LogiCORE IP Video Timing Controller v3.0

Product Guide

PG016 October 19, 2011

XILINX®

Table of Contents

LogiCORE IP Video Timing Controller v3.0 Product Guide

Introduction

The Xilinx Video Timing Controller LogiCORE™ IP is a general purpose video timing generator and detector. The input side of this core automatically detects horizontal and vertical synchronization pulses, polarity, blanking timing and active video pixels. While on the output, it generates the horizontal and vertical blanking and synchronization pulses used with a standard video system including support for programmable pulse polarity. The core is highly programmable through a comprehensive register set allowing control of various timing generation parameters. This programmability is coupled with a comprehensive set of interrupt bits which provides easy integration with the MicroBlaze™ Soft Processor for in-system control of the block in real-time. The Video Timing Controller is provided with either an AXI4-Lite compliant EDK pCore interface or a General Purpose Processor register interface.

Features

- Support for video frame sizes up to 8192 x 8192
- Direct regeneration of output timing signals with independent timing and polarity inversion
- Automatic detection and generation of horizontal and vertical video timing signals
- Support for multiple combinations of blanking or synchronization signals
- Automatic detection of input video control signal polarities
- Support for detection and generation of horizontal delay of vertical blank / sync
- Programmable output video signal polarities
- Generation of up to 16 additional independent output frame synchronization signals
- Selectable processor interface
  - AXI4-Lite
  - General Purpose Processor
- High number of interrupts and status registers for easy system control and integration

LogiCORE IP Facts Table	
Core Specifics	
Supported Device Family(1)	Virtex®-7, Kintex™-7, Virtex-6LX, Virtex-6LXT, Spartan®-6LX, Spartan-6LXT
Supported User Interfaces	AXI4-Lite, General Purpose Processor
Resources	See Table 1-1 through Table 1-4.
Provided with Core	
Design Files	Netlist, EDK pCore files, C Driver
Example Design	Not Provided
Test Bench	VHDL or Verilog(2)
Constraints File	Not Provided
Simulation Model	VHSIC Hardware Description Language (VHDL) or Verilog Structural model
Tested Design Tool	
Design Entry Tools	Integrated Software Environment (ISE) 13.3 Xilinx Platform Studio (XPS) 13.3
Simulation(3)	Mentor Graphics ModelSim, Xilinx® ISim 13.3
Synthesis Tools	Xilinx Synthesis Technology (XST) 13.3
Support	
Provided by Xilinx, Inc.	

1. For a complete listing of supported devices, see the release notes for this core.
2. HDL test bench and C Model available on the product page on Xilinx.com at http://www.xilinx.com/products/intellectual-property/EF-DI-VID-TIMING.htm
3. For the supported versions of the tools, see the ISE Design Suite 13: Release Notes Guide.

Chapter 1

Overview

All video systems require management of video timing signals, which are used to synchronize a variety of processes. The Video Timing Controller serves the function of both detecting and generating these timing signals.

Figure 1-1 shows a typical video frame including timing signals.

Note: All signals are shown with active high polarity.

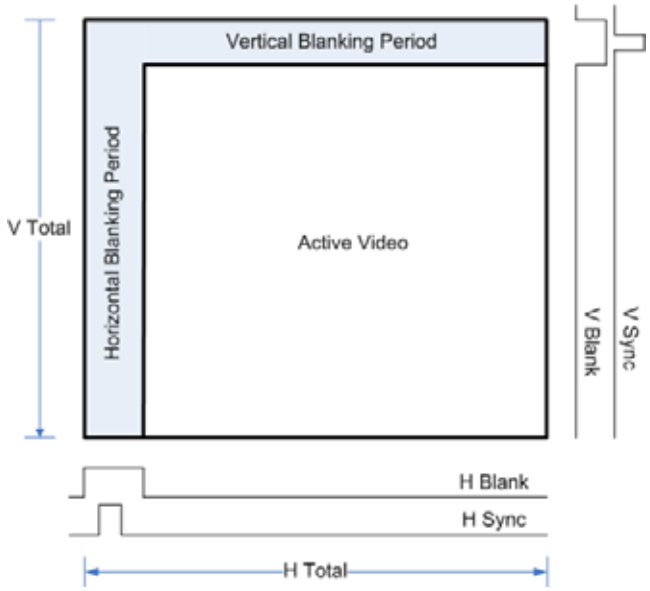

Figure 1-1: **Example Video Frame and Timing Signals**

A video frame can be completely described in terms of timing by only a few definitions. A video frame comprises active video and blanking periods. The vertical and horizontal synchronization signals describe the video frame timing, which includes active and blanking data. In addition, the frame synchronization signals can be used to synchronize video data from one processing block to another within a video system. There are additional signals that can also be used to control the video system, such as a signal to differentiate valid chroma samples.

Video systems may utilize different combinations of blank, synchronization or active signals with various polarities to synchronize processing and control video data. The Video Timing Controller makes this process easy by providing a highly programmable and flexible core that allows detection and generation of the various timing signals within a video system.

Standards Compliance

The Video Timing Controller core is compliant with the AXI4 standards as defined in the *AXI Reference Guide (UG761)*.

Feature Summary

The Video Timing Controller core supports the Advanced eXtensible Interface Lite (AXI4-Lite) and the General Purpose Processor interfaces. The AXI4-Lite interface can be easily incorporated into an EDK project. The General Purpose Processor interface exposes the core registers to the user. The user can wrap the exposed registers in an interface that is compliant with user system. These configurable interfaces allow the Video Timing Controller code to be integrated easily with AXI4 processor based systems, with non-AXI4-compliant processors systems with little logic (GPP), and with systems without a processor (GPP).

The Video Timing Controller core supports detecting video frame sizes up to 8192 clocks by 8192 lines (including horizontal and vertical blanking). This allows supporting HD (and well beyond) video frame sizes. In addition, the Video Timing Controller core allows automatically detecting the timing involved with horizontal / vertical blanks and syncs. The timing of the active_video and the active_chroma signals are also detected. This allows the user to easily determine the video frame size via the core register (AXI4-Lite or GPP) interface. The minimum set of signals used for detection is either vertical blank, horizontal blank and active video or vertical sync, horizontal sync and active video. The polarities of each input signal is also detected and reported via the register interface to allow easy use of each signal once the polarity is known.

The core also supports generating and regenerating (matching the detected input) video frame sizes up to 8192 clocks by 8192 lines (including blanking time). The generated output can be the same format or a different format as the detected input. This allows detecting one format and generating a different format. The generated output can also be synchronized to the detected input and has separate signal polarity settings as well. This allows regenerating the input with different signal polarities or with slight timing adjustments (such as delayed or shorted active video).

The Video Timing Controller core supports up to 16 frame sync output signals. These are toggled high for one clock cycle during each frame. These frame syncs allow triggering timing critical hardware processes at different times during a frame.

Applications

- Video Surveillance
- Industrial Imaging
- Video Conferencing
- Machine Vision
- Video Systems requiring timing detection or timing generation

Unsupported Features

The Video Timing Controller core does not automatically detect and regenerate timing signals with the same polarity at the output as on the input. Software that can read the

polarity of the input signals and set the polarity at the output is needed to configure the Video Timing Controller.

Licensing

The Xilinx Video Timing Controller LogiCORE IP system provides three licensing options. After installing the required Xilinx ISE software and IP Service Packs, choose a license option.

Simulation Only

The Simulation Only Evaluation license key is provided with the Xilinx CORE Generator tool. This key lets you assess the core functionality with either the provided example design or alongside your own design and demonstrates the various interfaces on the core in simulation. (Functional simulation is supported by a dynamically-generated HDL structural model.)

Full System Hardware Evaluation

The Full System Hardware Evaluation license is available at no cost and lets you fully integrate the core into an FPGA design, place-and-route the design, evaluate timing, and perform functional simulation of the Video Timing Controller core.

In addition, the license key lets you generate a bitstream from the placed and routed design, which can then be downloaded to a supported device and tested in hardware. The core can be tested in the target device for a limited time before timing out (ceasing to function), at which time it can be reactivated by reconfiguring the device.

Full

The Full license key is provided when you purchase the core and provides full access to all core functionality both in simulation and in hardware, including:

- Functional simulation support
- Back annotated gate-level simulation support
- Full implementation support including place and route and bitstream generation
- Full functionality in the programmed device with no time outs

Obtaining Your License Key

This section contains information about obtaining a simulation, full system hardware, and full license keys.

Simulation License

No action is required to obtain the Simulation Only Evaluation license key; it is provided by default with the Xilinx CORE Generator software.

Full System Hardware Evaluation License

To obtain a Full System Hardware Evaluation license:

1. Navigate to the product page for this core from:
   www.xilinx.com/products/ipcenter/EF-DI-VID-TIMING.htm
2. Click Evaluate.
3. Follow the instructions to install the required Xilinx ISE software and IP Service Packs.

Full License

To obtain a Full license key, you must purchase a license for the core. After doing so, click the "Access Core" link on the Xilinx.com IP core product page for further instructions.

Installing Your License File

The Simulation Only Evaluation license key is provided with the ISE CORE Generator system and does not require installation of an additional license file. For the Full System Hardware Evaluation license and the Full license, an email will be sent to you containing instructions for installing your license file. Additional details about IP license key installation can be found in the ISE Design Suite Installation, Licensing and Release Notes document.

Ordering Information

The Video Timing Controller v3.0 core is provided under the SignOnce IP Site License and can be generated using the Xilinx CORE Generator system v13.2 or higher. The CORE Generator system is shipped with Xilinx ISE Design Suite development software. To order Xilinx software, contact your local Xilinx sales representative. Information on additional Xilinx LogiCORE IP modules is available at the Xilinx Intellectual Property page.

Performance

The following sections detail the performance characteristics of the Video Timing Controller v3.0 core.

Maximum Frequencies

The following are typical clock frequencies for the target families. The maximum achievable clock frequency could vary and in most cases will be higher. The maximum achievable clock frequency and all resource counts may be affected by other tool options, additional logic in the FPGA device, using a different version of Xilinx tools, and other factors.

- Spartan-6: 150 MHz
- Virtex-6: 225 MHz
- Virtex-7: 225 MHz
- Kintex-7: 225 MHz

Latency

The Video Timing Controller core does not read or generate data, and therefore, does not have a specific data latency.

The Video Timing Controller core monitors and generates control signals. The output control signals can be configured to be the same as the input with no latency, or the output signals can be configured to incur a multi-clock or multi-line delay.

Throughput

The Video Timing Controller core does not read or generate data, and does not have a specific throughput.

Resource Utilization

Resource requirements for the Xilinx Timing Controller LogiCORE are estimated in tables 1-1-1-4 for Spartan®-6 DSP, Virtex®-6, Virtex-7, and Kintex-7 devices, respectively. Resource usage values were generated using the Xilinx CORE Generator tools v13.2. They are derived from post-synthesis reports, and may change during MAP and PAR. The resource usage values in the following tables are for the General Purpose Processor Interface. The EDK pCore Interface adds an estimated additional 370 LUTs and 300 flip-flops. The Xilinx Timing Controller LogiCORE does not utilize Block RAM.

Table 1-1: **Spartan-6 Device Resource Estimates**

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
256	256	No	Yes	No	Yes	No	No	160	122
256	256	No	Yes	No	Yes	Yes	No	236	177
256	256	No	Yes	Yes	No	Yes	No	167	130
256	256	Yes	No	No	Yes	Yes	No	313	271
256	256	Yes	No	Yes	No	Yes	No	312	242
256	256	Yes	Yes	Yes	Yes	Yes	Yes	748	611
512	512	No	Yes	No	Yes	No	No	175	133
512	512	No	Yes	No	Yes	Yes	No	274	196
512	512	No	Yes	Yes	No	Yes	No	180	141
512	512	Yes	No	No	Yes	Yes	No	216	296
512	512	Yes	No	Yes	No	Yes	No	238	263
512	512	Yes	Yes	Yes	Yes	Yes	Yes	778	668
1024	1024	No	Yes	No	Yes	No	No	123	144
1024	1024	No	Yes	No	Yes	Yes	No	164	211
1024	1024	No	Yes	Yes	No	Yes	No	131	152
1024	1024	Yes	No	No	Yes	Yes	No	368	321
1024	1024	Yes	No	Yes	No	Yes	No	392	284
1024	1024	Yes	Yes	Yes	Yes	Yes	Yes	852	725
2048	2048	No	Yes	No	Yes	No	No	140	155
2048	2048	No	Yes	No	Yes	Yes	No	189	228

Table 1-1: **Spartan-6 Device Resource Estimates *(Cont'd)***

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
2048	2048	No	Yes	Yes	No	Yes	No	155	163
2048	2048	Yes	No	No	Yes	Yes	No	487	346
2048	2048	Yes	No	Yes	No	Yes	No	360	305
2048	2048	Yes	Yes	Yes	Yes	Yes	Yes	1062	782
4096	4096	No	Yes	No	Yes	No	No	139	166
4096	4096	No	Yes	No	Yes	No	Yes	197	258
4096	4096	No	Yes	No	Yes	Yes	No	197	245
4096	4096	No	Yes	No	Yes	Yes	Yes	201	263
4096	4096	No	Yes	Yes	No	No	No	147	169
4096	4096	No	Yes	Yes	No	No	Yes	159	187
4096	4096	No	Yes	Yes	No	Yes	No	148	174
4096	4096	No	Yes	Yes	No	Yes	Yes	159	192
4096	4096	No	Yes	Yes	Yes	No	No	196	245
4096	4096	No	Yes	Yes	Yes	No	Yes	197	263
4096	4096	No	Yes	Yes	Yes	Yes	No	184	250
4096	4096	No	Yes	Yes	Yes	Yes	Yes	205	268
4096	4096	Yes	No	No	Yes	No	No	320	220
4096	4096	Yes	No	No	Yes	No	Yes	357	253
4096	4096	Yes	No	No	Yes	Yes	No	526	371
4096	4096	Yes	No	No	Yes	Yes	Yes	571	407
4096	4096	Yes	No	Yes	No	No	No	374	287
4096	4096	Yes	No	Yes	No	No	Yes	376	320
4096	4096	Yes	No	Yes	No	Yes	No	415	326
4096	4096	Yes	No	Yes	No	Yes	Yes	424	362
4096	4096	Yes	No	Yes	Yes	No	No	508	412
4096	4096	Yes	No	Yes	Yes	No	Yes	572	445
4096	4096	Yes	No	Yes	Yes	Yes	No	683	515
4096	4096	Yes	No	Yes	Yes	Yes	Yes	715	551
4096	4096	Yes	Yes	No	Yes	No	No	538	403
4096	4096	Yes	Yes	No	Yes	No	Yes	669	528
4096	4096	Yes	Yes	No	Yes	Yes	No	845	636
4096	4096	Yes	Yes	No	Yes	Yes	Yes	888	690
4096	4096	Yes	Yes	Yes	No	No	No	641	476
4096	4096	Yes	Yes	Yes	No	No	Yes	697	527

Table 1-1: **Spartan-6 Device Resource Estimates *(Cont'd)***

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
4096	4096	Yes	Yes	Yes	No	Yes	No	690	520
4096	4096	Yes	Yes	Yes	No	Yes	Yes	757	574
4096	4096	Yes	Yes	Yes	Yes	No	No	811	674
4096	4096	Yes	Yes	Yes	Yes	No	Yes	893	725
4096	4096	Yes	Yes	Yes	Yes	Yes	No	1003	785
4096	4096	Yes	Yes	Yes	Yes	Yes	Yes	1047	839

Table 1-2: **Virtex-6 Device Resource Estimates**

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
256	256	No	Yes	No	Yes	No	No	100	122
256	256	No	Yes	No	Yes	Yes	No	138	177
256	256	No	Yes	Yes	No	Yes	No	104	130
256	256	Yes	No	No	Yes	Yes	No	225	271
256	256	Yes	No	Yes	No	Yes	No	230	242
256	256	Yes	Yes	Yes	Yes	Yes	Yes	586	611
512	512	No	Yes	No	Yes	No	No	95	133
512	512	No	Yes	No	Yes	Yes	No	144	194
512	512	No	Yes	Yes	No	Yes	No	114	141
512	512	Yes	No	No	Yes	Yes	No	237	296
512	512	Yes	No	Yes	No	Yes	No	236	263
512	512	Yes	Yes	Yes	Yes	Yes	Yes	611	668
1024	1024	No	Yes	No	Yes	No	No	114	144
1024	1024	No	Yes	No	Yes	Yes	No	149	211
1024	1024	No	Yes	Yes	No	Yes	No	124	152
1024	1024	Yes	No	No	Yes	Yes	No	217	321
1024	1024	Yes	No	Yes	No	Yes	No	217	284
1024	1024	Yes	Yes	Yes	Yes	Yes	Yes	669	725
2048	2048	No	Yes	No	Yes	No	No	124	155
2048	2048	No	Yes	No	Yes	Yes	No	182	228
2048	2048	No	Yes	Yes	No	Yes	No	129	163
2048	2048	Yes	No	No	Yes	Yes	No	281	346
2048	2048	Yes	No	Yes	No	Yes	No	245	305
2048	2048	Yes	Yes	Yes	Yes	Yes	Yes	684	782

