8-Bit RISC Processor using VHDL

A Course End Project Report Submitted in the
Partial Fulfillment of the Requirements
for the Award of the Degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS AND COMMUNICATION ENGINEERING

Submitted by

D.Abhinay	**20881A04E1**
G.Chandrakanth	**20881A04E5**

SUPERVISOR
Dr.S.Rajender
Dean of Academics

Department of Electronics and Communication Engineering
VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD
An Autonomous Institute, Affiliated to JNTUH

February, 2022

VARDHAMAN COLLEGE OF ENGINEERING, HYDERABAD

An Autonomous Institute, Affiliated to JNTUH

Department of Electronics and Communication Engineering

CERTIFICATE

This is to certify that the project titled **8-Bit RISC Processor using VHDL** is carried out by

D.Abhinay	**20881A04E1**
G.Chandrakanth	**20881A04E5**

in partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Electronics and Communication Engineering** during the year 2021-22.

Signature of the Supervisor
Dr.S.Rajender
Dean of Academics

Signature of the HOD
Dr. G.A.E. Satish Kumar
Professor and Head, ECE

Kacharam (V), Shamshabad (M), Ranga Reddy (Dist.)–501218, Hyderabad, T.S.
Ph: 08413-253335, 253201, Fax: 08413-253482, www.vardhaman.org

Acknowledgement

The satisfaction that accompanies the successful completion of the task would be put incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crown all the efforts with success.

We wish to express our deep sense of gratitude to **Dr.S.Rajender** , Designation and Course Instructor, Department of Electronics and Communication Engineering, Vardhaman College of Engineering, for his able guidance and useful suggestions, which helped us in completing the Course End project in time.

We are particularly thankful to **Dr. G.A.E. Satish Kumar**, the Head of the Department, Department of Electronics and Communication Engineering, his guidance, intense support and encouragement, which helped us to mould our project into a successful one.

We show gratitude to our honorable Principal **Dr. J.V.R. Ravindra**, for providing all facilities and support.

We also thank all the staff members of Electronics and Communication Engineering department for their valuable support and generous advice. Finally thanks to all our friends and family members for their continuous support and enthusiastic help.

D.Abhinay

G.Chandrakanth

Abstract

This article describes an 8-bit RISC processor design using Verilog Hardware Description Language (HDL) on FPGA board. The proposed processor is designed using Harvard architecture, having separate instruction and data memory. The salient feature of proposed processor is pipelining, used for improving performance, such that on every clock cycle one instruction will be executed. Another important feature is that instruction set contains only 34 instructions, which is very simple, easy to learn and compact. The proposed processor has 8-bit ALU, Two 8-bit I/O ports, serial-in/serial-out ports, Eight 8-bit general-purpose registers, 4-bit flag register and priority based three vectored interrupts. Another advantage of the proposed processor is that it can execute programs with up to 262,144 instructions long, such that any practical programs can be fitted into it. The proposed processor is physically verified on Xilinx Spartan 3E Starter Board FPGA with 0.0517µs instruction cycle

Keywords: ECG, Arrhythmia, Discrete wavelet transform (DWT), Artificial neural network (ANN).

Table of Contents

List of Figures

Abbreviations

Abbreviation	Description
RISC	Reduced Instruction Set Computer
ALU	Arithmetic and Logic Unit
FPGA	Field Programmable Gate Array
ASIC	Application-Specific Integrated Circuits
UART	Universal Asynchronous Receiver Transmitter
CPI	Clock Cycles Per Instruction

CHAPTER 1

Introduction

Reconfigurable computing is computer architecture by associating some of the flexibility of software with the high performance of hardware by processing with very versatile high-speed computing fabrics like field-programmable gate arrays (FPGAs). The role of reconfigurable processor in embedded system design has increased greatly during the past decades. Due to the betterment of field programmable gate array (FPGA), we have reached a point where architecture of processor can be modified by programming [1]. The primary difference between reconfigurable processor and ordinary microprocessors is the potential to make significant changes to the datapath itself besides the control flow. On the contrary, the main distinction with application-specific integrated circuits (ASICs) is the feasibility to redesign the hardware during runtime by loading a new circuit on the reconfigurable fabric. Reduced Instruction Set Computer (RISC) is a design technique used to minimize the amount of area required, complexity of instruction set, instruction cycle and cost during the implementation of the design. With the expeditious development of the silicon technology and fall in the price of the integrated circuit, the usage of RISC processor is increasing extensively in all fields. Load and store operations are only used to access memory[1].

