**Summary**

This application note demonstrates the creation of video systems by using Xilinx native video IP cores such as AXI Video Direct Memory Access (VDMA), Video Timing Controller (VTC), test pattern generator (TPG), and the DDR3 memory controller to process configurable frame rates and resolutions in Kintex™-7 FPGAs. The reference design focuses on run time configuration of an on-board clock generator for a video pixel clock and video IPs for running selected combinations of video resolution and frame rate.

The system also displays system-level bandwidth utilization and video latency for each combination of frame rate and resolution, which are common metrics used by system designers. This application note discusses the configuration of each video IP in detail, helping designers to make effective use of video IPs for processing various video features.

The video datapath of the current design includes video TPG, VDMA (streaming to memory mapped), DDR, VDMA (memory mapped to streaming), and video on-screen display (OSD) IP blocks. Each of these video IP blocks are configured dynamically to process various combinations of frame rate and resolution. An onboard configurable clock generator (Si570) is used to generate the video clock for the desired refresh rate and resolution. The VDMA is driven from a TPG with a VTC block to set up the necessary video timing signals. Data read by the AXI VDMA is sent to the OSD. The output of the OSD core drives an onboard High Definition Media Interface (HDMI) video display through the color space converter.

A performance monitor block is added to capture the performance of DDR memory. Video frame data moved by the AXI VDMA blocks are buffered through a shared DDR3 SDRAM and are controlled by a MicroBlaze™ processor.

The reference design is targeted for the Kintex-7 FPGA XC7K325TFFG900-1 on the Xilinx KC705 evaluation board (Revision C).

---

**Included Systems**

The reference design is created and built using version 13.4 of the Xilinx Platform Studio (XPS) tool, which is part of the ISE® Design Suite: Embedded Edition. XPS helps simplify the task of instantiating, configuring, and connecting IP blocks together to form complex embedded systems. The design also includes software built using the Xilinx Software Development Kit (SDK). The software runs on the MicroBlaze processor subsystem and implements control, status, and monitoring functions. Complete XPS and SDK project files are provided with the reference design to allow the user to examine and rebuild the design or use it as a template for starting a new design.

---

**Introduction**

Xilinx IP cores implement various functions for many video applications. The use of AXI Interconnect, the Memory Interface Generator (MIG) tool, VDMA, onboard configurable clock generator, VTC, and OSD IP blocks can form the core of video systems capable of handling the various combinations of frame rates and resolution. AXI is a standardized IP interface protocol based on the Advanced Microcontroller Bus Architecture (AMBA®) specification. The AXI interfaces used in the reference design consist of AXI4, AXI4-Lite, and AXI4-Stream interfaces as described in the AMBA AXI4 specifications [Ref 1]. These interfaces provide a common IP interface protocol framework around which to build the design.
AXI VDMA implements a high-performance, video-optimized DMA engine with frame buffering, and two-dimensional (2D) DMA features. Together, the AXI Interconnect and AXI MIG implement a multi-port memory controller (MPMC) for sharing data from multiple sources through a common memory device, typically DDR3 SDRAM. The AXI VDMA transfers the buffered 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. An onboard SI570 clock generator is used to dynamically change the video pixel clock frequency at run time for varying video frame rates. High-level control of the system containing I/O peripherals and processor support IP is provided by an embedded MicroBlaze processor. To optimize the system to balance performance and area, multiple AXI Interconnect blocks are used to implement segmented or cascaded AXI Interconnect networks with each AXI Interconnect block individually tuned and optimized.

Hardware Requirements

The hardware requirements for this reference system are:

- Xilinx KC705 evaluation board (revision C or D)
- One USB Type-A to Mini-B 5-pin cable
- One USB Type-A to Micro-B 5-pin cable
- High-quality HDMI cable (colors are not displayed properly otherwise)
- Display monitors supporting configurable resolutions (up to 75 frames/sec, tested on a Dell P2210T monitor)

The installed software tool requirements for building and downloading this reference system are:

- Xilinx Platform Studio 13.4
- ISE Design Suite 13.4
- SDK 13.4

Reference Design Specifics

In addition to the MicroBlaze processor, the reference design includes these cores:

- MDM
- LMB block RAM
- AXI_INTERCONNECT
- Clock Generator
- PROC_SYS_RESET
- AXI_UARTLITE
- AXI IIC
- AXI_INTC
- AXI_V7_DDRX
- Video Timing Controller
- AXI_TPG
- AXI_VDMA
- AXI_PERFORMANCE_MONITOR
- AXI_OSD
- HDMI_Interface IP cores

An onboard configurable clock generator (SI570) is used to supply the video clock to the design.
Figure 1 shows a block diagram of the reference system.

![Reference System Block Diagram](image)

**Figure 1:** Reference System Block Diagram

In Figure 1, the PLB-based VTC is connected to the system through the AXI-to-PLB bridge. Table 1 shows the address map of the reference system.

