IO Controller Build on XILINX FPGA with Microblaze Soft Processor based Approach

T .Jyothi and Mr. D.Kishore

Abstract-The microprocessors available for use in Xilinx Field Programmable Gate Arrays (FPGAs) with Xilinx EDK (Embedded Development Kit) software tools can be divided into two broad categories. There are soft-core microprocessors (MicroBlaze) and the hard-core embedded microprocessor (PowerPC). Xilinx Platform Studio (XPS) provides an integrated environment for creating software and hardware specification flows for embedded processor systems based on MicroBlaze and PowerPC processors. MicroBlaze is a 32-bit RISC soft-core (synthesizable) processor core that enables embedded developers to tune performance to match the requirements of target applications. XPS offers customization of tool flow configuration options and provides a graphical system editor for connection of processors, peripherals, and buses. XPS tool can create a simple processor system and the process of adding a custom OPB peripheral to that processor system by using the Import Peripheral Wizard. The EDK allows the user to incorporate soft- core processors (MicroBlaze or PicoBlaze) to interface the built-in PowerPC processors with the reconfigurable FPGA resources. Such processors can also be used to interface the hardware system to a variety of input/output peripheral devices

Keywords- Input/output peripheral devices, MicroBlaze, serial communication, Xilinx Platform Studio, OPB peripheral.

I. INTRODUCTION

FIELD-PROGRAMMABLE GATE ARRAYS (FPGA's) are flexible and reusable high-density circuits that can be easily re-configured by the designer, enabling the VLSI design/validation/simulation cycle to be performed more quickly and less expensive. Increasing device densities have prompted FPGA manufacturers, such as Xilinx and Altera, to incorporate larger embedded components, including multipliers, DSP blocks and even embedded processors. One of the recent architectural enhancements in the Xilinx Spartan, Virtex family architectures is the introduction of the MicroBlaze (Soft IP) and PowerPC405 hard-core embedded processor (1l].The MicroBlaze processor is a 32-bit Harvard Reduced Instruction Set Computer (RISC) architecture optimized for implementation in Xilinx FPGAs with separate 32-bit instruction and data buses running at full speed to execute programs and access data from both on-chip and external memory at the same time(1]. The reference paper implements in C/C++ and also the interfacing in it is done by using SDK which operates slower and it is more complex due to LCD Display.

T.Jyothi, Sphoorthy Engineering College, Student, M. Tech, E.C.E Dept., Hyderabad, India.Email: thumula.jyothi@gmail.com .Under the guidance of Mr. D.Kishore, Asst. prof., ECE Dept. Email:dapartikishore@gmail.com

II. BACKGROUND

The backbone of the architecture is a single-issue, 3-stage pipeline with 32 general-purpose registers (does not have any address registers like the Motorola 68000 Processor), an Arithmetic Logic Unit (ALU), a shift unit, and two levels of interrupt. This basic design can then be configured with more advanced features to tailor to the exact needs of the target embedded application such as: barrel shifter, divider, multiplier, single precision floating-point unit (FPU), instruction and data caches, exception handling, debug logic, Fast Simplex Link (FSL) interfaces and others [4].

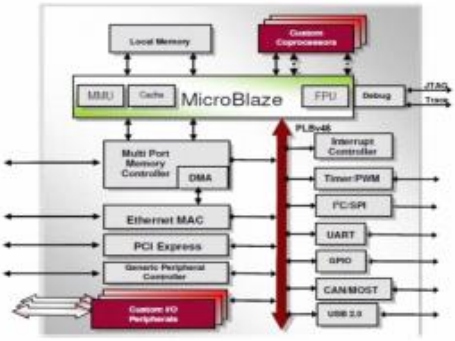


Figure I. Spartan 3E FPGA- MicroBlaze Processor Internal Architecture

This flexibility allows the user to balance the required performance of the target application against the logic area cost of the soft processor MicroBlaze also supports reset, interrupt, user exception, and break hardware exceptions. For interrupts, MicroBlaze supports only one external interrupt source (connecting to the Interrupt input port) (2). If multiple interrupts are needed, an interrupt controller must be used to handle multiple interrupt requests to MicroBlaze shown in figure(l).