CHAPTER 2

8-BIT RISC PROCESSOR ARCHITECTURE

The 8-bit RISC processor is designed using Harvard architecture, having separate instruction memory and data memory. The Fig. 2.1 shows the complete architecture of proposed system, which is having 8-bit ALU, Two 8-bit I/O ports and Eight 8-bit general registers, 3 interrupts, serialin/serial-out ports and 4-bit flag register having zero flag (Z), carry flag (C), borrow flag (B) and parity flag (P).

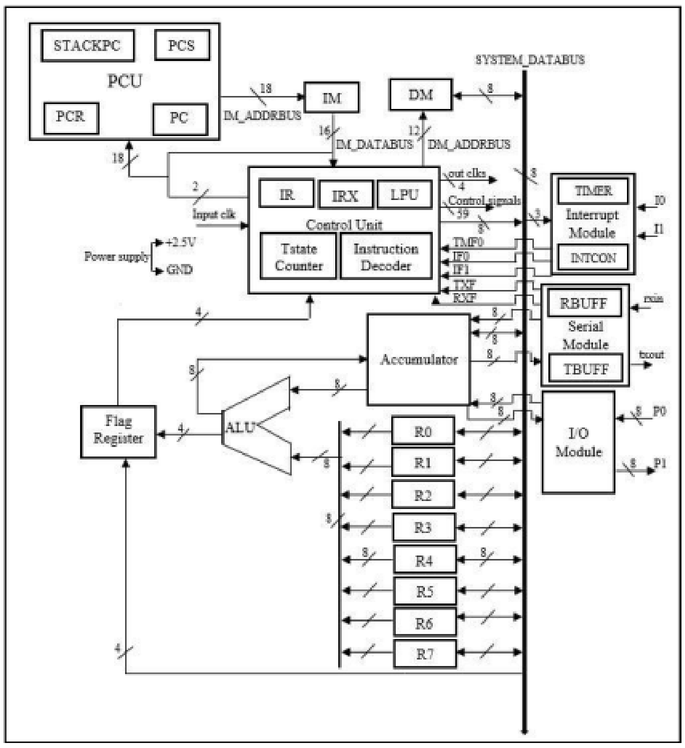


Figure 2.1: RISC processor architecture

The proposed RISC processor will work at 2.5 voltage supply and at clock frequency of 25MHz. The 8-bit SYSTEM DATABUS used for transferring data between different modules and 8-bit Accumulator used for arithmetic and logical operations are also integral part of the proposed processor. The Instruction Memory (IM) and Data Memory (DM) have different address and data buses for communicating between different modules.

The interrupt module contains three interrupts, which are priority based and one of the interrupt is mask able. Its most important feature is that its instruction set is very simple, contains only 34 instructions, which is easy to learn. The Serial Module is having ‘rxin’ as serial-in port and ‘txout’ as serial-out port, supports full duplex serial communication by using UART protocol. For efficient reduction of power, clock gating is used for specific modules which be clocked, only when it is required. Data memory and general-purpose registers are the modules where clock gating is used. All loading to the registers is taken place during falling edge of clock and all control signals are generated during rising edge of clock.

