ijecs
open access www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 5 May, 2014 Page No. 5741-5744

Design of an Application Specific Microcontroller

Naveen Kumar, Akshay Jindal, Prof. Alok Basu

B.Tech (CSE)
Bharati Vidyapeeth's College of Engg New Delhi
naveen_k098@yahoo.com
B.Tech (CSE)
Bharati Vidyapeeth's College of Engg New Delhi
akshayajddn@gmail.com
Professor
Bharati Vidyapeeth's College of Engg New Delhi
alok.basu@bharatividyapeeth.edu

ABSTRACT

This paper is concerned with the designing of an Application specific microcontroller that can be implemented on any FPGA. ASuC is a collection of several generic processor modules that can be integrated to facilitate any particular application in most optimized way. We are designing our project in Verilog HDL using XILINX Design Suite 14.1. The proposed architecture and simulation waveforms are presented in this paper.

General Terms

Application specific microcontroller, Verilog.

Keywords

ASuC, VLSI, HDL, ASIC, FPGA.

1. INTRODUCTION

In embedded system design, there are four prime factors that determines the performance and quality of any embedded system i.e. Cost, Size, Speed and Power consumption. Hardware designers focus on these parameters to optimize the performance of their system. In traditional embedded system, the complete design system is based on a processor and all the components are interfaced with the processor. Then, the processor is programmed to do all the operations based on various factors and devices interfaced with it. In this approach all the components such as timers, UART's and input-output interface are externally wired with the processor to form a complete system. But this approach resulted in increased cost, size and power consumption as all needed components is externally interfaced with the processor[1].

With the advent of VLSI technology, the integration of processor and other required components on a single IC package became possible, that truly revolutionized the world of embedded system design. This device i.e. Microcontroller (uC) miniaturized the world of embedded system by integrating almost complete system in a single chip IC. Despite of such level of integration involved in a microcontroller, it was just advancement to previous approach and provided just a generic solution to system design, it means that whole system needed to be designed around that generic microcontroller. It may be a better solution for simpler systems but as the complexity increases, microcontroller becomes a bottleneck for some applications. This consequently decreases the performance of entire system.

With the increase in complexity of embedded systems designs over time, designing each hardware component of the system from scratch soon became far too impractical and expensive for most designers. Therefore, the idea of pre-designed and pre-tested hardware description of modules became an attractive alternative.[2] Soft-core processors are microprocessors whose architecture and behavior are fully described using a synthesizable subset of a hardware description language (HDL). They can be synthesized for any Application-Specific Integrated Circuit (ASIC) or Field Programmable Gate Array (FPGA) technology; therefore they provide designers with substantial amount of flexibility[2].

In this paper, we propose a design of an Application specific microcontroller (ASuC) which is a soft-core implementation of a microcontroller that overcomes the limitations imposed by traditional microcontroller. With ASuC, one may create his own custom made microcontroller that can be completely optimized from MOS to Behavioral level.

2. PROPOSED ARCHITECTURE

The proposed concept ASuC presented in this paper can be visualized as shown in fig. 1. In this figure, various required modules are connected together using a common interface designed to facilitate a particular application.

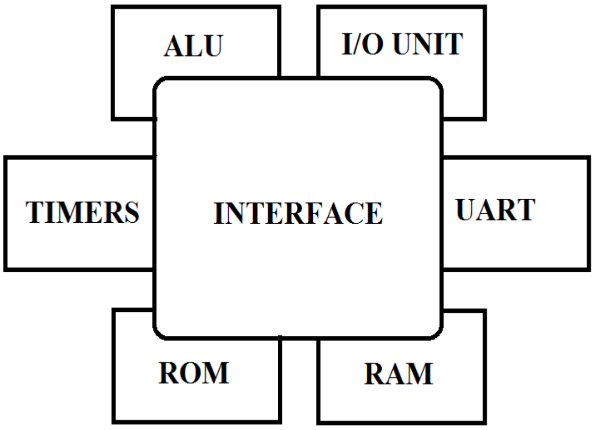

Fig 1: Simplified architecture of proposed system

2.1 Details of proposed system

A microcontroller is a miniaturized computer containing a processor along with ROM, RAM and other peripherals on a single integrated circuit. As microcontroller is a complete functional unit, it is readily used in almost all embedded applications.