**Table 1: Reference System Address Map**

<table>
<thead>
<tr>
<th>Peripheral</th>
<th>Instance</th>
<th>Base Address</th>
<th>High Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>axi_intc</td>
<td>microblaze_0_intc</td>
<td>0x81200000</td>
<td>0x8120FFFF</td>
</tr>
<tr>
<td>lmb_bram_if_cntlr</td>
<td>microblaze_0_ibram_ctrl</td>
<td>0x00000000</td>
<td>0x00001FFF</td>
</tr>
<tr>
<td>lmb_bram_if_cntlr</td>
<td>microblaze_0_ibram_ctrl</td>
<td>0x00000000</td>
<td>0x00001FFF</td>
</tr>
<tr>
<td>mdm</td>
<td>debug_module</td>
<td>0x74800000</td>
<td>0x7480FFFF</td>
</tr>
<tr>
<td>axi_uartlite</td>
<td>rs232_uart_1</td>
<td>0x80600000</td>
<td>0x8060FFFF</td>
</tr>
<tr>
<td>axi_7series_ddrx</td>
<td>ddr3_sdram</td>
<td>0x40000000</td>
<td>0x5FFFFFFF</td>
</tr>
<tr>
<td>Video Timing Controller</td>
<td>timebase_0</td>
<td>0x73820000</td>
<td>0x7382FFFF</td>
</tr>
<tr>
<td>Video Timing Controller</td>
<td>timebase_1</td>
<td>0x73800000</td>
<td>0x7380FFFF</td>
</tr>
<tr>
<td>axi_tpg</td>
<td>axi_tpg_0</td>
<td>0xBEE00000</td>
<td>0xBEE0FFFF</td>
</tr>
<tr>
<td>perf_axi_mm</td>
<td>perf_axi_mm_0</td>
<td>0x90000000</td>
<td>0x9000FFFF</td>
</tr>
<tr>
<td>axi_osd</td>
<td>axi_osd_0</td>
<td>0x73a00000</td>
<td>0x73a0FFFF</td>
</tr>
<tr>
<td>axi_vdma</td>
<td>axi_vdma_0</td>
<td>0xbe200000</td>
<td>0xbe20FFFF</td>
</tr>
</tbody>
</table>
Table 2 describes the detailed configurable frame rates and resolutions.

### Table 2: Configuration Frame Rates and Resolution

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Horizontal Active Video</th>
<th>Vertical Active Video</th>
<th>Video Clock Frequency (MHz)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>640</td>
<td>480</td>
<td>25.2</td>
</tr>
<tr>
<td>2</td>
<td>720</td>
<td>480</td>
<td>27.2</td>
</tr>
<tr>
<td>3</td>
<td>800</td>
<td>600</td>
<td>39.7</td>
</tr>
<tr>
<td>4</td>
<td>1024</td>
<td>768</td>
<td>64.9</td>
</tr>
<tr>
<td>5</td>
<td>1280</td>
<td>720</td>
<td>74.2</td>
</tr>
<tr>
<td>6</td>
<td>1280</td>
<td>1024</td>
<td>108.2</td>
</tr>
<tr>
<td>7</td>
<td>1920</td>
<td>1080</td>
<td>148.5</td>
</tr>
<tr>
<td>8</td>
<td>1600</td>
<td>1200</td>
<td>162</td>
</tr>
</tbody>
</table>

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 other configuration information is also provided. This information is applied to a video system, but the principles can be applied to a wide range of embedded systems.

This application note assumes the user has some general knowledge of XPS. See *EDK Concepts, Tools, and Techniques: A Hands-On Guide to Effective Embedded System Design* [Ref 3] for more information about the XPS tools.

### Video-Related IP

The reference design implements a video system that can run the desired video resolutions and frame rates. Each picture consists of four bytes per pixel to represent an upper bound for high-quality video streams such as RGBA with alpha channel information. The video system and its IPs run with a total aggregate read/write bandwidth as described in Table 3. The video traffic of the system is generated by the TPG block and displayed by the OSD core.

### Table 3: Frame Resolution Versus Bandwidth of Video Streams

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Horizontal Active Video</th>
<th>Vertical Active Video</th>
<th>Frame Rate</th>
<th>Bandwidth/One Stream (MB)</th>
<th>Bandwidth/Two Streams (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>640</td>
<td>480</td>
<td>60</td>
<td>73.728</td>
<td>147.45</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75</td>
<td>92.16</td>
<td>184.32</td>
</tr>
<tr>
<td>2</td>
<td>720</td>
<td>480</td>
<td>60</td>
<td>82.94</td>
<td>165.88</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75</td>
<td>103.68</td>
<td>207.36</td>
</tr>
<tr>
<td>3</td>
<td>800</td>
<td>600</td>
<td>60</td>
<td>115.2</td>
<td>230.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75</td>
<td>144</td>
<td>288</td>
</tr>
<tr>
<td>4</td>
<td>1024</td>
<td>768</td>
<td>60</td>
<td>188.743</td>
<td>377.486</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75</td>
<td>235.929</td>
<td>471.858</td>
</tr>
<tr>
<td>5</td>
<td>1280</td>
<td>720</td>
<td>60</td>
<td>221.184</td>
<td>442.368</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75</td>
<td>276.48</td>
<td>552.96</td>
</tr>
<tr>
<td>6</td>
<td>1280</td>
<td>1024</td>
<td>60</td>
<td>314.572</td>
<td>629.144</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75</td>
<td>393.2</td>
<td>786.4</td>
</tr>
</tbody>
</table>
This application note demonstrates a video system using two high-definition video streams of configurable resolutions and frame rates. Video systems typically include a source, some internal processing, and a destination. There can be multiple stages internally using a variety of IP modules. The canonical video system in Figure 2 shows that most video systems consist of input, pre-processing, main processing, post-processing, and output stages. Many of the video stages illustrated require memory access at video rates. Video data goes in and out of memory according to the requirements of the internal processing stages. In this application note, a TPG creates the internal IP block memory traffic to simulate typical conditions.