Table 1-2: **Virtex-6 Device Resource Estimates *(Cont'd)***

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
4096	4096	No	Yes	No	Yes	No	No	133	166
4096	4096	No	Yes	No	Yes	No	Yes	192	258
4096	4096	No	Yes	No	Yes	Yes	No	188	245
4096	4096	No	Yes	No	Yes	Yes	Yes	183	263
4096	4096	No	Yes	Yes	No	No	No	134	169
4096	4096	No	Yes	Yes	No	No	Yes	140	187
4096	4096	No	Yes	Yes	No	Yes	No	132	174
4096	4096	No	Yes	Yes	No	Yes	Yes	152	192
4096	4096	No	Yes	Yes	Yes	No	No	176	245
4096	4096	No	Yes	Yes	Yes	No	Yes	174	263
4096	4096	No	Yes	Yes	Yes	Yes	No	190	250
4096	4096	No	Yes	Yes	Yes	Yes	Yes	193	268
4096	4096	Yes	No	No	Yes	No	No	164	220
4096	4096	Yes	No	No	Yes	No	Yes	182	253
4096	4096	Yes	No	No	Yes	Yes	No	264	371
4096	4096	Yes	No	No	Yes	Yes	Yes	296	407
4096	4096	Yes	No	Yes	No	No	No	210	287
4096	4096	Yes	No	Yes	No	No	Yes	226	320
4096	4096	Yes	No	Yes	No	Yes	No	271	326
4096	4096	Yes	No	Yes	No	Yes	Yes	288	362
4096	4096	Yes	No	Yes	Yes	No	No	317	412
4096	4096	Yes	No	Yes	Yes	No	Yes	317	445
4096	4096	Yes	No	Yes	Yes	Yes	No	437	515
4096	4096	Yes	No	Yes	Yes	Yes	Yes	463	551
4096	4096	Yes	Yes	No	Yes	No	No	369	403
4096	4096	Yes	Yes	No	Yes	No	Yes	486	528
4096	4096	Yes	Yes	No	Yes	Yes	No	579	636
4096	4096	Yes	Yes	No	Yes	Yes	Yes	604	690
4096	4096	Yes	Yes	Yes	No	No	No	441	476
4096	4096	Yes	Yes	Yes	No	No	Yes	465	527
4096	4096	Yes	Yes	Yes	No	Yes	No	458	520
4096	4096	Yes	Yes	Yes	No	Yes	Yes	528	574
4096	4096	Yes	Yes	Yes	Yes	No	No	612	674
4096	4096	Yes	Yes	Yes	Yes	No	Yes	649	725

Table 1-2: **Virtex-6 Device Resource Estimates *(Cont'd)***

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
4096	4096	Yes	Yes	Yes	Yes	Yes	No	706	785
4096	4096	Yes	Yes	Yes	Yes	Yes	Yes	731	839

Table 1-3: **Virtex-7 Device Resource Estimates**

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
256	256	No	Yes	No	Yes	No	No	100	122
256	256	No	Yes	No	Yes	Yes	No	124	177
256	256	No	Yes	Yes	No	Yes	No	96	130
256	256	Yes	No	No	Yes	Yes	No	207	271
256	256	Yes	No	Yes	No	Yes	No	221	242
256	256	Yes	Yes	Yes	Yes	Yes	Yes	535	611
512	512	No	Yes	No	Yes	No	No	112	133
512	512	No	Yes	No	Yes	Yes	No	135	194
512	512	No	Yes	Yes	No	Yes	No	115	141
512	512	Yes	No	No	Yes	Yes	No	224	296
512	512	Yes	No	Yes	No	Yes	No	233	263
512	512	Yes	Yes	Yes	Yes	Yes	Yes	580	668
1024	1024	No	Yes	No	Yes	No	No	113	144
1024	1024	No	Yes	No	Yes	Yes	No	148	211
1024	1024	No	Yes	Yes	No	Yes	No	122	152
1024	1024	Yes	No	No	Yes	Yes	No	232	321
1024	1024	Yes	No	Yes	No	Yes	No	243	284
1024	1024	Yes	Yes	Yes	Yes	Yes	Yes	591	725
2048	2048	No	Yes	No	Yes	No	No	119	155
2048	2048	No	Yes	No	Yes	Yes	No	174	228
2048	2048	No	Yes	Yes	No	Yes	No	128	163
2048	2048	Yes	No	No	Yes	Yes	No	256	346
2048	2048	Yes	No	Yes	No	Yes	No	257	305
2048	2048	Yes	Yes	Yes	Yes	Yes	Yes	682	782
4096	4096	No	Yes	No	Yes	No	No	124	166
4096	4096	No	Yes	No	Yes	No	Yes	192	258
4096	4096	No	Yes	No	Yes	Yes	No	175	245
4096	4096	No	Yes	No	Yes	Yes	Yes	181	263

Table 1-3: **Virtex-7 Device Resource Estimates *(Cont'd)***

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
4096	4096	No	Yes	Yes	No	No	No	137	169
4096	4096	No	Yes	Yes	No	No	Yes	136	187
4096	4096	No	Yes	Yes	No	Yes	No	127	174
4096	4096	No	Yes	Yes	No	Yes	Yes	151	192
4096	4096	No	Yes	Yes	Yes	No	No	170	245
4096	4096	No	Yes	Yes	Yes	No	Yes	182	263
4096	4096	No	Yes	Yes	Yes	Yes	No	175	250
4096	4096	No	Yes	Yes	Yes	Yes	Yes	181	268
4096	4096	Yes	No	No	Yes	No	No	165	220
4096	4096	Yes	No	No	Yes	No	Yes	178	253
4096	4096	Yes	No	No	Yes	Yes	No	273	371
4096	4096	Yes	No	No	Yes	Yes	Yes	269	407
4096	4096	Yes	No	Yes	No	No	No	226	287
4096	4096	Yes	No	Yes	No	No	Yes	230	320
4096	4096	Yes	No	Yes	No	Yes	No	284	326
4096	4096	Yes	No	Yes	No	Yes	Yes	290	362
4096	4096	Yes	No	Yes	Yes	No	No	305	412
4096	4096	Yes	No	Yes	Yes	No	Yes	306	445
4096	4096	Yes	No	Yes	Yes	Yes	No	419	515
4096	4096	Yes	No	Yes	Yes	Yes	Yes	425	551
4096	4096	Yes	Yes	No	Yes	No	No	382	403
4096	4096	Yes	Yes	No	Yes	No	Yes	470	528
4096	4096	Yes	Yes	No	Yes	Yes	No	566	636
4096	4096	Yes	Yes	No	Yes	Yes	Yes	580	690
4096	4096	Yes	Yes	Yes	No	No	No	426	476
4096	4096	Yes	Yes	Yes	No	No	Yes	431	527
4096	4096	Yes	Yes	Yes	No	Yes	No	501	520
4096	4096	Yes	Yes	Yes	No	Yes	Yes	516	574
4096	4096	Yes	Yes	Yes	Yes	No	No	590	674
4096	4096	Yes	Yes	Yes	Yes	No	Yes	600	725
4096	4096	Yes	Yes	Yes	Yes	Yes	No	684	785
4096	4096	Yes	Yes	Yes	Yes	Yes	Yes	715	839
8192	8192	No	Yes	No	Yes	No	No	152	177
8192	8192	No	Yes	No	Yes	Yes	No	194	262

Table 1-3: **Virtex-7 Device Resource Estimates *(Cont'd)***

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
8192	8192	No	Yes	Yes	No	Yes	No	147	185
8192	8192	Yes	No	No	Yes	Yes	No	422	396
8192	8192	Yes	No	Yes	No	Yes	No	289	347
8192	8192	Yes	Yes	Yes	Yes	Yes	Yes	745	896

Table 1-4: **Kintex-7 Device Resource Estimates**

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
256	256	No	Yes	No	Yes	No	No	107	122
256	256	No	Yes	No	Yes	Yes	No	145	177
256	256	No	Yes	Yes	No	Yes	No	100	130
256	256	Yes	No	No	Yes	Yes	No	235	271
256	256	Yes	No	Yes	No	Yes	No	225	242
256	256	Yes	Yes	Yes	Yes	Yes	Yes	557	611
512	512	No	Yes	No	Yes	No	No	113	133
512	512	No	Yes	No	Yes	Yes	No	154	194
512	512	No	Yes	Yes	No	Yes	No	115	141
512	512	Yes	No	No	Yes	Yes	No	224	296
512	512	Yes	No	Yes	No	Yes	No	223	263
512	512	Yes	Yes	Yes	Yes	Yes	Yes	583	668
1024	1024	No	Yes	No	Yes	No	No	125	144
1024	1024	No	Yes	No	Yes	Yes	No	170	211
1024	1024	No	Yes	Yes	No	Yes	No	131	152
1024	1024	Yes	No	No	Yes	Yes	No	280	321
1024	1024	Yes	No	Yes	No	Yes	No	239	284
1024	1024	Yes	Yes	Yes	Yes	Yes	Yes	635	725
2048	2048	No	Yes	No	Yes	No	No	129	155
2048	2048	No	Yes	No	Yes	Yes	No	180	228
2048	2048	No	Yes	Yes	No	Yes	No	136	163
2048	2048	Yes	No	No	Yes	Yes	No	274	346
2048	2048	Yes	No	Yes	No	Yes	No	259	305
2048	2048	Yes	Yes	Yes	Yes	Yes	Yes	724	782
4096	4096	No	Yes	No	Yes	No	No	146	166
4096	4096	No	Yes	No	Yes	No	Yes	213	258

Table 1-4: **Kintex-7 Device Resource Estimates *(Cont'd)***

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
4096	4096	No	Yes	No	Yes	Yes	No	206	245
4096	4096	No	Yes	No	Yes	Yes	Yes	208	263
4096	4096	No	Yes	Yes	No	No	No	140	169
4096	4096	No	Yes	Yes	No	No	Yes	145	187
4096	4096	No	Yes	Yes	No	Yes	No	148	174
4096	4096	No	Yes	Yes	No	Yes	Yes	157	192
4096	4096	No	Yes	Yes	Yes	No	No	198	245
4096	4096	No	Yes	Yes	Yes	No	Yes	215	263
4096	4096	No	Yes	Yes	Yes	Yes	No	205	250
4096	4096	No	Yes	Yes	Yes	Yes	Yes	217	268
4096	4096	Yes	No	No	Yes	No	No	188	220
4096	4096	Yes	No	No	Yes	No	Yes	209	253
4096	4096	Yes	No	No	Yes	Yes	No	306	371
4096	4096	Yes	No	No	Yes	Yes	Yes	307	407
4096	4096	Yes	No	Yes	No	No	No	242	287
4096	4096	Yes	No	Yes	No	No	Yes	235	320
4096	4096	Yes	No	Yes	No	Yes	No	265	326
4096	4096	Yes	No	Yes	No	Yes	Yes	294	362
4096	4096	Yes	No	Yes	Yes	No	No	339	412
4096	4096	Yes	No	Yes	Yes	No	Yes	339	445
4096	4096	Yes	No	Yes	Yes	Yes	No	420	515
4096	4096	Yes	No	Yes	Yes	Yes	Yes	437	551
4096	4096	Yes	Yes	No	Yes	No	No	380	403
4096	4096	Yes	Yes	No	Yes	No	Yes	484	528
4096	4096	Yes	Yes	No	Yes	Yes	No	577	636
4096	4096	Yes	Yes	No	Yes	Yes	Yes	609	690
4096	4096	Yes	Yes	Yes	No	No	No	432	476
4096	4096	Yes	Yes	Yes	No	No	Yes	461	527
4096	4096	Yes	Yes	Yes	No	Yes	No	484	520
4096	4096	Yes	Yes	Yes	No	Yes	Yes	528	574
4096	4096	Yes	Yes	Yes	Yes	No	No	582	674
4096	4096	Yes	Yes	Yes	Yes	No	Yes	610	725
4096	4096	Yes	Yes	Yes	Yes	Yes	No	698	785
4096	4096	Yes	Yes	Yes	Yes	Yes	Yes	741	839

Table 1-4: **Kintex-7 Device Resource Estimates *(Cont'd)***

Maximum Clocks	Maximum Lines	Detection Enable	Generation Enable	H/V Blanks	H/V Syncs	Active Video	Active Chroma	LUTs	FFs
8192	8192	No	Yes	No	Yes	No	No	157	177
8192	8192	No	Yes	No	Yes	Yes	No	211	262
8192	8192	No	Yes	Yes	No	Yes	No	165	185
8192	8192	Yes	No	No	Yes	Yes	No	323	396
8192	8192	Yes	No	Yes	No	Yes	No	295	347
8192	8192	Yes	Yes	Yes	Yes	Yes	Yes	680	896

Note: The Video Timing Controller does not utilize block RAMs or Xilinx XtremeDSP™ slices.

XILINX®

Chapter 2

Core Interfaces and Register Space

This chapter provides detailed descriptions for each interface. In addition, detailed information about configuration and control registers is included.

Port Descriptions

Core Interfaces

Processor Interface

Video systems commonly use an integrated processor system to dynamically control the parameters within the system. This is especially important when several independent image processing cores are integrated into a single FPGA. The Video Timing Controller core can be configured with one of two interfaces: an EDK pCore Interface (AXI4-Lite) or a General Purpose Processor Interface (GPP).

Common I/O Signals

The EDK pCore interface and the General Purpose Processor interface share a number of the same Input / Output (I / O) signals. The signals that both interfaces share are specified in Table 2-1.

Table 2-1: **Common Port Descriptions**

Name	Direction	Description
sclr	Input	SYNCHRONOUS CLEAR / RESET System synchronous reset (active high). Asserting `sclr` synchronously with `video_clk_in` resets the video timing controller internal state machines. `sclr` has priority over `ce`.
ce	Input	CLOCK ENABLE Used to halt processing and hold current values.
Detector Interface		
video_clk_in	Input	INPUT CLOCK Core clock (active high edge). Always present.

Table 2-1: **Common Port Descriptions *(Cont'd)***

Name	Direction	Description
hsync_in	Input	INPUT HORIZONTAL SYNCHRONIZATION Used to set the det_hsync_start and the det_hbp_start registers. Polarity is auto-detected . Optional. Either horizontal blank or horizontal synchronization signal inputs must be present. Both do not have to be present.
hblank_in	Input	INPUT HORIZONTAL BLANK Used to set the det_hfp_start and the det_hactive_start registers. Polarity is auto-detected. Optional. Either horizontal blank or horizontal synchronization signal inputs must be present. Both do not have to be present.
vsync_in	Input	INPUT VERTICAL SYNCHRONIZATION Used to set the det_v0sync_start and the det_v0bp_start registers. Polarity is auto-detected . Optional. One of the following inputs must be present: active video, vertical blank or vertical synchronization.
vblank_in	Input	INPUT VERTICAL BLANK Used to set the det_v0fp_start and the det_v0active_start registers. Polarity is auto-detected . Optional. One of the following inputs must be present: active video, vertical blank or vertical synchronization.
active_video_in	Input	INPUT ACTIVE VIDEO Used to set the det_v0fp_start and the det_v0active_start registers. Polarity is auto-detected . Optional. One of the following inputs must be present: active video, vertical blank or vertical synchronization.
active_chroma_in	Input	INPUT ACTIVE CHROMA Used to set the det_v0achroma_start register and bit 4 in the detection status register. Polarity is auto-detected . Optional.
Generator Interface		
video_clk_out	Output	OUTPUT CLOCK Same as video_clk_in.
hsync_out	Output	OUTPUT HORIZONTAL SYNCHRONIZATION Generated horizontal synchronization signal. Polarity configured by the control register. Asserted active during the cycle set by the gen_hsync_start register and deasserted during the cycle set by the gen_hbp_start register.
hblank_out	Output	OUTPUT HORIZONTAL BLANK Generated horizontal blank signal. Polarity configured by the control register. Asserted active during the cycle set by the gen_hfp_start and deasserted during the cycle set by the gen_hactive_start register.

Table 2-1: **Common Port Descriptions *(Cont'd)***

Name	Direction	Description
vsync_out	Output	OUTPUT VERTICAL SYNCHRONIZATION Generated vertical synchronization signal. Polarity configured by the control register. Asserted active during the line set by the `gen_v0sync_start` register and deasserted during the line set by the `gen_v0bp_start` register.
vblank_out	Output	OUTPUT VERTICAL BLANK Generated vertical blank signal. Polarity configured by the control register. Asserted active during the line set by the `gen_v0fp_start` register and deasserted during the line set by the `gen_v0active_start` register.
active_video_out	Output	OUTPUT ACTIVE VIDEO Generated active video signal. Polarity configured by the control register. Active for non blanking lines. Asserted active during the cycle set by the `gen_hactive_start` register and deasserted during the cycle set by the `gen_hbp_start` register.
active_chroma_out	Output	OUTPUT ACTIVE CHROMA Generated active chroma signal. Denotes which lines contain valid chroma samples (used for YUV 4:2:0). Polarity configured by the control register. Active for non blanking lines after the line set by the `gen_v0achroma_start` register (inclusive). For valid chroma lines, asserted active during every cycle the `active_video_out` signal is set per line.
Frame Synchronization Interface		
fsync *[Frame Syncs - 1:0]*	Output	FRAME SYNCHRONIZATION OUTPUT Each Frame Synchronization bit toggles for only one clock cycle during each frame. The number of bits is configured with the Frame Syncs GUI parameter. Each bit is independently configured for horizontal and vertical clock cycle position with the fsync_hstart and fsync_vstart registers).

Notes:

1. All registers are little-endian.

EDK pCore Interface

The pCore interface creates a core that can be easily added to an EDK Project as a hardware peripheral. This section describes the I/O signals associated with the Video Timing Controller pCore.

Table 2-2: **AXI4-Lite Signals**

Pin Name	Dir	Width	Description
AXI Global System Signals(1)			
S_AXI_ARESETN	I	1	AXI Reset, active low
IP2INTC_Irpt	O	1	Interrupt request output
AXI Write Address Channel Signals(1)			
S_AXI_AWADDR	I	[(C_S_AXI_ADDR_WIDTH-1): 0]	AXI4-Lite Write Address Bus. The write address bus gives the address of the write transaction.

