Summary

This application note describes how to create video systems for digital visual interface (DVI) input and video test pattern generator (TPG) input using Xilinx® native IPs in the Zynq™-7000 All Programmable (AP) SoC. The reference design configures video IPs with a processing frame rate of 60 Hz and 1920 x 1080 resolution, and displays system level bandwidth utilization and video latency as metrics. This application allows designers to create complex and high-performance video systems using the Zynq-7000 AP SoC, with one input from the DVI and one input from the TPG.

The design uses two AXI Video Direct Memory Access (VDMA) cores to simultaneously move four streams (two transmit video streams and two receive video streams), each in a 1920 x 1080 frame size at 60 f/s and 24 data bits per pixel (RGB). One VDMA is driven from a TPG with a video timing controller (VTC) block. The other VDMA is driven by incoming video from DVI-In. The data from both stream to memory map (S2MM) paths of the VDMA cores are buffered into DDR, read back by the MM2S channel of the AXI VDMA, and sent to a common on-screen display (OSD) core that multiplexes or overlays multiple video streams to a single output video stream. The output of the OSD core drives the onboard High-Definition Multimedia Interface (HDMI™) video display interface through the color space converter.

FreeRTOS and Linux are the recommended operating systems for the Zynq-7000 AP SoC. This application note demonstrates the application of FreeRTOS. The reference design is targeted for the ZC702 evaluation board.

FreeRTOS is a free operating system that consists of only a few files. It is easy to port, use, and maintain. FreeRTOS supports multiple threads or tasks, mutexes, semaphores, and software timers. In the reference design, the main application runs in one of the FreeRTOS threads while another thread is created to gradually change the transparency of the OSD to show the blending effect when option 2 is selected. FreeRTOS implements multiple threads by having the host program call a thread tick method at regular short intervals. The thread tick method switches tasks depending on priority or in a round-robin scheduling scheme that can be set in the Xilinx Software Development Kit (SDK) under Board Support Package (BSP) options. For more information, see: http://www.freertos.org/

Included Systems

The reference design is created with the Xilinx Platform Studio (XPS) in the Vivado™ System Edition 2012.4. XPS simplifies instantiating, configuring, and connecting IP blocks to form complex embedded systems. The design also includes software created with the Xilinx SDK. The software runs on the ARM® processor and implements control, status, and monitoring functions. Complete XPS and SDK project files are provided with this application note to allow the user to examine and rebuild this design or to use it as a template for starting a new design.

Note: This application note assumes the user has general knowledge of XPS, Vivado Design Suite, and Zynq-7000 architecture.

© Copyright 2013-2016 Xilinx, Inc. Xilinx, the Xilinx logo, Artix, ISE, Kintex, Spartan, Virtex, Vivado, Zynq, and other designated brands included herein are trademarks of Xilinx in the United States and other countries. ARM, AMBA, and Cortex are trademarks of ARM in the EU and other countries. HDMI and High-Definition Multimedia Interface are trademarks of HDMI Licensing LLC. All other trademarks are the property of their respective owners.
Xilinx native IPs can be used to effectively process various features of video systems. The AXI Interconnect, VDMA, and OSD IP blocks can form the core of video systems that handle desired frame rates and resolutions through multiple video streams and frame buffers sharing a common DDR3 SDRAM. AXI is a standardized IP interface protocol based on the ARM Advanced Microcontroller Bus Architecture (AMBA®) specification. The reference design uses the AXI4, AXI4-Lite, and AXI4-Stream AXI interfaces, as described in the AMBA AXI4 documentation [Ref 1]. These interfaces provide a common IP interface protocol framework for building the design.

The AXI Interconnect and DDR3 implement a high-bandwidth, multi-ported memory controller (MPMC) for use in applications where multiple devices share a common memory controller. This is a requirement in many video, embedded, and communications applications where data from multiple sources moves through a common memory device, typically DDR3 SDRAM on the processing system. The Zynq-7000 AP SoC includes a Processing System (PS) block and a Programmable Logic (PL) block as shown in Figure 1.