### Onboard Configurable Clock Generator

In the reference design, clocks for the processor, DDR memory, and other slaves of the processor are derived from the internal mixed-mode clock manager (MMCM) of the FPGA. After the design is implemented, these frequencies cannot be changed at run time. However, to dynamically change the resolution and frame rates of the system, the video pixel clock must be configurable at run time. The onboard configurable clock generator (SI570), which is connected through the IIC bus, is used to generate the video pixel clock at run time. After selecting an option through HyperTerminal, the SI570 is configured to generate the desired video pixel clock at run time.

### AXI Interconnect

This design contains two AXI Interconnects, each targeted to balance throughput, area, and timing considerations (see LogiCORE IP AXI Interconnect (v1.05) Data Sheet [Ref 4]). The AXI_MM instance is used for high-speed masters and slaves that include high throughput and high \( F_{\text{MAX}} \) optimizations. The AXI_Lite is generally optimized for area and is used for the processor to access slave registers of various IP cores. If multiple video pipelines are added to

---

**Table 3: Frame Resolution Versus Bandwidth of Video Streams (Cont’d)**

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Horizontal Active Video</th>
<th>Vertical Active Video</th>
<th>Frame Rate</th>
<th>Bandwidth/One Stream (MB)</th>
<th>Bandwidth/Two Streams (MB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>1920</td>
<td>1080</td>
<td>60</td>
<td>497.664</td>
<td>995.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75</td>
<td>622.08</td>
<td>1244.16</td>
</tr>
<tr>
<td>8</td>
<td>1600</td>
<td>1200</td>
<td>60</td>
<td>460.8</td>
<td>921.6</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>75</td>
<td>576</td>
<td>1152</td>
</tr>
</tbody>
</table>

**Notes:**
1. The bandwidth for one stream is calculated as Vertical Active Video x Horizontal Active Video x Bytes per Pixel x Frame Rate.

**Figure 2:** Typical Video System
the system, multiple interconnects can be cascaded as described in *Designing High-Performance Video Systems with the AXI Interconnect* [Ref 5].

**AXI VDMA**

The AXI VDMA core is designed to provide video read/write transfer capabilities from the AXI4 memory-mapped domain to the AXI4-Stream domain, and vice versa. The AXI VDMA provides high-speed data movement between system memory and AXI4-Stream based target video IP. The AXI VDMA core incorporates video-specific functionality, i.e., Gen-Lock and Frame Sync, for fully synchronized frame DMA operations and 2D DMA transfers. In addition to synchronization, frame store numbers and register direct or scatter gather 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.

This design uses a single instance of AXI VDMA with three frame buffers using the AXI4 MM2S, AXI4 S2MM, AXI4-Stream MM2S, and AXI4-Stream S2MM interfaces.

AXI VDMA can be configured to process multiple resolutions through specifying horizontal and vertical resolutions in its register space. The 32-bit wide MM2S and S2MM interfaces from the AXI VDMA instance are connected to the AXI_MM instance of the AXI Interconnect. The masters run off the video clock (onboard clock generator) which requires asynchronous clock converters to the 200 MHz AXI Interconnect core frequency. Upsizers in the AXI Interconnect are used to convert 32-bit transactions from the AXI VDMA to the transaction width of the AXI Interconnect core.

In addition, line buffers inside the AXI VDMA for the read and write sides are set to 1K deep, and the store and forward feature of the AXI VDMA are enabled on both channels to improve system performance. See *LogiCORE IP AXI Video Direct Memory Access (axi_vdma) (v5.00.a) Data Sheet* [Ref 6] for more information.

The following workaround is needed if C_PRMRY_IS_ACLK_ASYNC to 1 AND two or more clocks are being driven from the same source:

1. Right-click on the core instance and select **Make This IP Local** to make the pcore local to the XPS project.
2. Navigate to the pcores/axi_vdma_v5_00_a/data/ directory.
3. Open the axi_vdma_2_1_0.tcl file.
4. Comment out any lines from 77 to 136 in the TCL file that incorrectly constrain signals in the same clock domain. For example, if the core is set to asynchronous mode (C_PRMRY_IS_ACLK_ASYNC=1) and m_axi_mm2s_aclk and s_axi_lite_aclk use the same clock source, comment out these timing ignore (TIG) constraints:
   
   ```tcl
   puts $outputFile "TIMESPEC
   TS_${instancetype}_from_s_axi_lite_aclk_to_m_axi_mm2s_aclk = FROM
   "s_axi_lite_aclk" TO "m_axi_mm2s_aclk" TIG;"
   puts $outputFile "TIMESPEC
   TS_${instancetype}_from_m_axi_mm2s_aclk_to_s_axi_lite_aclk = FROM
   "m_axi_mm2s_aclk" TO "s_axi_lite_aclk" TIG;"
   ```
5. Save the file.
6. In XPS, select **Project** and click **Rescan User Repositories**.