ASuC is a hardware description model of different modules of a microcontroller which are usually found in a microcontroller. It proposes reconfigurable soft-core modules that can be interfaced together in any desirable architecture to produce highly optimized devices. The architecture of a traditional microcontroller is fixed and cannot be changed to suit any particular application, rather the entire system is developed around its fixed architecture. Whereas ASuC provides the freedom of including only those modules which are required and discarding redundant modules this only increases the overhead in various operations. Once ASuC is configured, it can be implemented on any reconfigurable device such as ASIC or FPGA depending on the complexity of the system.

As mentioned above ASuC are soft-core models of highly configurable microcontroller components which can be interfaced together, these soft-core modules holds many advantages over traditional processors. First, soft-core processors are flexible and can be customized for a specific application with relative ease. Second, since soft-core processors are technology independent and can be synthesized for any given target ASIC or FPGA technology, they are therefore more immune to becoming obsolete when compared with circuit or logic level descriptions of a processor. Finally, since a soft-core processor's architecture and behavior are described at a higher abstraction level using an HDL, it becomes much easier to understand the overall design[3].

In many applications, ASuC may provide several advantages over traditional processors such as flexibility, reduced cost, platform independence, and immunity to obsolescence and reduced processing time.

For example, suppose a system that is intended to perform FFT (Fast Fourier Transform) over a large amount of real time data. So if a traditional sequential processor is used for such application, it will perform single FFT operation in many machine cycles as this operation requires many complex additions and multiplication over large array of data. So in order to improve its performance, one will choose costlier processors with pipelining and more internal memory. This may increase the speed of data processing but it will significantly increase cost, power consumption and size. In order to further increase the performance one will have to move for costlier options like superscalar processors which will increase all the prime factors.

In such situations ASuC can provide significantly better solution. It will allow developers to develop an FFT specific unit that can be programmed to include several FFT sub modules, which all receives parallel data and produces parallel output in few machine cycles as shown in fig.2. So using ASuC, the above operation can be executed much faster, which will be a significant increase in performance as compared to traditional processors.

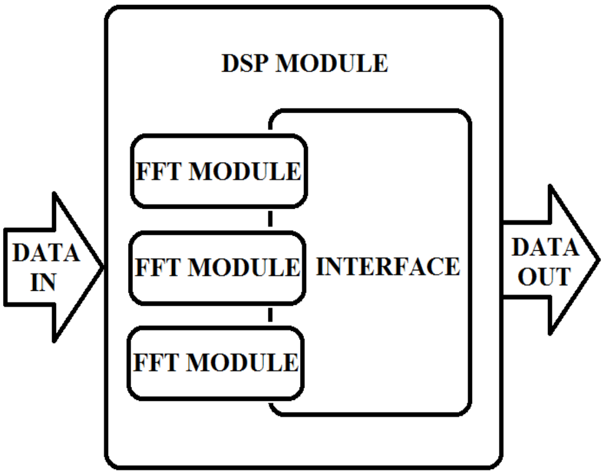

Fig 2: Proposed SIMD DSP sub module

3. IMPLEMENTATION

In this paper we propose to implement our ASuC design in Verilog HDL. We divide our design into three primary modules i.e. Arithmetic Logic Unit, Control Unit and Peripherals.

3.1 Arithmetic and Logic Unit

ALU is fundamental building block of the central processing unit of a microcontroller, which is responsible for all operations performed inside a microcontroller. The following figure depicts the design of our ALU.