![Zynq-7000 AP SoC Block Diagram](image_url)

**Figure 1: Zynq-7000 AP SoC Block Diagram**

The PS is the hard block and the PL is configurable. The PS consists of two ARM Cortex™-A9 cores with D cache and I cache. There are four high-performance ports from the PS to the PL to access the DDR. There are two general purpose interconnects to control the peripherals in the PL. The design uses two HP ports (HP0 and HP2) to access the DDR memory via the VDMA. The GP0 port is used to configure the various IPs on the PL.
Figure 2 shows the reference design block diagram.

- AXI2XSVI: AXI4-Stream bridge to XSVI bridge
- RGB2YCC: RGB to YCbCr converter in 4:2:2 format
- VTC: Video Timing Controller core
- TPG: Test Pattern Generator core
- OSD: On-screen Display core

Table 1 lists the address map for the reference design.

<table>
<thead>
<tr>
<th>Peripheral</th>
<th>Version</th>
<th>Instance</th>
<th>Base Address</th>
<th>High Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>processing_system7</td>
<td>4.02</td>
<td>processing_system7</td>
<td>N/A</td>
<td>N/A</td>
</tr>
<tr>
<td>processing_system7-DDR</td>
<td>4.02</td>
<td>processing_system7</td>
<td>0x00000000</td>
<td>0x3FFFFFFF</td>
</tr>
<tr>
<td>axi_vtc</td>
<td>5.01</td>
<td>axi_vtc_0</td>
<td>0x7de11000</td>
<td>0x7de11fff</td>
</tr>
<tr>
<td>axi_vtc</td>
<td>5.01</td>
<td>axi_vtc_1</td>
<td>0x7de10000</td>
<td>0x7de10fff</td>
</tr>
<tr>
<td>axi_tpg</td>
<td>4.00</td>
<td>v_tpg_0</td>
<td>0x74220000</td>
<td>0x7422ffff</td>
</tr>
<tr>
<td>axi_osd</td>
<td>5.01</td>
<td>v_osd_0</td>
<td>0x74200000</td>
<td>0x7420ffff</td>
</tr>
<tr>
<td>axi_vdma</td>
<td>5.04</td>
<td>axi_vdma_0</td>
<td>0x7de13000</td>
<td>0x7de13fff</td>
</tr>
<tr>
<td>axi_vdma</td>
<td>5.04</td>
<td>axi_vdma_1</td>
<td>0x7de12000</td>
<td>0x7de12fff</td>
</tr>
<tr>
<td>axi_timer</td>
<td>1.03</td>
<td>axi_time_0</td>
<td>0x42800000</td>
<td>0x4280ffff</td>
</tr>
<tr>
<td>axi_iic</td>
<td>1.02</td>
<td>axi_iic_0</td>
<td>0x41600000</td>
<td>0x4160ffff</td>
</tr>
</tbody>
</table>

The AXI VDMA implements a high-performance, video-optimized DMA engine with frame buffering, and two-dimensional DMA features. The AXI VDMA transfers video data streams to and from memory and operates under dynamic software control or static configuration modes. A clock generator and processor system reset block supplies clocks and resets throughout the system. High-level control of the system containing I/O peripherals and processor support IP is provided by an embedded ARM processor that is available as a hard block on the Zynq-7000 AP SoC.
Figure 3 shows the hardware connectivity for the reference design. The hardware requirements for the reference design are:

- Xilinx ZC702 Rev 1.0 board
- One USB (Type A to Type B)
- JTAG platform USB cable
- Two HDMI to DVI converter cables
- Display monitors supporting 1920 x 1080 at 60 f/s
- FMC DVI card for feeding in DVI-In (AES-FMC-DVI-G)
If a JTAG platform USB cable is used, the board switch settings for SW10 should be as shown in Figure 4. If an onboard USB JTAG cable is used, the SW10 settings should be the reverse of what is shown.