Table 2-2: **AXI4-Lite Signals *(Cont'd)***

S_AXI_AWVALID	I	1	AXI4-Lite Write Address Channel Write Address Valid. This signal indicates that valid write address is available. 1 = Write address is valid. 0 = Write address is not valid.
S_AXI_AWREADY	O	1	AXI4-Lite Write Address Channel Write Address Ready. Indicates core is ready to accept the write address. 1 = Ready to accept address. 0 = Not ready to accept address.
AXI Write Data Channel Signals(1)			
S_AXI_WDATA	I	[(C_S_AXI_DATA_WIDTH-1): 0]	AXI4-Lite Write Data Bus.
S_AXI_WSTRB	I	[C_S_AXI_DATA_WIDTH / 8-1:0]	AXI4-Lite Write Strobes. This signal indicates which byte lanes to update in memory.
S_AXI_WVALID	I	1	AXI4-Lite Write Data Channel Write Data Valid. This signal indicates that valid write data and strobes are available. 1 = Write data / strobes are valid. 0 = Write data / strobes are not valid.
S_AXI_WREADY	O	1	AXI4-Lite Write Data Channel Write Data Ready. Indicates core is ready to accept the write data. 1 = Ready to accept data. 0 = Not ready to accept data.

Table 2-2: **AXI4-Lite Signals *(Cont'd)***

Pin Name	Dir	Width	Description
AXI Write Response Channel Signals(1)			
S_AXI_BRESP(2)	O	[1:0]	AXI4-Lite Write Response Channel. Indicates results of the write transfer. 00b = OKAY - Normal access has been successful. 01b = EXOKAY - Not supported. 10b = SLVERR - Error. 11b = DECERR - Not supported.
S_AXI_BVALID	O	1	AXI4-Lite Write Response Channel Response Valid. Indicates response is valid. 1 = Response is valid. 0 = Response is not valid.
S_AXI_BREADY	I	1	AXI4-Lite Write Response Channel Ready. Indicates Master is ready to receive response. 1 = Ready to receive response. 0 = Not ready to receive response.
AXI Read Address Channel Signals(1)			
S_AXI_ARADDR	I	[(C_S_AXI_ADDR_WIDTH-1): 0]	AXI4-Lite Read Address Bus. The read address bus gives the address of a read transaction.
S_AXI_ARVALID	I	1	AXI4-Lite Read Address Channel Read Address Valid. 1 = Read address is valid. 0 = Read address is not valid.
S_AXI_ARREADY	O	1	AXI4-Lite Read Address Channel Read Address Ready. Indicates core is ready to accept the read address. 1 = Ready to accept address. 0 = Not ready to accept address.
AXI Read Data Channel Signals(1)			
S_AXI_RDATA	O	[(C_S_AXI_DATA_WIDTH-1): 0]	AXI4-Lite Read Data Bus.
S_AXI_RRESP(2)	O	[1:0]	AXI4-Lite Read Response Channel Response. Indicates results of the read transfer. 00b = OKAY - Normal access has been successful. 01b = EXOKAY - Not supported. 10b = SLVERR - Error. 11b = DECERR - Not supported.

Table 2-2: **AXI4-Lite Signals *(Cont'd)***

Pin Name	Dir	Width	Description
S_AXI_RVALID	O	1	AXI4-Lite Read Data Channel Read Data Valid. This signal indicates that the required read data is available and the read transfer can complete. 1 = Read data is valid. 0 = Read data is not valid.
S_AXI_RREADY	I	1	AXI4-Lite Read Data Channel Read Data Ready. Indicates master is ready to accept the read data. 1 = Ready to accept data. 0 = Not ready to accept data.

1. The function and timing of these signals are defined in the AMBA AXI Protocol Version: 2.0 Specification.

2.For signals S_AXI_RRESP[1:0] and S_AXI_BRESP[1:0], the core does not generate the Decode Error ('11') response.
Other responses such as '00' (OKAY) and '10' (SLVERR) are generated by the core based upon certain conditions.

General Purpose Processor Interface

The other interface option is the General Purpose Processor (GPP) interface. The GPP Interface consists of the Common I/O signals listed in Table 2-1 and the Dynamic Configuration Interface signals detailed in Table 2-3. The signals in Table 2-3 correspond to the registers in Table 2-5. The directly exposed Dynamic Configuration Interface signals allow you to wrap these signals with a user-defined bus interface targeting any arbitrary processor. It is recommended to disable the control[2] (Register Update Enable) signal of the control bus before updating the other Dynamic Configuration Interface signals. After the Dynamic Configuration Interface signals are ready to be updated in the core, the control[2] signal should be enabled. Values are written into the core on the falling edge of the frame_sync input.

Table 2-3: **General Purpose Processor Port Descriptions**

Name	Direction	Description
General Purpose Processor Interface		
control[31:0]	Input	CONTROL REGISTER Bit 0: Generation Enable. When low, the generation hardware will not generate video timing output signals. When high, enable hardware to generate output. Set this bit high only after the software has configured the generator registers. Bit 1: Detection Enable. When low, no detection will be performed. All 'locked' status bits will be driven low. When high, perform timing signal detection for enabled signals. Bit 2: Generator / Detector Synchronization Enable. When low, the generator will not be synchronized to the detector. When high, the generator will be synchronized to the detector. Bit 3: Lock Interrupt Polarity. When low, the lock interrupts will trigger an interrupt on the falling edge of the internal lock signals, signifying that the detected input has changed timing. When high, the lock interrupts will trigger an interrupt on the rising edge of the internal lock signals, signifying that a lock has been achieved on the detected input. Bit 4: Generated Active Chroma Line Skip. This is the number of lines to skip between each successive active chroma line. Low denotes not to skip lines. Used for YUV 4:2:2 or 4:4:4. High denotes to skip every other line. Used for 4:2:0. Bit 5: Generated Active Chroma Pixel Skip. This is the number of pixels to skip between each successive active chroma pixel. Low denotes not to skip pixels. Can be combined with the Active Chroma Line Skip.

Table 2-3: **General Purpose Processor Port Descriptions *(Cont'd)***

Name	Direction	Description
control[31:0] *(continued from previous page)*	Input	Bits 7-6: RESERVED Source Selects. Bits 8-18 select which register controls the generator outputs. Low denotes the detection register will be used. High denotes that the generation register will be used. These bits allow the video timing controller detector to control the generator outputs (when low) or allow the host processor to override each value independently (when high). Bit 8: Horizontal Total Register Source Select Bit 9: Horizontal Front Porch Start Register Source Select Bit 10: Horizontal Synchronization Start Register Source Select Bit 11: Horizontal Back Porch Start Register Source Select Bit 12: Horizontal Active Video Start Register Source Select Bit 13: Vertical Total Register Source Select Bit 14: Vertical Front Porch Start Register Source Select Bit 15: Vertical Synchronization Start Register Source Select Bit 16: Vertical Back Porch Start Register Source Select Bit 17: Vertical Active Video Start Register Source Select Bit 18: Start of Active Chroma Register Source Select Bit 19: RESERVED Generated Output Signal Polarities. Bits 20-26 configure the polarity of each output. High denotes active high polarity. Low denotes active low polarity. Bit 20: Horizontal Synchronization Output Polarity Bit 21: Horizontal Blank Output Polarity Bit 22: Vertical Synchronization Output Polarity Bit 23: Vertical Blank Output Polarity Bit 24: RESERVED Bit 25: Active Video Output Polarity Bit 26: Active Chroma Output Polarity Bits 27-31: RESERVED

Table 2-3: **General Purpose Processor Port Descriptions *(Cont'd)***

Name	Direction	Description
det_status[31:0]	Output	DETECTION STATUS REGISTER Bits 3-0: RESERVED Bit 4: Detected Active Chroma Line Skip. This is the number of lines skipped between each successive active chroma line. Low denotes no lines are skipped. Used for detecting YUV 4:2:2 or 4:4:4. High denotes every other line is skipped. Used for detecting YUV 4:2:0. Bit 5: Detected Active Chroma Pixel Skip. This is the number of pixels skipped between each successive active chroma pixel. Low denotes no pixels are skipped. Bits 19-6: RESERVED Detected Input Signal Polarities. Bits 20-26 denote the polarity of each input. High denotes active high polarity. Low denotes active low polarity. Bit 20: Horizontal Synchronization Input Polarity Bit 21: Horizontal Blank input Polarity Bit 22: Vertical Synchronization Input Polarity Bit 23: Vertical Blank Input Polarity Bit 24: RESERVED Bit 25: Active Video Input Polarity Bit 26: Active Chroma Input Polarity Bits 31-27: RESERVED
gen_htotal[X_{b2}-1:0]	Input	GENERATED HORIZONTAL TOTAL Total number of horizontal clock cycles (minus 1) per line including blanking and active cycles. This is the last pixel count on each line. Each line starts at count 0. Maximum allowable Horizontal Total is configured by the *MAX CLOCKS PER LINE* parameter.
gen_hfp_start[X_{b2}-1:0]	Input	GENERATED HORIZONTAL FRONT PORCH START Cycle count during which the Horizontal Front Porch starts. Also denotes the end of Active Video.
gen_hsync_start[X_{b2}-1:0]	Input	GENERATED HORIZONTAL SYNCHRONIZATION START Cycle count during which the Horizontal Synchronization starts. Also denotes the end of Horizontal Front Porch.
gen_hbp_start[X_{b2}-1:0]	Input	GENERATED HORIZONTAL BACK PORCH START Cycle count during which the Horizontal Back Porch starts. Also denotes the end of Horizontal Synchronization.

Table 2-3: **General Purpose Processor Port Descriptions *(Cont'd)***

Name	Direction	Description
gen_hactive_start[X_{b2}-1:0]	Input	GENERATED HORIZONTAL ACTIVE VIDEO START Cycle count during which the Horizontal Active Video starts. Also denotes the end of Horizontal Back Porch.
gen_v0total[Y_{b2}-1:0]	Input	GENERATED VERTICAL TOTAL LINES Total number of Vertical lines per frame (minus 1) including blanking and active cycles. This is the last line count in each frame. Each frame starts at line count 0. Maximum allowable Vertical Total is configured by the *MAX LINES PER FRAME* parameter.
gen_v0fp_start[Y_{b2}-1:0]	Input	GENERATED VERTICAL FRONT PORCH START Line count during which the Vertical Front Porch starts. Also denotes the end of Active Video.
gen_v0sync_start[Y_{b2}-1:0]	Input	GENERATED VERTICAL SYNCHRONIZATION START Line count during which the Vertical Synchronization starts. Also denotes the end of Vertical Front Porch.
gen_v0bp_start[Y_{b2}-1:0]	Input	GENERATED VERTICAL BACK PORCH START Line count during which the Vertical Back Porch starts. Also denotes the end of Vertical Synchronization.
gen_v0active_start[Y_{b2}-1:0]	Input	GENERATED VERTICAL ACTIVE VIDEO START Line count during which the Active Video starts. Also denotes the end of Vertical Back Porch.
gen_v0achroma_start [Y_{b2}-1:0]	Input	GENERATED ACTIVE CHROMA START Line count during which the Active Chroma starts. See bit 4 of the control register to configure for YUV 4:2:0 mode.
det_htotal[X_{b2}-1:0]	Output	DETECTED HORIZONTAL TOTAL Detected Total number of horizontal clock cycles per line including blanking and active cycles (minus 1). Maximum allowable horizontal Total is configured by the *MAX CLOCKS PER LINE* parameter.
det_hfp_start[X_{b2}-1:0]	Output	DETECTED HORIZONTAL FRONT PORCH START Detected cycle count during which the Input Horizontal Front Porch starts. Also denotes the end of Input Active Video.
det_hsync_start[X_{b2}-1:0]	Output	DETECTED HORIZONTAL SYNCHRONIZATION START Detected Cycle count during which the Input Horizontal Synchronization starts. Also denotes the end of Input Horizontal Front Porch.
det_hbp_start[X_{b2}-1:0]	Output	DETECTED HORIZONTAL BACK PORCH START Detected Cycle count during which the Input Horizontal Back Porch starts. Also denotes the end of Input Horizontal Synchronization.

Table 2-3: **General Purpose Processor Port Descriptions *(Cont'd)***

Name	Direction	Description
det_hactive_start[X_{b2}-1:0]	Output	DETECTED HORIZONTAL ACTIVE VIDEO START Cycle count during which the Input Horizontal Active Video starts. Also denotes the end of Input Horizontal Back Porch.
det_v0total[Y_{b2}-1:0]	Output	DETECTED VERTICAL TOTAL Total number of Input Vertical lines per frame including blanking and active cycles (minus 1). Maximum allowable Vertical Total is configured by the *MAX LINES PER FRAME* parameter.
det_v0fp_start[Y_{b2}-1:0]	Output	DETECTED VERTICAL FRONT PORCH START Line count during which the Input Vertical Front Porch starts. Also denotes the end of Input Active Video.
det_v0sync_start[Y_{b2}-1:0]	Output	DETECTED VERTICAL SYNCHRONIZATION START Line count during which the Input Vertical Synchronization starts. Also denotes the end of Input Vertical Front Porch.
det_v0bp_start[Y_{b2}-1:0]	Output	DETECTED VERTICAL BACK PORCH START Line count during which the Input Vertical Back Porch starts. Also denotes the end of Input Vertical Synchronization.
det_v0active_start[Y_{b2}-1:0]	Output	DETECTED VERTICAL ACTIVE VIDEO START Line count during which the Input Vertical Active Video starts. Also denotes the end of Input Vertical Back Porch.
det_v0achroma_start [Y_{b2}-1:0]	Output	DETECTED ACTIVE CHROMA START Line count during which the Input Active Chroma starts.
fsync_hstart [Frame Syncs*X_{b2}-1:0]	Input	FRAME SYNCHRONIZATION HORIZONTAL START REGISTER Bits Y_{b2}-1 to 0: Horizontal Cycle during which Frame Synchronization 0 is active. Bits $2X_{b2}$-1 to X_{b2}: Horizontal Cycle during which Frame Synchronization 1 is active.
fsync_vstart [Frame Syncs*Y_{b2}-1:0]	Input	FRAME SYNCHRONIZATION VERTICAL START REGISTER Bits Y_{b2}-1 to 0: Vertical line during which Frame Synchronization 0 is active. Bits $2Y_{b2}$-1 to Y_{b2}: Vertical line during which Frame Synchronization 1 is active. ***Note:*** Frame Syncs are not active during the complete line, only in the cycle during which both the `fsync_vstart` and `fsync_hstart` are valid each frame.
gen_v0blank_hstart	Input	GENERATED VERTICAL BLANK HORIZONTAL OFFSET START Denotes the horizontal cycle during which the vblank signal is asserted.
gen_v0blank_hend	Input	GENERATED VERTICAL BLANK HORIZONTAL OFFSET END Denotes the horizontal cycle during which the vblank signal deasserts.
gen_v0sync_hstart	Input	GENERATED VERTICAL SYNC HORIZONTAL OFFSET START Denotes the horizontal cycle during which the vsync signal is asserted.
gen_v0sync_hend	Input	GENERATED VERTICAL SYNC HORIZONTAL OFFSET END Denotes the horizontal cycle during which the vsync signal deasserts.

Table 2-3: **General Purpose Processor Port Descriptions *(Cont'd)***

Name	Direction	Description
det_v0blank_hstart	Output	DETECTED VERTICAL BLANK HORIZONTAL OFFSET START Denotes the horizontal cycle during which the vblank signal is asserted.
det_v0blank_hend	Output	DETECTED VERTICAL BLANK HORIZONTAL OFFSET END Denotes the horizontal cycle during which the vblank signal deasserts.
det_v0sync_hstart	Output	DETECTED VERTICAL SYNC HORIZONTAL OFFSET START Denotes the horizontal cycle during which the vsync signal is asserted.
det_v0sync_hend	Output	DETECTED VERTICAL SYNC HORIZONTAL OFFSET END Denotes the horizontal cycle during which the vsync signal deasserts.
version[31:0]	Output	CORE HARDWARE VERSION Bits 31-16: Set to 0x300a Bits 15 - 0: Reserved

Table 2-3: **General Purpose Processor Port Descriptions *(Cont'd)***

Name	Direction	Description
intr_status[31:0]	Output	INTERRUPT STATUS REGISTER Bit 0: Horizontal Synchronization Lock Status. When the lock polarity is low, set high when the horizontal synchronization timing has changed, signifying a signal lock has been lost. When the lock polarity is high, set high when the horizontal synchronization timing remains unchanged, signifying a signal lock. Bit 1: Horizontal Blank Lock Status. Set high when the horizontal blank timing has changed and the lock polarity is low. Set high when the horizontal blank timing remains unchanged and the lock polarity is high. Bit 2: Vertical Synchronization Lock Status. Set high when the vertical synchronization timing has changed and the lock polarity is low. Set high when the vertical synchronization timing remains unchanged and the lock polarity is high. Bit 3: Vertical Blank Lock Status. Set high when the vertical blank timing has changed and the lock polarity is low. Set high when the vertical blank timing remains unchanged and the lock polarity is high. Bit 4: Reserved. Bit 5: Active Video Lock Status. Set high when the active video timing has changed and the lock polarity is low. Set high when the active video timing remains unchanged and the lock polarity is high. Bit 6: Active Chroma Lock Status. Set high when the active chroma timing has changed and the lock polarity is low. Set high when the active chroma timing remains unchanged and the lock polarity is high. Bit 7: All Lock Status. Set high when bits 0-6 of the interrupt status register are high. When the lock polarity is high, a high on bit 7 indicates that all signals have been locked. When the lock polarity is low, a high on bit 7 indicates that all signal timing have changed. Bit 8: Detected Vertical Blank Interrupt Status. Set high during the first cycle the input vertical blank is asserted active after lock. Bit 9: Detected Active Video Interrupt. Set high during the first cycle the input active video is asserted active after lock. Bits 11-10: Reserved. Bit 12: Generated Vertical Blank Interrupt Status. Set high during the first cycle the output vertical blank is asserted. Bit 13: Generated Active Video Interrupt. Set high during the first cycle the output active video is asserted. Bits 15-14: Reserved. Bits 31-16: Frame Synchronization Interrupt Status. Bits 31-16 are set high when frame syncs 15-0 are set respectively.
intr_enable[31:0]	Input	INTERRUPT ENABLE REGISTER Same bit definitions as in the interrupt status register. Setting a bit high in the interrupt enable register enables the corresponding interrupt. Bits that are low mask the corresponding interrupt from triggering a host interrupt.