An interrupt controller is available for use with the Xilinx Embedded Development Kit (EDK) software tools. The processor will only react to interrupts if the Interrupt Enable (IE) bit in the Machine Status Register (MSR) is set to 1. On an

interrupt the instruction in the execution stage will complete, while the instruction in the decode stage is replaced by a branch to the interrupt vector (address Ox10). The interrupt return address (the PC associated with the instruction in the decode stage at the time of the interrupt) is automatically loaded into general-purpose register. In addition, the processor also disables future interrupts by clearing the IE bit in the MSR. The IE bit is automatically set again when executing the RTlD instruction. Writing software to control t he MicroBlaze processor can be done in C/C++ language.The EDK tools have built in VHDL to generate the necessary machine code for the MicroBlaze processor.

III. RECENT TRENDS IN RECONFIGURABLE TECHNOLOGY

Due to the advancement in the fabrication technology and the increase in the density of logic blocks on FPGA, the use of FPGA is not limited anymore to debugging and prototyping digital electronic circuits. Due to the enormous parallelism achievable on FPGA and the increasing density of logic blocks, it is being used now as a replacement to ASIC solutions in a few applications where the time to market is critical and also entire embedded processor systems are implemented on these devices with soft core processors embedded in the system. With the advancement of Field Programmable Gate Arrays (FPGAs), like addition of substantial amounts of memory, a new trend has emerged in the design community to implement the microprocessors on the FPGAs [5). These kind of processors implemented on a reconfigurable fabric are called soft-processors or soft cores as the design of the microprocessor is available in the form of software bitstream which can be downloaded on FPGA by the user. The users have the choice of selecting the resources on the processor and the memory hierarchy. Soft cores are designed to meet minimum performance specifications over a range of technology implementations, even though core performance varies across technologies. Soft cores are technology independent and require only simulation and timing verification after synthesized to a target technology.

This reduces the design cycle development time by a major factor as compared to the development cycle for a hard core processor and has the advantage of customizing the soft core design for a specific application. Currently there are a number of soft cores available in the markets that are developed by giants in the field of reconfigurable devices like Xilinx and AItera. Xilinx has their own architecture named MicroBlaze in this arena and they have also ported the popular PowerPC architecture for use in embedded systems. These soft cores are available in the form of synthesized HDL modules or gate level netlists. System designers can embed these cores into their designs and optionally add peripherals to the core.

IV. EXPERIMENTAL SETUP

A. Xilinx Platform Studio

The Xilinx Platform Studio (XPS) is the development environment or GUI used for designing the hardware portion of your embedded processor system.

B. Embedded Development Kit

Xilinx Embedded Development Kit (EDK) is an integrated software tool suite for developing embedded systems with Xilinx MicroBlaze and PowerPC CPUs [12]. EDK includes a variety of tools and applications to assist the designer to develop an embedded system right from the hardware creation to final implementation of the system on an FPGA. System design consists of the creation of the hardware and software components of the embedded processor system and the creation of a verification component is optional. A typical embedded system design project involves: hardware platform creation, hardware platform verification (simulation), software platform creation, software application creation, and software verification. Base System Builder is the wizard that is used to automatically generate a hardware platform according to the user specifications that is defmed by the MHS (Microprocessor Hardware Specification) file. The MHS file defines the system architecture, peripherals and embedded processors [9]. The Platform Generation tool creates the hardware platform using the MHS file as input. The software platform is defmed by MSS (Microprocessor Software Specification) file which defines driver and library customization parameters for peripherals, processor customization parameters, standard 110 devices, interrupt handler routines, and other software related routines. The MSS file is an input to the Library Generator tool for customization of drivers, libraries and interrupts handlers.