**Figure 4: Board Switch Setting**

---

**Software Requirements**

The software requirements for the reference design are:

- Vivado Design Suite 14.4
- SDK14.4

**Reference Design Specifics**

The reference design includes the Zynq-7000 AP SoC, MDM, LMB block RAM, AXI_INTERCONNECT, Clock Generator, PROC_SYS_RESET, AXI_UARTLITE, AXI_IIC, AXI_INTC, AXI_VTC, AXI_TPG, AXI_VDMA AXI_OSD, and HDMI_Interface IP cores. Onboard DDR that is accessible through the PS hard memory controller is used to store the frame.
**Hardware System Specifics**

This section describes the high-level features of the reference design, including how to configure the main IP blocks. Information about useful IP features, performance/area trade-offs, and configuration is also provided. The reference design is a video system, but the principles used to optimize system performance are applicable to a wide range of high-performance AXI systems. For information about AXI system optimization and design trade-offs, see the *AXI Reference Guide* [Ref 2].

**AXI Interconnect**

This design contains three AXI Interconnects, each aimed to balance throughput, area, and timing considerations (see the *LogiCORE IP AXI Interconnect (v1.06a) Data Sheet* [Ref 3]).

- The AXI_MM0 instance is used as a bridge between the VDMA0 and Zynq-7000 AP SoC HP0. The Zynq-7000 AP SoC HP0 is connected to the DDR. This bridge takes data from the VDMA0 (which takes input from the TPG) and stores it in the DDR, and then takes the same data and feeds it to the OSD via the VDMA0 MM2S channel. The data width of the interconnect is 64 bits and it runs at 150 MHz.

- The AXI_MM1 instance is used as a bridge between the VDMA1 and Zynq-7000 AP SoC HP1. The Zynq-7000 AP SoC HP1 is connected to the DDR. This bridge takes data from the VDMA1 (which takes input from the DVI) and stores it in the DDR, and then takes the same data and feeds it to the OSD via the VDMA1 MM2S channel. The data width of the interconnect is 64 bits and it runs at 150 MHz.

- The AXI4-Lite is generally optimized for area and is used by the processor to access slave registers. The AXI4-Lite is connected to the GP0 of the Zynq-7000 AP SoC device and is used to control the various IPs via the register map.

**AXI VDMA Instances**

The AXI VDMA core provides video read and write transfers from the AXI4 domain to the AXI4-Stream domain, and vice versa. The AXI VDMA core provides high-speed data movement between system memory and AXI4-Stream based target video IP. The AXI VDMA core incorporates video specific functionality (Gen-Lock and Frame Sync) for fully synchronized frame DMA operations and 2D DMA transfers. In addition to synchronization, frame store numbers and scatter gather or register direct mode operations are available for ease-of-control by the central processor.

Initialization, status, and management registers in the AXI VDMA core are accessed through an AXI4-Lite slave interface.

The reference design uses two instances of AXI VDMA, using the interfaces AXI4 MM2S, AXI4 S2MM, AXI4-Stream MM2S, and AXI4-Stream S2MM.

The 64-bit wide MM2S and S2MM interfaces from the AXI VDMA instance are connected to the AXI_MM0 and AXI_MM1 instance of the AXI Interconnect.

For maximum throughput for the AXI VDMA instances, the maximum burst length is set to 256. In addition, the master interfaces have a read and write issuance of four and a read and write FIFO depth of 512 to maximize throughput. These settings are conservatively set; for other applications and use cases, see the guidelines in the *AXI Reference Guide* [Ref 2].
AXI Video Timing Controller

The AXI Video Timing Controller (VTC) core 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. The output side of the core generates the horizontal and vertical blanking and synchronization pulses used in a standard video system, including support for programmable pulse polarity.

