Proc. Annual Symposium on Research & Industrial Training, 03(2016) 50-56
Department of Electronics – Wayamba University of Sri Lanka

STUDY OF THE HARDWARE-SOFTWARE CORE DESIGN ON FPGA

U. L. S. L. Perera*, W. A. S. Wijesinghe

Department of Electronics, Wayamba University of Sri Lanka, Kuliyapitiya, Sri Lanka
*samalika.perera@gmail.com**

ABSTRACT

Field Programmable Gate Arrays (FPGAs) provide designers with the ability to design hardware circuits faster with a lower cost. Increase in FPGA logic resources in modern devices allows us to incorporate microprocessors on the same FPGA. Combining a microprocessor with the other hardware on a single chip enables embedded hardware-software co-design. In this study we investigate implementation of Microblaze processor on Xilinx Spartan 3E FPGA and use it to control the hardware system that calculate Greatest Common Divisor (GCD) of given two integers. The results show that the system can be implemented on the FPGA a lower usage of logic resources. The system is capable of running about 60 MHz.

Keywords: Hardware-software co-design, GCD, FPGA, Microblaze.

1. INTRODUCTION

Field Programmable Gate Arrays are digital integrated circuits (ICs) that contain configurable logic blocks along with configurable interconnects between these blocks. The word Field in the name refers to the ability of the gate arrays to be programmed for a specific function by the user instead of by the manufacturer of the device. With the advancement of Field Programmable Gate Arrays (FPGAs) a new trend of implementing the microprocessors on the FPGAs has emerged in the design community[1]. Microprocessors implement in FPGAs are called soft-processors or soft cores. System designers can embedded these cores into their designs and have the option to customize them as required. There are a several soft core processors are provided by the FPGA vendors. There are several soft-core processors designed for FPGAs, both commercially and freely. Xilinx, the world's largest manufacturer of FPGAs designed two soft-cores, the Picoblaze and the Microblaze [2] cores.

1.1. Microblaze Soft Processor Core

The Microblaze is a soft processor with 32-bit Harvard Reduced Instruction Set Computer (RISC) architecture optimized for implementation in Xilinx FPGAs. A designer can create a system incorporating a Microblaze using the Xilinx Platform Studio in which a designer can quickly build a Microblaze processor system by instantiating and configuring cores from the provided libraries.

