



WP358 (v1.0) December 8, 2009

Simplifying Embedded Hardware and Software Development with Targeted Reference Designs

By: Navanee Sundaramoorthy

FPGAs are becoming the platform of choice for a growing number of hardware and software designers developing embedded processing applications for the industrial, medical, communications, aerospace, and defense markets. The combined value of design flexibility to support changing standards, reduced cost due to better system integration, and parallel processing performance has steadily fueled this trend of implementing complete systems on FPGAs.

In an effort to substantially simplify and accelerate the means by which both embedded software and hardware designers gain access to these benefits, Xilinx has developed fully-functional, tested, and supported *targeted reference designs* for both the Spartan®-6 and Virtex®-6 FPGAs.

Introduction

By definition, FPGAs have always offered the flexibility to integrate, adapt, and optimize performance, function, and power in embedded hardware designs. However, the inherent hardware design flexibility of the FPGA can be an impediment for the embedded software designer. Software developers need a known good foundation in a familiar design environment to keep their designs from becoming too complex.

Embedded software developers have very different and significantly less complex environmental needs than their hardware engineering counterparts. Specifically, the software developer wants a compiler, a debugger, an Eclipse-based integrated design environment (IDE), an industry-standard operating system (e.g., Linux, μ C/OS), a robust set of software libraries, and a stable processor system with well-defined peripherals.

Embedded hardware designers, however, are more concerned about system architecture, IP verification, and timing closure. They require pre-verified hardware reference designs as starting points to evaluate the underlying FPGA platform. From there, they might add standard IP, integrate some custom IP, and also remove IP blocks that they do not need. Ideally, they need a high-level environment to make these design changes quickly, without having to modify RTL code (VHDL or Verilog) for each change.

Xilinx has developed an embedded processing targeted reference design in the form of a fully functional and widely applicable MicroBlaze™ Processor Sub-System (PSS), accompanied by all of the requisite software design environment components that specifically and uniquely fulfill these requirements. Serving as the base of a new generation in Xilinx embedded platforms, the MicroBlaze PSS integrates essential processing, memory, and I/O functions and is supported by a strong software ecosystem, offering middleware stacks and industry-leading operating systems such as Linux and μ C/OS-II.

When used with the new Spartan-6 and Virtex-6 FPGA Embedded Kits, the MicroBlaze PSS allows software application developers to begin writing and porting their application code to the embedded platform before any hardware development has begun. Moreover, application software developers can immediately evaluate the performance of the processor, internal and external memory, bus architectures, and I/Os against their system requirements. The developers can even begin optimizing their application software to the MicroBlaze PSS in parallel with the hardware design effort.

Similarly, the new embedded kits provide everything the hardware design team needs to quickly evaluate the Xilinx embedded platform capabilities, as well as modify and extend these to their specific application needs. Thus, the new Xilinx® embedded kits create the opportunity for both the hardware and software design teams to move quickly and simultaneously through their respective evaluation and development processes.

The Embedded Targeted Reference Design

While the FPGA provides a plethora of ways to innovate in terms of feature usage, performance, configuration, and optimization, these options require decisions at every level that designers might not want to or need to consider. It is here that the MicroBlaze PSS saves time in getting started and becoming productive. To understand

the completeness of the MicroBlaze PSS, one must appreciate the current method of making those setup decisions in developing and using a processor system in an FPGA. To start with, the base processor used in the MicroBlaze PSS is the 32-bit Xilinx MicroBlaze processor CPU core. The MicroBlaze processor core is a parameterizable IP block delivered by Xilinx in the Embedded Development Kit (EDK) tool suite that offers various options that users might want to configure. Selectable features include enabling instruction and/or data caches as well as the sizes of the caches. The user can also configure and add features such as a hardware multiplier, hardware divide function, barrel shifter, memory management unit, and an IEEE floating point coprocessor to the MicroBlaze processor block. While this flexibility delivers optimum feature, performance, and area (cost) trade-offs, actually creating the design takes some effort that might not be required. At this first step, the MicroBlaze PSS design has made those decisions that are suitable for a variety of applications and is ready to use without any further configuration.

