16 Bit Data Processor Implemented by FSM

Dheera Chaudhary [1], Rajiv Sharma [2], Seema [3]
[1, 2, 3] Computer Science and Engineering, Maharshi Dayanand University, Rohtak, India.

Abstract – The goal of the presented work is to design a 16-bit data processor implemented by FSM (Finite State Machine) which can be used in several applications. Motivation for my research is simple interest in the processor and also shows that how Digital Circuit works. This project is to design and simulate a processor using a specialized Hardware Descriptive Language, which is used for arithmetic and logical operation.

Index Terms – FSM, RISC, ASM, ALU, VHDL

1. INTRODUCTION

Processors are the heart of all "smart" devices, whether they be electronic devices or otherwise. Their smartness comes as a direct result of the decisions and controls that processor makes. For example, one can usually consider a car to be an electronic device. However, it certainly has many complex, smart electronic systems, such as the anti-lock brakes and the fuel-injection systems. Each of these systems is controlled by a processor.

There are generally two types of processors: general-purpose processors and dedicated processors. General-purpose processors such as the Pentium CPU can perform different tasks under the control of software instructions. General-purpose processors are used in all personal computers. Dedicated processors also known as application-specific integrated circuits (ASICs), on the other hand, are designed to perform just one specific task.

For example, inside the cell phone, there is a dedicated processor that controls its entire operation. The embedded processor inside the cell phone does nothing but controls the operation of the phone. Dedicated processors are, therefore, usually much small and not as complex as general-purpose processors.

1.1 FINITE STATE MACHINE(FSM)

Implementing state machines in VHDL is fun and easily provided to stick to some fairly well-established forms. This style for state machine coding given here is not intended to be especially clever. They are intended to be portable, easily understandable, clean, and give consistent results with almost any synthesis tool. The format for coding state machines follows the general structure for a state machine.

Finite-state machine (FSM) or finite-state automation is a mathematical abstraction sometimes used to design digital logic or computer programs shown in figure.

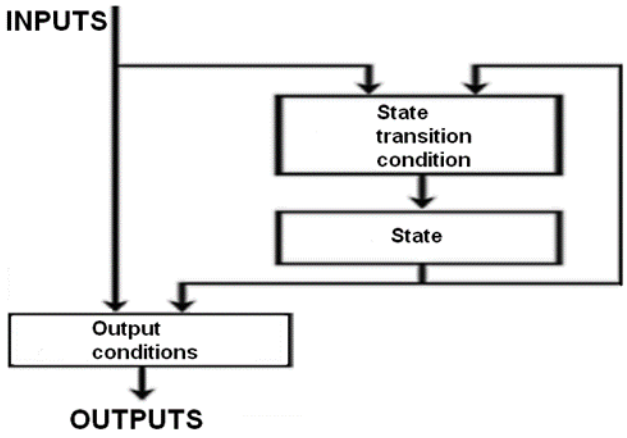

Figure: -Finite State Machine

A FSM is defined by the following:

- a finite non empty set of states
- an initial state
- a finite non empty set of distinct input events or their categories
- state transitions
- actions

Every digital logic circuit is categorized as either a combinational circuit or a sequential circuit. A combinational circuit is one where the output of the circuit is dependent only on the current inputs to the circuit. For example, an adder circuit is a combinational circuit. It takes two numbers as inputs. The adder evaluates the sum of these two numbers and outputs the result. A sequential circuit, on the other hand, is dependent not only on the current inputs, but also on all the previous inputs. In other words, a sequential circuit has to remember its past history. For example, the up-channel button on a TV remote is part of a sequential circuit. Pressing the up-channel button is the input to the circuit. However, just having this input is not enough for the circuit to determine what TV channel to display next. In addition to the up-channel button input, the circuit must also know the current channel that is being displayed, which is the history. If the current channel is channel 3, then pressing the up-channel button will change the channel to channel 4.