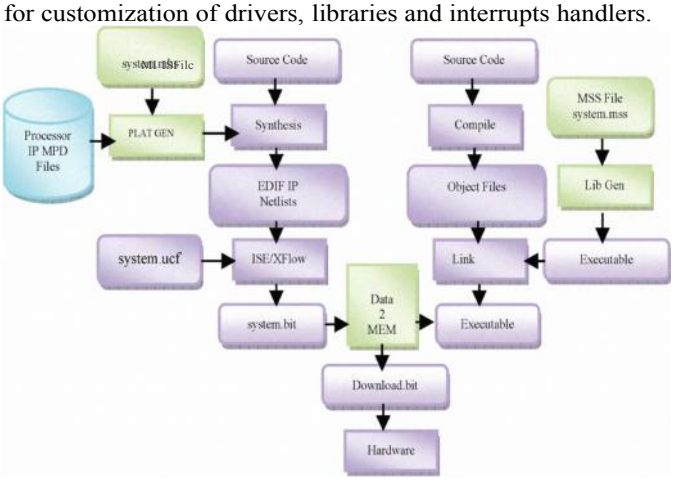


Figure 2. Embedded Development Kit Design Flow

The creation of the verification platform is optional and is based on the hardware platform. The MHS file is taken as an input by the Simgen tool to create simulation files for a specific simulator. Three types of simulation models can be generated by the Simgen tool: behavioral, structural and timing models. Some other useful tools available in EDK are Platform Studio which provides the GUI for creating the MHS and MSS files. Create / Import IP Wizard which allows the creation of the designer's own peripheral and import them into EDK

projects. Platform Generator customizes and generates the processor system in the form of hardware netlists. Library Generator tool configures libraries, device drivers, file systems and interrupt handlers for embedded processor system. Bitstream Initializer tool initializes the instruction memory of processors on the FPGA shown in figure2. GNU Compiler tools are used for compiling and linking application executables for each processor in the system [6]. There are two options available for debugging the application created using EDK namely: Xilinx Microprocessor Debug (XMD) for debugging the application software using a Microprocessor Debug Module (MDM) in the embedded processor system, and Software Debugger that invokes the software debugger corresponding to the compiler being used for the processor.

C. Software Development Kit

Xilinx Platform Studio Software Development Kit (SDK) is an integrated development environment, complimentary to XPS, that is used for VHDL software application creation and verification (8). SDK is built on the Eclipse open-source framework. Soft Development Kit (SDK) is a suite of tools that enables you to design a software application for selected Soft IP Cores in the Xilinx Embedded Development Kit (EDK).The software application can be written in a "VHDL" then the complete embedded processor system for user application will be completed, else debug & download the bit file into FPGA. Then FPGA behaves like processor implemented on it in a Xilinx Field Programmable Gate Array (FPGA) device [9].

V. DESIGN METHODOLOGY

The hardware developed for this project has been under the Xilinx EDK environment. Within this environment, the system developed contains a MicroBlaze soft-core processor connected to local memory bus (LMB) lines to Block RAM (BRAM) for the system's memory. The processor takes inputs and sends outputs to the general purpose input/output (GPIO) devices via the on-chip peripheral bus (OPB) line [6]. This is the general MicroBlaze soft-core processor set-up for the system. Beyond this is adding the necessary components for the UART, PS/2 keyboard port, and the RS-232 serial port. Next Phase the Netlist to be generated and add the software applicat ion project by following the EDK flow. By selecting Generate Netlist from the Hardware menu, the Xilinx EDK will begin to compile the hardware. Once the hardware assembles with zero errors, next select Generate Libraries and BSPs from the Software menu. Then, software application added to the project.Xilinx EDK environment supports the VHDL Language and compile it to the MicroBlaze soft-core processor [7].