The AXI VTC core contains an AXI4-Lite Interface to access slave registers from a processor. For more information about the AXI VTC core, see the LogiCORE IP Video Timing Controller Product Guide [Ref 4]. In this design, two AXI VTC instances are used without detection. The first instance is used for the video input portion of the video pipeline. The second instance is used for the AXI OSD, which is the read portion of the video pipeline.

The AXI VTC v5.01 core is provided under license and can be generated using the CORE Generator™ tool v14.4 or higher.

AXI Test Pattern Generator

The AXI Test Pattern Generator (TPG) core contains an AXI4-Lite Interface to access slave control registers from a processor. In the reference design, the video traffic to DDR3 memory is generated by a TPG core. The TPG block can generate several video test patterns that are commonly used in the video industry for verification and testing. In the design, the TPG core is used as a replacement for other video IP because only the amount of traffic generated to demonstrate the performance of the system is of interest. The control software demonstrates the generation of 11 possible video patterns, such as color bars, horizontal and vertical burst patterns, flat colors, and zone plates. For any selected test pattern, the amount of data generated is the same for a particular resolution and frame rate.

Several operating modes are accessible through software control. In this application note, the AXI TPG core always generates one of 11 possible test patterns through user input. These patterns are for testing purposes only and are not calibrated for broadcast industry standards.

AXI On-Screen Display

The AXI On-Screen Display (OSD) core provides a flexible, video-processing block for alpha blending, compositing up to eight independent layers, and generating simple text and graphics capable of handling images up to 4K x 4K of various image formats and bit width (for more information, see the LogiCORE IP Video On-Screen Display Product Guide [Ref 5]. In this application note, the OSD core is configured to display two video layers only, but can be configured for multiple video streams as separate display layers.

The AXI OSD core contains an AXI4-Lite interface to access the slave registers from a processor. For more information about the AXI OSD core, see the LogiCORE IP Video On-Screen Display Product Guide [Ref 5]. The AXI OSD v5.01 core is provided under the Sign Once IP site license and can be generated using the CORE Generator tool. 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.
Software Applications

The application is ported to FreeRTOS and works as a child thread for various configurations. The software application starts the video pipeline and allows the user to select either the DVI input or the TPG. Because the design contains an OSD, the user can extend this application note to multiple video pipelines and display separate layers or alpha blend all layers on the LCD screen.

Application-level software for controlling the system is written in C using the provided drivers for each IP. The programmer’s model for each IP describes the particular API used by the drivers. Alternatively, application software can be written to use the IP control registers directly and handle the interrupts at the application level, but using the provided drivers and a layer of control at the application level is a more convenient option.

The application software in the reference design performs the following actions in the order listed:

- Software application initializes the system, caches, UART, and VDMA
- HDMI port through the IIC interface
- TPG instances are configured to generate the various patterns
- AXI VDMA instance is started by configuring VDMA read and write channels to begin the transfers for the VDMA instances
- VTC instances are started for 1080p60 timing configuration
- AXI OSD is configured for above resolution
- TPG instance in the design is configured to write tartan bars test patterns

After the initial setup sequence, the user can enter an option with the HyperTerminal to select the input source. If the selected source is TPG, the tartan bars are displayed on the screen. The video patterns and DVI inputs are routed to the HDMI port through the color space (RGB to YCbCr) converter. For multiple video pipelines, OSD registers can also be configured to blend input channels to the required level. Different values are specified for the alpha blending register for each layer to show all layers on the LCD screen at the same time.

Set up and configure the hardware as follows:

1. Connect a USB cable from the host PC to the USB JTAG port. Ensure the appropriate device drivers are installed.
2. Connect a second USB cable from the host PC to the USB UART port. Ensure the USB-UART drivers described previously are installed.
3. Connect the ZC702 HDMI connector to a video monitor capable of displaying frame rates up to a 120 Hz video signal.
4. Connect power supply cable.
5. Set power to On.
6. Start a terminal program (for example, HyperTerminal) on the host PC set for:
   a. Baud Rate: 115200
   b. Data Bits: 8
   c. Parity: None
   d. Stop Bits: 1
   e. Flow Control: None
Running the Reference Design

This section describes running the reference design using the pre-built bitstream and the compiled software application.