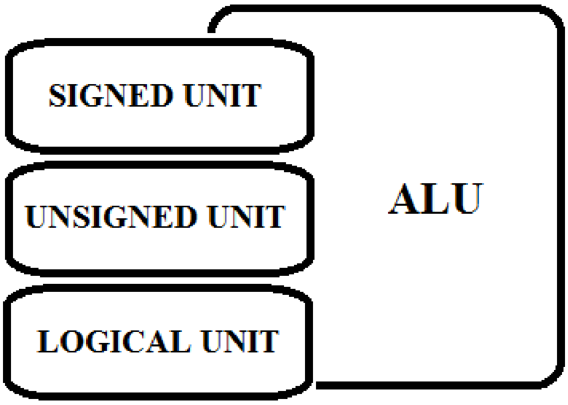

Fig 3: Components of ALU

3.1.1 Unsigned ALU

In our design to process unsigned data, we designed Unsigned ALU sub module. This module is responsible for all arithmetic and logical operations on signed data; fig 4 depicts our model of an unsigned ALU. This module receives two 8-bit operands, one from accumulator and one either directly from instruction or from RAM. The input sl0, sl1, sl2, sl3, sl4 are the control signals received from control unit, these inputs are used to select a particular operation to be performed on data. After receiving a particular input, this module generate two outputs i.e. carry out and output. This output is then fed to an 8 bit Accumulator.

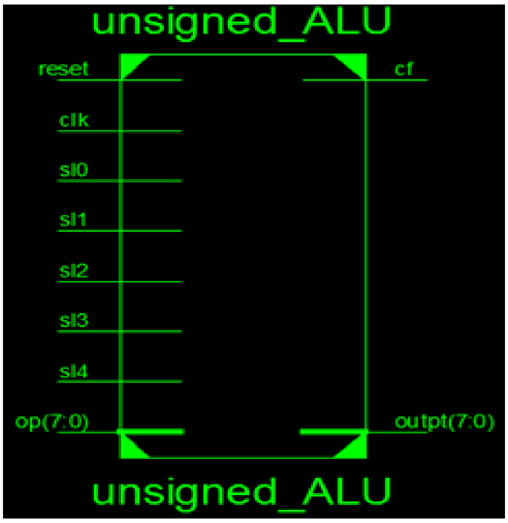

Fig 4: Synthesized Block Diagram of Unsigned ALU

3.1.2 Signed ALU

We designed signed ALU to process signed data, this module also receives two 8-bit inputs and generate up to 16-bit data output depending on the operation selected.

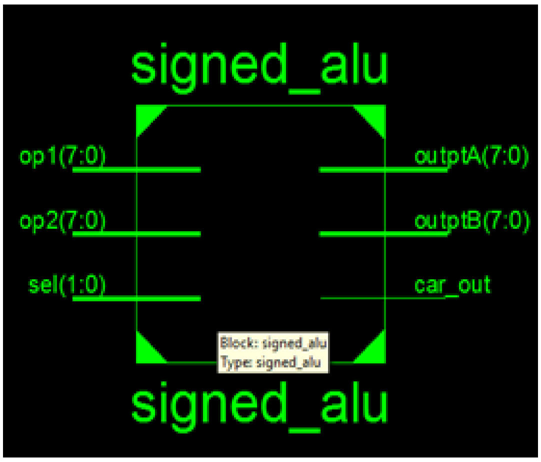

Fig 5: Synthesized Block Diagram of Signed ALU

3.2 Control Unit

Control unit is the most important module of our ASuC, which is designed to adapt according to dynamic architecture of ASuC. This module receives operation in form of 8-bit opcode, along with clock and other asynchronous signals and generates various control signals to control the operations of whole microcontroller. In fig 6, the block diagram of synthesized control unit is show with various inputs and outputs.

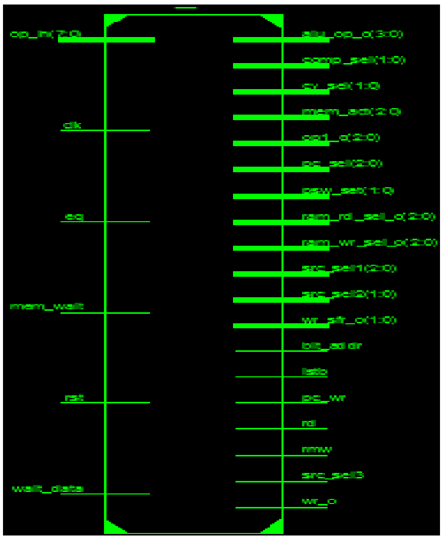

Fig 6: Synthesized Block Diagram of Control Unit

3.3 Peripherals

The peripherals are the supporting components that are embedded inside microcontroller to provide additional facilities. In our design we provided two supporting peripherals i.e. Universal asynchronous receiver transmitter and timer/counter.