The processor core is the basic foundation for embedded processing use. To be useful, it must have some basic support (such as reset and interrupt circuits) as well as key peripherals and interconnects that are required in many applications. Basic peripherals such as memory controllers, timers, interrupt controllers, and serial interfaces are all available to be configured and added within the framework of the EDK tool suite. The challenge here is that one might not want to spend time configuring these basic peripherals (and the settings associated with them) but instead, just use them. Here again, the MicroBlaze PSS has been pre-configured to use a basic set of peripherals for most applications. The result is a pre-configured embedded processing system that is ready to be programmed, as illustrated in [Figure 1](#).

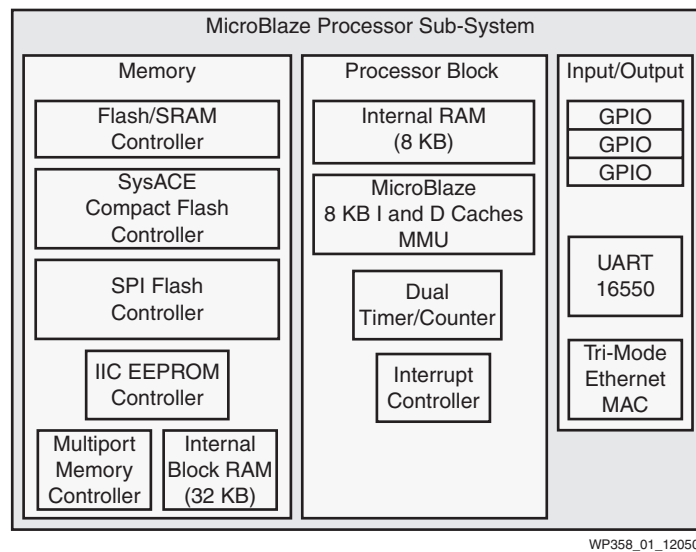


Figure 1: MicroBlaze Processor Sub-System

With this basic system configuration, designers can start evaluating, programming, or both, for their application development. In addition, from an FPGA hardware perspective, designers can use this targeted reference design as the core of their complete system design, as shown in [Figure 2](#). The user-defined portion of the logic in [Figure 2](#) is where the hardware designer can customize and extend the targeted reference design using the Xilinx EDK tool suite. It is for this reason that Xilinx has named this design the MicroBlaze Processor Sub-System, because it truly is a sub-system to be used in a more robust system driven by application requirements.