VI. IMPLEMENTATION

This Paper is implemented in Spartan3E Starter Kit (XC3S500E-Device Family, FG320-Package, and 4- Speed Grade). General Purpose Input Output's (GPIO's) Peripheral Devices like, Dip Switches, Push Buttons, Led's, Seven Segment Display shown in figure 3, are controlled through Serial Communication (UART) via RS-232 interface using MicroBlaze Processor [51.AII the Peripheral Devices can control through commands from an external keyboard using RS-232 via Data Communication Equipment (DCE) interface. UART is a soft IP core available in Spartan FPGA fabric, also it has inbuilt embedded processor connected with OPB & PLB buses. MicroBlaze processor is a Soft IP which is available in Xilinx Platform Studio Tool [11].

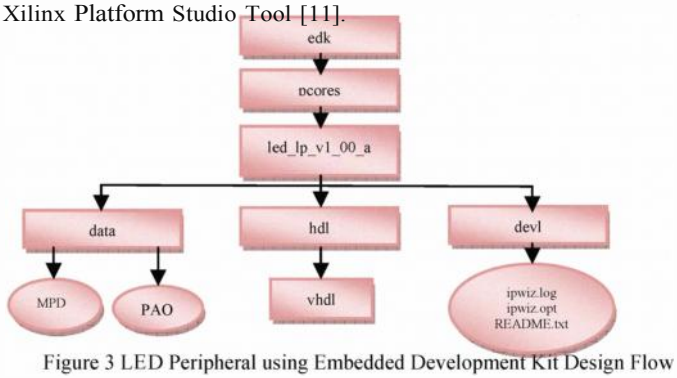
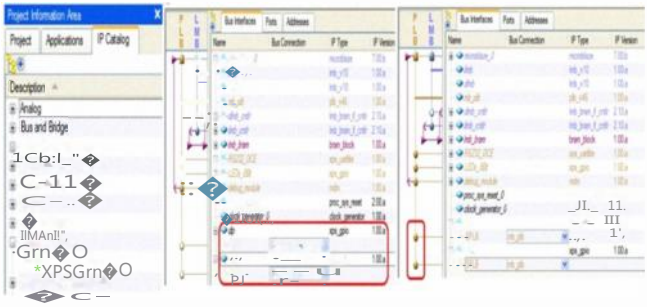


Figure 3 LED Peripheral using Embedded Development Kit Design Flow

While creating a new project using Xilinx Platform Studio-Embedded Development Kit through Base System Builder wizard user can incorporate the wide variety of Soft IP cores and Peripherals according the application oriented view. The Spartan 3E FPGA Board effectively communicates with PC through RS232 serial interface[lO].GPIO Peripherals initialization is shown in below figure 4(a, b, c).



la)GPIO Initialization Ib) Peripherals without Bus Connection Ie) Peripherals with Bus Connection

Figure 4(a, b, c) GPIO Peripherals with & without Bus Connections

The general purpose I/O's like Led's, Dip Switches, Push Buttons, Seven Segment Display, and RS-232 Data Communication Equipment & Data Terminal Equipment etc., are Controlled through Commands from Pc. Commands may be either alphanumeric or characters from key board [12]. When user enters a command from keyboard particular Discrete Inputs & Discrete Outputs are on & off. The status of the Discrete Input Peripherals like (Dipswitches, Push buttons) and Discrete Output Peripherals like (Led's, Seven Segment Display) are ON, OFF and there hex values can be displayed in HyperTerminal. A 'VHDL' Program has been written for the above application to control particular Input /Output Peripheral Device through Serial

Communication using MicroBalze Processor. It can be done through address of particular I /0 peripheral Device. EDK Tools offers flexibility to users, can configure through the process of adding a custom XPS peripheral to a processor system by using the Create or Import Peripheral wizard [2]. In this paper LED and some other peripherals were created using Create or Import Peripheral wizard.

VII. RESULTS

Figure 5 Initializing the MicroBlaze Processor.

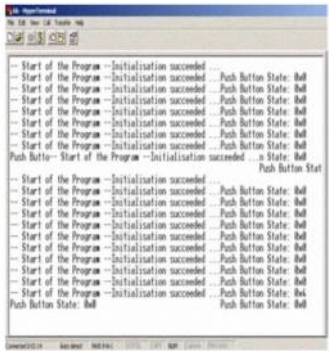


Figure 6 Initializing of Push Buttons and Display their Status.

