Pre-tested System-on-Chip Design

Accelerates PLD Development

A Lattice Semiconductor White Paper

March 2010

Lattice Semiconductor
5555 Northeast Moore Ct.
Hillsboro, Oregon 97124 USA
Telephone: (503) 268-8000
www.latticesemi.com

Introduction

Many moderate size Programmable Logic Device (PLD) designs, especially those in control plane applications, consist of a number of interfaces interconnected via an on-chip bus to a microprocessor that may be on- or off-chip. Although each interface is often relatively simple, the task of building all the on-chip interconnections and debugging them can be time consuming and frustrating. An increasing number of designers are using development boards with pre-designed processor-based systems to accelerate the development process.

This approach has several benefits. First, it is not necessary to assemble much of the IP or build shims from the chosen IP to the on-chip bus structure. Also, the on-chip bus structure does not need to be selected and created. Second, it is not necessary to manually connect the various IP to the on-chip bus. Even the relatively small design discussed later in this white paper, which uses approximately 1000 Look-Up Tables (LUTs), requires approximately 300 lines of Verilog code for the top level hook-up. Third, the time to functioning hardware is rapidly reduced with this approach. It can take days to get the basic section of the hardware and the all important debug interface operating. Adding or removing interfaces becomes much simpler once the design has been stabilized with the debug interface operating. And, if modifications result in the failure of the debug interface, it easy to return to the last know good version of the design.

Typical Development Board

An example of a development environment that allows designers to start with a pre-designed system and then modify it to build a prototype before transferring the design into their final system is the MachXO Mini Development board from Lattice. Although suitable for many applications, this board was tailored for the control plane applications that small, non-volatile PLDs are often used for. Figure 1 illustrates functions that are typically implemented in PLDs performing control plane applications.