CHAPTER 3

PIPELINING ARCHITECTURE

Pipelining is a standard trait used in RISC based architecture design for improving performance and provides a way to reduce average execution time per instruction. The reduction can be observed as decreasing the number of clock cycles per instruction (CPI), as falling off in the clock cycle time, or as a combinational effect. The pipelining architecture used for the proposed processor is shown in Fig. 3.1. In the proposed processor, pipelining makes the architecture more efficient when tested on Xilinx Spartan 3E Starter Board FPGA with performance of 19.3 MIPS at 25MHz. The proposed processor requires only two clock cycles for the execution of an instruction (branching instruction is an exception (TX2 also)), i.e. one fetch (TF1) and one execution cycle (TX1), such that they are mutually exclusive. By the pipelining technique, while executing one instruction next instruction is fetched, such that on every clock cycle one instruction will be executed.

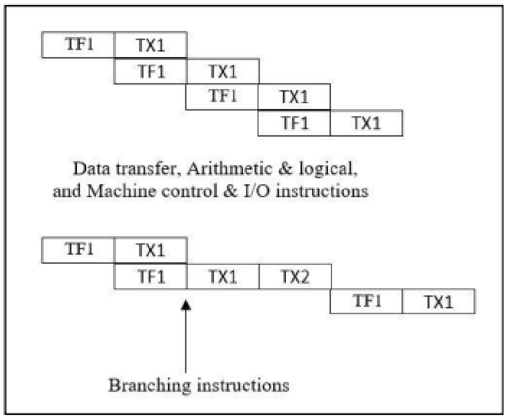

TF1
TX1
TF1
TX1
TF1
TX1
TF1
TX1
Data transfer, Arithmetic & logical, and Machine control & I/O instructions
TF1
TX1
TF1
TX1
TX2
TF1
TX1
Branching instructions

Figure 3.1: Pipelining architecture

CHAPTER 4
DESCRIPTION OF FUNCTIONAL MODULES

In the present work, the RISC processor consists of blocks namely, Program Counter Unit (PCU), Instruction Memory, Data Memory, Control Unit, Register set, Arithmetic Logical Unit (ALU), I/O module and Interrupt module and Serial Module

4.1 Program Counter Unit (PCU)

Program Counter unit consists of Program Counter (PC), Program Counter Save (PCS), Program Counter Register (PCR) and STACKPC. In the proposed processor, Program Counter (PC) is 18-bit wide register that contains the address (location) of the instruction being executed at the current time. As each instruction is fetched, the program counter increases its stored value by one. In fetch cycle, an 18-bit address bus known as IM_ADDRBUS transfers Program Counter content to Instruction Memory (IM) when the corresponding control signal is enabled. The Program Counter Save (PCS) stores (18-bit), Program Counter incremented by two value, when SAV PC instruction is executed. A Program Counter Register (PCR) is used to store address of the instruction, when a jump instruction is executed. STACKPC is used to store the current PC value during the execution of interrupt service routine.

4.2 Instruction Memory (IM)

Instruction Memory is 16-bit wide and having 262,144 address locations. In fetch cycle, when the corresponding control signals are enabled, a 16-bit data bus known as IM_DATABUS transfers Instruction Memory (IM) content to Instruction Register (IR), if IM_ADDRBUS provides an address location to IM at the same instance of time. .

4.3 Control Unit

The control unit contains Instruction Register (IR), Instruction RegisterX (IRX), tstate counter, Low Power Unit (LPU) and instruction decoder. The IR gets the instruction for decoding during fetch cycle. While executing one instruction next instruction is being fetched, therefore IR content should be stored in another register for execution of the instruction. For this purpose, IR content is moved to IRX during every rising edge of clock of execution cycle. Tstate counter generates fetch and execution cycles required for the proper working of processor. The tstates TF1 (fetch cycle), TX1 (execution cycle), TX2 (execution cycle2 only for branching instruction) are generated at rising edge of clock. The pipelining feature is applied in the tstate counter module. Interrupt priority and exceptions in instructions, like jump, are also taken in account in tstate counter module. Instruction decoder will generate controls signals required for the modules whenever IRX is loaded with a valid instruction. As IRX is loaded at every rising edge of execution cycle, so control signals are generated at the same instance of time.

