panel View

Use	Con...	Module Name	Description	Clock	Base	End	IRQ
<input checked="" type="checkbox"/>		altmemddr	DDR2 SDRAM High Performance Cont...	mem_clk	0x00000000	0x1fffffff	
<input checked="" type="checkbox"/>		sdi_in_1	Clocked Video Input BETA	vip_clk	0x00205a00	0x00205bff	0
<input checked="" type="checkbox"/>		my_alt_vip_clip_1	Clipper	vip_clk	0x00205d00	0x00205dff	
<input checked="" type="checkbox"/>		pipeline_bridge_1	Avalon-MM Pipeline Bridge	altmemddr...	0x00000000	0x1fffffff	
<input checked="" type="checkbox"/>		alt_vip_dil_1	Deinterlacer BETA	multiple			
<input checked="" type="checkbox"/>		crs_to444_1	Chroma Resampler	vip_clk			
<input checked="" type="checkbox"/>		my_alt_vip_scl_1	Scaler	vip_clk	0x00205400	0x002057ff	
<input checked="" type="checkbox"/>		crs_to422_1	Chroma Resampler	vip_clk			
<input checked="" type="checkbox"/>		alt_vip_itl_1	Interlacer BETA	vip_clk	0x00206080	0x002060ff	
<input checked="" type="checkbox"/>		fb_pipeline_bridge_1	Avalon-MM Pipeline Bridge	altmemddr...	0x00000000	0x1fffffff	
<input checked="" type="checkbox"/>		alt_vip_vfb_1	Frame Buffer BETA	multiple			
<input checked="" type="checkbox"/>		sdi_out_1	Clocked Video Output BETA	vip_clk	0x00200000	0x00201fff	1
<input checked="" type="checkbox"/>		sdi_in_2	Clocked Video Input BETA	vip_clk	0x00205800	0x002059ff	2
<input checked="" type="checkbox"/>		my_alt_vip_clip_2	Clipper	vip_clk	0x00205e00	0x00205eff	
<input checked="" type="checkbox"/>		pipeline_bridge_2	Avalon-MM Pipeline Bridge	altmemddr...	0x00000000	0x1fffffff	
<input checked="" type="checkbox"/>		alt_vip_dil_2	Deinterlacer BETA	multiple			
<input checked="" type="checkbox"/>		crs_to444_2	Chroma Resampler	vip_clk			
<input checked="" type="checkbox"/>		my_alt_vip_scl_2	Scaler	vip_clk	0x00205000	0x002053ff	
<input checked="" type="checkbox"/>		crs_to422_2	Chroma Resampler	vip_clk			
<input checked="" type="checkbox"/>		alt_vip_itl_2	Interlacer BETA	vip_clk	0x00206100	0x0020617f	
<input checked="" type="checkbox"/>		fb_pipeline_bridge_2	Avalon-MM Pipeline Bridge	altmemddr...	0x00000000	0x1fffffff	
<input checked="" type="checkbox"/>		alt_vip_vfb_2	Frame Buffer BETA	multiple			
<input checked="" type="checkbox"/>		sdi_out_2	Clocked Video Output BETA	vip_clk	0x00202000	0x00203fff	3
<input checked="" type="checkbox"/>		cpu	Nios II Processor				
		instruction_master	Avalon Memory Mapped Master	cpu_clk			
		data_master	Avalon Memory Mapped Master				
		jtag_debug_module	Avalon Memory Mapped Slave		0x00204800	0x00204fff	31
<input checked="" type="checkbox"/>		tristate_bridge	Avalon-MM Tristate Bridge	cpu_clk			
<input checked="" type="checkbox"/>		ssram	Cypress CY7C1380C SSRAM	cpu_clk	0x00000000	0x001fffff	
<input checked="" type="checkbox"/>		sysid	System ID Peripheral	cpu_clk	0x00206500	0x0020653f	
<input checked="" type="checkbox"/>		jtag_uart	JTAG UART	cpu_clk	0x00206540	0x0020657f	4
<input checked="" type="checkbox"/>		timer	Interval Timer	cpu_clk	0x00205c00	0x00205cff	5
<input checked="" type="checkbox"/>		dip	PIO (Parallel I/O)	cpu_clk	0x00205f00	0x00205ff7f	
<input checked="" type="checkbox"/>		leds	PIO (Parallel I/O)	cpu_clk	0x00205f80	0x00205fff	
<input checked="" type="checkbox"/>		buttons	PIO (Parallel I/O)	cpu_clk	0x00206000	0x0020607f	
<input checked="" type="checkbox"/>		sync_speed	PIO (Parallel I/O)	cpu_clk	0x00206480	0x002064ff	

Clocked Video Output (BETA) MegaCore Function

Two instances of the Clocked Video Output MegaCore function are required. Both instances have the following configuration:

- 20-Bit Y'CbCr, 4:2:2, embedded syncs, for transmission over SDI.
- Avalon-ST Video protocol input.
- FIFO depth of one line of the maximum video line length supported (1,920 pixels).
- Run-time control is enabled by turning on **Use control port** in the **Parameter Settings** GUI. This provides an Avalon-MM slave interface.
- Three run-time configuration video modes to allow run-time switching between different output resolutions. The resolution, sync, and blanking data for each mode is loaded from software executed on the Nios II processor, using the Avalon-MM slave interface.

720p60, and 1080i60 are demonstrated in the V2 reference design.

- Clock-crossing between the system clock and the pixel clock which is enabled when **Video in and out use the same clock** is turned off in the **Parameter Settings** GUI.
- Status information, such as underflow status and FIFO fill level is exposed through the register map to the software control code.
- Support for locking to an external sync reference (genlock). (BETA functionality.)



For more information about the Clocked Video Output MegaCore function, refer to the *Video and Image Processing Suite User Guide*.

Output SDI MegaCore Function

The SDI MegaCore function is configured as a triple-rate transmitter serial digital interface. Two instances are required, one for each output. The SDI output clock frequency is 148.5/148.35 MHz for 3Gb/s (3G-SDI), 74.25/74.175 MHz for 1.5Gb/s (HD-SDI), or 27.0 MHz (SD-SDI).

System Peripherals

The system contains the following useful memory-mapped peripheral and parallel I/O components to provide information on the status of the design:

- A JTAG UART to display software `printf` output.
- Push buttons configured to allow switching between video output resolutions (refer to “*Output Resolution*” on page 18).
- Dual in-line package (DIP) switches (not used).
- LEDs for system status (refer to “*Board LEDs Status Information*” on page 18).
- Quad seven-segment display to show the standard of the input video streams (refer to “*Quad Seven-Segment Display Status Information*” on page 19).

Clock Domains

The video datapath is clocked by the system clock `vip_clk` at 148.5 MHz, to allow the processing functions to process progressive frames of video with resolutions up to 1920×1080 pixels at a rate of 60 frames per second.

The Nios II processor, which executes the software that configures the video processing functions, is clocked at a lower clock rate of 62.5 MHz.

The DDR2 SDRAM is run at its maximum rate of 266 MHz. with the memory controller run in half-rate mode clocked at 133 MHz. The Deinterlacer and the Frame Buffer MegaCore functions de-couple the data processing clock rate (148.5 MHz) from the memory controller clock domain (133.33 MHz).