Figure 7 Initializing the MicroBlaze Processor & Multiplication Operation is performed.

Figure 8 Multiplication Operation is performed and Result is displayed.

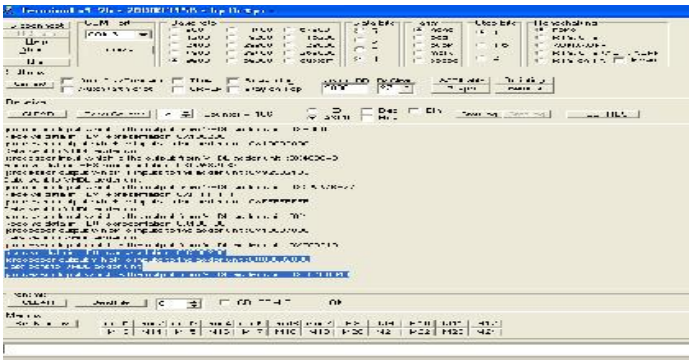


Figure 9 Division Operation is performed and Result is displayed.

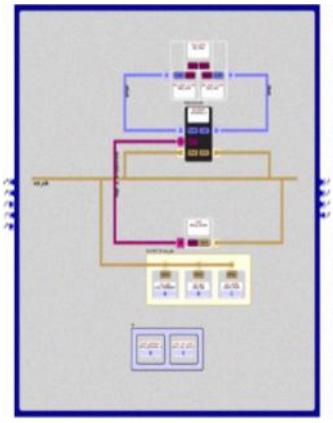


Figure 10.Xilinx Platform Studio Generated Block Diagram.

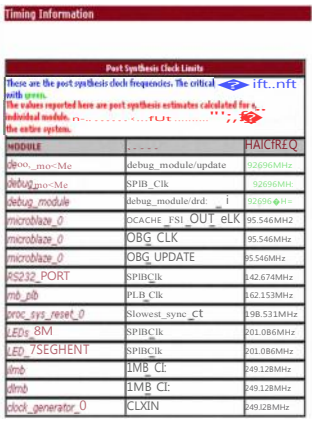


Figure 11.Post Synthesis Clock Report.