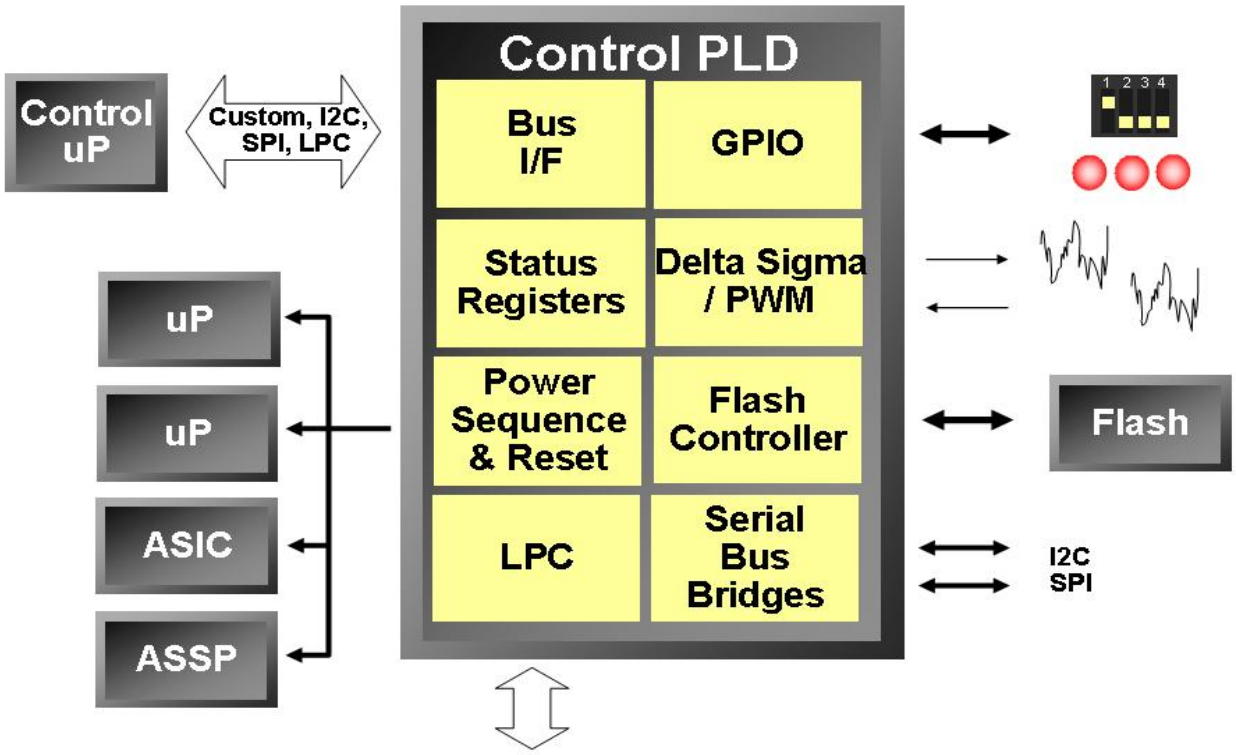

Figure 1 - Typical Control Plane Functions Implemented in PLDs

At the center of the Mini Development board is the MachXO 2280, a non-volatile, Flash-based, PLD that provides 2280 LUTs for logic implementation. Figure 2 shows the MachXO Mini Development board and annotates the major interfaces. The board provides five major classes of functionality:

- GPIO and status indicators are provided through a bank of LEDs, Headers, DIP switches and push button switches.
- A SPI Flash memory and I2C temperature sensor serve as examples of these common serial interfaces.
- A SRAM memory provides additional scratch pad memory.
- RS232 over USB provides an interface for debug.
- JTAG over USB serves as an interface for device programming.

The RS232 over USB and JTAG over USB interfaces are critical for the development process. Both utilize mini "B" type USB connectors and can be connected via cable to USB ports on any personal computer. Once the appropriate drivers have been loaded

(at this time available for Windows XP and later) the JTAG over USB interface allows the Lattice ispVM software to program the MachXO as required with modified designs. With the appropriate drivers the RS232 over USB interface allows emulation of RS232 over USB. Drivers are included in the latest version of Linux distributions and are available for Windows XP and later. The serial port simply appears as a “COM” port on the host PC. This interface provides a convenient method for the on-chip design to send status and, with appropriate modifications, debug information to the host processor.