The Clocked Video Input and Clocked Video Output MegaCore functions provide clock domain crossing, which allows the DVI output and SDI input to run at the speed of the relevant standard used.

Run Time Resolution Changes

You can make the following changes at run time:

- The resolution of the input video streams
- The output resolution. Using the push-buttons on the Stratix II GX audio video development board.

Input Resolution

The Clocked Video Input MegaCore function retains a count of the height, width and format (progressive or interlaced) of the input video, and detects changes to the resolution or format. After stable video is captured, the Clocked Video Input MegaCore function generates control packets containing the new resolution and progressive/interlaced format. The control packets propagate through each function in the data path, with each MegaCore function configuration ready to receive the new video data.

The Clocked Video Input MegaCore function also generates an interrupt when a resolution change occurs, with the interrupt line connected to the Nios II processor. When an interrupt is generated, the Nios II interrupt service routine performs the following functions:

- Calculate new coefficients based on the new scaling ratio
- Writes new clipping area parameters
- Writes new scaling coefficients to the scaler control interface

Output Resolution

The PB0 and PB1 push-button switches on the Stratix II GX audio video development board can be used to change the resolution on the displays.

Pressing PB1 successively, cycles through the output resolutions for the first SDI display. The supported resolutions are:

- Interlaced 1920×1080 pixels @ 60 Hz
- Progressive 1280×720 pixels @ 60 Hz

Pressing PB0 successively, cycles through the output resolutions for the second SDI display. The supported resolutions are:

- Interlaced 1920×1080 pixels @ 60 Hz
- Progressive 1280×720 pixels @ 60 Hz



A monitor capable of displaying a 1920×1080 pixel resolution is required. The monitor should be configured to display the video without scaling to the native resolution (1:1 pixel mapping).

Board LEDs Status Information

The LEDs on the Stratix II GX audio video development board provide the status information listed in [Table 3 on page 19](#).

Table 3. Board LED Status Information

LED	Status
0	Overflow at the input or underflow at the output has occurred.
1	Video stream 1 is active.
2	Video stream 2 is active.
3	Heartbeat (flashes if the software is running).
4	The Output 1 resolution is progressive 1280×720 pixels.
5	The Output 1 resolution is interlaced 1920×1080 pixels.
6	The Output 2 resolution is progressive 1280×720 pixels.
7	The Output 2 resolution is interlaced 1920×1080 pixels.

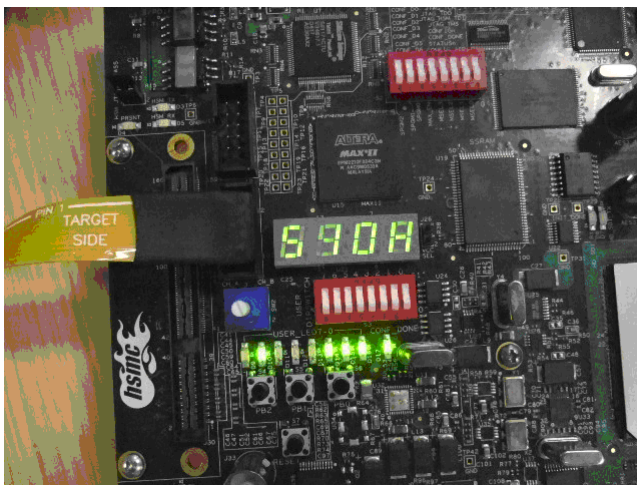
Quad Seven-Segment Display Status Information

Table 4 shows the quad seven-segment display on the Stratix II GX audio video development board provide information about the input standard.

Table 4. Quad Seven-Segment Display Status Information

Segment	Status
0	Video stream 1 standard received (S = SD-SDI, H = HD-SDI, 3 = 3G-SDI)
1	Video stream 1 clock rate (3 = 148.5 MHz, 2 = 148.35 MHz, 1 = 74.25 MHz, 0 = 74.175 MHz) or SD
2	Video output reference clock (0 = 148.5MHz, 1 = 148.35 MHz)
3	Video stream 2 standard received (S = SD-SDI, H = HD-SDI, 3 = 3G-SDI)

For example, the seven-segment display in Figure 7 on page 19 displays S, 1, 0, H, which means high definition video on stream 1, at 74.175 MHz clock rate, with a 148.35 MHz output reference clock (refer to “Clocking” on page 22 for an explanation of the output reference clock), and standard definition video on stream 2.

Figure 7. Quad Seven-Segment Display Showing Input Video Standard

External Memory Bandwidth Calculations

Table 5 summarizes the maximum bandwidth requirement for each MegaCore function that accesses external DDR2 SDRAM in the V2 reference design.

Table 5. MegaCore Function Bandwidth Requirements

MegaCore Function	Maximum Input Rate	Maximum Output Rate	Master Type	Maximum Bandwidth (Gbps)
Motion Adaptive Deinterlacer (Stream1)	(1080i60) $1920 \times 1080 \times 21.3 \times 3 = 1.327$ Gbps	(1080p60) $1920 \times 1080 \times 21.3 \times 60 = 2.654$ Gbps	1 × write at input rate	1.327
			2 × read at output rate	5.308
			1 motion data read at 1/4 output rate	0.637
			1 motion data write at 1/4 output rate	0.637
Frame Buffer (Stream 1)	(1080p60) $1920 \times 1080 \times 21.3 \times 60 = 2.654$ Gbps	(1080p60) $1920 \times 1080 \times 21.3 \times 60 = 2.654$ Gbps	1 × write at input rate	2.654
			1 × read at output rate	2.654
Motion Adaptive Deinterlacer (Stream2)	(1080i60) $1920 \times 1080 \times 21.3 \times 30 = 1.327$ Gbps	(1080p60) $1920 \times 1080 \times 21.3 \times 60 = 2.654$ Gbps	1 × write at input rate	1.327
			2 × read at output rate	5.308
			1 motion data read at 1/4 output rate	0.637
			1 motion data write at 1/4 output rate	0.637
Frame Buffer (Stream 2)	(1080p60) $1920 \times 1080 \times 21.3 \times 60 = 2.654$ Gbps	(1080p60) $1920 \times 1080 \times 21.3 \times 60 = 2.654$ Gbps	1 × write at input rate	2.654
			1 × read at output rate	2.654

Note to Table 5:

- (1) The motion data is 1/4 the output rate because it is created for each input field (1/2 the output rate) and is stored using a single 10-bit value (not 20-bits as used for the video data).

The maximum memory bandwidth is achieved is when the input resolution is 1080i60 and the output resolution is 1080p60 on both video streams. The maximum total bandwidth is then the sum of the bandwidths for the motion adaptive Deinterlacer and Frame Buffer MegaCore functions.

$$\begin{aligned} \text{MaxBandwidth} &= \text{Motion Adaptive Deinterlacer} + \text{Frame Buffers} \\ &= 7.909 + 7.909 + 5.308 + 5.308 = 26.434 \text{ Gbps} \end{aligned}$$

The Stratix II GX audio video development board when used with the Micron MT9HTF6472AY- 53EB3 high-performance DDR2 SDRAM provides a maximum theoretical bandwidth of:

$$266.7 \text{ MHz} \times 64 \text{ bits} \times 2 \text{ (both clock edges used)} = 34.133 \text{ Gbps}$$

Thus a memory access efficiency requirement of 77.5%.