**MicroBlaze Processor ICache and DCache**

The MicroBlaze processor ICache and DCache masters run at 100 MHz because the MicroBlaze processor runs a software application from main memory that sets up and monitors the video datapath. The 100 MHz clock setting ensures that synchronous integer ratio clock converters in the AXI Interconnect can be used, which offers lower latency and less area than
asynchronous converters. Setting the MicroBlaze processor to run at 100 MHz synchronous improves design timing closure and reduces area.

**AXI 7 Series DDRX**

The single slave connected to the Interconnect is the AXI 7 series DDR3 memory controller (a block that integrates MIG into XPS). The memory controller’s AXI Interface is 256 bits wide, running at 200 MHz, and disables narrow burst support for optimal throughput and timing. See the 7 Series FPGAs Memory Interface Solutions User Guide [Ref 7] for more information about the memory controller.

**AXI Interconnect (AXI_Lite)**

The MicroBlaze processor data peripheral (DP) interface master writes and reads to all AXI4-Lite slave registers in the design for control and status information.

A single AXI4-Lite interconnect is sufficient for one video pipeline design. It is possible to extend the number of interconnects with AXI2AXI connectors if there are more than 16 AXI4-Lite slaves.

The AXI_Lite interconnect block is configured for shared access mode because high throughput is not required in this portion of the design. Therefore, area can be optimized over performance on these interconnect blocks. Also, this interconnect is clocked at 50 MHz to ensure that synchronous integer ratio clock converters in the AXI Interconnect can be used, which offer lower latency and less area than asynchronous clock converters.

**AXI_Lite Interconnect**

The slaves on the AXI_Lite Interconnect are for MDM, AXI_UARTLITE, AXI_IIC, AXI_INTC, VTC (two instances), AXI_PERFORMANCE_MONITOR, AXI OSD, and AXI_VDMA.

**Video Timing Controller**

The VTC 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. The output side of the core generates the horizontal and vertical blanking and synchronization pulses used in a standard video system and includes support for programmable pulse polarity.

The VTC contains an PLBV46 interface to access slave registers from a processor. For more information about the VTC, see the LogiCORE IP Video Timing Controller v3.0 Data Sheet [Ref 8]. 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 output portion of the video pipeline.

The VTC core is provided under the SignOnce IP site license and can be generated using the CORE Generator™ tool, which is a part of the ISE Design Suite. A simulation evaluation license for the core is shipped with the CORE Generator tool. To access the full functionality of the core, including FPGA bitstream generation, a full license must be obtained from Xilinx.

**Note:** The PLB-based pcore timebase_v2_01_a (VTC) generated by the CORE Generator tool v13.4 is copied into the local pcores directory of the system. The MPD file of the pcore is modified to support Kintex-7 devices by adding *kintex7= PREFERRED to the OPTION ARCHITECTURE_MAP. The AXI-to-PLB bridge was used to connect timebase_v2_01 to the rest of the system.

**AXI TPG**

The AXI TPG contains an AXI4-Lite Interface to access slave control registers from a processor. In this reference design, the video traffic to DDR3 memory is generated by a TPG. The TPG block can generate several video test patterns that are commonly used in the video industry for verification and testing. In the reference design, the TPG is used as a replacement to a video source because only the amount of traffic generated to demonstrate the performance...
of the system is of interest. The control software demonstrates generation of nine possible video patterns such as color bars, horizontal and vertical burst patterns, flat colors, and zone plates. No matter which test pattern is chosen, 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 TPG always generates one of nine possible test patterns through user input. These patterns are meant for testing purposes only, and are not calibrated to broadcast industry standards.

**AXI OSD**

The Video On-Screen Display LogiCORE™ IP 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 sizes in YUVA 4:4:4 or RGBA image formats in 8, 10, or 12 bits per color component. In this application note, the OSD is configured to display one video layer only but can be configured for multiple video streams as separate display layers. Figure 3 shows a three-layer block diagram of the OSD core.

![Sample Three-Layer OSD Core Block Diagram](image)

*Figure 3: Sample Three-Layer OSD Core Block Diagram*

The AXI OSD contains an AXI4-Lite interface allowing access to core configuration registers from a processor. For more information about the AXI OSD, see the *LogiCORE IP Video On-Screen Display v2.0 Data Sheet* [Ref 9].

The Video On-Screen Display core is provided under the SignOnce IP site license and can be generated using the CORE Generator tool, which is a part of the ISE Design Suite.
A simulation evaluation license for the core is shipped with the CORE Generator tool. To access the full functionality of the core, including FPGA bitstream generation, a full license must be obtained from Xilinx.

**AXI Performance Monitor**

The AXI performance monitor core measures throughput for a DDR3 memory connected to the AXI Interconnect. The processor accesses the AXI performance monitor core registers through a slave AXI4-Lite interface contained in the core. The AXI performance monitor core only monitors the read and write channels between the AXI slave and the AXI Interconnect. The core does not modify or change any of the AXI transactions it is monitoring. The AXI performance monitor core also calculates the glass-to-glass delay of the system by connecting appropriate signals to the core.