Table 2-3: **General Purpose Processor Port Descriptions *(Cont'd)***

Name	Direction	Description
intr_clr[31:0]	Input	INTERRUPT CLEAR REGISTER Same bit definitions as in the interrupt status register. Setting a bit high in the interrupt clear register clears the corresponding bit in the interrupt status register. Bits in the interrupt status register are cleared only on the rising edge of the corresponding bits in the interrupt clear register. Therefore, each bit in the interrupt clear register must be driven low before being driven high to clear the status register bits.
intr_out	Output	HOST INTERRUPT Active high host interrupt output. This output is set active high when an interrupt occurs (an enabled bit in the status register is high) and cleared to low when all enabled status bits in the `intr_status` register have been cleared by writing to the `intr_clr` register.

1. X_{b2} is the $\log_2$(*Max Clocks per Line*) GUI parameter. Y_{b2} is the $\log_2$(*Max Lines per Frame*) GUI parameter.

Dynamic Register Interface

There are 16 dynamic inputs as listed in Table 2-1 (see General Purpose Processor Interface). They may be driven by the user as desired. New values take effect immediately. It is recommended to disable Video Timing Generation while updating these inputs.

EDK pCore (AXI4-Lite) Interface

The Xilinx Video Timing Controller, when configured as an EDK pCore, uses the AXI4-Lite Interface to interface to a microprocessor. See the AMBA AXI4 Interface Protocol Web site for more information on the AXI4 and AXI4-Lite interface signals.

When the developer selects the EDK pCore interface, Xilinx CORE Generator creates a pCore and all support files that can be added to an EDK project as a hardware peripheral. This pCore provides a memory mapped interface for the programmable registers within the core and a complete device driver to enable rapid application development.

Xilinx CORE Generator will place all EDK pCore source files in the "pcores" subdirectory located in the core output directory. The core output directory is given the same name as the component. For example, if the component name is set to "v_tc_v3_0_u0," then the EDK pCore source files will be located in the following directory:

```
<coregen project directory>/v_tc_v3_0_u0/pcores/axi_vtc_v3_00_a
```

The pCore should be copied to the user's `<EDK_Project>/pcores` directory or to a user pCores repository.

Parameter Modification in CORE Generator

EDK pCore parameters found in the `<coregen project directory>/v_tc_v3_0_u0/pcores/axi_vtc_v3_00_a/data/axi_vtc_v2_1_0.mpd` file cannot be modified in the Xilinx CORE Generator tool. Parameters shown on the CORE Generator Graphical User Interface will be disabled if the EDK pCore (AXI4-Lite) Interface is selected. Xilinx recommends that all parameter changes be made with the Video Timing Controller pCore GUI in the EDK environment.

Port Descriptions

Table 2-4 shows the I/O signals on the Xilinx Video Timing Controller when the core is configured with an EDK pCore Interface. The AXI4-Lite signals are specified in Table 2-5.

Table 2-4: **EDK pCore Port Descriptions**

Name	Direction	Description
ce	Input	CLOCK ENABLE Used to halt processing and hold current values.
Detector Interface		
video_clk_in	Input	INPUT CLOCK Core and AXI interface clock (active high edge).
hsync_in	Input	INPUT HORIZONTAL SYNCHRONIZATION Used to set the det_hsync_start and the det_hbp_start registers. Polarity is auto-detected . Optional. Either horizontal blank or horizontal synchronization signal inputs must be present. Both do not have to be present.
hblank_in	Input	INPUT HORIZONTAL BLANK Used to set the det_hfp_start and the det_hactive_start registers. Polarity is auto-detected. Optional. Either horizontal blank or horizontal synchronization signal inputs must be present. Both do not have to be present.
vsync_in	Input	INPUT VERTICAL SYNCHRONIZATION Used to set the det_v0sync_start and the det_v0bp_start registers. Polarity is auto-detected. Optional. One of the following inputs must be present: active video, vertical blank or vertical synchronization.
vblank_in	Input	INPUT VERTICAL BLANK Used to set the det_v0fp_start and the det_v0active_start registers. Polarity is auto-detected. Optional. One of the following inputs must be present: active video, vertical blank or vertical synchronization.
active_video_in	Input	INPUT ACTIVE VIDEO Used to set the det_v0fp_start and the det_v0active_start registers. Polarity is auto-detected . Optional. One of the following inputs must be present: active video, vertical blank or vertical synchronization.
active_chroma_in	Input	INPUT ACTIVE CHROMA Used to set the det_v0achroma_start register and bit 4 in the detection status register. Polarity is auto-detected . Optional.
Generator Interface		

Table 2-4: **EDK pCore Port Descriptions *(Cont'd)***

Name	Direction	Description
video_clk_out	Output	OUTPUT CLOCK Same as `video_clk_in`.
hsync_out	Output	OUTPUT HORIZONTAL SYNCHRONIZATION Generated horizontal synchronization signal. Polarity configured by the control register. Asserted active during the cycle set by the `gen_hsync_start` register and deasserted during the cycle set by the `gen_hbp_start` register.
hblank_out	Output	OUTPUT HORIZONTAL BLANK Generated horizontal blank signal. Polarity configured by the control register. Asserted active during the cycle set by the `gen_hfp_start` and deasserted during the cycle set by the `gen_hactive_start` register.
vsync_out	Output	OUTPUT VERTICAL SYNCHRONIZATION Generated vertical synchronization signal. Polarity configured by the control register. Asserted active during the line set by the `gen_v0sync_start` register and deasserted during the line set by the `gen_v0bp_start` register.
vblank_out	Output	OUTPUT VERTICAL BLANK Generated vertical blank signal. Polarity configured by the control register. Asserted active during the line set by the `gen_v0fp_start` register and deasserted during the line set by the `gen_v0active_start` register.
active_video_out	Output	OUTPUT ACTIVE VIDEO Generated active video signal. Polarity configured by the control register. Active for non blanking lines. Asserted active during the cycle set by the `gen_hactive_start` register and deasserted during the cycle set by the `gen_hbp_start` register.
active_chroma_out	Output	OUTPUT ACTIVE CHROMA Generated active chroma signal. Denotes which lines contain valid chroma samples (used for YUV 4:2:0). Polarity configured by the control register. Active for non blanking lines after the line set by the `gen_v0achroma_start` register (inclusive). For valid chroma lines, asserted active during every cycle the `active_video_out` signal is set per line.

Table 2-4: **EDK pCore Port Descriptions *(Cont'd)***

Name	Direction	Description
Frame Synchronization Interface		
fsync *[Frame Syncs - 1:0]*	Output	FRAME SYNCHRONIZATION OUTPUT Each Frame Synchronization bit toggles for only one clock cycle during each frame. The number of bits is configured with the Frame Syncs GUI parameter. Each bit is independently configured for horizontal and vertical clock cycle position with the fsync_hstart and fsync_vstart registers).

1. The function and timing of these signals are defined in the AMBA AXI Protocol Version: 2.0 Specification.
2. For signals S_AXI_RRESP[1:0] and S_AXI_BRESP[1:0], the core does not generate the Decode Error ('11') response.
Other responses such as '00' (OKAY) and '10' (SLVERR) are generated by the core based upon certain conditions.

EDK pCore Register Set

The EDK pCore Interface provides a memory mapped interface for all programmable registers within the core. All registers default to 0x00000000 on Power-on/Reset unless otherwise noted.

Table 2-5: **EDK pCore Address Map**

Address Offset	Name	Read/Write	Description
0x0000	Control	R/W	General control register
0x0004	Generator Horizontal 0	R/W	Horizontal total and front porch
0x0008	Generator Horizontal 1	R/W	Horizontal sync and back porch
0x000c	Generator Horizontal 2	R/W	Horizontal Active Video
0x0010	Generator Vertical 0	R/W	Vertical total and front porch
0x0014	Generator Vertical 1	R/W	Vertical sync and back porch
0x0018	Generator Vertical 2	R/W	Vertical Active Video and Active Chroma
0x001C	Reserved	-	Reserved
0x0020	Reserved	-	Reserved
0x0024	Reserved	-	Reserved
0x0028	Detector Status	R	Detector polarities and chroma format status
0x002c	Detector Horizontal 0	R	Horizontal total and front porch (detected)
0x0030	Detector Horizontal 1	R	Horizontal sync and back porch (detected)
0x0034	Detector Horizontal 2	R	Horizontal Active Video (detected)
0x0038	Detector Vertical 0	R	Vertical total and front porch (detected)
0x003c	Detector Vertical 1	R	Vertical sync and back porch (detected)
0x0040	Detector Vertical 2	R	Vertical Active Video and Active Chroma (detected)
0x0044	Reserved	-	Reserved
0x0048	Reserved	-	Reserved
0x004c	Reserved	-	Reserved

Table 2-5: **EDK pCore Address Map *(Cont'd)***

Address Offset	Name	Read/Write	Description
0x0050 … 0x008c	Frame Sync 0 - 15 Config	R/W	Horizontal start clock and vertical start line of Frame Sync 0 - 15
0x0090 … 0x009c	Reserved	-	Reserved
0x00a0	Generator Horizontal Offset 0	R/W	Generated vblank horizontal offset
0x00a4	Generator Horizontal Offset 1	R/W	Generated vsync horizontal offset
0x00a8 … 0x00ac	Reserved	-	Reserved
0x00b0	Detector Horizontal Offset 0	R	Detected vblank horizontal offset
0x00b4	Detector Horizontal Offset 1	R	Detected vsync horizontal offset
0x00b8 … 0x00ec	Reserved	-	Reserved
0x00f0	Version Register	R	Core Hardware Version
0x0100	Software Reset	R/W	Resets pCore when written with 0xa000_0000
0x021c	GIER	R/W	Global Interrupt Enable Register
0x0220	ISR	R/W	Interrupt Status/Clear Register
0x0228	IER	R/W	Interrupt Enable Register

Note: The registers of the EDK pCore Interface are big-endian. The registers of the General Purpose Processor Interface are little-endian.

Table 2-6: **Control Register (Address Offset 0x0000)**

0x0000	Control Register		R/W
Name	**Bits**	**Description**	
Reserved	31:27	Reserved	
Active_Chroma_pol[a]	26	Active Chroma Output Polarity	
Active_Video_pol[(a)]	25	Active Video Output Polarity	
Reserved	24	Reserved	
Vblank_pol[(a)]	23	Vertical Blank Output Polarity	
Vsync_pol[(a)]	22	Vertical Synchronization Output Polarity	

Table 2-6: **Control Register (Address Offset 0x0000) *(Cont'd)***

Hblank_pol[a]	21	Horizontal Blank Output Polarity
Hsync_pol[a]	20	Horizontal Synchronization Output Polarity
Reserved	19	Reserved
Vchroma_src_sel[b]	18	Start of Active Chroma Register Source Select
Vactive_src_sel[b]	17	Vertical Active Video Start Register Source Select
Vbp_src_sel[b]	16	Vertical Back Porch Start Register Source Select
Vsync_src_sel[b]	15	Vertical Synchronization Start Register Source Select
Vfp_src_sel[b]	14	Vertical Front Porch Start Register Source Select
Vtotal_src_sel[b]	13	Vertical Total Register Source Select
Hactive_src_sel[b]	12	Horizontal Active Video Start Register Source Select
Hbp_src_sel[b]	11	Horizontal Back Porch Start Register Source Select
Hsync_src_sel[b]	10	Horizontal Synchronization Start Register Source Select
Hfp_src_sel[b]	9	Horizontal Front Porch Start Register Source Select
Htotal_src_sel	8	Horizontal Total Register Source Select
Reserved	7:6	Reserved
Gen_achroma_pixel_skip	5	Generated Active Chroma Pixel Skip. This is the number of pixels to skip between each successive active chroma pixel. Low denotes not to skip pixels. Can be combined with the Active Chroma Line Skip.
Gen_achroma_line_skip	4	Generated Active Chroma Line Skip. This is the number of lines to skip between each successive active chroma line. Low denotes not to skip lines. Used for YUV 4:2:2 or 4:4:4. High denotes to skip every other line. Used for 4:2:0.
Lock_pol	3	Bit 3: Lock Interrupt Polarity. When low, the lock interrupts trigger an interrupt on the falling edge of the internal lock signals. When high, the lock interrupts trigger an interrupt on the rising edge of the internal lock signals.
Sync_en	2	Generator / Detector Synchronization Enable. When low, the generator will not be synchronized to the detector. When high, the generator will be synchronized to the detector.

Table 2-6: **Control Register (Address Offset 0x0000) *(Cont'd)***

0x0000	Control Register		R/W
Name	**Bits**	**Description**	
Det_en	1	Detection Enable. When low, no detection will be performed. All 'locked' status bits will be driven low. When high, perform timing signal detection for enabled signals.	
Gen_en	0	Generation Enable. When low, the generation hardware will not generate video timing output signals. When high, enable hardware to generate output. Set this bit high only after the software has configured the generator registers.	

a. Bits 20-26 configure the polarity of each output. High denotes active high polarity. Low denotes active low polarity.

b. Bits 8-18 select which register controls the generator outputs. Low denotes the detection register will be used. High denotes that the generation register will be used. These bits allow the video timing controller detector to control the generator outputs (when low) or allow the host processor to override each value independently (when high).

Table 2-7: **Generator Horizontal 0 Register (Address Offset 0x0004)**

0x0004	Generator Horizontal 0		R/W
Name	**Bits**	**Description**	
Reserved	31:29	Reserved	
HFP_start	28:16	GENERATED HORIZONTAL FRONT PORCH START Cycle count during which the Horizontal Front Porch starts. Also denotes the end of Active Video.	
Reserved	15:13	Reserved	
HTotal	12:0	GENERATED HORIZONTAL TOTAL Total number of horizontal clock cycles (minus 1) per line including blanking and active cycles. This is the last pixel count on each line. Each line starts at count 0. Maximum allowable Horizontal Total is configured by the *MAX CLOCKS PER LINE* parameter.	

Table 2-8: **Generator Horizontal 1 Register (Address Offset 0x0008)**

0x0008	Generator Horizontal 1		R/W
Name	**Bits**	**Description**	
Reserved	31:29	Reserved	
HBP_start	28:16	GENERATED HORIZONTAL BACK PORCH START Cycle count during which the Horizontal Back Porch starts. Also denotes the end of Horizontal Synchronization.	
Reserved	15:13	Reserved	
HSync_start	12:0	GENERATED HORIZONTAL SYNCHRONIZATION START Cycle count during which the Horizontal Synchronization starts. Also denotes the end of Horizontal Front Porch.	

Table 2-9: **Generator Horizontal 2 Register (Address Offset 0x000C)**

0x000C	Generator Horizontal 2	R/W
Name	**Bits**	**Description**
Reserved	32:13	Reserved
HActive_start	12:0	GENERATED HORIZONTAL ACTIVE VIDEO START Cycle count during which the Horizontal Active Video starts. Also denotes the end of Horizontal Back Porch.

Table 2-10: **Generator Vertical 0 Register (Address Offset 0x0010)**

0x0010	Generator Vertical 0	R/W
Name	**Bits**	**Description**
Reserved	31:29	Reserved
V0FP_start	28:16	GENERATED VERTICAL FRONT PORCH START Line count during which the Vertical Front Porch starts. Also denotes the end of Active Video.
Reserved	15:13	Reserved
V0Total	12:0	GENERATED VERTICAL TOTAL LINES Total number of Vertical lines per frame (minus 1) including blanking and active cycles. This is the last line count in each frame. Each frame starts at line count 0. Maximum allowable Vertical Total is configured by the *MAX LINES PER FRAME* parameter.

Table 2-11: **Generator Vertical 1 Register (Address Offset 0x0014)**

0x0014	Generator Vertical 1	R/W
Name	**Bits**	**Description**
Reserved	31:29	Reserved
V0BP_start	28:16	GENERATED VERTICAL BACK PORCH START Line count during which the Vertical Back Porch starts. Also denotes the end of Vertical Synchronization.
Reserved	15:13	Reserved
V0Sync_start	12:0	GENERATED VERTICAL SYNCHRONIZATION START Line count during which the Vertical Synchronization starts. Also denotes the end of Vertical Front Porch.

Table 2-12: **Generator Vertical 2 Register (Address Offset 0x0018)**

0x0018	Generator Vertical 2		R/W
Name	**Bits**	**Description**	
Reserved	31:29	Reserved	
V0chroma_start	28:16	GENERATED ACTIVE CHROMA START Line count during which the Active Chroma starts. See bit 4 of the control register to configure for YUV 4:2:0 mode.	
Reserved	15:13	Reserved	
V0active_start	12:0	GENERATED VERTICAL ACTIVE VIDEO START Line count during which the Active Video starts. Also denotes the end of Vertical Back Porch.	