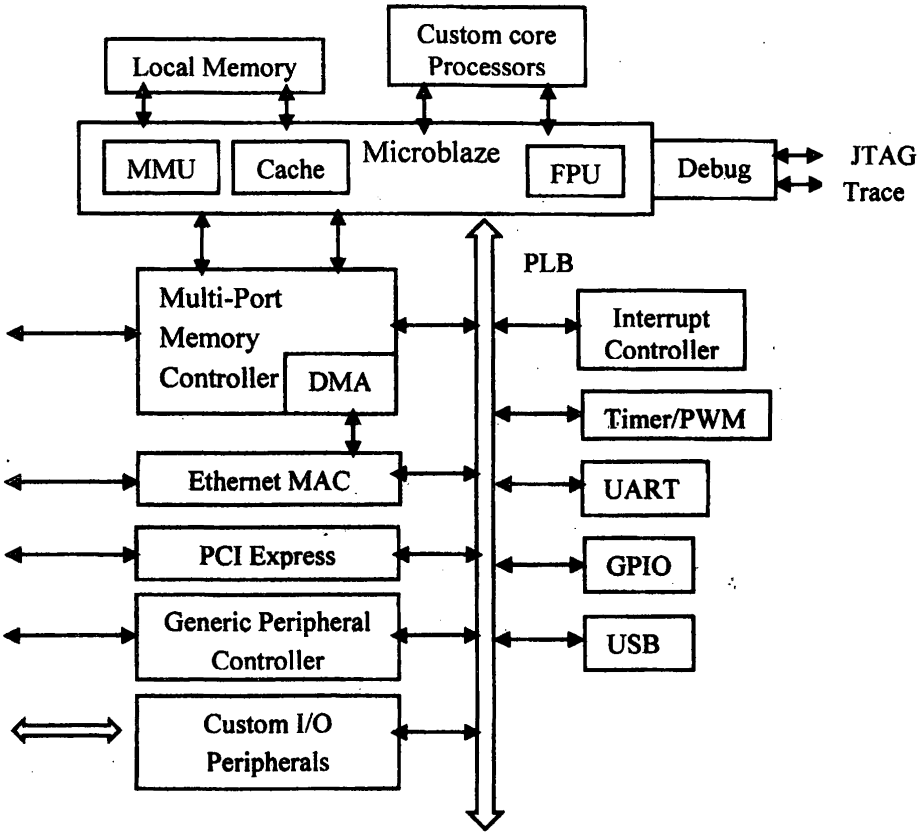

Figure 1: Microblaze Processor Architecture.

The main blocks of the Microblaze processor is shown in Figure 1. Some of the features available in this architecture can be selected by the designer as the requirement of the target embedded application. Because micro blaze is a soft-core microprocessor, any optional features, which are not used will not be implemented and will not take up any of the FPGAs resources. Besides the Microblaze, other components consists with architecture are GPIOs, a timer, an interrupt controller and a serial communication device UART[3,4].

1.2. Implementation of Hardware-Software Core Design

Software programs execute instructions in a serial and they are lacking parallelism. Some applications, such as real-time image processing, demand high performance which cannot be achieved from software implementation. FPGAs allow us to achieve parallelism for high performance over software implementations. However, implementing complex algorithms in FPGA hardware is not straightforward as software implementation. The way forward is to use

both hardware and software in a single embedded system that is hardware-software co-design. Hardware-software co-design is a process of achieving higher performance through partitioning functionality to hardware and software in an appropriate way. Designing a soft core processor is versatile task due enormous flexibility available to the designer. The Xilinx platform studio offers this flexibility to the designer. Hardware portion of the design is implemented by using XPS and software portion of the design is created by using Xilinx Software Development Kit. The soft-core processor is the core processor which is implemented on the hardware of Xilinx FPGA families according of to the peripherals required by the designer. In this project we focus on the study of hardware-software co-designing for the Xilinx Spartan 3E FPGA with the implementation of Greatest Common Devisor (GCD) algorithm.

2. EXPERIMENTAL METHODOLOGY

3.1. Greatest Common Devisor (GCD) Algorithm.

The greatest common divisor of two integers A and B is the greatest integer that divides both A and B with no remainder. For example, GCD(3,5)=1, GCD(12,60)=12 and GCD(12,90)=6. The greatest common divisor GCD (a, b, c...) can also be defined for three or more positive integers as the largest divisor shared by all of them [5]. Pseudocode for the algorithm is shown below. Figure 2 shows the flowchart diagram for calculating GCD.

```
Inputs int A, B;
Output int gcd;
while (A! = B) {
        if (A>B)
                A=A-B;
        else
                B =B-A;
}
gcd = A;
```