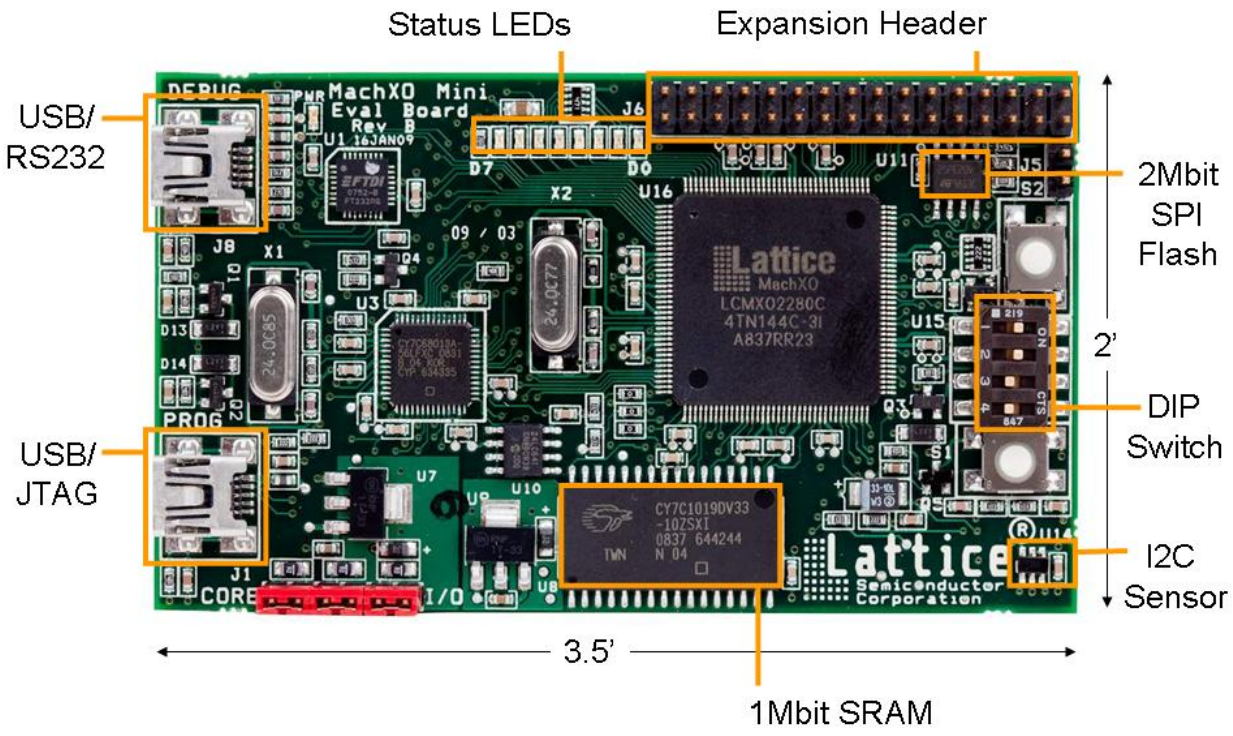

Figure 2 – MachXO Mini Development Board

Pre-tested System-on-Chip Design

The key to how the Mini Development board accelerates development is the pre-developed, pre-loaded System-on-Chip (SOC) design that can serve as the starting point for the development of a PLD implementing control plane functions. The following is a typical development flow and associated times, assuming that the appropriate software tools and drivers have been loaded onto the host PC:

1. Verify that the hardware and interfaces operate as expected. (Typically 5 minutes.) At this point the designer has *known good hardware*.
2. Confirm that the FPGA configuration file can reloaded into the MachXO using the ispVM programming software. (Typically 5 minutes.)
3. Recompile the design using ispLEVER design software and confirm that the resulting configuration operates as expected. (Typically 10 minutes.) At this point the designer has a *known good starting point*.
4. Make the first modifications to the design, recompile and load the new configuration. (Typically 60 minutes, although the actual time depends on the number of modifications that are made.)

By following this development flow, weeks of initial development can be reduced to hours. An overview of the SOC design is shown in Figure 3. As can be seen the design consists of six interfaces and a microprocessor interconnected by an on-chip bus.

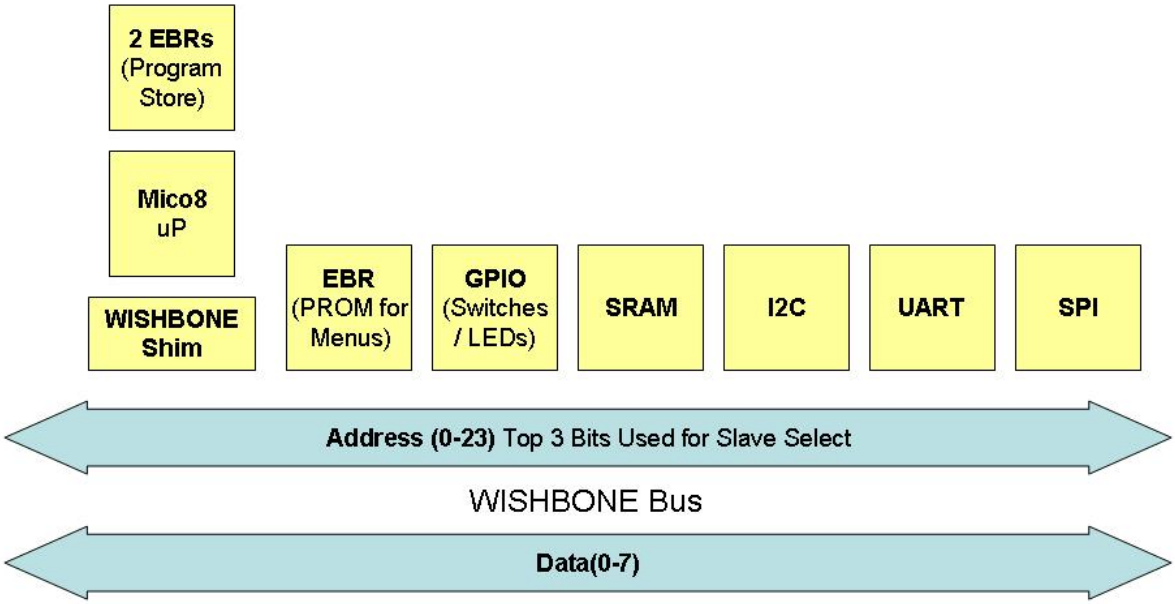

Figure 3 – MachXO Mini Board Pre-loaded SOC Design

The center of this design is the on-chip bus. The WISHBONE standard is used: WISHBONE is an Open Source bus standard that provides an ideal infrastructure for SOC designs. Benefits of the WISHBONE standard include:

- WISHBONE is Open Source and so can be used with a variety of FPGAs or ASICs.
- There is a large number of Open Source IPs available with WISHBONE interfaces.
- Simple but flexible definition allows designers to trade complexity for performance.

As the performance demands of many control plane PLDs are relatively low, the WISHBONE bus is implemented as a shared bus with the microprocessor acting as the master. The address bus is 24 bits with the top 3 bits reserved for generating the slave select signals.

The LatticeMico8 (Reference design RD1026) and the associated WISHBONE adapter (RD1043) are used for the processor that forms the core of the design. The Mico8 has the benefit of being Open Source. This is important because often a substantial amount of code is developed for embedded processors. The Open Source nature of the LatticeMico8 means that it can be implemented in a variety of FPGA fabrics or ASICs, protecting investments in embedded code.

The Mico8 has an 8-bit data path, 18-bit wide instructions and either 16 or 32 general purpose registers. External memory is paged with the first 8-bits of address included in the relevant op codes. The other 16-bits of the address bus are controllable with registers R15 and R16. Program memory can be set between 512 and 4096 deep. In this case up to 1024 locations of 18 bits are used. Program memory is stored within two Embedded Block RAMs (EBR.) These can be initialized to the desired contents at device start-up. Note that each MachXO EBR supports up to 1024 words of 9-bits. The Mico8 uses approximately 300 LUTs.

New code for the Mico8 can be easily generated by modifying the assembly listings that are provided as part of the project files. Modified code can then be passed through the LatticeMico8 tool chain and the appropriate HEX files generated. These HEX files can be used by the ispLEVER tool to appropriately initialize the EBRs used for program storage. The Mico8 tool chain is also Open Source, with both source and executable forms downloadable from the Lattice website.

The other EBR memory is connected to the WISHBONE bus. This EBR is configured as 1024 8-bit words. The space is used to share data for the various menus that are displayed by the Mico8 via the RS232 interface.

The SPI interface utilizes a modified version of Lattice's SPI WISHBONE Controller reference design (RD1044). The SPI interface connects an external 2 Mbit SPI Flash memory to the WISHBONE bus. The SPI WISHBONE controller can be used to control up to eight slaves if required. Receive and transmit resisters are configurable from 1 to 32-bits long, although in this case 8-bit registers are used. Registers are double buffered to allow data to be received or sent while the microprocessor is servicing the interrupt request to clear the other buffer. This reference design uses approximately 113 LUTs. The I2C interface uses the Lattice I2C Master with WISHBONE Bus interface reference design (RD1046). The design supports 7- or 10-bit addressing modes and an 8-bit receive / transmit buffer. The I2C reference design uses approximately 234 LUTs.

The UART design utilizes the Lattice WISHBONE UART reference design (RD1042.) This design implements a flexible UART with similar features to the common standalone NS16450 UART. The design consumes around 291 LUTs. The SRAM interface is relatively simple and was coded for this project. It is a good example of a simple WISHBONE peripheral.

Pulling It All Together

The goal of the MachXO Mini Development board is to accelerate development time for designs of the type typically found in small, non-volatile FPGAs. In order to achieve this, all the information that describes the SOC design discussed in this white paper is downloadable as a single package from the Lattice website. However, in addition to the documentation and source files, a complete ispLEVER project is also provided for the SOC design discussed, as well as for demonstrations of TransFR and the use of sleep mode to reduce operating power. Provided information includes the project file and all necessary preferences and constraints so that the design will compile the first time.

So, if its time to overhaul the Control PLD design being used in your system, take a look at some of the development systems and pre-tested designs that are out there. It could save weeks off your next design cycle.

###