Since sequential circuits are dependent on the history, they must therefore contain memory elements for remembering the history; whereas combinational circuits do not have memory elements. Examples of combinational circuit's inside the microprocessor include the next-state logic and output logic in the control unit, and the ALU, multiplexers, tri-state buffers, and comparators in the data path. Examples of sequential circuits include the register for the state memory in the controller and the registers in the data path. Model is also a sequential circuit. Regardless of whether a circuit is combinational or sequential, they are all made up of the three basic logic gates: AND, OR, and NOT gates. From these three basic gates, the most powerful computer can be made. Furthermore, these basic gates are built using transistors the fundamental building blocks for all digital logic circuits. Transistors are just electronic binary switches that can be turned on or off. The on and off states of a transistor are used to represent the two binary values: 1 and 0.

It is a behavior model composed of a finite number of states, transitions between those states, and actions, similar to a flow graph in which one can inspect the way logic runs when certain conditions are met. It has finite internal memory, an input feature that reads symbols in a sequence, one at a time without going backward; and an output feature, which may be in the form of a user interface, once the model is implemented. The operation of an FSM begins from one of the states (called a start state), goes through transitions depending on input to different states and can end in any of those available, however only a certain set of states mark a successful flow of operation (called accept states).

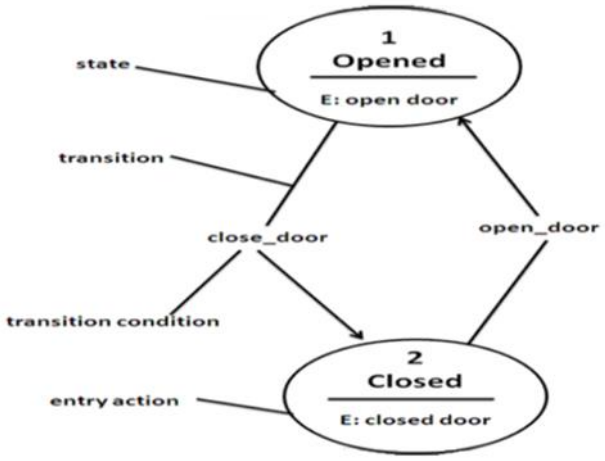

Figure:-FSM with opened and closed door

An FSM can be represented using a (or state transition diagram) as in figure. A state diagram for a door that can only be opened and closed.

The control unit is a finite state machine (FSM). It is a machine that executes by going from one state to another and that there are only a finite number of states for the machine to go to. The control unit is made up of three parts: the next-state logic, the state memory, and the output logic. The purpose of the state memory is to remember the current state that the FSM is in. The next-state logic is the circuit for determining what the next state should be for the machine. And the output logic is the circuit for generating the actual control signals for controlling the data path.

1.2 TYPES OF FSM

Finite-state machines are classified into two main types:

Moore type FSM: A Moore type FSM is one where the output of the machine is dependent only on the current state. Three processes for Moore machine:

- One process is used to model the state registers to decide the next state.
- Second process models to update the next state.
- Third process models the output logic.
- Or second and third combined into one process.

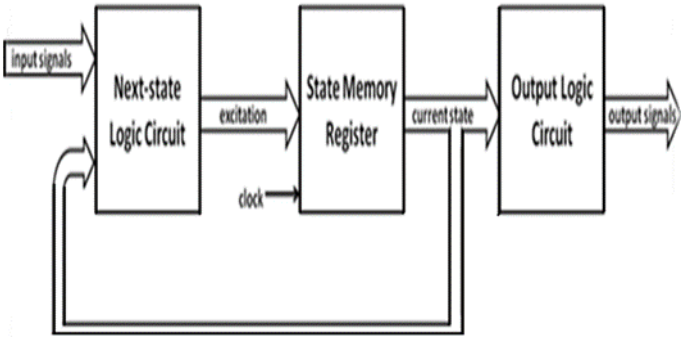

Figure: -Moore Type FSM

Mealy type FSM: A Mealy type FSM is one where the output is dependent on both the current state and the input signals. This FSM is shown in figure 3.4.