To run the design:

1. In a command shell or terminal window, go to the \ready_for_download\ directory:
   
   \cd <unzip dir>/video_streaming_xapp/ready_for_download

2. Run the Xilinx Microprocessor Debugger (XMD) by entering the \xmd\ command.

3. Enter \source xmd.tcl\.

Results from Running Hardware and Software

The LCD monitor connected to the ZC702 board displays a color bar pattern and the HyperTerminal screen displays the output shown in Figure 5.

The user can choose '0' to select TPG as the input source or '1' to select DVI as the input source. The '2' option displays the gradual blending of two layers on the monitor.

*Note:* The design was tested on the P2210T Dell monitor only.

Building Hardware

This section describes building the hardware design.

1. Unzip the provided video_streaming_xapp directory and copy the init.tcl file to the home directory in the .Xilinx\Vivado folder. The init.tcl file includes commands to enable the Zynq-7000 AP SoC in the Vivado tools.

2. In the Vivado Design Suite, open video_streaming_xapp/xapp.xpr.

3. In the left panel, click Generate Bitstream.

*Note:* The system_stub.bit file is generated in the video_streaming_xapp/xapp.runs/impl_1 directory.
Compiling Software and Running Design

1. Start SDK. In Linux, enter `xsdk` to start SDK.
2. In the Workspace Launcher, unzip and select this workspace:
   `video_streaming_xapp/xapp.sdk/app`
3. In SDK, select `Xilinx Tools > Repository`. Unzip and add this path as the repository:
   `video_streaming_xapp/xapp.sdk/app/repository`
4. Click `OK`.
5. Click `Project > Clean` and build the entire project.
   **Note:** Make sure the armgcc compiler path is set in the system environment. The BSP and software applications are compiled in this step and the process takes a few minutes.
6. At this point, the user can modify existing software applications and create new software applications in SDK.
7. The ELF file is generated at:
   `video_streaming_xapp/xapp.sdk/app/xapp/Debug/xapp.elf`

Running the Hardware and Software

1. Open the XMD terminal.
2. Change the path in `xmd.tcl` to the user path for `xapp.elf` and `system_stub.bit`.
3. Change the path of `ps7_init.tcl` to the user path on the PC. Usually this file is located at `video_streaming_xapp/xapp.sdk/app/xapp_hw_platform`.
4. Go to the folder where `xmd.tcl` is located.
5. Run `source xmd.tcl`.
   **Note:** The baud rate of the hyperterminal should be set to 115200 and the FMC DVI card must be in the slot as shown in Figure 3, page 4.

Reference Design

The reference design has been fully verified and tested on hardware. The design includes details on the various functions of the different modules. The interface has been successfully placed and routed at 150 MHz on the main AXI Interfaces to the memory controller using the Vivado tools.

The reference design files for this application note can be downloaded at:


Table 2 shows the reference design matrix.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Developer name</td>
<td>Dinesh Kumar</td>
</tr>
<tr>
<td>Target devices (stepping level, ES, production, speed grades)</td>
<td>Zynq-7000 (ZC702) AP SoC</td>
</tr>
<tr>
<td>Source code provided</td>
<td>Yes</td>
</tr>
<tr>
<td>Source code format</td>
<td>VHDL/Verilog (some sources encrypted)</td>
</tr>
<tr>
<td>Design uses:</td>
<td></td>
</tr>
</tbody>
</table>
  - Code |
  - IP from existing Xilinx application note |
  - Reference designs |
  - CORE Generator (or third party software) |
  - Reference design fuses core generated from EDK/XPS (part of Vivado Design Suite v2012.4) |

Implementation
Utilization and Performance

The device resource utilization information in Table 3 is from the Design Summary tab in the Vivado tool. The utilization information is approximate due to cross-boundary logic optimizations and logic sharing between modules.