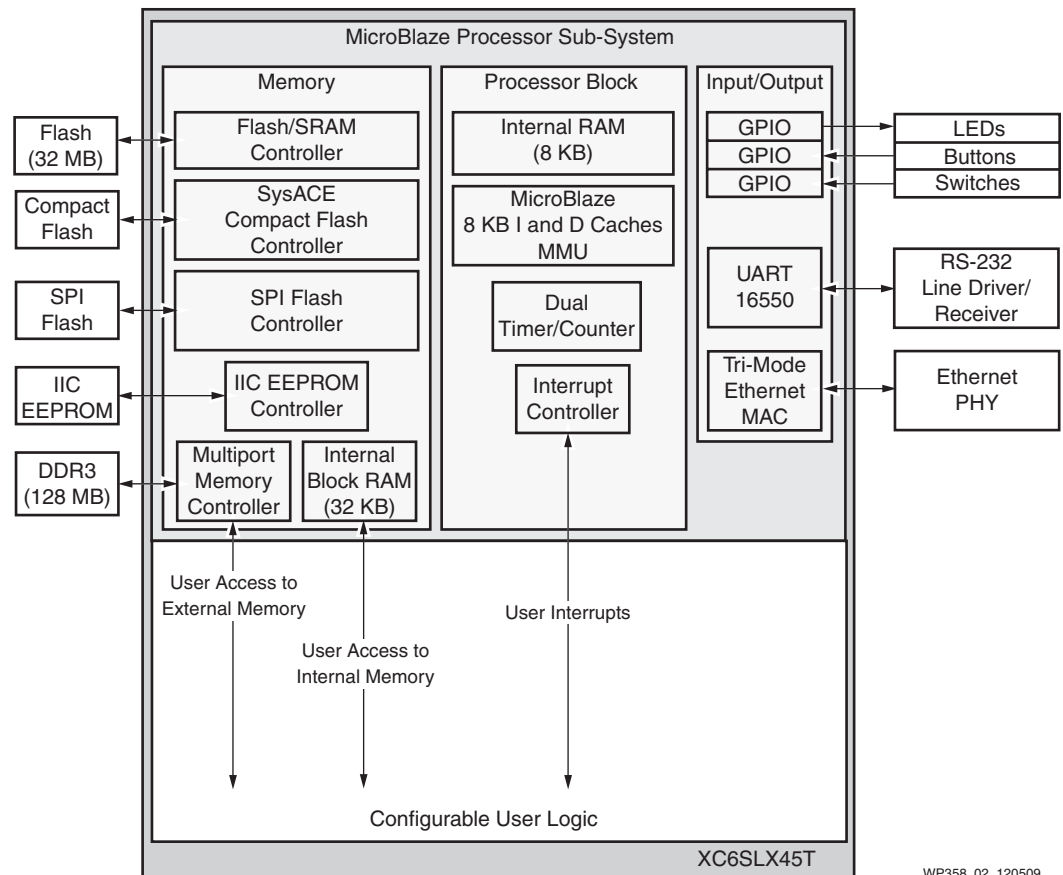


Figure 2: MicroBlaze Processor Sub-System within the FPGA

The Embedded Targeted Reference Design Features

The MicroBlaze processor is configured with a Memory Management Unit (MMU) and various parameters optimized for performance, including the 5-stage pipeline option (as opposed to the slower 3-stage pipeline option) and a hardware barrel shifter. The hardware barrel shifter can shift or rotate a data word by any number of bits in a single clock cycle. Data shifting is a required element of many key operations such as address generation and arithmetic functions. The action of a barrel shifter can be emulated in software, but this takes valuable time not available in real-time applications. The instruction cache and data cache are both enabled, each with a cache size of 8 KB.

The MMU is configured in Virtual mode with two memory protection zones to run full-fledged embedded operating systems like Linux. In Virtual mode, the MMU controls effective address-to-physical address mapping and supports memory protection. Virtual mode provides greater control over memory protection. The MMU provides memory protection and relocation, which are useful for multi-tasking environments. Multi-tasking gives the appearance of simultaneous or near-simultaneous execution of multiple programs.

For memory support, a variety of features are pre-configured to support various types of memory devices. The Multi-Port Memory Controller (MPMC), also delivered with the EDK tool suite, is configured to provide four 32-bit bidirectional ports for access to the external DDR3 memory. In addition, support for 32 KB internal block RAM, 32 MB

linear (parallel) flash, 8 MB serial flash, compact flash using the System ACE™ technology, and 1 KB IIC EEPROM is included in the design.

The general-purpose I/O (GPIO) uses the Xilinx Platform Studio (XPS) GPIO IP core provided with EDK and is instantiated three times in the system to enable a variety of uses such as control pushbuttons, DIP switches, and LEDs on the associated development boards that run the design.

The Tri-Mode Ethernet MAC (TEMAC) core is configured to support a GMII/MII PHY interface and contains internal 4 KB transmit and receive FIFOs. The TEMAC core can run at 10 Mb/s, 100 Mb/s, or 1,000 Mb/s depending on the network to which it is attached.

For timers, the XPS Timer core delivered with the EDK tool suite is configured to provide two 32-bit timers. For serial communication, there is an integrated 16550-compatible UART core that is pre-configured to use interrupts. The baud rate, data bits, and parity settings of this UART core are controlled by software.

The features of the MicroBlaze Processor Sub-system are summarized here:

- Processor Block:
 - 32-bit MicroBlaze processor with 8 KB I cache and 8 KB D cache, consisting of:
 - Hardware barrel shifter
 - Memory management unit(The MicroBlaze processor and system bus run at 100 MHz)
 - 8 KB local memory for instructions and data
 - Debug module
 - Interrupt controller
 - Dual 32-bit timer/counter
- Memory:
 - 128 MB DDR3 SDRAM interface operating at 400 MHz
 - 32 KB internal block RAM
 - 32 MB linear (parallel) flash
 - 8 MB serial flash
 - Compact flash using System ACE™ technology
 - 1 KB IIC EEPROM
 - Multi-port memory controller with one available port for the user logic interface to the external DDR3 SODIMM memory
- I/O:
 - Three GPIO controllers
 - 16550 UART core
 - 10/100/1000 TEMAC core
 - Serial Peripheral Interface (SPI) and Inter-Integrated Circuit (I²C) serial interface cores

Embedded Targeted Reference Design Deliverables

The MicroBlaze PSS is delivered as an XPS project that contains the MicroBlaze PSS design. All MicroBlaze PSS peripherals and IP used are included with the EDK delivered in the ISE® Design Suite: Embedded Edition. Design and project settings have been optimized for implementation on Spartan-6 and Virtex-6 FPGAs and for use with the Xilinx SP605 and ML605 boards, respectively.