The memory bandwidth efficiency is determined by a number of factors, such as randomness of addresses, refresh rate, turnaround times between reads and writes, and burst lengths. Altera's memory controllers can reach an efficiency of up to about 90% if the access conditions are right (long bursts of writes to the same column followed by long bursts of reads).

Using a half-rate memory controller to satisfy the memory bandwidth requirements means that the local interface width between memory controller and internal FPGA logic is 256 bits ($= 4 \times 64$ bits). Both the DDR2 SDRAM and memory controller run at 266 MHz, while the internal FPGA blocks run at half this rate, 133 MHz.

Memory Subsystem Architecture

To achieve the 77.5% memory access efficiency required by the V2 reference design, the memory subsystem has been architected to minimize the impact of bank management commands (non-data transfer) commands sent to the DDR2 SDRAM, by using:

- Long bursts
- Large on-chip buffers
- Bank interleaving

Figure 8 shows the Avalon-Memory Mapped Interfaces settings panel for the Deinterlacer and Frame Buffer MegaCore functions.

Figure 8. Avalon-MM Interfaces

Avalon Memory-Mapped Interfaces

☒ Use separate clocks for the Avalon-MM master interfaces

External memory port width : 256

Write-only master interface FIFO depth : 128

Write-only master interface burst target : 64

Read-only master interface FIFO depth : 128

Read-only master interface burst target : 64

Base address of frame buffers : 0x08000000

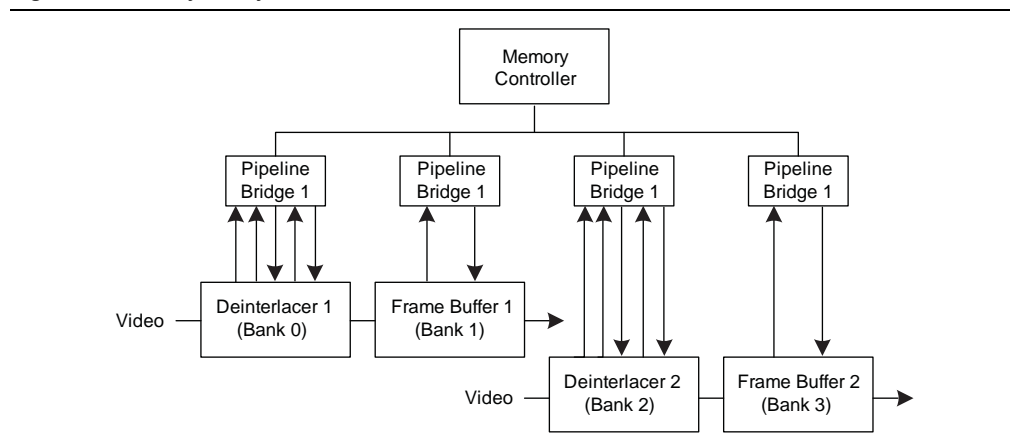
3 frame buffers are required, a total of 16200 kilobytes.

Long bursts (within a row) minimize the impact of the initial setup commands. Making the Deinterlacer and Frame Buffer burst size a factor of the row size (64 words in the GUI) ensures that a burst does not cross a row boundary.

Large on-chip buffers (in the Deinterlacer and Frame Buffer) are necessary to buffer enough data to create/receive the long bursts. By making the buffer size twice the burst size (128 words in the GUI), a burst can be being transferred to and from the DDR2 whilst the next or previous burst is processed. This action allows the video processing data path to cope with the longer latencies created by the arbitration of a multi-master system all doing long bursts. The on-chip buffers are also used for crossing the clock domains between the Video and Image Processing Suite MegaCore functions and the memory controller. This action allows the memory to be run at a higher frequency without affecting the speed that the data path runs at.

Bank interleaving further reduces the penalty for switching rows by overlapping the bank management commands of one bank with the data transfer to and from another bank. Figure 9 shows the grouping of masters into banks using pipeline bridges.

Figure 9. Memory Subsystem



This arrangement allows the round robin arbitration of SOPC Builder to switch between banks efficiently. For example if Deinterlacer 1 needs to perform a write, Frame Buffer 1 a read, Deinterlacer 2 a read, and Frame Buffer 2 a write, the accesses would be scheduled as:

1. Pipeline bridge 1—Deinterlacer 1 write to Bank 0
2. Pipeline bridge 2—Frame Buffer 1 read to Bank 1
3. Pipeline bridge 3—Deinterlacer 2 read to Bank 2
4. Pipeline bridge 4—Frame Buffer 2 write to Bank 3

The Deinterlacer and Frame Buffer can be allocated to different banks by setting their base addresses in the GUI.

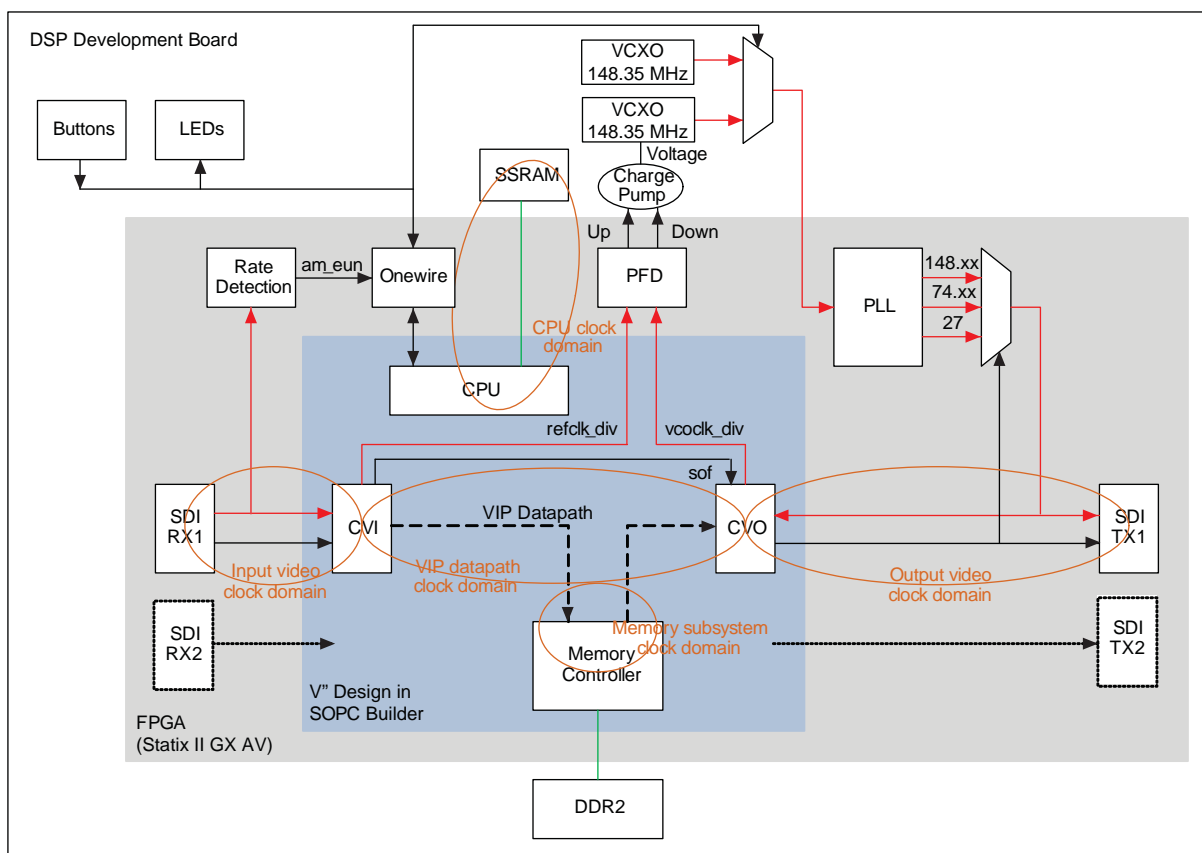
Designing a system in this way works best when:

- The bandwidth requirements for each master are equal
- The bandwidth requirements for each group of masters are equal
- There are enough banks for each Frame Buffer and Deinterlacer to have one each
- There are no random access masters, Nios program code is kept in a separate memory

Clocking

The V2 reference design uses a number of different clock domains. Figure 10 on page 23 shows how these domains interact.

The video data path runs at a fixed frequency of 148.5 MHz. The two SDI MegaCore functions create separate recovered video clocks, extracted from the video inputs, that can run up to 148.5 MHz. The two Clocked Video Input MegaCore functions handle the clock crossing from the input video frequency to the fixed video data path frequency using clock crossing FIFOs.

Figure 10. Top Level of the V2 Reference Design

The two output video clock frequencies are selected from a clock multiplexer, one per output, and can be independently set to 148.35 or 148.5 MHz, 74.175 or 74.25 MHz, or 27 MHz. The value of the selectable output frequencies for both outputs is based on the detected frequency of video input 1.

Table 6 shows a list of the available output frequencies.

Table 6. Available Output Frequencies

Input Frequency (MHz)	Output Frequencies (MHz)
27	148.35, 74.175 or 27
74.175	148.35, 74.175 or 27
74.25	148.5, 74.25 or 27
148.35	148.35, 74.175 or 27
148.5	148.5, 74.25 or 27

The two Clocked Video Output MegaCore functions handle the clock crossing from the fixed video data path frequency to the output video frequency using clock crossing FIFOs.

The DDR2 SDRAM for buffering video frames in the Deinterlacer and Frame Buffer, runs at a frequency of 266.67 MHz. The memory subsystem, consisting of the slave port of the DDR2 High Performance Memory Controller, the Avalon Switch fabric and the master ports of the Deinterlacer and Frame Buffer run at 133.33 MHz. The Deinterlacer and Frame Buffer handle the clock crossing from the video data path frequency to the memory subsystem using clock crossing FIFOs.

The CPU and its SSRAM are run at a frequency of 62.5 MHz. SOPC Builder automatically handles the insertion of clock crossing adapters so that the CPU can access the control ports of the Video and Image Processing Suite MegaCore functions.

Generator Lock

Generator lock is the technique of locking the timing of video outputs to a reference source. Sources that are locked to the same reference can be switched between cleanly, on a frame boundary. There are two derivatives of generator lock:

- Genlock—aligning the horizontal (hsync) and vertical (vsync) timing of the output video to a reference source with the same format and frame rate. For example, 1080p60 to 1080p60.
- Framelock (crosslock)—aligning the start of frame of the output video to a reference source of a different format with a lockable frame rate. For example, 720p60 to 1080i60, or 720p30 to 1080p60.

For the rest of this section, the term genlock is used to refer to both genlock and framelock.

Video output 1 is locked using video input 1 as a reference source. Locking is achieved in two stages:

1. Clock locking—aligning the output video clock to the input video clock and tracking any changes.
2. Frame locking—aligning the start of frame (SOF) of the output video to the SOF of the input video.

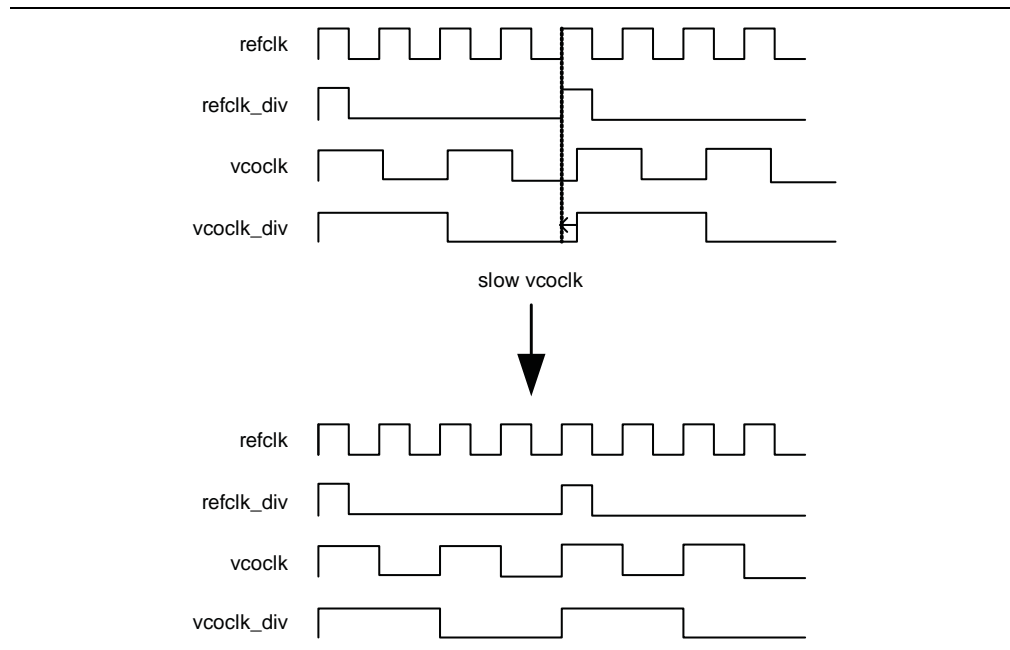
Clock Locking

The V2 reference design utilizes the voltage controlled crystal oscillators (VCXO) on the Stratix II GX audio video development board to perform clock locking. The control software reads back the clock frequency of the input video from the registers of the Clock Video Input MegaCore function. The software compares this clock frequency to the clock frequency of the output video and determines the divider values for both the input and output video clocks that produce the same period.

For example, when comparing 27 MHz to 74.25 MHz, the divider values are 91 and 250 respectively. The Clocked Video Input MegaCore function outputs `refclk_div`, a divided down version of the input video clock, and the Clocked Video Output MegaCore function outputs `vcoclk_div`, a divided down version of the output video clock.

Figure 10 on page 23 shows how `refclk_div` and `vcoclk_div` are connected. The phase frequency detector (PFD) block compares these two clocks, `refclk_div` and `vcoclk_div`, and matches the period of `vcoclk_div` to `refclk_div` using the VCXO to speed up or slow down `vcoclk_div` (Figure 11).