Clock gating for Data Memory and general purpose register set are done in LPU of Control Unit. Control Unit will receive inputs from flag register, Serial Module and Interrupt Module. The Control Unit will take input clock from source clock of FPGA and generate 59 control signals and 4 clock signals for the proper working of all modules. The four clock signals include gated clock signals for register set and Data Memory modules, baud rate clock for Serial Module and 25 MHz output clock signal for all other modules. The LPU mainly uses clock-gating technique to reduce power dissipation. The Data Memory and Register set are the main memory elements, which consume power, so gated clock is provided to these modules. Whenever loading to a memory/general-purpose register corresponding module will be activated.

4.3.1 Data Memory (DM)

Data Memory is 8-bit wide and having 4096 address locations. Data Memory gets the required address location by 12-bit address bus known DM_ADDRBUS from control unit. DM provides read and write control, can be accessed by 8-bit data bus known as SYSTEM _DATABUS.

4.3.2 ALU unit

ALU is connected to Accumulator and general-purpose registers by its 8-bit buses AL_DATABUS_A and ALU_DATABUS_B. ALU unit consists of AND, OR, XOR, ADD, SUB operations. The result of operation is stored in Accumulator. The result also contains zero flag (Z), Carry flag(C), Borrow flag (B) and parity flag, which will be stored in 4-bit Flag Register. Parity flag is set when resultant of ALU operation contains odd number of ones.

4.4 Accumulator (A)

Accumulator is an 8-bit wide register, integral part of the proposed processor, as data transfer, ALU operations and I/O operations takes place through it. Accumulator also contains increment, decrement, rotate right, rotate left and compliment operations. Accumulator is connected to serial module for sending the data required for transmission through serial-out port and storing the data received through serial-in port.

4.5 Register set

Register set contains eight 8-bit registers R0, R1, R2, R3, R4, R5, R6 and R7, which can be used for storing data that are frequently used. Register set is connected to ALU unit for arithmetic and logic operations. It is also connected to SYSTEM_DATABUS for loading and storing data.

4.6 Interrupt module

The interrupt module contains two external interrupts I0 and I1 and one timer interrupt. The interrupts are priority based, the I0 interrupt is having highest priority, followed by I1 interrupt and timer interrupt is having least priority. The I1 interrupt is maskable and I0 is not maskable. The interrupt module is having a 3-bit register INTCON to control the functions in interrupt module. Its bit 2 is used for enabling and disabling timer interrupt, bit 1 is used for enabling and disabling external interrupts and bit 0 is used for masking I1. The timer module has 10-bit TIMER register for counting the predefined interval. When the TIMER register reaches the maximum value, timer flag TMF0 gets set. Thus in the interrupt service routine we need to clear the TMF0 flag by the instruction CLRTMRF.

4.7 I/O Module

The I/O module has one 8-bit input port and one 8-bit output port for communicating with external environment. The input port and output port are directly connected to Accumulator. It will transfer data to and from Accumulator when corresponding control signals are enabled.

4.8 Serial Module

The serial module contains rxin as serial-in port and txout as serial-out port. The serial communication is full duplex that is it can transmit and receive simultaneously, which is based on UART protocol is shown is Fig. 4.1. The baud rate used for this serial Module is 115200. The data transmission starts with a start bit of 0, followed by the individual data bits of the word with the Least Significant Bit (LSB) being sent first and then stop bit 1.

The Serial Port Module consists of two 8-bit registers TBUFF and RBUFF for storing data while transmission and reception. The data stored in TBUFF register is shifted out during serial data transmission and process is vice versa for RBUFF register. The baud rate (clock) required for serial communication is provided by the control unit.