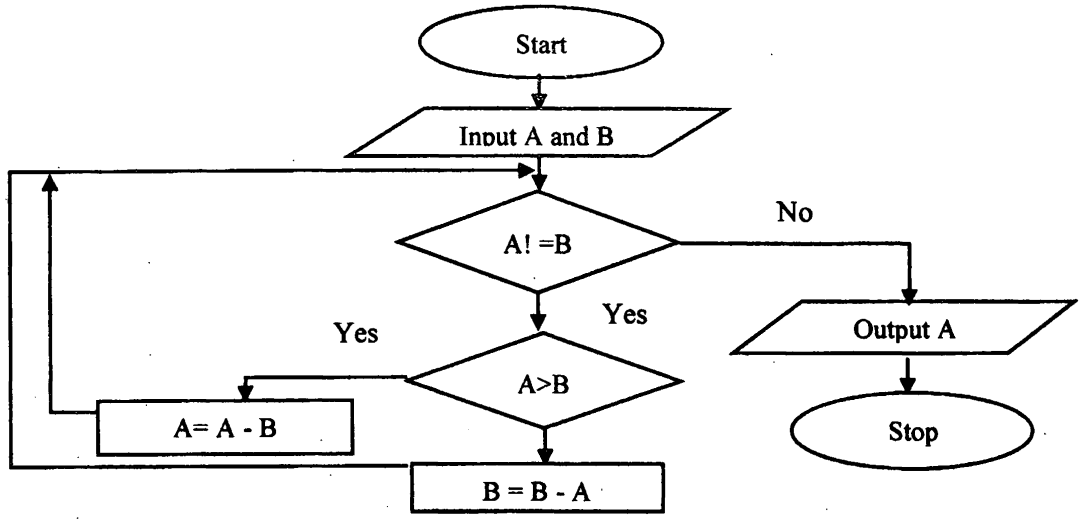

Figure 2: Flowchart to find GCD of two numbers

3.2. Implementation

To implement the GCD algorithm in hardware, first we described the algorithm using VHDL hardware description language and simulated the design. Once we have working module of GCD algorithm, we created a custom peripheral using Xilinx Platform Studio. The GCD peripheral is then connected to the Microblaze. A software program written in C will run on the Microblaze to control the functionality of the GCD peripheral. Xilinx Software Development Kit (XSDK) is used write the control software.

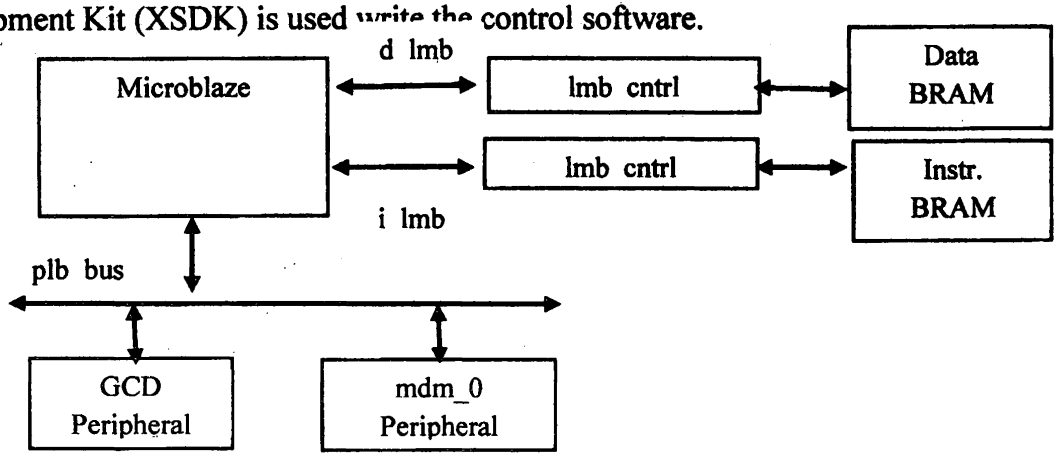

Figure 3: Block Diagram of the Design

Figure 3 shows the block diagram of the GCD implementation. It includes two Block RAMs (BRAM), one for instruction memory and the other for data memory, whose sizes are user defined. A local memory bus to BRAM interface connects the Microblaze with the instruction and data memories. The system also includes two peripherals connected via the PLB.The Microblaze Debug Module (mdm_0) is an IP that can be used for debugging the system. Further it creates a communication link between the FPGA board and the host PC, which we use for see the outputs of the embedded system [6].

3. RESULTS AND DISCUSSION

We implemented GCD algorithm in hardware and simulated to check the correct functionality. Figure 4 shows the simulated waveform of the hardware GCD algorithm.