Figure 11. Clock Locking



The Up and Down signals output from the PFD are used to control the frequency of the `vcoclk_div` signal. The Down signal is held high whilst the Up signal is pulsed to charge the capacitors of the Charge Pump.

Increasing the high portion of the Up pulse results in the voltage output from the Charge Pump increasing and the frequency of `vcoclk_div` increasing.

Decreasing the size of the high portion of the Up pulse results in the voltage output from the Charge Pump decreasing and the frequency of `vcoclk_div` decreasing.

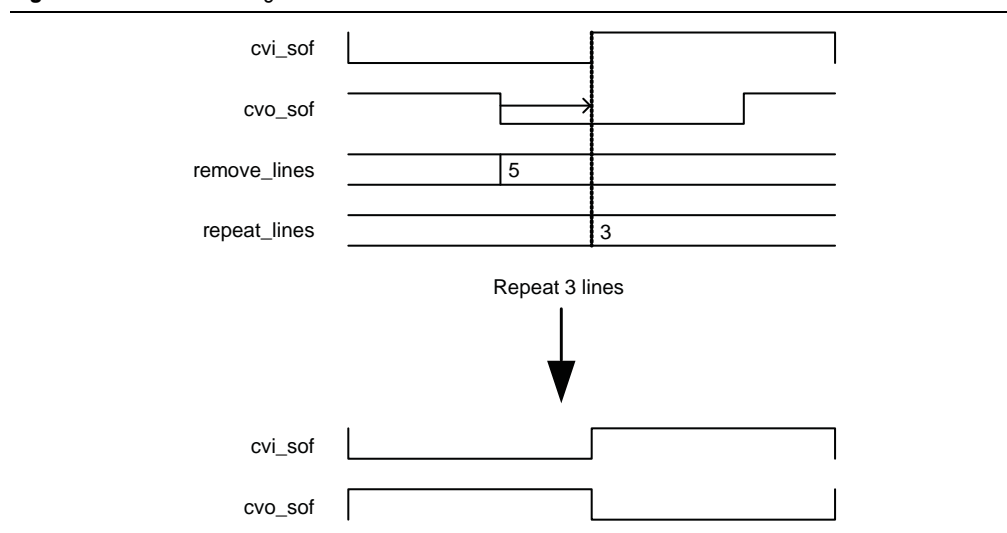
Frame Locking

The Clocked Video Output MegaCore function performs frame locking by aligning the SOF of the output video to the SOF produced by the Clocked Video Input MegaCore function.

The Clocked Video Input MegaCore function outputs a SOF signal that is connected to the Clocked Video Output MegaCore function. A SOF is indicated by an edge transition (0 to 1 or 1 to 0). The SOF position can be set anywhere within the input video frame by loading the registers of the Clock Video Input MegaCore function (and Clocked Video Output MegaCore function) with the sample and line (measured from the rising edge of the V sync) on which the SOF is to occur.

Figure 12 shows an example of the SOF transitions.

Figure 12. Frame Locking



The Clocked Video Output MegaCore function compares the two SOF signals to determine how far apart they are. It then repeats or removes that number of samples and lines in the output video to align the two sof signals. If the SOFs are less than a threshold value of samples apart (the value of the `vcoclk` divider register), the Clocked Video Output MegaCore function does not alter the output video.

Table 7 shows the modified control register map used by the Clocked Video Input BETA MegaCore function.

Table 7. Clocked Video Input BETA MegaCore Function Control Register Map

Address	Register	Description
0	Control	Same as user guide.
1	Status	Same as User Guide except bit 10 which indicates a valid resolution in the sample and line count registers.
2	UsedW	Same as user guide.
3	ActiveSampleCount	The detected sample count of the video streams excluding blanking.
4	F0ActiveLineCount	The detected line count of the video streams F0 field excluding blanking.
5	ActiveSampleCount	Same as address 3.
6	F1ActiveLineCount	The detected line count of the video streams F1 field excluding blanking.
7	TotalSampleCount	The detected sample count of the video streams including blanking.
8	F0TotalLineCount	The detected line count of the video streams F0 field including blanking.
9	F1TotalLineCount	The detected line count of the video streams F1 field including blanking.
10	Standard	The contents of the <code>vid_std</code> signal.
11	SofSample	Start of frame sample and subsample register. The sample and sub-sample upon which the SOF occurs (and the <code>sof</code> signal triggers).
12	F0SofLine	Field 0 start of frame line register. The line upon which the SOF occurs (in field 0).
13	–	Reserved.
14	RefclkDivider	Number of cycles of <code>vid_clk</code> (<code>refclk</code>) before <code>refclk_div</code> signal triggers.

Table 8 shows the modified control register map used by the Clocked Video Output BETA MegaCore function.

Table 8. Clocked Video Output BETA MegaCore Function Control Register Map

Address	Register	Description
0 to 22	–	Same as user guide.
23	SofSample	Start of frame sample and subsample register. The sample and sub-sample upon which the SOF occurs (and the vid_sof signal triggers).
24	F0SofLine	Field 0 start of frame line register. The line upon which the SOF occurs (in field 0).
25	–	Reserved.
26	VcoclDivider	Number of cycles of vid_clk (vcocl) before refclk_div signal triggers.

Reviewing the V2 Reference Design

This section is a walkthrough that demonstrates how to construct a video processing application using the Altera video and image processing framework. This section includes a description of the top level design file in the Quartus II software, the full hardware system in SOPC Builder, and the configuration and control software code in the Nios II IDE environment.

To review the complete reference design, perform the following steps:

1. Run the Quartus II software to ensure the QUARTUS_ROOTDIR environment variable is correctly set.
2. Close the Quartus II software.
3. Install the ACS patch as described in “Installing the Reference Design” on page 3.
4. In Windows Explorer, browse to the install directory.
5. Double-click on **make_project_V2.bat** to copy the **V2.sopc** and **config.v** files to the top level, and create the quartus project **s2gxav.qpf** and **s2gxav.qsf**.

Quartus II Top Level

1. To review the top level Quartus II system file, perform the following steps:
2. Launch the Quartus II software.
3. On the File menu in the Quartus II software, click **Open Project**, browse to the **V2_<revision>** directory where the reference design is installed and open the Quartus II project file: **s2gxav.qpf**.
4. On the File menu, click **Open**, browse into the **s2gxav** subdirectory and open the top-level Verilog HDL file **config.v**.



This file controls which interfaces appear in the top-level Verilog HDL file **s2gxav.v**. The interfaces enabled in the **config.v** file must match those that appear in the SOPC Builder system. If interfaces are removed in the SOPC Builder system they must be commented out in the **config.v** file to remove them from the top level Verilog HDL file **s2gxav.v**.

5. On the File menu, click **Open**, browse into the **top** subdirectory and open the top level verilog file **s2gxav.v**.