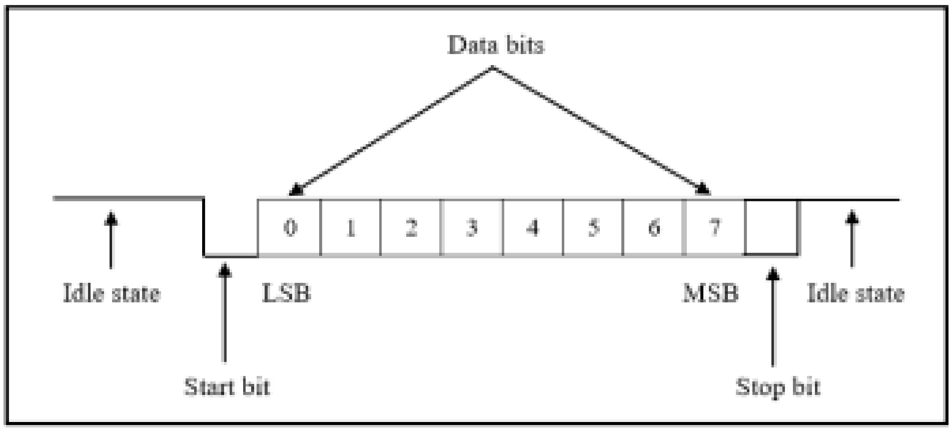


Figure 4.1: UART serial communication protocol

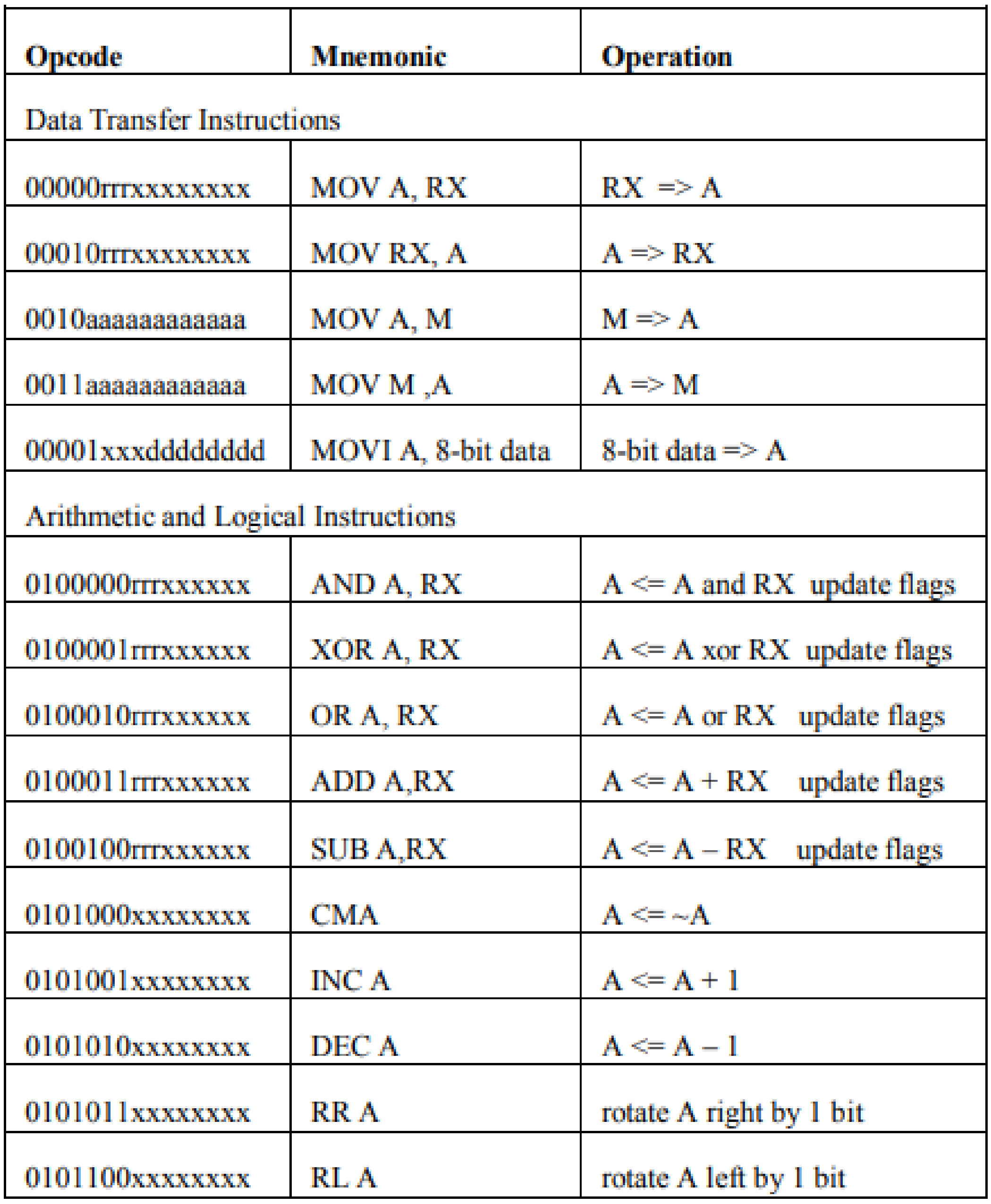
CHAPTER 5

INSTRUCTION SET ARCHITECTURE

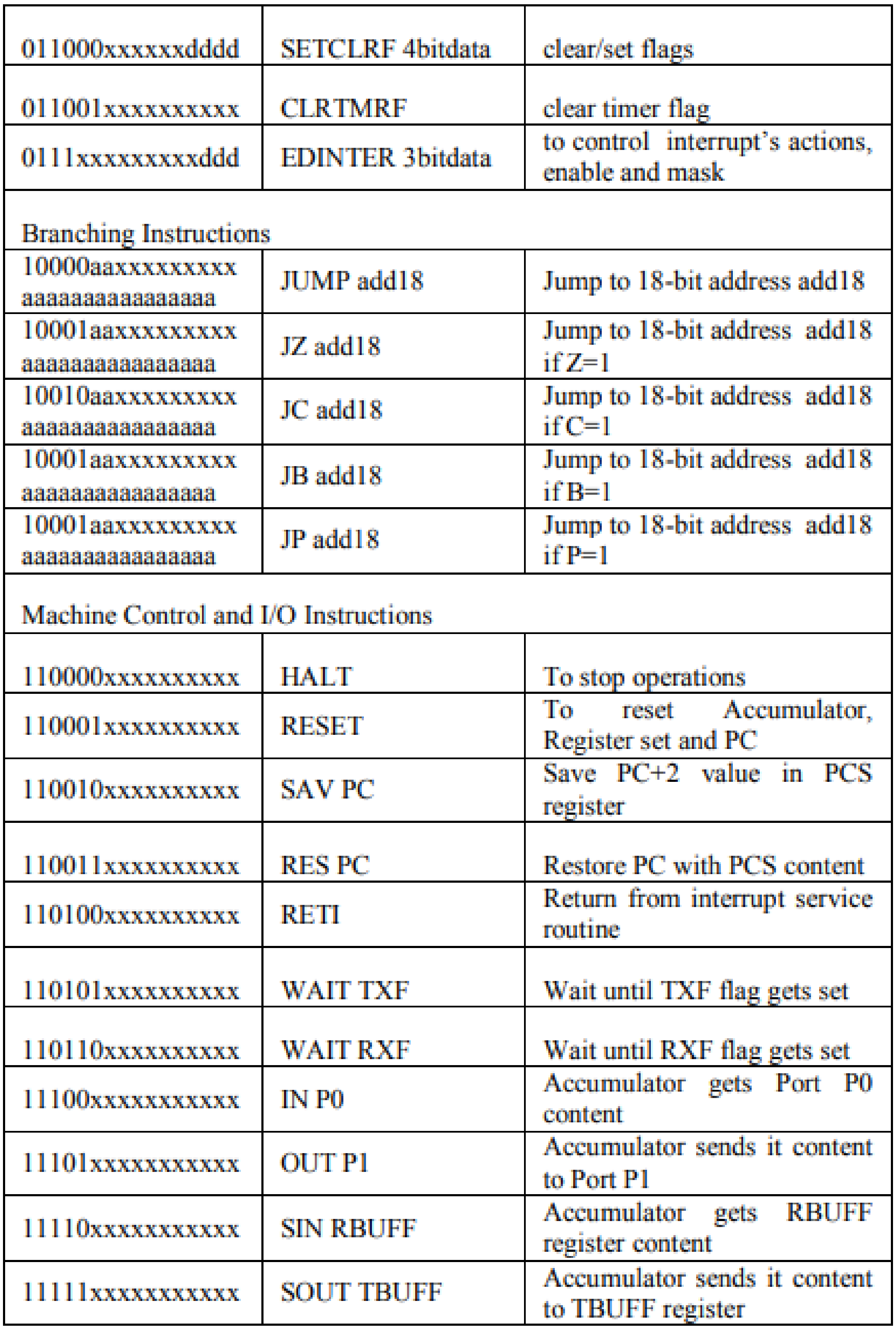
The Instruction set architecture contains four type of instructions, data transfer instructions, arithmetic and logic instructions, branching instructions and machine control and I/O instructions. The instruction set architecture contains only 34 basic instructions and total number of opcodes is 83. The main advantage of the proposed processor is use of SAV PC instruction that saves PC incremented by two value in PCS. SAV PC can be used before jump instruction so that after jumping to another location, using RES PC (loads PCS content to PC) we can come back to next instruction after jump instruction. The combined use of SAV PC and jump instruction, will act like a CALL instruction thus reduces the number of instruction required but functionality is maintained. We can use jump instruction alone to jump to specific location and never come back. Another special instruction is RETI that is used to restore the PC value after the execution of interrupt service routine. SETCLRF is a special instruction, which can be used to clear or set every flag in flag register. EDINTER is another special one, which can be used to enable/disable interrupts and mask the I1 interrupt when needed. CLRTMRF should be used in the interrupt service routine of timer interrupt.