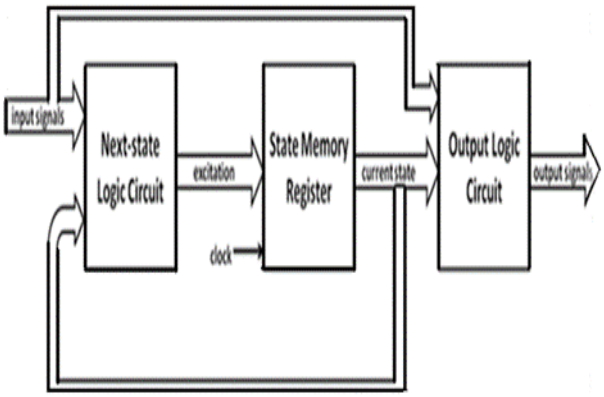

Figure: -Mealy Type FSM

Two processes for Mealy machine:

- One process is used to model the state registers to decide the next state.
- Second process models to update the next state and output logic.

2. LITERATURE SURVEY

IEEE Papers Review with Critique

The Sweet-16 Processor was introduced by Venelin Angelov, Volker Lindenstruth in Jan 2009. A 16-bit fully functional one cycle RISC processor was developed for the illustration and use in computer architecture classes. It is simple enough so that it can be designed by entry level students without any prerequisites. In addition, its architecture is optimized to support the computer architecture curriculum with concrete practical hands-on experiments. The processor is subsequently used for assembly language work. The architecture has upgrade options for advanced studies, such as pipelining, interrupts, etc. This paper presents the architecture of unique processor Sweet-16.

16-Bit Teaching Microprocessor Design and Application was proposed by Xiao Tiejun, Liu Fang in Feb 2008. The existing commercial microprocessors are provided as black box units, with which users are unable to monitor internal signals and operation process and could neither modify the original structure. So they are unsuitable for users. In order to solve this problem, the present paper designs a simplified but fully functional 16-bit teaching microprocessor using the micro programmed control method. The microprocessor is easy to modify as a VHDL and Verilog HDL modular, hierarchical description gives access to every internal signal and is open for user modifications. The design of ALU was optimized so that it consumes fewer resources. The microprocessor was implemented on Xilinx FPGA and has been applied in the Computer Organization and Architecture.

8 Bit and 16 Bit Multi Microprocessor Systems has been proposed by Soegijardjo Soegijok, Lie Tjeng Chiao, Djaka Sundan. This paper describes the design and implementation of two types of multi microprocessor systems.

A twin multiprocessor system which was first developed and consists of two modules of exactly the same configuration. Each module has a pair of 8-bit 6802 monolithic microprocessors in a master-slave configuration. A multi microprocessor system based on 16-bit 68000 microprocessors implemented recently, consists of three processing elements, two 16 Kbytes global memories interconnected through two global buses.

James E. Cross and Raphael A. Soetan introduced Teaching Microprocessor Design using the 8086 Microprocessor. The 16-bit microprocessor has essentially replaced the 8-bit microprocessor as standard for engineering design. Those instructors' microprocessors are obligated to respond to the changing times. The data processing capability of the 16-bit machine is an order of magnitude above that of its 8-bit predecessor. However, in many cases, a 16-bit microprocessor offers practically no advantage over an 8-bit machine for I/O processing. Some comparisons will be made between the 8086 and the 8085 microprocessors. However, the thrust of this presentation will be to share some of the experiences of the authors in teaching a laboratory using the 8086 microprocessors.

Morris Chang proposed the processor on Teaching Top-down Design Using VHDL and CPLD. This paper presents a teaching experience in using VHDL and CPLD in the senior digital design course. The courses focus on the top-down design methodology through hands-on experiments. The industrial available tool such as Altera, made possible through Altera's University Program, provide our students a smooth transition from academic concepts to industrial practice. VHDL, the industrial standard language (IEEE-1076), is used as the design entry. Thus, the students are forced to learn the practical aspect of writing a synthesizable VHDL code. The hands-on weekly projects are exercised on the integrated CPLD design tool which has VHDL compiler, logic synthesizer, functional and timing simulator, floor plan editor and programmer. With the help of programmable devices, one can bypass the waiting period for IC fabrication and obtain ASIC designs after the devices have been programmed.

Design of an Algorithmic State Machine Controlled, Field Programmable Gate Array Based 16-bit Microprocessor proposed by Shirazy Md. Shorab Muslim, Zahir Uddin Ahmad in April 2007. A 16-bit microprocessor is designed by top-down design methodology which is controlled by an ASM State Machine) chart and fitted it into an FPGA (Field Programmable Gate Array). The verified simulation result and LDC post route simulation result has been shown in this paper. The synthesis result of the design is also described in this paper.