*Note:* Glass-to-glass delay is defined as the number of clock cycles consumed to display a frame from the TPG (video source) to an LCD screen (video sync).

Several signals must be connected in the system to measure the throughput. DDR slave interconnect (AXI_MM) is connected to one of the four slots of the monitor. In addition to the signals of the DDR slave interconnect and AXI4-Lite bus interface, the core clock (the higher of two bus interface clock frequencies) must also be connected. To evaluate the glass-to-glass delay of the system, Vid_clk, Vtc0_Fsync, Vsync_osd, Tpg_Active_video_in, Tpg_Data, Osd_Active_Video_In, and Osd_data are also connected. Data and enable signals output from the TPG (the starting block of the datapath) and OSD (the ending block of the datapath) are used to evaluate the number of clock cycles to process a frame between these two IPs.

The core can also measure various performance metrics such as total read byte count, write byte count, read requests, write requests, and write responses. Count start and count end conditions come from the processor through the register interface. The global clock counter of the core measures the number of clocks between the count start and end events. The counters used for the performance monitor can be configured for 32 or 64 bits through the register interface. Final user-selectable metrics can also be read through the register interface.

In this application note, the AXI performance monitor core is connected to the DDR3 memory controller to measure the throughput of the core. Valid, ready, strobe, and other AXI signals of the DDR3 slave are used to enable various counters for measuring events on the bus.

---

**Software Applications**

The software application starts up the video pipeline allowing the user to change resolution and frame rate and examine bandwidth in real time.

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 application program interface (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 far more convenient option.

The application software in the reference design performs these actions:

1. The system, caches, UART, and VDMA are initialized.
2. The HDMI port is initialized.
3. The onboard SI570 clock generator is configured through the IIC interface.
4. The onboard SI570 clock generator is configured to generate the video clock for the desired frame rate and resolution.
5. The TPG instances are set to write a default color bar pattern that does not start until the VTC instances are started. TPG and VTC instances are configured to generate the desired frame resolutions.
6. The VTC instances are started for the desired resolution.
7. The AXI OSD is configured for the desired resolution.

8. The AXI VDMA instance is started by configuring VDMA read and write channels to begin the transfers for the VDMA instances.

9. The TPG instance in the design is configured to write one of the 9 possible test patterns:
   - Vertical ramp
   - Horizontal ramp
   - Flat red
   - Flat green
   - Flat blue
   - Color bars
   - Zone plates
   - Tartan bars
   - Cross hatch

After the initial setup sequence, the user can enter an option through HyperTerminal to select one of the combinations of resolution and frame rate (0–7 for resolutions, a or b for frame rates). Because OSD is configured for a single channel, its input data is directly transferred to its output. Video patterns output from the OSD are routed to the HDMI port through a color space (RGB to YCbCr) converter. For multiple video pipelines, OSD registers can also be configured to blend input channels to the required level, and different values are given to the alpha blending register for each layer to show all layers on the LCD screen at the same time.

After configuring the video pipeline for the desired frame rate and resolution, the user can select one of the 9 possible video patterns by entering a number from 0 through 8. Options p and d set up the performance monitoring IP instance to measure the performance of the DDR memory and glass-to-glass delay, respectively, in this sequence:

1. Connect a USB cable from the host PC to the USB JTAG port (item (6) in Figure 4). Ensure that the appropriate device drivers are installed (refer to Kintex-7 FPGA KC705 Evaluation Kit Getting Started Guide [Ref 10] for additional information).

2. Connect a second USB cable from the host PC to the USB UART port (6). Ensure that the USB-UART drivers described in Hardware Requirements, page 2 have been installed.

3. Connect the KC705 HDMI connector (18) to a video monitor. To display the higher frame rate configurations, the monitor should be capable of displaying frame rates up to 120 Hz video signal.

4. Connect a power supply cable.

5. Turn the power on (27).

6. Start a terminal program (e.g., HyperTerminal) on the host PC with these settings:
   - Baud rate: 9600
   - Data bits: 8
   - Parity: None
   - Stop bits: 1
   - Flow control: None
Executing the Reference Design on Hardware

This section provides instructions to execute the reference design in hardware. The reference design runs on the KC705 board shown in Figure 4. In these instructions, numbers in parentheses correspond to the callout numbers in Figure 4. Not all callout numbers are referenced.

Executing the Reference System Using the Pre-Built Bitstream and the Compiled Software Application

These are the steps to execute the system using files in the `ready_for_download` directory of the `<unzip_dir>/vdma_ref_design` directory:

1. In a Xilinx command shell (ISE Design Suite 32- or 64-bit command prompt) or terminal window, change directories to the `ready_for_download` directory.
   ```
   % cd <unzip dir>/vdma_ref_design/ready_for_download
   ```
2. Invoke the Xilinx Microprocessor Debugger (XMD) tool:
   ```
   % xmd
   ```
3. Download the bitstream inside XMD:
   ```
   XMD% fpga -f download.bit
   ```
4. Connect to the processor inside XMD:
   ```
   XMD% connect mb mdm
   ```
5. Download the processor code (ELF) file:
   ```
   XMD% dow vdma_display.elf
   ```
6. Run the software:
   ```
   XMD% run
   ```

Figure 4: KC705 Board
Note: The user must hit the CPU reset pushbutton (number 28 in Figure 4) before selecting a new resolution. This makes the processor rerun the application from reset (the BIT and ELF files do not need to be reloaded).