Table 2-13: **Detector Status Register (Address Offset 0x0028)**

0x0028	Detector Status		R
Name	**Bits**	**Description**	
Reserved	31:27	Reserved	
Active_Chroma_pol	26	Active Chroma Input Polarity	
Active_Video_pol	25	Active Video Input Polarity	
Field_id_pol	24	Field ID Input Polarity	
Vblank_pol	23	Vertical Blank Input Polarity	
Vsync_pol	22	Vertical Synchronization Input Polarity	
Hblank_pol	21	Horizontal Blank Input Polarity	
Hsync_pol	20	Horizontal Synchronization Input Polarity	
Reserved	19:6	Reserved	
Det_achroma_pixel_skip	5	Detected Active Chroma Pixel Skip. This is the number of pixels skipped between each successive active chroma pixel. Low denotes no pixels are skipped. High denotes every other pixel is skipped.	
Det_achroma_line_skip	4	Detected Active Chroma Line Skip. This is the number of lines skipped between each successive active chroma line. Low denotes no lines are skipped. Used for detecting YUV 4:2:2 or 4:4:4. High denotes every other line is skipped. Used for detecting YUV 4:2:0.	
Reserved	3:0	Reserved	

Note: Bits 20-26 denote the polarity of each input. High denotes active high polarity. Low denotes active low polarity.

Table 2-14: **Detector Horizontal 0 Register (Address Offset 0x002C)**

0x002C	Detector Horizontal 0	R
Name	**Bits**	**Description**
Reserved	31:29	Reserved
HFP_start	28:16	DETECTED HORIZONTAL FRONT PORCH START Detected cycle count during which the Input Horizontal Front Porch starts. Also denotes the end of Input Active Video.
Reserved	15:13	Reserved
HTotal	12:0	DETECTED HORIZONTAL TOTAL Detected Total number of horizontal clock cycles per line including blanking and active cycles (minus 1). Maximum allowable horizontal Total is configured by the *MAX CLOCKS PER LINE* parameter.

Table 2-15: **Detector Horizontal 1 Register (Address Offset 0x0030)**

0x0030	Detector Horizontal 1	R
Name	**Bits**	**Description**
Reserved	31:29	Reserved
HBP_start	28:16	DETECTED HORIZONTAL BACK PORCH START Detected Cycle count during which the Input Horizontal Back Porch starts. Also denotes the end of Input Horizontal Synchronization.
Reserved	15:13	Reserved
HSync_start	12:0	DETECTED HORIZONTAL SYNCHRONIZATION START Detected Cycle count during which the Input Horizontal Synchronization starts. Also denotes the end of Input Horizontal Front Porch.

Table 2-16: **Detector Horizontal 2 Register (Address Offset 0x0034)**

0x0034	Detector Horizontal 2	R
Name	**Bits**	**Description**
Reserved	31:13	Reserved
HActive_start	12:0	DETECTED HORIZONTAL ACTIVE VIDEO START Cycle count during which the Input Horizontal Active Video starts. Also denotes the end of Input Horizontal Back Porch.

Table 2-17: **Detector Vertical 0 Register (Address Offset 0x0038)**

0x0038	Detector Vertical 0	R
Name	**Bits**	**Description**
Reserved	31:29	Reserved
V0FP_start	28:16	DETECTED VERTICAL FRONT PORCH START Line count during which the Input Vertical Front Porch starts. Also denotes the end of Input Active Video.
Reserved	15:13	Reserved
V0Total	12:0	DETECTED VERTICAL TOTAL Total number of Input Vertical lines per frame including blanking and active cycles (minus 1). Maximum allowable Vertical Total is configured by the *MAX LINES PER FRAME* parameter.

Table 2-18: **Detector Vertical 1 Register (Address Offset 0x003C)**

0x003C	Detector Vertical 1	R
Name	**Bits**	**Description**
Reserved	31:29	Reserved
V0BP_start	28:16	DETECTED VERTICAL BACK PORCH START Line count during which the Input Vertical Back Porch starts. Also denotes the end of Input Vertical Synchronization
Reserved	15:13	Reserved
V0Sync_start	12:0	DETECTED VERTICAL SYNCHRONIZATION START Line count during which the Input Vertical Synchronization starts. Also denotes the end of Input Vertical Front Porch.

Table 2-19: **Detector Vertical 2 Register (Address Offset 0x0040)**

0x0040	Detector Vertical 2	R
Name	**Bits**	**Description**
Reserved	31:29	Reserved
V0chroma_start	28:16	DETECTED ACTIVE CHROMA START Line count during which the Input Active Chroma starts.
Reserved	15:13	Reserved
V0active_start	12:0	DETECTED VERTICAL ACTIVE VIDEO START Line count during which the Input Vertical Active Video starts. Also denotes the end of Input Vertical Back Porch.

Table 2-20: **Frame Sync 0 Register (Address Offset 0x0050)**

0x0050	Frame Sync 0	R/W
Name	**Bits**	**Description**
Reserved	31:29	Reserved
V_start	28:16	FRAME SYNCHRONIZATION VERTICAL START REGISTER Vertical line during which Frame Synchronization 0 is active. ***Note:*** Frame Syncs are not active during the complete line, only in the cycle during which both the `fsync_vstart` and `fsync_hstart` are valid each frame.
Reserved	15:13	Reserved
H_start	12:0	FRAME SYNCHRONIZATION HORIZONTAL START REGISTER Horizontal Cycle during which Frame Synchronization 0 is active.

Note: Frame Sync 1-15 Registers (address offset 0x54 - 0x8c) have the same format as the Frame Sync 0 Register.

Table 2-21: **Generator Vblank Horizontal Offset Register (Address Offset 0x00a0)**

0x00A0	Generator VBlank Horizontal Offset	R/W
Name	**Bits**	**Description**
Reserved	31:29	Reserved.
V0blank_hend	28:16	Vertical blank horizontal offset end. Denotes the horizontal cycle during which the vblank signal deasserts.
Revision	15:13	Revision Number. Set to 0XA.
V0blank_hstart	12:0	Vertical blank horizontal offset start. Denotes the horizontal cycle during which the vblank signal is asserted.

Table 2-22: **Generator VSync Horizontal Offset Register (Address Offset 0x00a4)**

0x00A4	Generator VSync Horizontal Offset	R/W
Name	**Bits**	**Description**
Reserved	31:29	Reserved.
V0sync_hend	28:16	Vertical sync horizontal offset end. Denotes the horizontal cycle during which the vsync signal deasserts.
Reserved	15:13	Reserved.
V0sync_hstart	12:0	Vertical sync horizontal offset start. Denotes the horizontal cycle during which the vsync signal is asserted.

Table 2-23: **Detector Vblank Horizontal Offset Register (Address Offset 0x00b0)**

0x00B0	Detector VBlank Horizontal Offset		R
Name	**Bits**	**Description**	
Reserved	31:29	Reserved.	
V0blank_hend	28:16	Vertical blank horizontal offset end. Denotes the horizontal cycle during which the vblank signal deasserts.	
Reserved	15:13	Reserved.	
V0blank_hstart	12:0	Vertical blank horizontal offset start. Denotes the horizontal cycle during which the vblank signal is asserted.	

Table 2-24: **Detector Vsync Horizontal Offset Register (Address Offset 0x00b4)**

0x00B4	Detector VBlank Horizontal Offset		R
Name	**Bits**	**Description**	
Reserved	31:29	Reserved.	
V0sync_hend	28:16	Vertical sync horizontal offset end. Denotes the horizontal cycle during which the vsync signal deasserts.	
Reserved	15:13	Reserved.	
V0sync_hstart	12:0	Vertical sync horizontal offset start. Denotes the horizontal cycle during which the vsync signal is asserted.	

Table 2-25: **Version Register (Address Offset 0x00F0)**

0x00F0	Version Register		R
Name	**Bits**	**Description**	
Major Version	31:29	Major Version Number. Set to 0x3.	
Minor Version	28:21	Minor Version Number. Set to 0x00.	
Revision	20:17	Revision Number. Set to 0xA.	
Reserved	16:0	Reserved	

Table 2-26: **Software Reset Register (Address Offset 0x0100)**

0x0100	Software Reset		R/W
Name	**Bits**	**Description**	
Soft_Reset_Value	31:0	Soft Reset to reset the registers and IP Core, data Value provided by the EDK create peripheral utility. (0xa000_0000)	

Table 2-27: **Global Interrupt Enable Register (Address Offset 0x021c)**

0x00F0	Version Register	R/W
Name	**Bits**	**Description**
GIER	31	Global Interrupt Enable. Writing a 1 to this bit will enable all interrupts. Set to 0 (all interrupts disabled) by default.
Reserved	30:0	Reserved

Table 2-28: **ISR (Interrupt Status/Clear) Register (Address Offset 0x0220)**

0x0220	ISR - Interrupt Status/Clear	R/W
Name	**Bits**	**Description**
Fsync	31:16	Frame Synchronization Interrupt Status. Bits 16-31 are set high when frame syncs 0-15 are set respectively.
Reserved	15:14	Reserved
Gen_active_video	13	Generated Active Video Interrupt. Set high during the first cycle the output active video is asserted.
Gen_blank	12	Generated Vertical Blank Interrupt Status. Set high during the first cycle the output vertical blank is asserted.
Reserved	11:10	Reserved
Det_active_video	9	Detected Active Video Interrupt. Set high during the first cycle the input active video is asserted active after lock.
Det_vblank	8	Detected Vertical Blank Interrupt Status. Set high during the first cycle the input vertical blank is asserted active after lock.
All_lock	7	All Lock Status. Set High when bits 0-6 are high, signifying that all enabled detection signals have locked. Signals that have detection disabled will not affect this bit.
Active_chroma_lock	6	Active Chroma Lock Status. Set high when the active chroma timing has changed and the lock polarity is low. Set high when the active chroma timing remains unchanged and the lock polarity is high.
Active_video_lock	5	Active Video Lock Status. Set high when the active video timing has changed and the lock polarity is low. Set high when the active video timing remains unchanged and the lock polarity is high.
Reserved	4	Reserved
Vblank_lock	3	Vertical Blank Lock Status. Set high when the vertical blank timing has changed and the lock polarity is low. Set high when the vertical blank timing remains unchanged and the lock polarity is high.
Vsync_lock	2	Vertical Synchronization Lock Status. Set high when the vertical synchronization timing has changed and the lock polarity is low. Set high when the vertical synchronization timing remains unchanged and the lock polarity is high.
Hblank_lock	1	Horizontal Blank Lock Status. Set high when the horizontal blank timing has changed and the lock polarity is low. Set high when the horizontal blank timing remains unchanged and the lock polarity is high.

Table 2-28: **ISR (Interrupt Status/Clear) Register (Address Offset 0x0220) *(Cont'd)***

0x0220	ISR - Interrupt Status/Clear	R/W
Name	**Bits**	**Description**
Hsync_lock	0	Horizontal Synchronization Lock Status. When the lock polarity is low, set high when the horizontal synchronization timing has changed, signifying a signal lock has been lost. When the lock polarity is high, set high when the horizontal synchronization timing remains unchanged, signifying a signal lock.

Note: Setting a bit high in the ISR will clear the corresponding interrupt.

Table 2-29: **IER (Interrupt Enable) Register (Address Offset 0x0228)**

0x0228	IER - Interrupt Enable	R/W
Name	**Bits**	**Description**
Fsync	31:16	Frame Synchronization Interrupt Enable.
Reserved	15:14	Reserved
Gen_active_video	13	Generated Active Video Interrupt Enable.
Gen_blank	12	Generated Vertical Blank Interrupt Enable.
Reserved	11:10	Reserved
Det_active_video	9	Detected Active Video Interrupt Enable.
Det_vblank	8	Detected Vertical Blank Interrupt Enable.
All_lock	7	All Lock Enable.
Active_chroma_lock	6	Active Chroma Lock Enable.
Active_video_lock	5	Active Video Lock Enable.
Reserved	4	Reserved
Vblank_lock	3	Vertical Blank Lock Enable.
Vsync_lock	2	Vertical Synchronization Lock Enable.
Hblank_lock	1	Horizontal Blank Lock Enable.
Hsync_lock	0	Horizontal Synchronization Lock Enable.

Setting a bit high in the interrupt enable register enables the corresponding interrupt. Bits that are low mask the corresponding interrupt from triggering a host interrupt.

Customizing and Generating the Core

This chapter includes information on using Xilinx tools to customize and generate the core.

Graphical User Interface (GUI)

The Xilinx Video Timing Controller core is easily configured to meet the developer's specific needs through the CORE Generator graphical user interface (GUI). See Figure 3-1. This section provides a quick reference to parameters that can be configured at generation time.

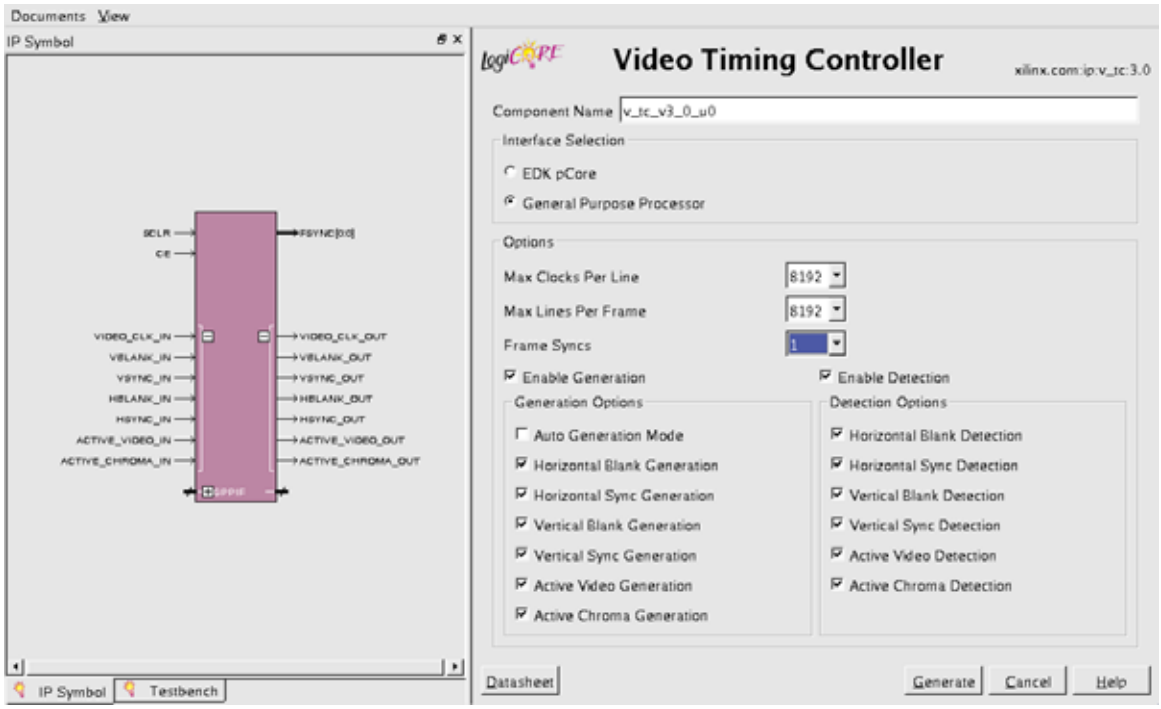

Figure 3-1: **Video Timing Controller Graphical User Interface**

The GUI displays a representation of the IP symbol on the left side and the parameter assignments on the right side, described as follows:

- **Component Name:** The component name is used as the base name of output files generated for the module. Names must begin with a letter and must be composed from characters: a to z, 0 to 9 and "_".

  Note: The name **`v_tc_v3_0`** is not allowed.

- Interface Selection: The Video Timing Controller is generated with one of two interfaces
  - EDK pCore Interface: The CORE Generator tool will generate the Video Timing Controller as a pCore which can be easily imported into an EDK project as a hardware peripheral. The core registers can then be programmed in real-time via the MicroBlaze processor. See the Port Descriptions in Chapter 2 section.
  - General Purpose Processor Interface: The CORE Generator tool will generate a set of ports that can be used to program the Video Timing Controller. See the General Purpose Processor Interface section.
- **Maximum Clocks per Line:** This parameter sets the maximum number of clock cycles per video line that the Video Timing Controller can generate or detect. Values of 128, 256, 512, 1024, 2048, 4096 and 8192 are valid.
- **Maximum Lines per Frame:** This parameter sets the maximum number of lines per video frame that the Video Timing Controller can generate or detect. Values of 128, 256, 512, 1024, 2048, 4096 and 8192 are valid.
- **Frame Syncs**: This parameter sets the number of frame synchronization outputs to generate and supports up to 16 independent outputs.
- **Enable Generation:** This parameter enables or disables the video timing outputs.
- **Auto Mode Generation:** When enabled, this parameter will cause the generated video timing outputs to change based on the detected inputs. If this parameter is disabled, the video timing outputs will be generated based on only the first detected input format. The output for the generated synchronization signals will continue even if the detection block loses lock. This parameter is available only if both the *Enable Generation* and *Enable Detection* parameters are enabled.

  Note: This parameter has an effect only if one or more of the source select control register bits are set to low.
- **Horizontal Blank Generation:** This parameter enables or disables generating the horizontal blank output.
- **Horizontal Sync Generation:** This parameter enables or disables generating the horizontal synchronization output.
- **Vertical Blank Generation:** This parameter enables or disables generating the vertical blank output.
- **Vertical Sync Generation:** This parameter enables or disables generating the vertical synchronization output.
- **Active Video Generation:** This parameter enables or disables generating the active video output.
- **Active Chroma Generation:** This parameter enables or disables generating the active chroma output.
- **Enable Detection:** This parameter enables or disables the detecting the timing of the video inputs.
- **Horizontal Blank Detection:** This parameter enables or disables detecting the horizontal blank input.
- **Horizontal Sync Detection:** This parameter enables or disables detecting the horizontal synchronization input.
- **Vertical Blank Detection:** This parameter enables or disables detecting the vertical blank input.
- **Vertical Sync Detection:** This parameter enables or disables detecting the vertical synchronization input.