3.3.1 UART

UART is a supporting component of a microcontroller which is used for serial communication with external devices or peripherals[4]. The fig 7 gives an overview of our UART module. This module receives parallel data from CPU and sends data serially to external devices on a configurable baud rate. Similarly, it also revives serial data asynchronously from outside devices and sends as parallel data to the CPU.

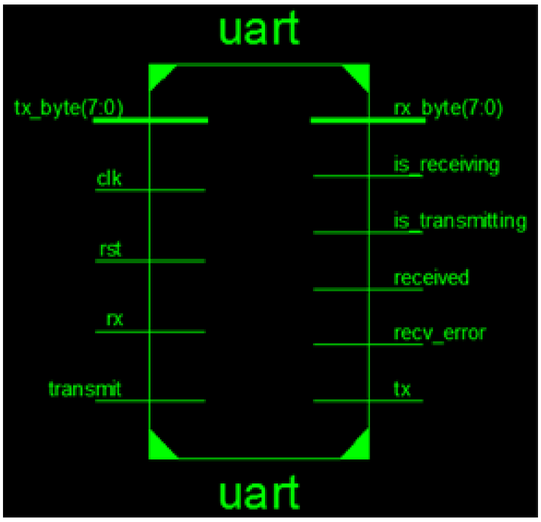

Fig 7: Synthesized Block Diagram of UART

3.3.2 Timer/Counter

Embedded systems often require mechanisms for counting the occurrence of events and for performing tasks at regular intervals.[5] Embedded processors are often equipped with hardware support for this functionality. By providing this support, we can ensure that events are not missed and that timing of behavior occurs at regular intervals[5]. To fulfill these requirements, timer/counter is used in microcontrollers. In fig 4, synthesized block diagram of timer/counter is given. This module works in two modes i.e. timer and counter. In timer mode, this module is used to generate accurate time delay and in counter mode, this module is used to count the occurrence of any event on any particular input pin.

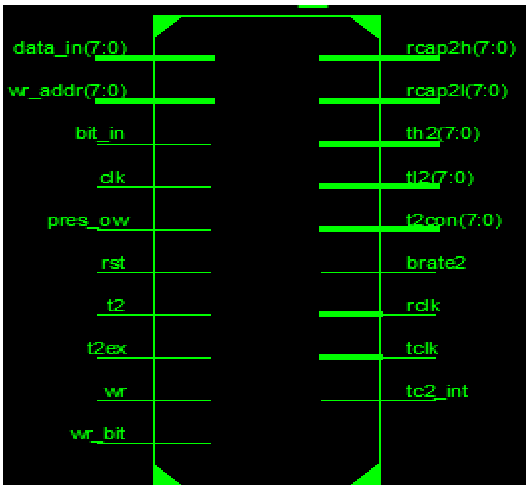

Fig.4: Synthesized Block Diagram of Timer/Counter

4. CONCLUSION

In this paper we have proposed designed of a customizable microcontroller, which can be integrated according to any architecture to facilitate any particular application.

Currently, we have developed basic CPU core, along with memory resources and other supporting peripherals. Our goal is to further advance the knowledge of our design and implementation of ASuC.

5. REFERENCES

[1] John L. Hennessy, and David A. Patterson, B. 2006. "Computer Architecture A Quantitative Approach, 4th Edition" ISBN-13: 978-1558605961.

[2] Jason G. Tong, Ian D. L. Anderson and Mohammed A. S. Khalid, J. 2006. "Soft-Core Processors for Embedded Systems", ISM-2006.

[3] Vijay G. Savani, Piyush M. Bhatasana and Akash I. Mecwan, J. 2012. "Implementation of Data Compression Algorithms on FPGA using Soft-core Processor", ISSN 0976-4860

[4] Muhammad A. Mazidi, Janice G. Mazidi and Rolin D. McKinlay, B. 2008."The 8051 Microcontroller and Embedded Systems", ISBN 978-81-317-1026-5

[5] Andrew H.Fagg, L. 2005 "Timer/Counter Notes".