Results from Running Hardware and Software

The LCD monitor connected to the KC705 board displays a color bar pattern, and the HyperTerminal screen displays the output shown in Figure 5.

Figure 5: HyperTerminal Menu for Selecting Resolution

The user can select between the 60 and 75 frames/sec options, as shown in Figure 6.

Figure 6: HyperTerminal Menu for Selecting Frame Rate
The user can choose from nine pattern options displayed on the HyperTerminal screen, as shown in Figure 7:

- 0 = Horizontal ramp
- 1 = Vertical ramp
- 2 = Flat red
- 3 = Flat green
- 4 = Flat blue
- 5 = Color bars
- 6 = Zone plates
- 7 = Tartan bars
- 8 = Cross hatch

Figure 7: HyperTerminal Menu for Selecting Video Pattern
Performance

The DDR3 PHY is set for 64 bits with a memory clock frequency of 400 MHz. The theoretical throughput on the DDR3 memory is 6.4 GB/s, which is the total bandwidth available in the design.

Selecting option p causes the software to display this output (the numbers might vary slightly from the values shown):

----------DDR3, AXI4 Slave Profile Summary........

Theoretical DDR Bandwidth = 6400000000 bytes/sec

Practical DDR Bandwidth = 1093621153 bytes/sec

Percentage of DDR Bandwidth consumed by frame of resolution (1920x1080 @ 60 Hz) (Approx.)= 17.0873

Selecting option d causes the software to display this output (the numbers might vary slightly from the values shown):

Processing Time Per Frame of resolution (1920x1080 @ 60 Hz) (Glass to Glass delay) = 13.45879 ms

Building Hardware

This section covers rebuilding the hardware design.

Before rebuilding the project, the user must ensure that the full system hardware evaluation or full licenses for AXI OSD and VTC are installed. To obtain evaluation licenses for the AXI OSD or VTC, refer to these websites:

• Xilinx On-Screen Display LogiCORE IP [Ref 11]
• Xilinx VTC [Ref 12]

Note: The generated bitstream is at <unzipdir>/vdma_ref_design/HW/vdma_ref_design/implementation/download.bit

To rebuild the hardware design:

1. Open vdma_ref_design/HW/vdma_ref_design/system.xmp in XPS.
2. Select Hardware > Generate Bitstream to generate a bitstream for the system.
3. Run Device Configuration > Update Bitstream to initialize the block RAM with a bootloop program to ensure the processor boots with a stable program in memory.
4. Select Project > Export Hardware Design to SDK... .
5. Select Export Only.

Compiling Software in SDK

1. Start SDK. (In Linux, type xsdk to start SDK.)
2. In the Workspace Launcher, select the following Workspace: <unzip dir>/vdma_ref_design/SW/SDK_Workspace
3. Click OK.
4. Set the repository by selecting Xilinx Tools > Repositories
   • For Local Repositories click on New...
   • Change directories to <unzip dir>/vdma_ref_design/SW/repository
   • Click OK.

The board support package (BSP) and software applications compile at this step. The process takes 2 to 5 minutes. The user can now modify existing software applications and create new software applications in SDK. (The following steps are only needed if the goal
is to rebuild the hardware as described in Building Hardware.

5. Right-click on the vdma_ref_design_hw_platform and select Change Hardware Platform Specification.

6. Click Yes.

7. Select the XPS SDK export system.xml file (<unzip dir>/vdma_ref_design/HW/vdma_ref_design/SDK/SDK_Export/hw/system.xml).

8. Click OK.

The BSP and software applications compile again at this step. The process takes 2 to 5 minutes.

Running the Hardware and Software through SDK

1. Select Xilinx Tools > Program FPGA.

   Ensure bootloop is used for microblaze_0.

2. Click Program.

3. In the Project Explorer window, right click vdma_display > Run As > Launch on Hardware.

Design Characteristics

This reference design is implemented in a Kintex-7 FPGA XC7K325TFFG900-1 using the ISE Design Suite: Embedded Edition 13.4.

The resources used are:

- Total LUTs used: 34,100 out of 203,800 (16%)
- Total I/Os used: 138 out of 600 (23%)
- Total internal memory used:
  - RAMB36E1s: 16 out of 445 (3%)
  - RAMB18E1s: 3 out of 890 (1%)

Note: Device resource utilization results depend on the implementation tool versions. Exact results can vary. These numbers should be used as a guideline.

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 200 MHz on the main AXI Interfaces to the memory controller using the ISE Design Suite 13.4.

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

https://secure.xilinx.com/webreg/clickthrough.do?cid=185348

The reference design matrix is shown in Table 4.