- **Active Video Detection:** This parameter enables or disables detecting the active video input.
- **Active Chroma Detection:** This parameter enables or disables detecting the active chroma input.

EDK pCore Graphical User Interface (GUI)

When the Xilinx Video Timing Controller core is generated from the CORE Generator software as an EDK pCore, it is generated with each option set to the default value. All customizations of a Video Timing Controller pCore are done with the EDK pCore graphical user interface (GUI). Figure 3-2 illustrates the EDK pCore GUI for the VideoTiming Controller pCore. All of the options in the EDK pCore GUI for the Video

Timing Controller core correspond to the same options in the CORE Generator software GUI. See Graphical User Interface (GUI) for details about each option.

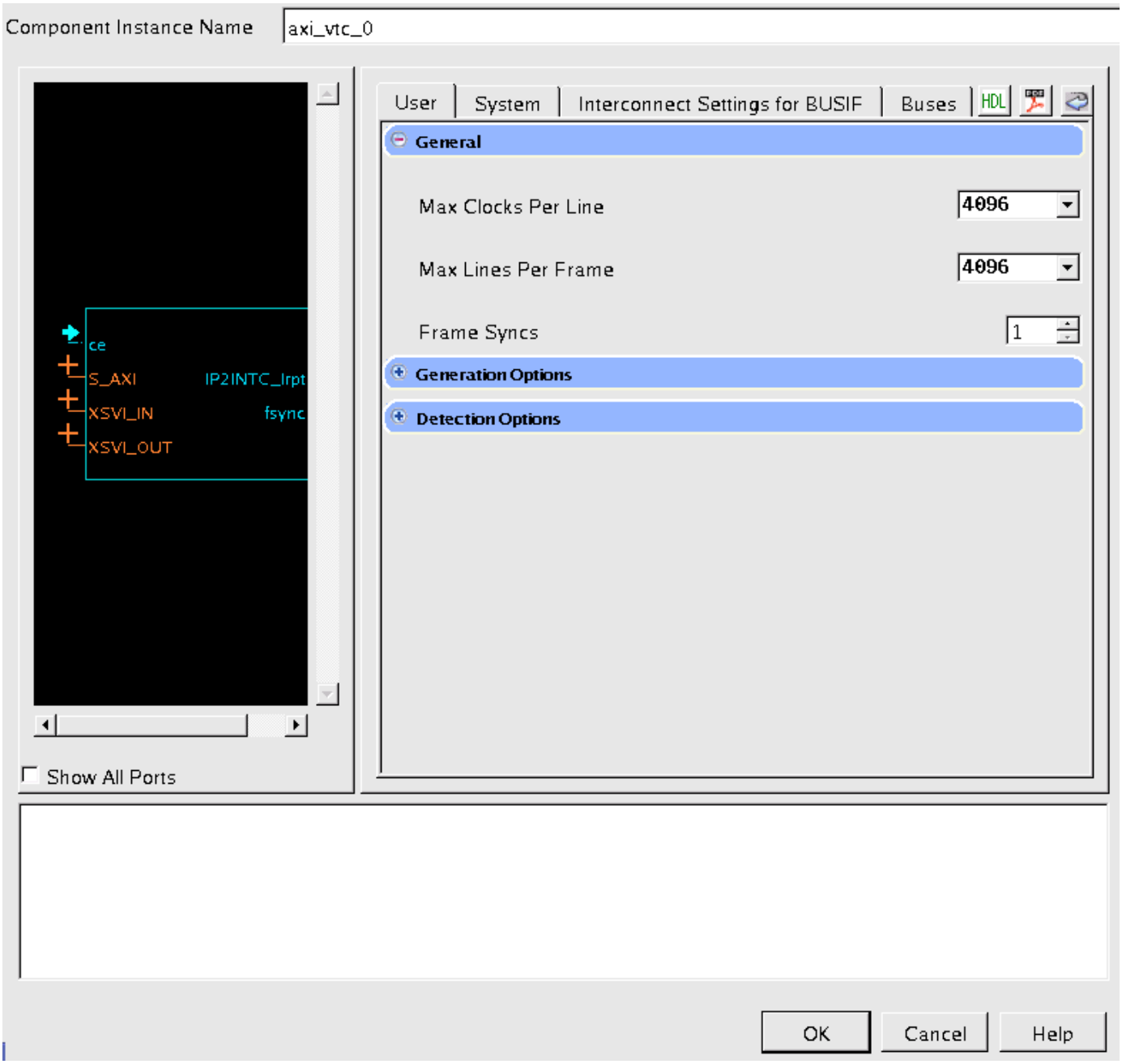

Figure 3-2: **EDK pCore GUI**

Parameter Values in the XCO File

Table 1 defines valid entries for the Xilinx CORE Generator (XCO) parameters. Xilinx strongly suggests that XCO parameters are not manually edited in the XCO file; instead, use the CORE Generator software GUI to configure the core and perform range and parameter value checking. The XCO parameters are helpful in defining the interface to other Xilinx tools.

Table 3-1: XCO Parameters

Table 3-1:

XCO Parameter	Default	Valid Values
component_name	v_tc_v3_0_u0	ASCII text using characters: a..z, 0..9 and "_" starting with a letter. Note: "v_osd_v4_0" is not allowed.
interface_selection	EDK_pCore	EDK_pCore, General_Purpose_Processor
max_clocks_per_line	4096	128,256,512,1024,2048,4096,8192
max_lines_per_frame	4096	128,256,512,1024,2048,4096,8192
enable_detection	false	true,false
vertical_blank_detection	true	true,false
vertical_sync_detection	true	true,false
horizontal_blank_detection	true	true,false
horizontal_sync_detection	true	true,false
active_video_detection	true	true,false
active_chroma_detection	false	true,false
enable_generation	true	true,false
vertical_blank_generation	true	true,false
vertical_sync_generation	true	true,false
horizontal_blank_generation	true	true,false
horizontal_sync_generation	true	true,false
active_video_generation	true	true,false
active_chroma_generation	false	true,false
auto_generation_mode	false	true,false
frame_syncs	1	1-16

Output Generation

The output files generated from the Xilinx CORE Generator software for the Video Timing Controller core depend upon whether the interface selection is set to EDK pCore or General Purpose Processor. The output files are placed in the project directory.

EDK pCore Files

When the interface type is set to EDK pCore, CORE Generator then outputs the core as a pCore that can be easily incorporated into an EDK project. The pCore output consists of a hardware pCore and a software driver. The pCore has the following directory structure:

<Component_Name>

drivers

```
vtc_v2_00_a
    data
    doc
        html
            api
    example
    src
pcores
    axi_vtc_v3_00_a
        data
        hdl
            vhdl
```

File Details

- <project directory>

  This is the top-level directory. It contains xco and other assorted files.

Name	Description
<component_name>.xco	Log file from CORE Generator software describing which options were used to generate the core. An XCO file can also be used as an input to the CORE Generator software.
<component_name>_flist.txt	A text file listing all of the output files produced when the customized core was generated in the CORE Generator software.

- <project directory>/<component_name>/pcores/axi_vtc_v3_00_a/data

  This directory contains files that EDK uses to define the interface to the pCore.

- < project directory>/<component_name>/pcores/axi_vtc_v3_00_a/hdl/vhdl

  This directory contains the Hardware Description Language (HDL) files that implement the pCore.

- < project directory>/<component_name>/drivers/vtc_v2_00_a/data

  This directory contains files that Software Development Kit (SDK) uses to define the operation of the pCore's software driver.

- < project directory>/<component_name>/drivers/vtc_v2_00_a/doc/html/api

  This directory contains HTML documentation files for the pCore's software driver.

- < project directory>/<component_name>/drivers/vtc_v2_00_a/src

This directory contains the source code of the pCore's software driver.

Name	Description
xvtc.c	Provides the Application Program Interface (API) access to all features of the Video Timing Controller device driver.
xvtc.h	Provides the API access to all features of the Video Timing Controller device driver.
xvtc_g.c	Contains a template for a configuration table of Video Timing Controller core.
xvtc_hw.h	Contains identifiers and register-level driver functions (or macros) that can be used to access the Video Timing Controller core.
xvtc_intr.c	Contains interrupt-related functions of the Video Timing Controller device driver.
xvtc_sinit.c	Contains static initialization methods for the Video Timing Controller device driver.

General Purpose Processor Files

When the interface selection is set to General Purpose Processor, CORE Generator then outputs the core as a netlist that can be inserted into a processor interface wrapper or instantiated directly in an HDL design. The output is placed in the <project directory>.

File Details

The CORE Generator software output consists of some or all the following files.

Table 3-2: **CORE Generator Software Output**

Name	Description
<component_name>_readme.txt	Readme file for the core.
<component_name>.ngc	The netlist for the core.
<component_name>.veo	The HDL template for instantiating the core.
<component_name>.vho	
<component_name>.v	The structural simulation model for the core. It is used for functionally simulating the core.
<component_name>.vhd	
<component_name>.xco	Log file from CORE Generator software describing which options were used to generate the core. An XCO file can also be used as an input to the CORE Generator software.
<component_name>_flist.txt	A text file listing all of the output files produced when the customized core was generated in the CORE Generator software.
<component_name>.asy	IP symbol file
<component_name>.gise	ISE® software subproject files for use when including the core in ISE software designs.
<component_name>.xise	

Chapter 4

Designing with the Core

Basic Architecture

The Video Timing Controller core contains three modules: the video timing detector, the video timing generator and the interrupt controller. See Figure 4-1.

Either the detector or the generator module can be disabled with the CORE Generator GUI to save resources.

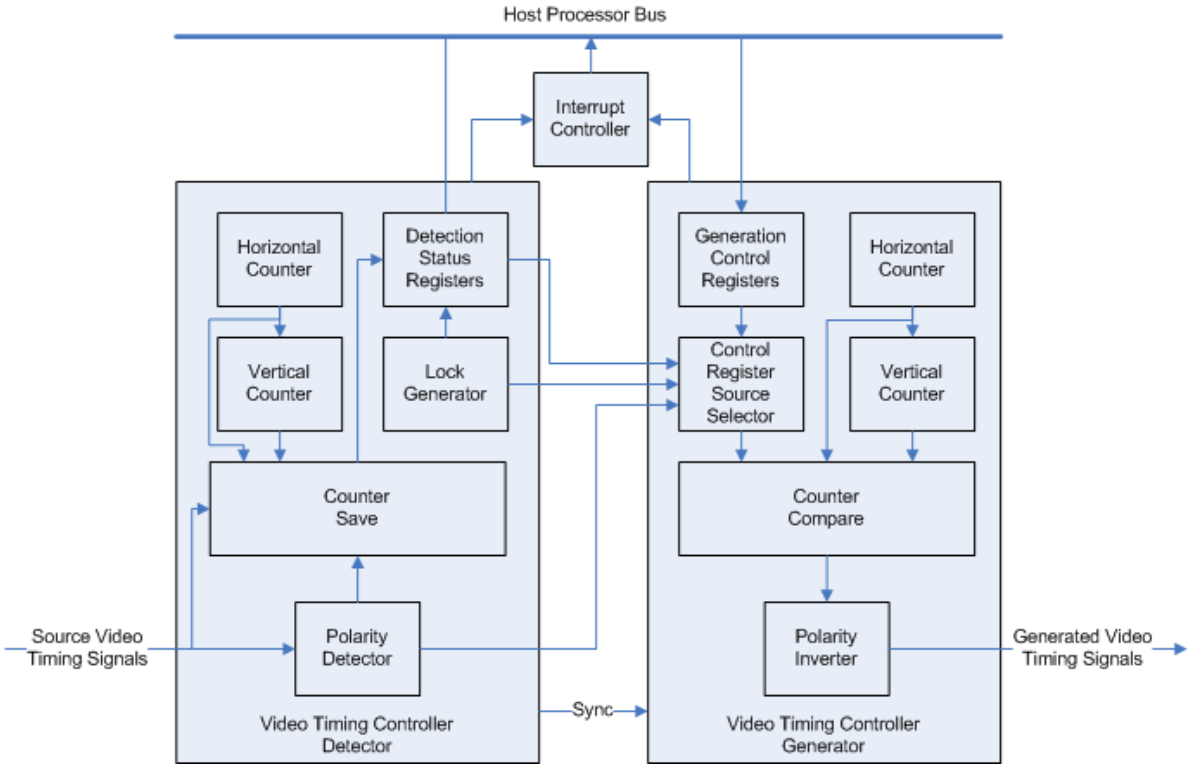

Figure 4-1: **Video Timing Controller Block Diagram**

Control Signals and Timing

The Video Timing Controller Inputs and Outputs are discussed and shown with timing diagrams in the following sections.

The blanking and active period definitions were discussed in Chapter 1, Overview. In addition to these definitions, the period from the start of blanking (or end of active video) to the start of synchronization is called the front porch. The period from the end of synchronization to the end of blanking (or start of active video) is called the back porch. The total horizontal period (including blanking and active video) can also be defined, and similarly the total vertical period.

Figure 4-2 shows the start of the horizontal front porch (HFP_Start), synchronization (HSync_Start), back porch (HBP_Start) and active video (Hactive_Start). It also shows the start of the vertical front porch (VFP_Start), synchronization (VSync_Start), back porch (VBP_Start) and active video (Vactive_Start). The total number of horizontal clock cycles is H Total and the total number of lines is the V Total.

These definitions of video frame periods are used for both Video Timing Detection and Video Timing Generation.

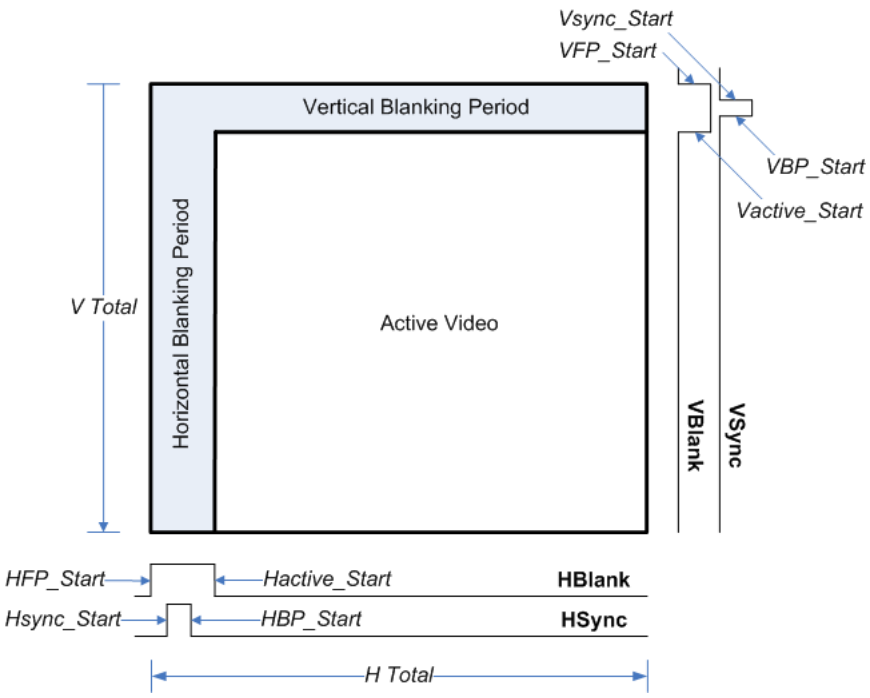

Figure 4-2: **Example Video Frame and Timing Signals with Front and Back Porch**

Video Timing Detection

The Video Timing Controller has six optional inputs for detecting the timing of the input video signal: vertical blank, vertical synchronization, horizontal blank, horizontal synchronization, active video and active chroma (see Detector Interface in Table 2-1). The minimum set of inputs required to detect is either vertical blank, horizontal blank and active video or vertical sync, horizontal sync and active video. To enable detection, the *Enable Detection* GUI parameter must be set, and the control register bit 1 must also be set. The GUI parameter allows saving FPGA resources. The *Control Register* allows run-time

flexibility. Other GUI parameters can be set to selectively disable detection of one or more input video timing signals (see Graphical User Interface (GUI) in Chapter 3).

The detected polarity of each input signal is shown by bits 26-20 of the Detection Status Register. High denotes active high polarity, and low denotes active low polarity. Bit 4 of the Detection Status Register shows the number of lines skipped between each active chroma line. High denotes that every other line is skipped (4:2:0), and low denotes that no lines are skipped (4:4:4 or 4:2:2).

The Video Timing Controller also has 11 little-endian output busses to show the status and timing of the input signals. Horizontal Detection Status busses have a width of $\log_2$(*Max Clocks per Line*). Vertical Detection Status busses have a width of $\log_2$(*Max Lines per Frame*).

Video Timing Generation

The Video Timing Controller can generate up to six output video signals: vertical blank, vertical synchronization, horizontal blank, horizontal synchronization, active video and active chroma (see Generator Interface in Table 2-1). To enable generation of these signals, the *Enable Generation* GUI parameter must be set, and the control register bit 0 must also be set. Other GUI parameters can be set to selectively disable generation of one or more video timing signals (see Graphical User Interface (GUI) in Chapter 3).

The polarity of each output signal can be set by bits 26-20 of the *Control Register*. High denotes active high polarity, and low denotes active low polarity. Bit 4 of the Control Register also sets the number of lines skipped between each active chroma line. High denotes that every other line is skipped (4:2:0), and low denotes that no lines are skipped (4:4:4 or 4:2:2).