Table 2: Reference Design Matrix (Cont’d)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthesis software tools/version used</td>
<td>Vivado Design Suite 2012.4</td>
</tr>
<tr>
<td>Implementation software tools/versions used</td>
<td>Vivado Design Suite 2012.4</td>
</tr>
<tr>
<td>Static timing analysis performed</td>
<td>Y (passing timing in PAR/TRCE)</td>
</tr>
</tbody>
</table>

Hardware Verification

| Hardware verified | Y                        |
| Hardware platform used for verification  | ZC702 board               |

Table 3: Device and Utilization

<table>
<thead>
<tr>
<th>Device</th>
<th>Package</th>
<th>Slice Registers</th>
<th>Slice LUTs</th>
<th>Bonded IOB</th>
<th>RAMB36E1s</th>
<th>RAMB18E1s</th>
</tr>
</thead>
<tbody>
<tr>
<td>xc7z020</td>
<td>clg484-1</td>
<td>20,726 (19.47%)</td>
<td>17,356 (32.62%)</td>
<td>54 (27%)</td>
<td>33(23.57%)</td>
<td>13(4.64%)</td>
</tr>
</tbody>
</table>

References

2. UG761, AXI Reference Guide
3. DS768, LogiCORE IP AXI Interconnect (v1.06a) Data Sheet
5. PG010, LogiCORE IP Video On-Screen Display Product Guide
6. XAPP741, Designing High-Performance Video Systems in 7 Series with the AXI Interconnect Application Note

Revision History

The following table shows the revision history for this document.

<table>
<thead>
<tr>
<th>Date</th>
<th>Version</th>
<th>Description of Revisions</th>
</tr>
</thead>
<tbody>
<tr>
<td>02/22/2013</td>
<td>1.0</td>
<td>Initial Xilinx release.</td>
</tr>
<tr>
<td>03/04/2016</td>
<td>2.0</td>
<td>Document is obsolete.</td>
</tr>
</tbody>
</table>
Notice of Disclaimer

The information disclosed to you hereunder (the “Materials”) is provided solely for the selection and use of Xilinx products. To the maximum extent permitted by applicable law: (1) Materials are made available "AS IS" and with all faults, Xilinx hereby DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and (2) Xilinx shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under, or in connection with, the Materials (including your use of the Materials), including for any direct, indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or Xilinx had been advised of the possibility of the same. Xilinx assumes no obligation to correct any errors contained in the Materials or to notify you of updates to the Materials or to product specifications. You may not reproduce, modify, distribute, or publicly display the Materials without prior written consent. Certain products are subject to the terms and conditions of the Limited Warranties which can be viewed at http://www.xilinx.com/warranty.htm; IP cores may be subject to warranty and support terms contained in a license issued to you by Xilinx. Xilinx products are not designed or intended to be fail-safe or for use in any application requiring fail-safe performance; you assume sole risk and liability for use of Xilinx products in Critical Applications: http://www.xilinx.com/warranty.htm#critapps.

Automotive Applications Disclaimer

XILINX PRODUCTS ARE NOT DESIGNED OR INTENDED TO BE FAIL-SAFE, OR FOR USE IN ANY APPLICATION REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS APPLICATIONS RELATED TO: (I) THE DEPLOYMENT OF AIRBAGS, (II) CONTROL OF A VEHICLE, UNLESS THERE IS A FAIL-SAFE OR REDUNDANCY FEATURE (WHICH DOES NOT INCLUDE USE OF SOFTWARE IN THE XILINX DEVICE TO IMPLEMENT THE REDUNDANCY) AND A WARNING SIGNAL UPON FAILURE TO THE OPERATOR, OR (III) USES THAT COULD LEAD TO DEATH OR PERSONAL INJURY. CUSTOMER ASSUMES THE SOLE RISK AND LIABILITY OF ANY USE OF XILINX PRODUCTS IN SUCH APPLICATIONS.