Table 4: Reference Design Matrix

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>General</td>
<td></td>
</tr>
<tr>
<td>Developer name</td>
<td>Sateesh Reddy Jonnalagada, Vamsi Krishna</td>
</tr>
<tr>
<td>Target devices (stepping level, ES, production, speed grades)</td>
<td>Kintex-7 FPGAs</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>
</tbody>
</table>
**Utilization and Performance**

*Table 4: Reference Design Matrix (Cont'd)*

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design uses code and IP from existing Xilinx application note and reference designs, CORE Generator software, or third party</td>
<td>Reference designs provided for EDK and video cores generated from the CORE Generator tool</td>
</tr>
</tbody>
</table>

**Simulation**

- Functional simulation performed: N/A
- Timing simulation performed: N/A
- Test bench used for functional and timing simulations: N/A
- Test bench format: N/A
- Simulator software/version used: N/A
- SPICE/IBIS simulations: N/A

**Implementation**

- Synthesis software tools/version used: XST 13.4
- Implementation software tools/versions used: ISE Design Suite 13.4: System Edition
- Static timing analysis performed: Yes (passing timing in PAR/TRCE)

**Hardware Verification**

- Hardware verified: Yes
- Hardware platform used for verification: KC705 board

---

**Utilization and Performance**

*Table 5 shows the device and utilization information.*

**Table 5: Device Utilization**

<table>
<thead>
<tr>
<th>Device</th>
<th>Speed Grade</th>
<th>Package</th>
<th>Slice Registers</th>
<th>Occupied Slices</th>
<th>Slice LUTs</th>
<th>I/Os</th>
<th>RAMB36E1s</th>
<th>RAMB18E1s</th>
</tr>
</thead>
<tbody>
<tr>
<td>XC7K325T</td>
<td>-1</td>
<td>FFG900</td>
<td>41,308 (10%)</td>
<td>15,577 (30%)</td>
<td>34,100 (16%)</td>
<td>146</td>
<td>16 (3%)</td>
<td>3 (1%)</td>
</tr>
</tbody>
</table>

Device resource utilization is detailed in Table 5 for the IP cores shown in Figure 1, page 3. The information in Table 6 is taken from the Design Summary tab in XPS under the Design Overview > Module Level Utilization report selection. The utilization information is approximate due to cross-boundary logic optimizations and logic sharing between modules.

**Table 6: Module Level Resource Utilization**

<table>
<thead>
<tr>
<th>IP Core</th>
<th>Instance Name</th>
<th>Slices</th>
<th>Slice Registers</th>
<th>LUTs</th>
<th>LUTRAM</th>
<th>Block RAM/FIFO</th>
<th>DSP48E1 Slices</th>
<th>BUFG</th>
<th>BUF</th>
<th>MMC</th>
<th>MMC ADV</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXI 7 series DDR3 controller</td>
<td>DDR3_SDRAM</td>
<td>6,257</td>
<td>9,955</td>
<td>11,623</td>
<td>3,010</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AXI Interconnect</td>
<td>AXI_Lite</td>
<td>377</td>
<td>294</td>
<td>1,456</td>
<td>140</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>AXI_MM</td>
<td>5,735</td>
<td>17,357</td>
<td>11,012</td>
<td>271</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Video Timing Controller</td>
<td>timebase_0</td>
<td>447</td>
<td>1,104</td>
<td>744</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>timebase_1</td>
<td>452</td>
<td>1,105</td>
<td>731</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
Table 6: Module Level Resource Utilization (Cont’d)

<table>
<thead>
<tr>
<th>IP Core</th>
<th>Instance Name</th>
<th>Slices</th>
<th>Slice Registers</th>
<th>LUTs</th>
<th>LUTRAM</th>
<th>Block RAM/FIFO Slices</th>
<th>DSP48E1</th>
<th>BUFG</th>
<th>BUFR</th>
<th>MMCM ADV</th>
</tr>
</thead>
<tbody>
<tr>
<td>AXI TPG (includes associated glue logic)</td>
<td>axi_tpg_0</td>
<td>403</td>
<td>1,088</td>
<td>898</td>
<td>14</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>xsvi2axi_0</td>
<td>8</td>
<td>27</td>
<td>11</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AXI VDMA</td>
<td>axi_vdma_0</td>
<td>2,227</td>
<td>4,778</td>
<td>2,873</td>
<td>301</td>
<td>8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AXI OSD (includes associated glue logic and display driver)</td>
<td>osd_0</td>
<td>685</td>
<td>1,475</td>
<td>955</td>
<td>30</td>
<td>1</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>csc_rgb_to_yccrbc422_0</td>
<td>150</td>
<td>377</td>
<td>286</td>
<td>105</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>hdmi_interface_0</td>
<td>7</td>
<td>19</td>
<td>19</td>
<td>19</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Clock, Reset, and miscellaneous system logic</td>
<td>clock_generator_0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>proc_sys_reset_0</td>
<td>17</td>
<td>31</td>
<td>21</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>MicroBlaze Processor Subsystem (includes local memory and debug module for JTAG-based debug)</td>
<td>microblaze_0</td>
<td>1,211</td>
<td>1,388</td>
<td>1,828</td>
<td>244</td>
<td>6</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>debug_module</td>
<td>93</td>
<td>128</td>
<td>116</td>
<td>23</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>microblaze_0_bram_block</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>microblaze_0_d_bram_ctrl</td>
<td>5</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>microblaze_0_dlimb</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>microblaze_0_i_bram_ctrl</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>microblaze_0_limb</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>microblaze_0_intc</td>
<td>62</td>
<td>72</td>
<td>93</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AXI_IIC</td>
<td>axi_iic_0</td>
<td>255</td>
<td>259</td>
<td>347</td>
<td>20</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>AXI UartLite</td>
<td>RS232_Uart_1</td>
<td>74</td>
<td>85</td>
<td>100</td>
<td>18</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td></td>
<td>19,553</td>
<td>41,183</td>
<td>34,100</td>
<td>4,126</td>
<td>19</td>
<td>13</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Notes:
1. Slices can be packed with basic elements from multiple IP cores and hierarchies. Therefore, a slice is counted in every hierarchical module that each of its packed basic elements belong to. This results in some double counting of slice counts when adding up the slice counts across modules.