For software deliverables, the Xilinx Software Development Kit (SDK) is the environment for embedded software development with Xilinx FPGAs. SDK is included in the ISE Design Suite: Embedded Edition. The SDK, which is Eclipse-based, includes a workspace and projects for the pre-built MicroBlaze PSS design. In addition, SDK includes all required drivers and libraries for the MicroBlaze PSS.

A MicroBlaze PSS data sheet, similar to any traditional microprocessor or microcontroller data sheet, is provided with the embedded kits. This data sheet provides an overview of the design, a detailed description of the MicroBlaze PSS operation, address map, block diagram, and performance. In addition, a hardware tutorial provides step-by-step instructions to modify the MicroBlaze PSS using XPS. The tutorial also describes recommended flows to add standard peripherals and IP provided with EDK or from third parties, along with instructions and options for debug using XPS. A software tutorial provides step-by-step instructions to use SDK to compile, debug, and profile stand-alone software applications, as well as to boot PetaLogix PetaLinux for the Linux kernel of the MicroBlaze processor.

Spartan-6 and Virtex-6 FPGA Embedded Kit Deliverables

To jump-start the evaluation and design process, the Spartan-6 and Virtex-6 FPGA Embedded Kits provide all of the hardware and software required to evaluate and begin designing with the targeted reference design. The components have been conveniently bundled in easy-to-use kits that provide many options to get started. The kits also provide a demonstration based on the MicroBlaze PSS that runs a simple industrial image processing application. By following a three-step procedure of setup, evaluate, and customize, both software and hardware engineers can quickly begin evaluating the performance and capabilities of the embedded platform. The key components of the kits include:

- The MicroBlaze PSS targeted reference design
- Either the Xilinx SP605 or ML605 evaluation board, depending on the selection of the Spartan-6 or Virtex-6 FPGA kit
- Integrated design tools:
 - ISE Design Suite: Embedded Edition, including EDK with XPS, SDK, and the ChipScope™ analyzer
- Embedded peripherals and IP provided with the ISE Design Suite and EDK
- Partner software tools, real-time operating systems (RTOSs), and middleware
- Documentation
- Getting started demonstration
- A USB stick containing all the design files to get started quickly

Using and Modifying the Embedded Targeted Reference Design

The embedded targeted reference design is the starting point for embedded hardware and software design with the Spartan-6 and Virtex-6 Embedded kits. The key steps in using the embedded targeted reference design are:

- Hardware modifications
- Software modifications and programming
- Integrating the design
- Profiling and debugging

Hardware Modifications

Before modifying the MicroBlaze PSS, the exact system configuration can be seen in the MicroBlaze PSS data sheet included in the embedded kits. This data sheet provides a traditional processor-centric view of the system in terms of block diagrams, address maps, external ports, and design clock frequencies. The MicroBlaze PSS is provided in the embedded kit as an XPS project that can be configured graphically using the XPS IDE. When the XPS design is opened, a system assembly panel allows viewing of the system topology between the MicroBlaze processor and the IP blocks included in the PSS. By double-clicking on the MicroBlaze processor, various processor parameters like caches and MMU configuration can be modified. In addition, by double-clicking on the clockgen block, features such as the system clock frequencies for the processor and memory interfaces can be modified quickly.

XPS includes a comprehensive catalog of industry-standard IP cores from simple general-purpose parallel I/O cores to the high-performance Gigabit Ethernet MAC IP core. Within XPS, IP can be included by dragging and dropping any of the available peripheral IPs into the system. The system assembly view then enables connection of the IP to the MicroBlaze PSS. The address of the included peripheral can be assigned in the address map tab of the system assembly view. Navigation to the ports tab then enables I/O signal connection of the IP core to the external FPGA pins. This last step is done to assign pin constraints to the external pins of the added peripherals.

XPS provides a Create IP wizard to assist in the creation of custom embedded IP that can be interfaced to the MicroBlaze PSS. The embedded kit contains some example user IP, and the steps for creating them are described in the hardware tutorial. In addition, designers can import their own IP in the XPS pcore standard, or IP from the Xilinx partner ecosystem.