The top level verilog file **s2gxav.v** instantiates the following modules:

- Phase Locked Loop (PLL) module to generate clocks for different video output standards (SD, HD and 3 Gbps).
- PLL reconfiguration and control modules to allow reconfiguration of the PLL, for the DVI output, at runtime.
- Two SDI modules configured in triple rate mode to receive data input in the SD-SDI, HD-SDI, or 3G-SDI standard.
- Quad seven-segment display configuration module to display video standard input and system status.
- A onewire interface handling clock multiplexing, switch input and LED output.
- SOPC Builder system.
- Two SDI modules configured in triple-rate mode to transmit data output in the SD-SDI, HD-SDI, or 3G-SDI standard.

The Quartus II project also contains the following timing constraint files: **s2gxav.sdc**, **altmemddr_phy_ddr_timing.sdc**, **alt_vip_cvi.sdc**, and **alt_vip_cvo.sdc**.

SOPC Builder System

To review the SOPC Builder system, perform the following steps:

1. Launch SOPC Builder by clicking **SOPC Builder** on the Tools menu in the Quartus II software, with the reference design project open.

Figure 13 on page 29 shows the complete SOPC Builder for the V2 reference design.

The system contains the following fully parameterizable components:

- Video processing and buffering functions.
- DDR2 external memory controller.
- Nios II processor for system configuration and control.
- Peripheral and parallel I/O components.
- Bridge components (pipeline and tri-state bridges).

All components in the SOPC Builder system are connected by either Avalon-MM or Avalon-ST interfaces. The Clocked Video Input and Clocked Video Output MegaCore functions form the boundary of the video processing datapath, exporting signals for connection to video interfaces at the top level.

Figure 13. Complete SOPC Builder System for the V2 Reference Design

	Module Name	Description	Clock
Video Stream 1	+ altmemddr	DDR2 SDRAM High Performance Controller	mem_clk
	+ sdi_in_1	Clocked Video Input BETA	vip_clk
	+ my_alt_vip_clip_1	Clipper	vip_clk
	+ pipeline_bridge_1	Avalon-MM Pipeline Bridge	altmemddr...
	+ alt_vip_dil_1	Deinterlacer BETA	multiple
	+ crs_to444_1	Chroma Resampler	vip_clk
	+ my_alt_vip_scl_1	Scaler	vip_clk
	+ crs_to422_1	Chroma Resampler	vip_clk
	+ alt_vip_itl_1	Interlacer BETA	vip_clk
	+ fb_pipeline_bridge_1	Avalon-MM Pipeline Bridge	altmemddr...
Video Stream 2	+ alt_vip_vfb_1	Frame Buffer BETA	multiple
	+ sdi_out_1	Clocked Video Output BETA	vip_clk
	+ sdi_in_2	Clocked Video Input BETA	vip_clk
	+ my_alt_vip_clip_2	Clipper	vip_clk
	+ pipeline_bridge_2	Avalon-MM Pipeline Bridge	altmemddr...
	+ alt_vip_dil_2	Deinterlacer BETA	multiple
	+ crs_to444_2	Chroma Resampler	vip_clk
	+ my_alt_vip_scl_2	Scaler	vip_clk
	+ crs_to422_2	Chroma Resampler	vip_clk
	+ alt_vip_itl_2	Interlacer BETA	vip_clk
Processor	+ fb_pipeline_bridge_2	Avalon-MM Pipeline Bridge	altmemddr...
	+ alt_vip_vfb_2	Frame Buffer BETA	multiple
Peripherals	+ sdi_out_2	Clocked Video Output BETA	vip_clk
	+ cpu	Nios II Processor	cpu_clk
	+ tristate_bridge	Avalon-MM Tristate Bridge	cpu_clk
	+ ssram	Cypress CY7C1380C SSRAM	cpu_clk
	+ sysid	System ID Peripheral	cpu_clk
	+ jtag_uart	JTAG UART	cpu_clk
	+ timer	Interval Timer	cpu_clk
	+ dip	PIO (Parallel I/O)	cpu_clk
	+ leds	PIO (Parallel I/O)	cpu_clk
	+ buttons	PIO (Parallel I/O)	cpu_clk
	+ sync_speed	PIO (Parallel I/O)	cpu_clk

- To review the configuration of the IP components in SOPC Builder, select the IP module name (for example **my_alt_vip_scl_1**) and click **Edit**. This displays the corresponding MegaWizard interface **Parameter Settings** GUI page.
- In SOPC Builder, generate the SOPC Builder system by clicking on **Generate**. Progress messages are issued in the SOPC Builder System Generation window and should complete with a message:
Info: System generation was successful
- Close SOPC Builder and click **Start Compilation** on the Processing menu to compile the Quartus II project.

Nios II Software

The V-Series of reference designs are highly configurable at run time by software executing on a Nios II processor.

The C++ source code provided performs the following functions:

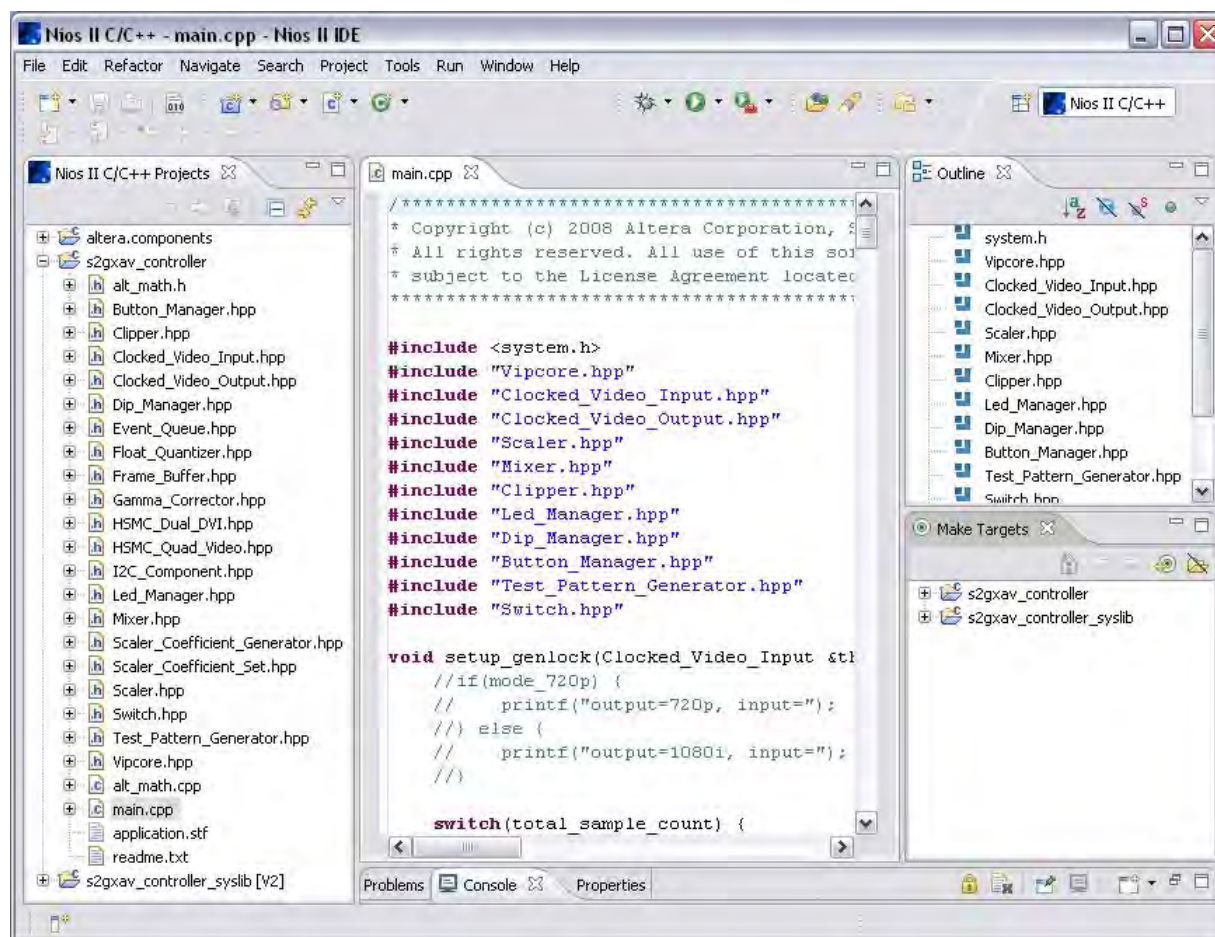
- Initializes the Video and Image processing Suite MegaCore functions:
 - Clocked Video Inputs to generate an interrupt on stable resolution changes.
 - Clocked Video Outputs with the resolution, sync and blanking data for the supported output modes (1080i60 and 720p60).
 - Starts all run time controllable functions.
- Configures board LEDs to provide status information. For example to show that the software is running, overflow or underflow has occurred, the current display mode, and so on. (Refer to “[Board LEDs Status Information](#)” on page 18 for more information.)
- Manages interrupts and button presses with event and button manager classes.
- Manages output switching triggered by button presses. (Refer to “[Run Time Resolution Changes](#)” on page 18). These changes include setting scaler output resolution and recalculating coefficients.
- Sets the clipper regions based on the video standard. For example, if the resolution detected is NTSC, a rectangle of 720×240 pixels is clipped from the interlaced video, with an offset of three lines from the top of the input stream.