3. CONCLUSION AND FUTURE WORK

3.1 CONCLUSION

The 16-bit data processor implemented by FSM is described in terms of its architecture and its functional capabilities. This processor is modified by Very High Speed Integrated Circuit Hardware Description Language (VHDL) and gives access to every internal signal. The key architecture elements are being described, as well as the hardware block diagram and internal structure.

Design of a circuit to implement a state diagram requires sequential diagram, which consists of drawing an implementation model with a state register and combinational logic block. Finally design of a single-processor circuit to implement a program requires us to first schedule the program

statement into a complex state diagram, construct a data path from the diagram, create a new state diagram that replaces complex actions and conditions by data path control actions, and then design control path for the new state diagram using sequential design.

The design architecture is written in Very High Speed Integrated Circuit Hardware Description Language (VHDL) code using Xilinx ISE 9.2i tool for synthesis and simulation. Functional simulation shows that the processor executes for all the various instructions. All the results have been verified and found them too correctly.

3.2 FUTURE WORK

1. Improve the performance of design: Processor is designed for few instructions. It should have more strength full instruction set, which is capable to perform all the operations.

2. Increase the speed of Processor: By using pipelining, speed of processor can be improved.

3. Improve timing performance: By finding and optimizing the critical path of the design, timing performance can be improved.

4. Reduce the area and power consumption: Area and power consumption can be reduced by reducing logic utilization of the design.

REFERENCES

[1] J. Eyre and J. Bier, DSP Processors Hit the Mainstream, IEEE Micro, August 1988.

[2] Development and Directions in Computer Architecture Lipovski, G.J. Doty, K.L. University of Texas Aug. 1978 Volume: 11, Issue: 8. (Pp.54-67)

[3] Andrei-Sorin F., Corneliu B., 2010 "Savage 16-16-bit RISC Architecture General Purpose Microprocessor" in Proc. IEEE Journal. (Pp.3-8)

[4] Venelin Angelov, Volker L., 2009 "The Educational Processor Sweet-16" in Proc. IEEE Conference. (Pp.555-559)

[5] Gordon Bell, "RISC: Back to the Future?", Datamation, Vol. 32, No. 11, June 1, 1986 (Pp.96-108)

[6] Gin-Der Wu Kuei-Ting Kuo, "Dual-ALU structure processor for speech reorganization" Publication Date: 24-26, April 2006

[7] Tseyun Feng, Dharma P. Agrawal, "A Microprocessor-controlled asynchronous circuit switching network", 1979 (Pp.202-215)

[8] Mochizuki, A. Kitamura, T. Shirahama, "Design of a Microprocessor Datapath Using four valued Differential-Pair Circuits", May 2006 (Pp.14-21)

[9] Xiao Tiejun, Liu Fang, 2008 "16-bit Teaching Microprocessor Design and Application" in Proc. IEEE International Symposium on It in Medicine and Education. (Pp.160-163)

[10] Cross, J.E. and Soetan, R. A., 1988 "Teaching Microprocessor Design using the 8086 Microprocessor" in Proc. IEEE Conference on Southeastcon'88. (Pp. 175-180)

[11] Soegijardjo Soegijok , Lie Tjeng Chiao, Djaka Sundan, 1991, "8 Bit and 16 Bit Multi Microprocessor Systems" in Proc. IEEE Journal.

[12] Gerry Kane, MIPS R2000 RISC Architecture, Prentice Hall, Englewood Cliffs, 1987.

[13] Young-Ho Cha, Chang-Su Park, Gyeong-Yeon Cho and Hyek-Hwan Choi, 2000, "An Embedded 16 bit Microprocessor" in Proc. IEEE Journal.

[14] Tseyun Feng, Dharma P. Agrawal, 1979, "A microprocessor-controlled asynchronous circuit switching network" in Proc. IEEE Journal. (Pp.202-215)

[15] Gin-Der Wu, Kuei-Ting Kuo, April 2006, "Dual ALU structure processor for speech recognization" in Proc. IEEE Journal.