After the embedded processing system is modified, the design is ready to be transferred to the FPGA. Many details of the FPGA implementation flow are abstracted in the XPS design environment. By clicking on the **Generate Netlist** button, the tools automatically generate RTL wrappers for each IP, create the bus connections, generate the synthesis scripts, and call the synthesis tools to generate the netlist. Similarly, by clicking on the **Generate Bitstream** button, the underlying FPGA implementation tools are automatically called to generate the FPGA design bitstream.

Software Development Flow

Before developing any software, the MicroBlaze system configuration can be seen in the MicroBlaze PSS data sheet included in the embedded kits. The pre-configured MicroBlaze PSS is the starting point for software development with the embedded kits.

To get started with the Xilinx SDK (Eclipse IDE), along with the MicroBlaze PSS, a ready-made Eclipse workspace is provided with the embedded kits customized with the right target settings. Starting with this workspace, a stand-alone software platform is created for low-level software development. In addition, an OS/RTOS board support package (BSP) can be created if the relevant plug-ins have been obtained from PetaLogix for Linux on the MicroBlaze processor, or Micrium for μ C/OS-II.

To create software applications using the SDK new project wizard, a simple “hello world” application or more complex board test applications can be built. The wizard not only creates the sample application code but also configures the right compiler and linker script settings for the address map of the MicroBlaze PSS design. After creation, SDK includes a complete application debug environment built on Eclipse. By right-clicking on the specific application and selecting the option to debug the application on the hardware board, SDK automatically connects to the hardware board using a USB-based JTAG debug cable and identifies and connects to the MicroBlaze processor on the board. SDK then loads the application, begins executing it, and stops at the first line of `main()` in the C program. Using standard software debugging steps, the user can single-step, run, and stop the application. The program can be inspected, and system run-time parameters can be viewed graphically in the register, variable, and memory views in the IDE.

After the application is functionally correct, the program can be profiled for performance hot spots for further optimization. SDK enables easy setup to run the program to collect run-time statistics on the board. SDK also includes tools to graphically display the profile information to easily identify hot spots in the application. Because the target is an FPGA, the option to offload some of the critical processing steps onto hardware running on the FPGA can be selected. Coprocessors can be created in RTL to offload these critical tasks; custom coprocessors can also be created. In addition, Xilinx tools such as System Generator for DSP (the evaluation version is included in the embedded kits) can be used to create signal processing based coprocessors, if desired.

After the hardware and software development tool flows have been completed, extending and customizing the targeted reference design can be done easily. The tutorials included in the embedded kits are intended to serve as examples of how designs are modified and configured with the EDK flows. As an example, step-by-step instructions to build one such system for a video-processing application are shown in [Figure 3](#).