C++ classes are included in the **software/s2gxav_controller** directory that provide a software API between the Nios II control code and the Video and Image Processing Suite MegaCore functions as well as peripherals and board components (such as DIP Switches, buttons, LEDs, and JTAG UART). The classes provide many member functions to accelerate software development and increase visibility of the data flow. For example, the clocked video input class (**Clocked_Video_Input.hpp**) member functions can report the level in a FIFO in the video system, or the member functions in the frame buffer class (**Frame_Buffer.hpp**) can report the number of frames that have been dropped or repeated by the Frame Buffer MegaCore function in the data path.

All Video and Image Processing Suite MegaCore function classes are derived from the base class **Vipcore.hpp** which contains methods common to all functions such as starting and stopping the video function processing at a frame boundary.

The main function in **main.cpp** demonstrates how to use the C++ class member functions to configure and control the data path functions.

[Figure 14 on page 31](#) shows the Nios II IDE environment and Nios II application project source files.

Figure 14. Nios II IDE Environment and Application Project Source Files

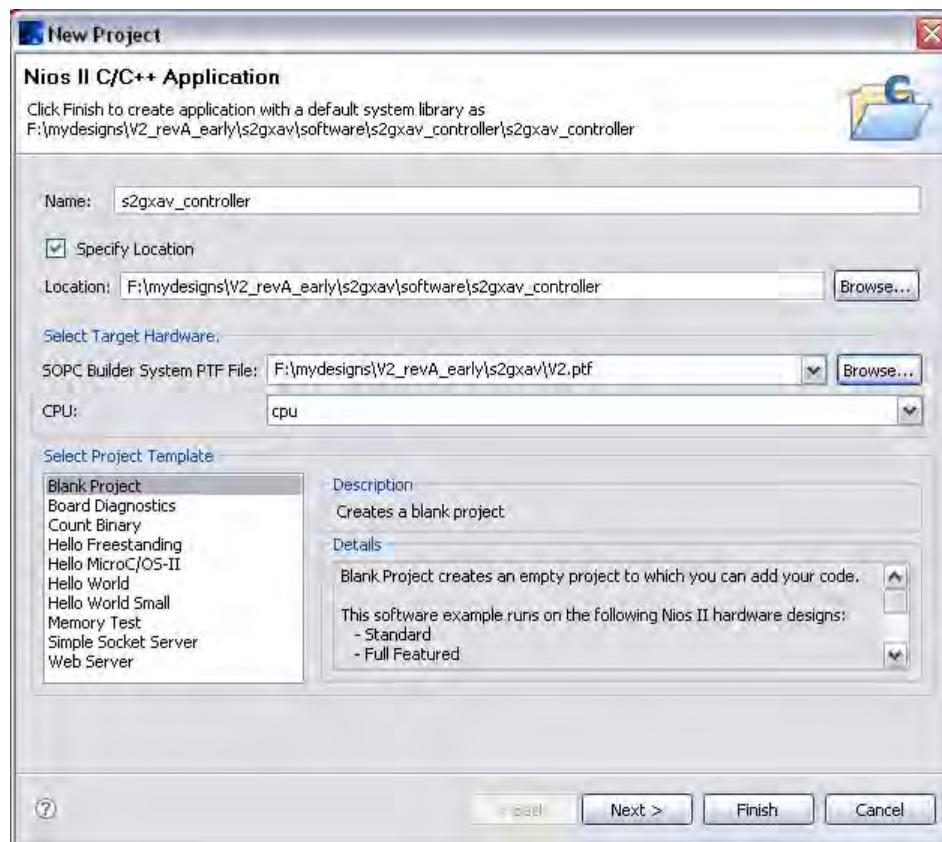
Building the Software in the Nios II IDE

Perform the following steps to build the software in the Nios II Integrated Development Environment (IDE):

1. Start the Nios II IDE software.
2. Click **Switch Workspace** on the File menu and browse to the **software** subdirectory of the V2 design install directory.
3. Click **OK** to create a new workspace.
4. When the Nios IDE has restarted, click the **Workbench** icon.
5. On the File menu, point to **New** and click **Nios II C/C++ Application**.
6. In the **New Project** dialog box (Figure 15 on page 32):
 - a. Select **Blank Project** as the **Project Template**.
 - b. Type `s2gxav_controller` in the **Name** field.
 - c. Turn on **Specify Location** and type in the path to the directory `<install directory>\V2_<revision>\s2gxav\software\s2gxav_controller`.

- d. Browse to the SOPC Builder System PTF File **V2.ptf** (in the Quartus II project).
- e. Click **Finish**.

Figure 15. Nios II IDE New Project Dialog Box



7. Verify that the application project **s2gxav_controller** appears in the **Nios II C/C++ Projects** list.
8. Browse the application source code files listed below **s2gxav_controller**.
9. To configure the System Library project, right-click on the **s2gxav_controller** project and click **System Library Properties**.
10. In the **System Library** dialog box:
 - a. Verify that **Program never exits**, **Clean exit (flush buffers)**, **Support C++** and **Reduced device drivers** are turned on.
 - b. Click **OK**.
11. To program the Nios II application executable to SRAM from the Nios II IDE, right-click on the **s2gxav_controller** project, point to **Run As** and click **Nios II Hardware**.