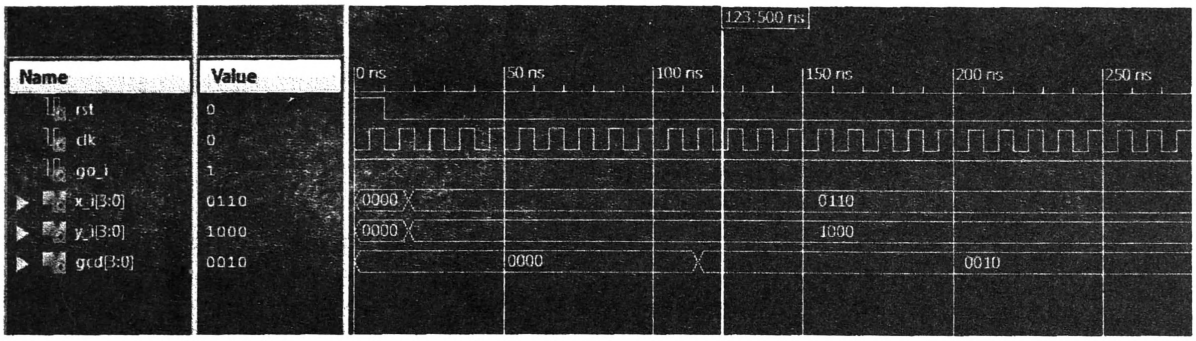

Figure 4: Simulation results for GCD implementation

After creating the custom peripheral for the GCD module to the Microblaze processor, programing file, bitstream, was generated. The bistream file contains the configuration details to implement the hardware system on the FPGA. Both the bitstream and the executable file of the software which runs on the Microblaze processor were downloaded to the FPGA on the Spartan 3E Starter Kit. The output of the test data were obtained from the host PC and samples are shown in Figure 5 and Figure 6.

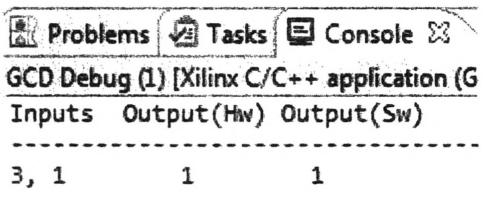

Figure 5: GCD Results on host PC

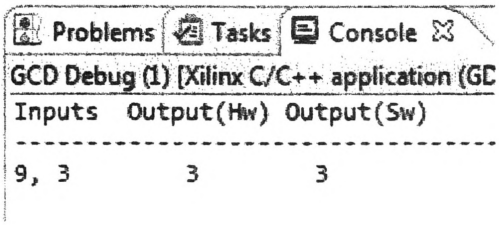

Figure 6: GCD Results on host PC

Implementation of Microblaze processor system on FPGA consumes fewer amounts of logic resources. Table 1 shows the resource utilization of the system.

Table 1: Device Utilization Summary

Device Utilization Summery			
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	1,303	9,312	13%
Number of 4 input LUTs	2,351	9,312	25%
Number of occupied Slices	1,722	4,656	36%
Number of bonded IOBs	8	232	3%

The number of flip-flops that are implemented in slices utilized by the design as 1,303(13%) and the number of flip-flops lookup tables implemented in slices utilized by the design as 2,351(25%). The number of occupied slices utilized by the design as 1,722(36%). Total number of pins configured with the inputs and outputs of the design as 8 from 232 pins. According to above table, utilization of resources of FPGA is lower than of its maximum capacity[7].According to the timing reports of the design, the maximum frequency that the system can run is 61.463MHz.

4. CONCLUSION

This paper described the study of hardware-software co-design for the Xilinx Spartan 3E FPGA. For our study, we considered the Greatest Common Divisor (GCD) algorithm and partitioned it to implement both in hardware and software. The calculation of the GCD algorithm was implemented in hardware while the software controller was implemented on Microblaze processor which was also implemented on FPGA hardware. In this study we have considered a very simple algorithm. The results show that the FPGA logic resource used for the system was very low. In future studies more complex algorithms will be considered for implementation as hardware-software co-design to achieve higher performance.

REFERENCES

[1]. Xilinx Inc, MicroBlaze Reference Manual, UG081 (v13.2).

[2]. Microblaze Soft Processor Core, Xilinx. http://www.xilinx.com/, [Accessed 2 March 2016].

[3].Spartan-3E FPGA Starter Kit Board User Guidehttp://www.xilinx.com/support/documentation/boards_and_kits/ug230.pdf.

[4]. David Sheldon, Rakesh Kumar, Frank Vahid, Dean Tullsen, Roman Lysecky, Conjoining Soft-Core FPGA Processors .

[5]GCD: http://mathworld.wolfram.com/GreatestCommonDivisor.html, [Accessed 7 February 2016].

[6]. Mohammed Zaheer1, A.M Khan, Development of Softcore Processor, 2015.

[7].XilinxTutorial,www.ee.iitb.ac.in/vlsi/resources/resource/logic_synthesis/xilinx_tutorial.pdf, [Accessed 5 March 2016].