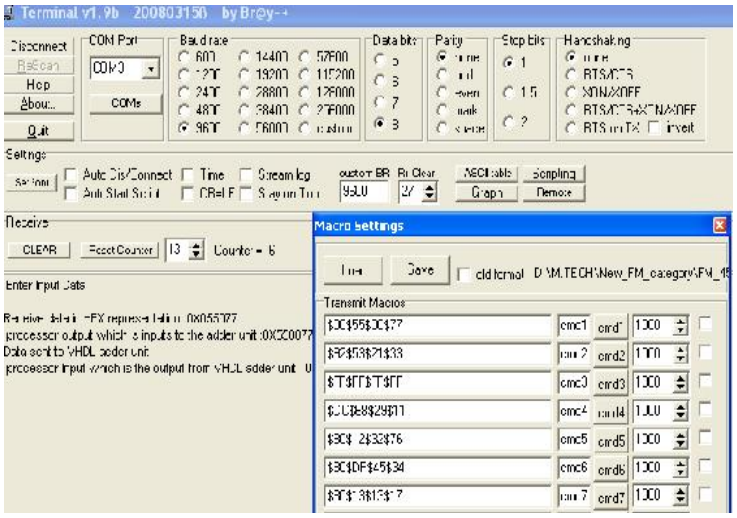


Figure 12. Displays the commands or inputs from .tmf file

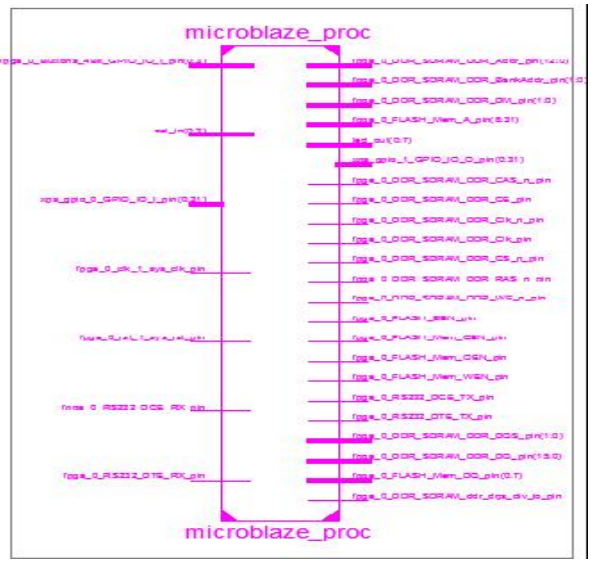


Figure 13.RTL schematic for the total project

VIII PHYSICAL VIEW OF IMPLEMENTATION RESULTS ON FPGA BOARDS

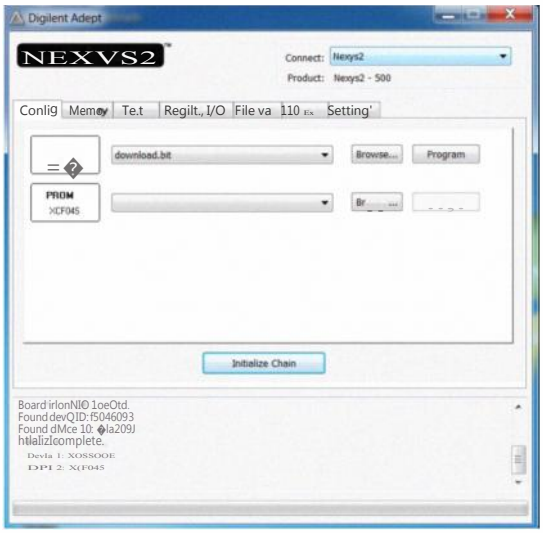


Figure 14. Nexys2Digilent Adept Software Settings for FPGA Board

Figure 15. Xilinx Spartan 3E FPGA Board with Serial (RS-232) Communication

CONCLUSION

The Base System Builder (BSB) can be used in XPS to create an embedded microprocessor project with the EOK tools. Several files, including an MHS file representing the processor system and a PBO file representing the schematic view, are created. The Import Peripheral Wizard can be used to integrate your user peripheral into an existing processor system. The wizard creates the necessary directory structure and adds the necessary files (MPD, PAO) to the project directory. After the peripheral is imported, user can use the peripheral in the design by using the XPS flow process. The software application can be written in a "C or C++" to control lIO peripherals through serial communication then the complete embedded processor system for user application will be completed and verified by generating and downloading the bit file into actual hardware.

REFERENCES

[I] Xilinx, http://www.xilinx.comlproducts/design.
[2] Xilinx Inc., PicoBlaze 8-bit Embedded Microcontroller UserGuide.http://www.xilinx.comlsupport/documentation/ userJluides/ugl29.
[3] Digilent Inc., Digilent Nexys2 Board Reference Manual
[4] Xilinx. Inc., Platform Specification Format Reference Manual, Embedded Development Kit EDK 9.2i
[5] Xilinx Inc. MicroBlaze Reference Manual, version 10.1.
[6] Xilinx Inc. Xilinx ISE and Xilinx EDK tools.
[7] Spartan-3 Starter Kit Board User Guide, Xilinx, Inc.
[8] Embedded System Tools Reference Manual, Xilinx, Inc
[9] Spartan-3 FPGA Family: Complete Data Sheet
[10] Platform Studio User Guide, Xilinx, Inc.
[II] Xilinx. Inc., Platform Specification Format Reference Manual, Embedded Development Kit EDK 9.2i
[12] Xilinx, Embedded System Example, XAPP433, version 2.2, 2006.