Altera recommends that you review the **.hpp** files included in the application project to quickly understand the control capability from the Nios II software environment.

Setting Up the Hardware and Configuring the FPGA

To set up the Stratix II GX audio video development board, perform the following steps:

1. Remove power from the development board by disconnecting the power cable.
2. Connect one end of the USB cable to the USB port on your PC.
3. Connect the other end to the 10-pin header labelled SYSTEM_JTAG on the development board.
4. Connect an SDI video source cable to the BNC input connector labelled SDI_IN0.
5. Connect a second SDI video source cable to the BNC input connector marked SDI_IN1.

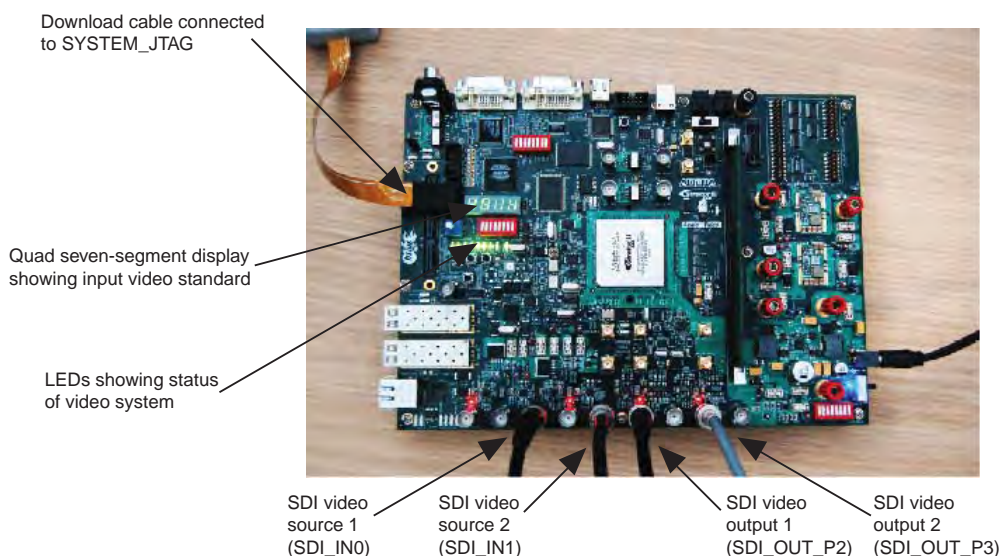


The design automatically detects the input video streams and produces output when the video sources are active.

6. Connect one end of a SDI cable to a SDI monitor (or a SDI to DVI convertor box if your monitor only supports DVI interfaces) capable of 1920×1080 @ 60Hz. Connect the other end to the SDI_OUT_P2 connector on the development board.
7. Connect a second SDI cable from the BNC output connector marked SDI_OUT_P3 to a SDI monitor.

Figure 16 shows the completed connections for the V2 reference design to the Stratix II GX audio video development board.

Figure 16. Stratix II GX Audio Video Development Board Connections



8. Re-apply power to the development board.



For details of installing the USB Blaster software driver on the host PC (located at `<quartus_install_dir>\drivers\usb-blaster`), refer to the [USB-Blaster Download Cable User Guide](#).

You can configure the Stratix II GX90 device by downloading the SRAM object file (**.sof**) image to the development board. To do so, perform the following steps:

1. In the Quartus II software, choose **Programmer** from the Tools menu.
2. In the **Mode** list of the **Programming** window, verify that **JTAG** is selected.
3. Click **Hardware Setup** to configure the programming hardware.
4. Verify that the **Hardware Setup** dialog box appears.
5. In the **Hardware** column, select **USB Blaster**.
6. Click **Close** to exit the **Hardware Setup** window.
7. In the **Programming** window, click **Add File**.
8. Select the file **s2gxav.sof** and click **Open**.
9. In the **Programmer** window, turn on **Program/Configure** on the same line as **s2gxav.sof**.
10. Click **Start**.

The programmer begins to download the configuration data to the FPGA. The **Progress** field displays the percentage of data that is downloaded. A message appears when the configuration is complete.



No video appears on the screen until the software executable has been downloaded to SRAM. Refer to [“Downloading the Nios II Software”](#).

Downloading the Nios II Software

To download the Nios II executable to SRAM, from a Nios II Command Shell perform the following steps:

1. Launch the Nios II Command Shell by clicking **Nios II Command Shell** in the **Nios II EDS <version>** menu under **Altera** in the Windows Start menu.
2. Change directory to `<install directory>\V2_<revision>\` containing **s2gxav_controller.elf**



If you have re-built your software program, change directory to the appropriate executable and linkable format file (**.elf**) file.

3. Enter the following command:

```
nios2-download -r -g s2gxav_controller.elf; nios2-terminal
```

Verify that video stream displays on the monitor as shown in [Figure 17](#).

Figure 17. V2 Reference Design Output Video



Troubleshooting

This section describes some problems that may occur and suggested solutions.

Error When Recompiling the Design

1. When re-compiling the design in the Quartus II software, compilation fails with an error:

Error: Node instance "sdi_megacore_top_inst" instantiates undefined entity "sdi_megacore_top"

Solution: the library path to the SDI MegaCore function in the Quartus II software is incorrect. Update the project library path to point to the **Altera/sdi/lib** directory (**Assignments->Settings->Libraries->Project Libraries->Add**). Remove any incorrect paths by selecting the path and selecting **Remove**.

2. When re-compiling the design in Quartus II compilation fails with the error:

Error: VHDL Use Clause error at *_GN.vhd(8): design library "altera" does not contain primary unit "alt_cusp81_package"

Solution: remove the project **db** directory and re-compile. This issue will be fixed in a future version of the tools.

Output Video Does Not Appear on Display

The output video does not appear on the display after downloading the Nios II .elf file.

Solution: press the reset button (RESET).

Delay Between a Change of Input and Output

Coefficient calculation is performed at run time so there may be a slight delay between a change of input format and the output video becoming sharp.

Solution: the software can be changed to precompute known coefficient sets.

Revision History

Table 9 shows the revision history for the *AN-581: High Definition (HD) Video Reference Design (V2)* application note.

Table 9. AN-581 Revision History

Version	Date	Errata Summary
1.0	November 2009	First release of this application note.



101 Innovation Drive
San Jose, CA 95134
www.altera.com
Technical Support
www.altera.com/support

Copyright © 2009 Altera Corporation. All rights reserved. Altera, The Programmable Solutions Company, the stylized Altera logo, specific device designations, and all other words and logos that are identified as trademarks and/or service marks are, unless noted otherwise, the trademarks and service marks of Altera Corporation in the U.S. and other countries. All other product or service names are the property of their respective holders. Altera products are protected under numerous U.S. and foreign patents and pending applications, maskwork rights, and copyrights. Altera warrants performance of its semiconductor products to current specifications in accordance with Altera's standard warranty, but reserves the right to make changes to any products and services at any time without notice. Altera assumes no responsibility or liability arising out of the application or use of any information, product, or service described herein except as expressly agreed to in writing by Altera Corporation. Altera customers are advised to obtain the latest version of device specifications before relying on any published information and before placing orders for products or services.