The Video Timing Controller has 11 little-endian input control busses to set the timing of the output signals. Each bus has a corresponding bit in the *Control Register* (bits 18-8) called *Source Selects* to select the internal detection bus or the external input generation bus. These bits allow the detected timing (if enabled) to control the generated outputs or allow the host processor to override each value independently via the generation input control busses (see Control Register in Table 2-1). Horizontal Generation Control busses have a width of $\log_2$(*Max Clocks per Line*). Vertical Generation Control busses have a width of $\log_2$(*Max Lines per Frame*).

Table 4-1 through Table 4-6 show example settings of the input control busses and the resultant video timing output signals.

Programming the horizontal generation registers to the values shown in Table 4-1 will result in the video timing signal outputs shown in Figure 4-3.

Notice that in Table 4-1 the Control Register bit 0 is set to enable generation, that all source selects are set to 1 to select the Generation Registers and that the polarity bits are all set to 1 to configure the outputs for active high polarity. (See Control Register in Table 2-1for a description of this register).

Table 4-1: **Example Horizontal Generation Register Inputs**

Generation Register Input	Value
gen_htotal	0x006
gen_hfp_start	0x000
gen_hsync_start	0x001
gen_hbp_start	0x002

Table 4-1: **Example Horizontal Generation Register Inputs**

Generation Register Input	Value
gen_hactive_start	0x004
control	0x07f7_ff05

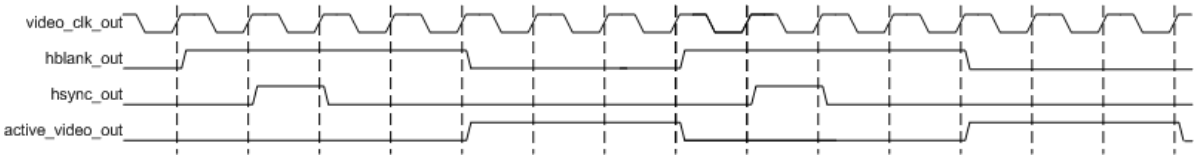

Figure 4-3: **Generated Horizontal Timing**

Note: All signals are shown active high. The polarities of the output signals can be changed at any time in the control register.

Next, an example vertical generation configuration is given. Programming the vertical generation registers to the values shown in Table 4-2 will result in the video timing signal outputs shown in Figure 4-4.

Notice that in Table 4-2 the Control Register bit 4 is set to 0 to configure the number of lines skipped between each active chroma line to be 0. This configures the Active Chroma output signal for 4:4:4 or 4:2:2 mode in which every line contains valid chroma samples. (See Control Register in Table 2-1 for a description of this register.

Table 4-2: **Example Vertical Generation Register Inputs**

Generation Register Input	Value
gen_v0total	0x006
gen_v0fp_start	0x000
gen_v0sync_start	0x001
gen_v0bp_start	0x002
gen_v0active_start	0x003
gen_v0achroma_start	0x003
control	0x07f7_ff05

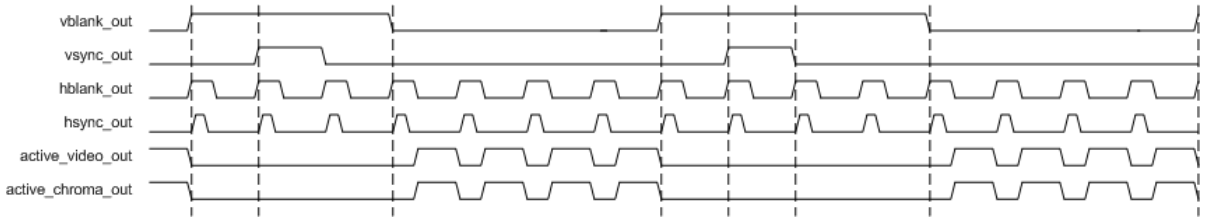

Figure 4-4: **Generated Vertical Timing (4:4:4 Chroma)**

Next is a vertical generation example similar to the previous except that the Active Chroma output is configured to for YUV 4:2:0. Programming the vertical generation registers to the

values shown in Table 4-3 will result in the video timing signal outputs shown in Figure 4-5.

Notice that in Table 4-3 the Control Register bit 4 is set to 1 to configure the number of lines skipped between each active chroma line to be one line. This configures the Active Chroma output signal for 4:2:0 mode in which only every other line contains valid chroma samples. (See Control Register in Table 2-1 for a description of this register.)

Table 4-3: **Example Vertical Generation Register Inputs (4:2:0 Chroma)**

Generation Register Input	Value
gen_v0total	0x006
gen_v0fp_start	0x000
gen_v0sync_start	0x001
gen_v0bp_start	0x002
gen_v0active_start	0x003
gen_v0achroma_start	0x003
control	0x07f7_ff15

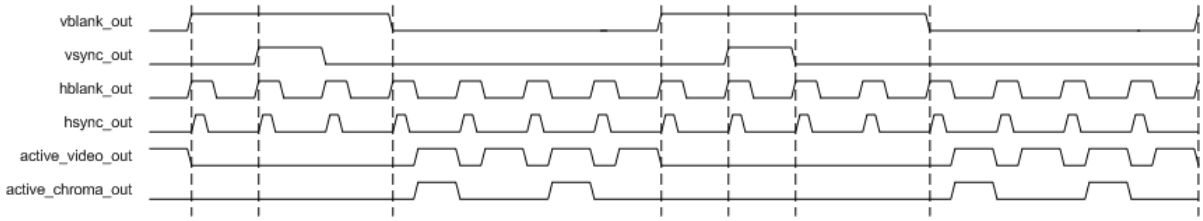

Figure 4-5: **Generated Vertical Timing (4:2:0 Chroma)**

Next is a vertical generation example similar to the previous except that the Active Chroma output is configured to be active for odd lines instead of even lines. Programming the vertical generation registers to the values shown in Table 4-4 will result in the video timing signal outputs shown in Figure 4-6.

Notice that the Generated Active Chroma Start Register is set to 4 instead of 3, as in the previous example. This configures the Active Chroma output signal for 4:2:0 mode, but with the opposite line set.

Table 4-4: **Example Vertical Generation Register Inputs (Alternate 4:2:0 Chroma)**

Generation Register Input	Value
gen_v0total	0x006
gen_v0fp_start	0x000
gen_v0sync_start	0x001
gen_v0bp_start	0x002
gen_v0active_start	0x003
gen_v0achroma_start	0x004
control	0x07f7_ff15

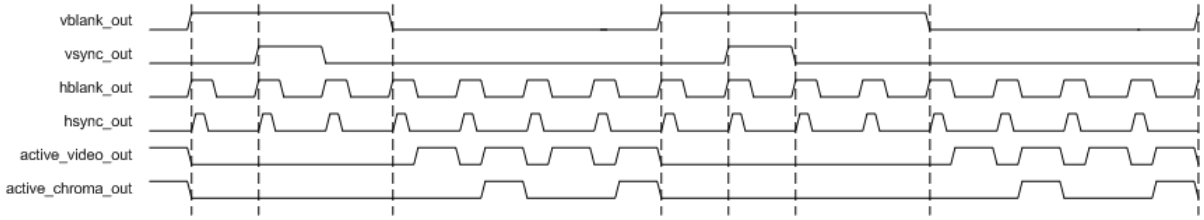

Figure 4-6: **Generated Vertical Timing (Alternate 4:2:0 Chroma)**

The next example shows how the Video Timing Controller can be configured to regenerate timing signals to selectively override individual characteristics. Table 4-5 shows the detection output register output signals. Programming the horizontal generation registers to the values shown in Table 4-6 will result in the video timing signal outputs shown in Figure 4-7.

Table 4-5: **Example Horizontal Detection Register Outputs**

Detection Register Output	Value
det_htotal	0x006
det_hfp_start	0x000
det_hsync_start	0x001
det_hbp_start	0x002
det_hactive_start	0x004
det_status	0x07f0_000

Notice that all polarities bits are high in the Detection Status Register, signifying that all inputs are detected to have an active high polarity.

Table 4-6: **Example Horizontal Generation Register Inputs**

Generation Register Input	Value
gen_hfp_start	0x006
gen_hactive_start	0x005
control	0x07e0_1207

Notice, in the Control Register, that bit 0 is set to enable generation, bit 1 is set to enable detection and bit 2 is set to enable synchronizing the generated output to the detected inputs.

The Horizontal Front Porch Start Register Source Select (bit 9 of the Control Register) is set to 1 and the Horizontal Active Video Start Register Source Select (bit 12 of the Control Register) is set to 1. This signifies that the `gen_hfp_start` and the `gen_hactive_start` registers will be used instead of the `det_hfp_start` and the `det_hactive_start` registers since these values are being overridden. All other source selects are low, signifying that the detection register should be used.

Also notice that the polarity of the output horizontal synchronization has been changed to active low by clearing bit 20 of the Control Register.

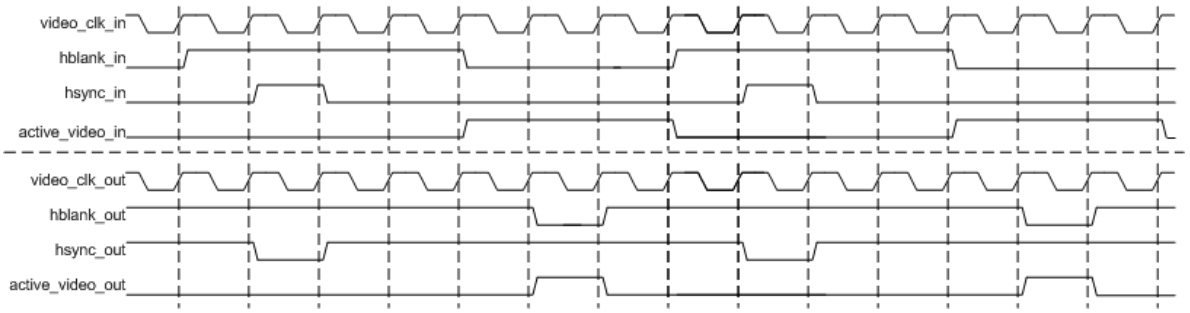

Figure 4-7: **Detected and Regenerated Horizontal Timing**

Note: All generated outputs remain synchronized to the inputs. The only changes made to the output are to the horizontal synchronization polarity and to the active video start and stop times.

Synchronization

Generation of the video timing output signals can be synchronized to the detected video timing input signals or generated independently. Synchronization of the output to the input allows the developer to override each individual timing signal with different settings such as signal polarity or start time. For example, the active video signal could be regenerated shifted one cycle earlier or later. This provides a flexible method for regenerating video timing output signals with different settings while remaining synchronized to the input timing.

The Video Timing Controller also has a GUI parameter, called Auto Mode Generation, to control the behavior of the generated outputs based on the detected inputs. When the Auto Mode Generation parameter is set, the generated video timing outputs will change based on the detected inputs. If this parameter is not set, then the video timing outputs will be generated based on only the first detected input format. (If the detector loses lock, the generated outputs will continue to be generated.) To change output timing while Auto Mode Generation is set, timing detection must first be disabled by clearing bit 1 in the Control Register and then re-enabling, if any of the Source Select bits are low.

Frame Syncs

The Video Timing Controller has a frame synchronization output bus. Each bit can be configured to toggle high for any one clock cycle during each video frame. Each bit is independently configured for horizontal and vertical clock cycle position with the `fsync_hstart` and `fsync_vstart` registers. Table 4-7 shows which bits in the `fsync_hstart` and `fsync_vstart` registers control which frame synchronization output.

Table 4-7: **Frame Synchronization Control Registers**

Frame Synchronization Output	Horizontal Position (fsync_hstart) Bits	Vertical Position (fsync_vstart) Bits
fsync[0]	[log2(x) - 1] to [0]	[log2(y) - 1] to [0]
fsync[1]	[2*log2(x) - 1] to [log2(x)]	[2*log2(y) - 1] to [log2(y)]
fsync[2]	[3*log2(x) - 1] to [2*log2(x)]	[3*log2(y) - 1] to [2*log2(y)]

Table 4-7: **Frame Synchronization Control Registers**

Frame Synchronization Output	Horizontal Position (fsync_hstart) Bits	Vertical Position (fsync_vstart) Bits
fsync[3]	[4*log2(x) - 1] to [3*log2(x)]	[4*log2(y) - 1] to [3*log2(y)]
fsync[4]	[5*log2(x) - 1] to [4*log2(x)]	[5*log2(y) - 1] to [4*log2(y)]
fsync[5]	[6*log2(x) - 1] to [5*log2(x)]	[6*log2(y) - 1] to [5*log2(y)]
fsync[6]	[7*log2(x) - 1] to [6*log2(x)]	[7*log2(y) - 1] to [6*log2(y)]
fsync[7]	[8*log2(x) - 1] to [7*log2(x)]	[8*log2(y) - 1] to [7*log2(y)]
fsync[8]	[9*log2(x) - 1] to [8*log2(x)]	[9*log2(y) - 1] to [8*log2(y)]
fsync[9]	[10*log2(x) - 1] to [9*log2(x)]	[10*log2(y) - 1] to [9*log2(y)]
fsync[10]	[11*log2(x) - 1] to [10*log2(x)]	[11*log2(y) - 1] to [10*log2(y)]
fsync[11]	[12*log2(x) - 1] to [11*log2(x)]	[12*log2(y) - 1] to [11*log2(y)]
fsync[12]	[13*log2(x) - 1] to [12*log2(x)]	[13*log2(y) - 1] to [12*log2(y)]
fsync[13]	[14*log2(x) - 1] to [13*log2(x)]	[14*log2(y) - 1] to [13*log2(y)]
fsync[14]	[15*log2(x) - 1] to [14*log2(x)]	[15*log2(y) - 1] to [14*log2(y)]
fsync[15]	[16*log2(x) - 1] to [15*log2(x)]	[16*log2(y) - 1] to [15*log2(y)]

Notes:

1. x is the Max Clocks per Line GUI parameter. y is the Max Lines per Frame GUI parameter.
2. The width of the frame synchronization bus is configured with the Frame Syncs GUI parameter. Frame syncs can be used for various control applications including controlling the timing of processing of external modules.

Host CPU Interrupts

The Video Timing Controller has an active high host CPU interrupt output. This output is set high when an interrupt occurs and set low when the interrupt event has been cleared by the host CPU. The Video Timing Controller also contains three 32-bit registers for configuring and reporting status of interrupts: the Interrupt Status, the Interrupt Enable and the Interrupt Clear Registers. A logical AND is performed on the Interrupt Enable Register and the Interrupt Status Register to set the interrupt output high. The Interrupt Clear Register is used to clear the Interrupt Status Register. Interrupt Status Register bits are cleared only on the rising edge of the corresponding Interrupt Clear Register. Therefore, each bit in the Interrupt Clear Register must be driven low before being driven high to clear the status register bits.

The polarity of the lock interrupts is configurable by bit 3 in the Control Register (see Table 2-1). When this bit is low, the lock interrupts will trigger an interrupt on the falling edge of the internal lock signals, signifying that the detected input has changed timing. When high, the lock interrupts will trigger an interrupt on the rising edge of the internal lock signals, signifying that a lock has been achieved on the detected input.

Use Model

This section illustrates a likely usage scenario for the Xilinx Video Timing Controller core.

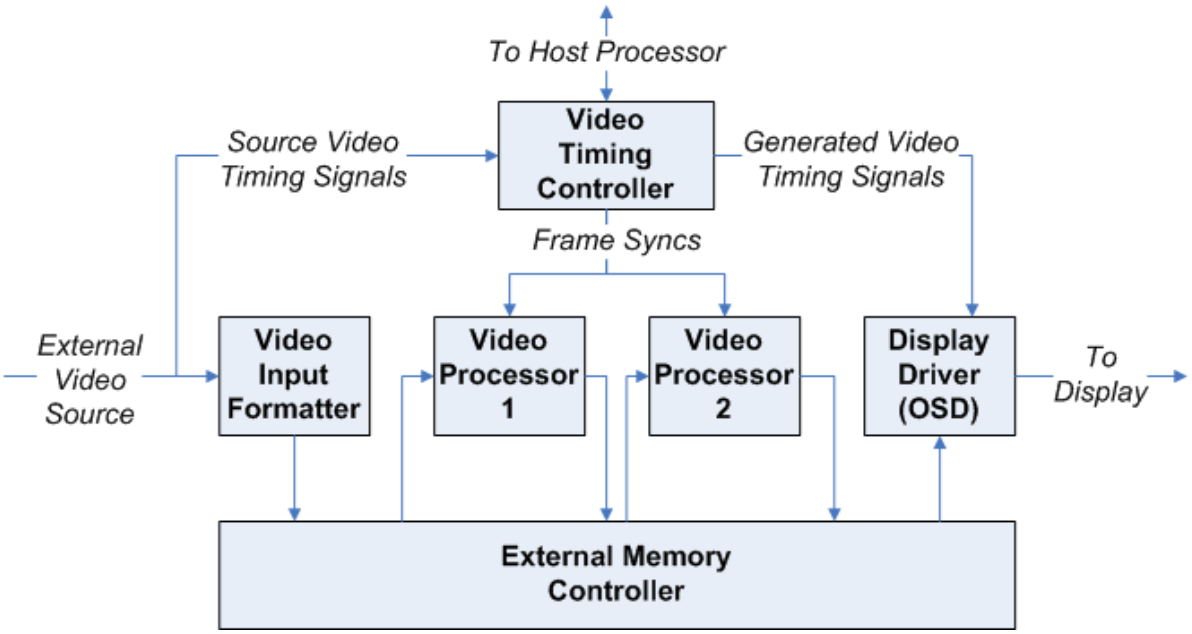

Figure 4-8: **Example Video Timing Controller Use Model**

Figure 4-8 shows four features of the Video Timing Controller being utilized in a video system:

1. Detection of the source video frame timing
2. Generation of video timing signals
3. Generation of two Frame Syncs to control the Video Processors
4. Connection to a Host Processor via the General Purpose Processor Interface

To detect the timing of the source video, the timing signals are connected to the Video Timing Controller Detection Module. Both the timing and the signal polarity of the timing signals are captured and easily read by the host processor.