Table 7 summarizes the bandwidth calculations for the physical memory interface.

Table 7: Physical DDR3 Memory Interface Maximum Theoretical Bandwidth

<table>
<thead>
<tr>
<th>Data Width</th>
<th>Data Rate</th>
<th>Maximum Theoretical Bandwidth</th>
</tr>
</thead>
<tbody>
<tr>
<td>64 bits (SODIMM)</td>
<td>800 Mb/s</td>
<td>6.40 GB/s (51.2 Gb/s)</td>
</tr>
</tbody>
</table>

Table 8 summarizes the total bandwidth of video data moved through memory (resolution = 1920 x 1080 and frame rate = 60 Hz).

Table 8: Average Bandwidth Used for Video Traffic

<table>
<thead>
<tr>
<th>Frame Resolution</th>
<th>Refresh Rate</th>
<th>Bits Per Pixel</th>
<th>Number of Video Streams</th>
<th>Video Throughput (Total Available Bandwidth)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1920 x 1080</td>
<td>60 Hz</td>
<td>32</td>
<td>2</td>
<td>0.996 GB/s (7.968 Gb/s)</td>
</tr>
</tbody>
</table>
Table 9 summarizes the percentage of the maximum theoretical bandwidth used by the video streams for various resolutions.

Table 9: Percentage of the Maximum Theoretical Bandwidth Used

<table>
<thead>
<tr>
<th>Resolution</th>
<th>Horizontal Active Video x Vertical Active Video</th>
<th>Frame Rate (Hz)</th>
<th>Maximum Theoretical Bandwidth (MB/s)</th>
<th>Percentage of the Maximum Theoretical Bandwidth Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>640 x 480</td>
<td>60</td>
<td>147.45</td>
<td>2.2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75</td>
<td>184.32</td>
<td>2.8</td>
</tr>
<tr>
<td>2</td>
<td>720 x 480</td>
<td>60</td>
<td>165.88</td>
<td>2.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75</td>
<td>207.36</td>
<td>3.2</td>
</tr>
<tr>
<td>3</td>
<td>800 x 600</td>
<td>60</td>
<td>230.4</td>
<td>3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75</td>
<td>288</td>
<td>4.5</td>
</tr>
<tr>
<td>4</td>
<td>1024 x 768</td>
<td>60</td>
<td>377.486</td>
<td>5.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75</td>
<td>471.858</td>
<td>7.3</td>
</tr>
<tr>
<td>5</td>
<td>1280 x 720</td>
<td>60</td>
<td>442.368</td>
<td>6.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75</td>
<td>552.96</td>
<td>8.6</td>
</tr>
<tr>
<td>6</td>
<td>1280 x 1024</td>
<td>60</td>
<td>629.144</td>
<td>9.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75</td>
<td>786.4</td>
<td>12.2</td>
</tr>
<tr>
<td>7</td>
<td>1920 x 1080</td>
<td>60</td>
<td>995.3</td>
<td>15.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75</td>
<td>1244.16</td>
<td>19.4</td>
</tr>
<tr>
<td>8</td>
<td>1600 x 1200</td>
<td>60</td>
<td>921.6</td>
<td>14.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>75</td>
<td>1152</td>
<td>18</td>
</tr>
</tbody>
</table>

Conclusion

The reference design accompanying this application note demonstrates the configuration of Xilinx video IP cores through their register interface for processing various desired combinations of video resolutions and frame rates. Video clocks for processing different frame rates and resolutions of the design are supplied by the onboard configurable clock generator. The AXI performance monitor IP core connected to DDR memory is used to measure bandwidth and latencies of the system for various resolutions and frame rates. This application note thus details the effective configuration of Xilinx video IP cores implement various functions for many video applications.

References

This application note uses the following references:

1. AMBA AXI4 specifications
2. UG761, AXI Reference Guide
4. DS768, LogiCORE IP AXI Interconnect (v1.05) Data Sheet
5. XAPP740, Designing High-Performance Video Systems with the AXI Interconnect
6. DS799, LogiCORE IP AXI Video Direct Memory Access (axi_vdma) (v5.00.a) Data Sheet
7. UG586, 7 Series FPGAs Memory Interface Solutions User Guide
8. DS857, LogiCORE IP Video Timing Controller v3.0 Data Sheet
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>05/03/2012</td>
<td>1.0</td>
<td>Initial Xilinx release.</td>
</tr>
</tbody>
</table>

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](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](http://www.xilinx.com/warranty.htm#critapps).