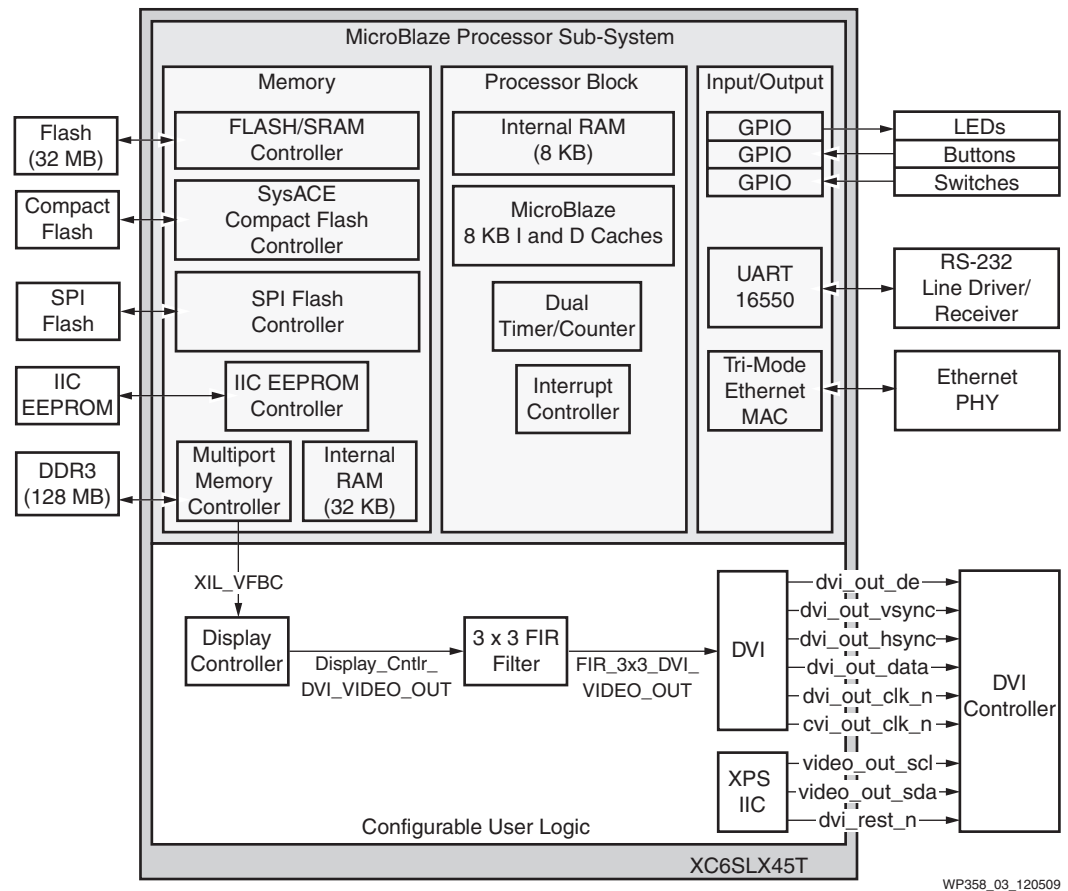


Figure 3: **FPGA-Based Video Processing Using Embedded Targeted Reference Design**

Summary

The challenge to make FPGA-based embedded platforms more accessible to both embedded software and hardware designers has precipitated the creation of the Xilinx MicroBlaze Processor Sub-System—the first embedded processing targeted reference design. This fully-functional, tested, and supported embedded processing sub-system simplifies and accelerates both platform evaluation and application development for all embedded designers, regardless of their prior experience working with FPGAs.

To learn more about the embedded kits, visit <http://www.xilinx.com/embedded>.

Revision History

The following table shows the revision history for this document:

Date	Version	Description of Revisions
12/08/09	1.0	Initial Xilinx release.

Notice of Disclaimer

The information disclosed to you hereunder (the "Information") is provided "AS-IS" with no warranty of any kind, express or implied. Xilinx does not assume any liability arising from your use of the Information. You are responsible for obtaining any rights you may require for your use of this Information. Xilinx reserves the right to make changes, at any time, to the Information without notice and at its sole discretion. Xilinx assumes no obligation to correct any errors contained in the Information or to advise you of any corrections or updates. Xilinx expressly disclaims any liability in connection with technical support or assistance that may be provided to you in connection with the Information. XILINX MAKES NO OTHER WARRANTIES, WHETHER EXPRESS, IMPLIED, OR STATUTORY, REGARDING THE INFORMATION, INCLUDING ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT OF THIRD-PARTY RIGHTS.

CRITICAL APPLICATIONS DISCLAIMER

XILINX PRODUCTS (INCLUDING HARDWARE, SOFTWARE AND/OR IP CORES) ARE NOT DESIGNED OR INTENDED TO BE FAIL-SAFE, OR FOR USE IN ANY APPLICATION REQUIRING FAIL-SAFE PERFORMANCE, SUCH AS IN LIFE-SUPPORT OR SAFETY DEVICES OR SYSTEMS, CLASS III MEDICAL DEVICES, NUCLEAR FACILITIES, APPLICATIONS RELATED TO THE DEPLOYMENT OF AIRBAGS, OR ANY OTHER APPLICATIONS THAT COULD LEAD TO DEATH, PERSONAL INJURY OR SEVERE PROPERTY OR ENVIRONMENTAL DAMAGE (INDIVIDUALLY AND COLLECTIVELY, "CRITICAL APPLICATIONS"). FURTHERMORE, XILINX PRODUCTS ARE NOT DESIGNED OR INTENDED FOR USE IN ANY APPLICATIONS THAT AFFECT CONTROL OF A VEHICLE OR AIRCRAFT, 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. CUSTOMER AGREES, PRIOR TO USING OR DISTRIBUTING ANY SYSTEMS THAT INCORPORATE XILINX PRODUCTS, TO THOROUGHLY TEST THE SAME FOR SAFETY PURPOSES. TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CUSTOMER ASSUMES THE SOLE RISK AND LIABILITY OF ANY USE OF XILINX PRODUCTS IN CRITICAL APPLICATIONS.