Video timing signals are generated to control a display driver module and an external display. The timing of these output signals is controlled by the host processor. The Video Timing Controller can be configured in real-time to replicate the source video format or to slightly change the format on the output, for example, in cases where the input signals are positive polarity yet the display requires negative polarity synchronization signals. The Video Timing Controller can also be reconfigured in real-time to output a completely different format from the input source.

Two Frame Sync outputs are generated to control Video Processor 1 and Video Processor 2. These outputs could be used to control when Video Processor 2 starts processing relative to when Video Processor 1 starts processing. These Frame Syncs can be reconfigured in real-time as well.

The Video Timing Controller is connected to a Host Processor in this example. General Purpose Processor Interface allows for easy connection between status/control registers and the host processor. In addition, the Video Timing Controller interrupt output can also be used to synchronize the software with hardware events.

Clocking

The Video Timing Controller core has one clock ("video_clk_in") that is used to clock the entire core. This includes the AXI4-Lite interface and the core logic.

Resets

When configured with the GPP Interface, the Video Timing Controller core has one reset ("sclr") that is used for the entire core. The reset is active-High.

When configured with the AXI4-Lite Interface, the Video Timing Controller core has one reset ("s_axi_aresetn") that is used for the entire core. The reset is active-Low.

Protocol Description

For the pCore version of the Video Timing Controller core, the register interface is compliant with the AXI4-Lite interface.

Chapter 5

Constraining the Core

Required Constraints

There are no required constraints for the Video Timing Controller core.

Device, Package, and Speed Grade Selections

This core has not been characterized for low power devices.

Clock Frequencies

There are no specific clock frequency requirements for this core other than the Maximum Frequency discussed in the Performance section.

Clock Management

There is only one clock, clk, for the Video Timing Controller core. When using the AXI4-Lite EDK pCore interface of the Video Timing Controller core, the AXI Interconnect core will handle the asynchronous clock domain crossing from the video to the processor clock domain.

Clock Placement

There are no specific clock placement requirements for this core.

Banking

There are no specific Banking rules for this core.

Transceiver Placement

There are no Transceiver Placement requirements for this core.

I/O Standard and Placement

There are no specific I/O standards and placement requirements for this core.

Chapter 6

Detailed Example Design

Directory and File Contents

- Expected
  - reg_out.txt
  - timebase_out.txt
- Stimuli
  - reg_in.txt
  - timebase_in.txt
- Results
- src
  - v_tc_v3_0_u0.vhd
  - v_tc_v3_0_u0.xco
- tb_src
  - tb_v_tc_v3_0VHT.vhd
  - VHTSimPack.vhd
- isim_wave.wcfg - Waveform configuration file for iSim
- mti_wave.do - Waveform configuration for ModelSim
- run_isim.bat - Runscript for iSim in Windows OS
- run_isim.sh - Runscript for iSim in Linux OS
- run_mti.bat - Runscript for ModelSim in Windows OS
- run_mti.sh - Runscript for ModelSim in Linux OS

The Expected directory contains the pre-generated expected / golden data used by the test bench to compare to the actual output data. The Stimuli directory contains the pre-generated input data used by the test bench to simulate the core. The Results directory is where the actual simulation output data file are written. The src directory contains the .vhd and .xco files of the core. The .vhd file is a netlist generated using CORE Generator™ software. The .xco file can be used with the CORE Generator software to regenerate the netlist. The tb_src directory contains the top-level test bench design. This directory also contains other packages used by the test bench.

Demonstration Test Bench

This demonstration test bench is provided as a simple introductory package that enables core users to observe the core generated by the CORE Generator tool operating in a waveform simulator. The user is encouraged to observe core-specific aspects in the waveform, make simple modifications to the test conditions, and observe the changes in the waveform.

Simulation

Simulation using ModelSim for Linux:

- From the console, Type "source run_mti.sh".

Simulation using ModelSim for Windows:

- Double-click on "run_mti.bat" file.

Simulation using iSim for Linux:

- From the console, Type "source run_isim.sh".

Simulation using iSim for Linux:

- Double-click on "run_isim.bat" file.

Messages and Warnings

"Memory Collision Errors" have been observed when running some demonstration tests. The issue has been investigated and it has been determined that these errors can be safely ignored. This error message can be suppressed in ModelSim when the global "SIM_COLLISION_CHECK" option is set to "NONE".

XILINX®

Appendix A

Verification, Compliance, and Interoperability

Simulation

A highly parameterizable test bench was used to test the Video Timing Controller core. Testing included the following:

- Register accesses
- Processing of multiple frames of data
- Testing of various frame sizes including 1080p, 720p, and 480p
- Varying instantiations of the core
- Varying the polarity of input and output signals
- Varying the horizontal offset of the vertical timing signals
- Regenerating the input on the output
- Testing of various interrupts

Hardware Testing

The Video Timing Controller core has been tested in a variety of hardware platforms at Xilinx to represent a variety of parameterizations, including the following:

- A test design was developed for the core that incorporated a MicroBlaze™ processor, AXI4 Interconnect and various other peripherals. The software for the test system included live video input for the Video Timing Controller core. The Video Timing Controller, in addition to live video, was also connected in loopback allow the generator to feed the detector for a robust loopback test. Various tests could be supported by varying the configuration of the Timing Controller core or by loading a different software executable. The MicroBlaze processor was responsible for:
  - Initializing the appropriate input and output buffers in external memory.
  - Initializing the Video Timing Controller core.
  - Initializing the HDMI/DVI input and output cores for live video.
  - Launching the test.
  - Configuring the Video Timing Controller for various input frame sizes and checking the detection/generation loopback connection for correct video detection
  - Controlling the peripherals including the UART and AXI VDMAs.

XILINX®

Appendix B

Migrating

Migrating to the EDK pCore AXI4-Lite Interface

The Video Timing Controller v3.0 changed from the PLB processor interface to the EDK pCore AXI4-Lite interface. As a result, all of the PLB-related connections have been replaced with an AXI4-Lite interface. For more information, see the *AXI Reference Guide* at: www.xilinx.com/support/documentation/ip_documentation/ug761_axi_reference_guide.pdf

Parameter Changes in the XCO File

There are no parameter changes in the .xco file.

Port Changes

The Video Timing Controller v3.0 added the ability to operate on video frame sizes up to 8192 x 8192. Previous versions supported 4096 x 4096 maximum. If the maximum sizes of 8192 are selected, some GPP ports will be 13 bits wide where on previous versions of the core, these ports were 12 bits.

The Video Timing Controller v3.0 also added the ability to detect and generate vertical signals with a horizontal offset. In order to report the horizontal start cycle of these vertical signals, the Video Timing Controller v3.0 added the following new ports:

- gen_v0blank_hstart
- gen_v0blank_hend
- gen_v0sync_hstart
- gen_v0sync_hend
- det_v0blank_hstart
- det_v0blank_hend
- det_v0sync_hstart
- det_v0sync_hend

Functionality Changes

The Video Timing Controller v3.0 added the ability to operate on video frame sizes up to 8192 x 8192. Previous versions supported 4096 x 4096 maximum.

The Video Timing Controller v3.0 also added the ability to detect and generate vertical signals with a horizontal delay offset.

Special Considerations when Migrating to AXI

The Video Timing Controller v3.0 added the support for the AXI4-Lite interface with this version. When using the Video Timing COntroller v3.0, please note that the pcore name changed from "timebase" to "axi_vtc". All software driver functions, data structures and filenames also changed from a "xtimebase" prefix to "xvtc" prefix.

Appendix C

Debugging

When debugging, check the following:

- Verify that the clock pin, video_clk_in, is connected to the video clock source and is running.
- Verify that reset pin for the AXI4-Lite interface, s_axi_aresetn, is active low and has asserted and deasserted properly.
- Verify that reset pin for the General Purpose Processor interface, sclr, is active high and has asserted and deasserted properly.
- Can the Version register be read properly? See Table x for register definitions.
- Check the interrupt status register lock status (if using detector) or for specific errors. Check Table x for definitions of each bit.
- Verify that the vblank_in, hblank_in and active_video_in inputs are properly driven or that the vsync_in, hsync_in and active_video_in inputs are properly driven . These are the minimum required port connections for the XSVI input to perform detection.
- Verify that bits 0 and 1 of the Control register are set to "1". Bit 0 is the generator enable

See Solution Centers in Appendix E for information helpful to the debugging progress.

XILINX®

Appendix D

Application Software Development

Device Drivers

The Xilinx Video Timing Controller pCore includes a software driver written in the C Language that the user can use to control the Xilinx Video Timing Controller devices. A high-level API is provided and can be used without detailed knowledge of the Xilinx Video Timing Controller devices. Application developers are encouraged to use this API to access the device features. A low-level API is also provided in case applications prefer to access the devices directly through the system registers described in the previous section.

Table D-1 lists the files that are included with the Xilinx Video Timing Controller pCore driver and their description.

Table D-1: **Device Driver Source Files**

File Name	Description
xvtc.h	Contains all prototypes of high-level API to access all of the features of the Xilinx Video Timing Controller devices.
xvtc.c	Contains the implementation of high-level API to access all of the features of the Xilinx Video Timing Controller devices except interrupts.
xvtc_intr.c	Contains the implementation of high-level API to access interrupt feature of the Xilinx Video Timing Controller devices.
xvtc_sinit.c	Contains static initialization methods for the Xilinx Video Timing Controller device driver.
xvtc_g.c	Contains a template for a configuration table of Xilinx Video Timing Controller devices. This file is used by the high-level API and will be automatically generated to match the Video Timing Controller device configurations by Xilinx EDK/SDK tools when the software project is built.
xvtc_hw.h	Contains low-level API (that is, register offset/bit definition and register-level driver API) that can be used to access the Xilinx Video Timing Controller devices.
example.c	An example that demonstrates how to control the Xilinx Video Timing Controller devices using the high-level API.

Xilinx CORE Generator software will place all EDK pCore driver files in the “drivers” subdirectory located in the core output directory. The core output directory is given the same name as the component. For example, if the component name is set to “v_tc_v3_0_u0,” then the device driver source files will be located in the following directory:

```
<coregen project directory>/v_tc_v3_0_u0/drivers/vtc_v2_00_a/
```

The driver software should be copied to the user's <EDK_Project> / drivers directory or to a user pCores repository.

pCore API Functions

This section describes the functions included in the pcore Driver files generated for the Video Timing Controller pCore. The software API is provide to allow easy access to the registers of the pCore as defined in Table 2-2 in the Register Space section. To utilize the API functions provided, the following header files must be included in the user's C code:

- #include "xparameters.h"
- #include "xvtc.h"

The hardware settings of your system, including the base address of your Video Timing Controller core are defined in the xparameters.h file. The xvtc.h file provides the API access to all of the features of the Object Segmentation device driver. More detailed documentation of the API functions can be found by opening the file index.html in the pCore directory vtc_v2_00_a / doc / html / api.

Functions in xvtc.c

- int XVtc_CfgInitialize (XVtc *InstancePtr, XVtc_Config *CfgPtr, u32 EffectiveAddr)
  This function initializes a VTC device.
- void XVtc_Enable (XVtc *InstancePtr, u32 Type)
  This function enables a VTC device.
- void XVtc_Disable (XVtc *InstancePtr, u32 Type)
  This function disables a VTC device.
- void XVtc_SetPolarity (XVtc *InstancePtr, XVtc_Polarity *PolarityPtr)
  This function sets up the output polarity of a VTC device.
- void XVtc_GetPolarity (XVtc *InstancePtr, XVtc_Polarity *PolarityPtr)
  This function gets the output polarity setting used by a VTC device.
- void XVtc_SetSource (XVtc *InstancePtr, XVtc_SourceSelect *SourcePtr)
  This function sets up the source selecting of a VTC device.
- void XVtc_GetSource (XVtc *InstancePtr, XVtc_SourceSelect *SourcePtr)
  This function gets the source select setting used by a VTC device.
- void XVtc_SetSkipLine (XVtc *InstancePtr, int GeneratorChromaSkip)
  This function sets up the line skip setting of the Generator in a VTC device.
- void XVtc_GetSkipLine (XVtc *InstancePtr, int *GeneratorChromaSkipPtr)
  This function gets the line skip setting used by the Generator in a VTC device.
- void XVtc_SetSkipPixel (XVtc *InstancePtr, int GeneratorChromaSkip)
  This function sets up the pixel skip setting of the Generator in a VTC device.
- void XVtc_GetSkipPixel (XVtc *InstancePtr, int *GeneratorChromaSkipPtr)
  This function gets the pixel skip setting used by the Generator in a VTC device.
- void XVtc_SetDelay (XVtc *InstancePtr, int VertDelay, int HoriDelay)

This function sets up the Generator delay setting of a VTC device.

- void XVtc_GetDelay (XVtc *InstancePtr, int *VertDelayPtr, int *HoriDelayPtr)

  This function gets the Generator delay setting used by a VTC device.

- void XVtc_SetFSync (XVtc *InstancePtr, u16 FrameSyncIndex, u16 VertStart, u16 HoriStart)

  This function sets up the SYNC setting of a Frame Sync used by VTC device.

- void XVtc_GetFSync (XVtc *InstancePtr, u16 FrameSyncIndex, u16 *VertStartPtr, u16 *HoriStartPtr)

  This function gets the SYNC setting of a Frame Sync used by VTC device.

- void XVtc_SetGeneratorHoriOffset (XVtc *InstancePtr, XVtc_HoriOffsets *HoriOffsets)

  This function sets the VBlank / VSync Horizontal Offsets for the Generator in a VTC device.

- void XVtc_GetGeneratorHoriOffset (XVtc *InstancePtr, XVtc_HoriOffsets *HoriOffsets)

  This function gets the VBlank / VSync Horizontal Offsets currently used by the Generator in a VTC device.

- void XVtc_GetDetectorHoriOffset (XVtc *InstancePtr, XVtc_HoriOffsets *HoriOffsets)

  This function gets the VBlank / VSync Horizontal Offsets detected by the Detector in a VTC device.

- void XVtc_SetGenerator (XVtc *InstancePtr, XVtc_Signal *SignalCfgPtr)

  This function sets up VTC signal to be used by the Generator module in a VTC device.

- void XVtc_GetGenerator (XVtc *InstancePtr, XVtc_Signal *SignalCfgPtr)

  This function gets the VTC signal setting used by the Generator module in a VTC device.

- void XVtc_GetDetector (XVtc *InstancePtr, XVtc_Signal *SignalCfgPtr)

  This function gets the VTC signal setting used by the Detector module in a VTC device.

- void XVtc_GetVersion (XVtc *InstancePtr, u16 *Major, u16 *Minor, u16 *Revision)

  This function returns the version of a VTC device.

Functions in xvtc_sinit.c

- XVtc_Config * XVtc_LookupConfig (u16 DeviceId)

  XVtc_LookupConfig returns a reference to an XVtc_Config structure based on the unique device id, DeviceId.

Functions in xvtc_intr.c

- void XVtc_IntrHandler (void *InstancePtr)

  This function is the interrupt handler for the VTC driver.

- int XVtc_SetCallBack (XVtc *InstancePtr, u32 HandlerType, void *CallBackFunc, void *CallBackRef)

  This routine installs an asynchronous callback function for the given HandlerType:.

XILINX®

Appendix E

Additional Resources

Xilinx Resources

For support resources such as Answers, Documentation, Downloads, and Forums, see the Xilinx Support website at:

http://www.xilinx.com/support.

For a glossary of technical terms used in Xilinx documentation, see:

http://www.xilinx.com/support/documentation/sw_manuals/glossary.pdf.

Solution Centers

See the Xilinx Solution Centers for support on devices, software tools, and intellectual property at all stages of the design cycle. Topics include design assistance, advisories, and troubleshooting tips.

References

These documents provide supplemental material useful with this user guide:

- AXI Reference Guide.
- AMBA AXI4 Interface Protocol.

Technical Support

Xilinx provides technical support at www.xilinx.com/support for this LogiCORE™ IP product when used as described in the product documentation. Xilinx cannot guarantee timing, functionality, or support of product if implemented in devices that are not defined in the documentation, if customized beyond that allowed in the product documentation, or if changes are made to any section of the design labeled DO NOT MODIFY.

See the IP Release Notes Guide (XTP025) for more information on this core. For each core, there is a master Answer Record that contains the Release Notes and Known Issues list for the core being used. The following information is listed for each version of the core:

- New Features
- Resolved Issues
- Known Issues

Ordering Information

The Video Timing Controller core is provided under the Xilinx Core License Agreement and can be generated using the Xilinx® CORE Generator™ system. The CORE Generator system is shipped with Xilinx ISE® Design Suite software.

A simulation evaluation license for the core is shipped with the CORE Generator system. To access the full functionality of the core, including FPGA bitstream generation, a full license must be obtained from Xilinx.

Contact your local Xilinx sales representative for pricing and availability of additional Xilinx LogiCORE IP modules and software. Information about additional Xilinx LogiCORE IP modules is available on the Xilinx IP Center.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
10/19/2011	1.0	Initial Xilinx release.

Notice of Disclaimer