Serial data transmission is possible by the use of SOUT TBUFF and WAIT TXF. Initially use SOUT TBUFF to send data to be send from accumulator to TBUFF. Then use WAIT TXF for waiting until data is completely transmitted serially from TBUFF register. For serial data reception, initially use WAIT RXF for waiting until data is completely received by RBUFF register and then use SIN RBUFF to store the serial data received in RBUFF register to accumulator. IN P0 is used to store 8-bit parallel input data from input Port P0 to accumulator and OUT P1 is used to send 8-bit parallel data from accumulator to output Port P1.

TABLE I shows instruction set of proposed RISC processor. In the TABLE I Opcode column, rrr is register select from 000 to 111, x is don't care, a is address and d is data. In the TABLE I Mnemonic column, RX is register can be any one of R0, R1, R2, R3, R4, R5, R6 and R7, A is accumulator, M is Data Memory address and add18 is Instruction Memory address. In Operation column, update flags means update the flags corresponding to that operation.



Opcode	**Mnemonic**	**Operation**
Data Transfer Instructions		
00000rrrxxxxxxxx	MOV A, RX	RX => A
00010rrrxxxxxxxx	MOV RX, A	A => RX
0010aaaaaaaaaaaa	MOV A, M	M => A
0011aaaaaaaaaaaa	MOV M ,A	A => M
00001xxxdddddddd	MOVI A, 8-bit data	8-bit data => A
Arithmetic and Logical Instructions		
0100000rrrxxxxxx	AND A, RX	A <= A and RX update flags
0100001rrrxxxxxx	XOR A, RX	A <= A xor RX update flags
0100010rrrxxxxxx	OR A, RX	A <= A or RX update flags
0100011rrrxxxxxx	ADD A,RX	A <= A + RX update flags
0100100rrrxxxxxx	SUB A,RX	A <= A – RX update flags
0101000xxxxxxxxx	CMA	A <= ~A
0101001xxxxxxxxx	INC A	A <= A + 1
0101010xxxxxxxxx	DEC A	A <= A – 1
0101011xxxxxxxxx	RR A	rotate A right by 1 bit
0101100xxxxxxxxx	RL A	rotate A left by 1 bit



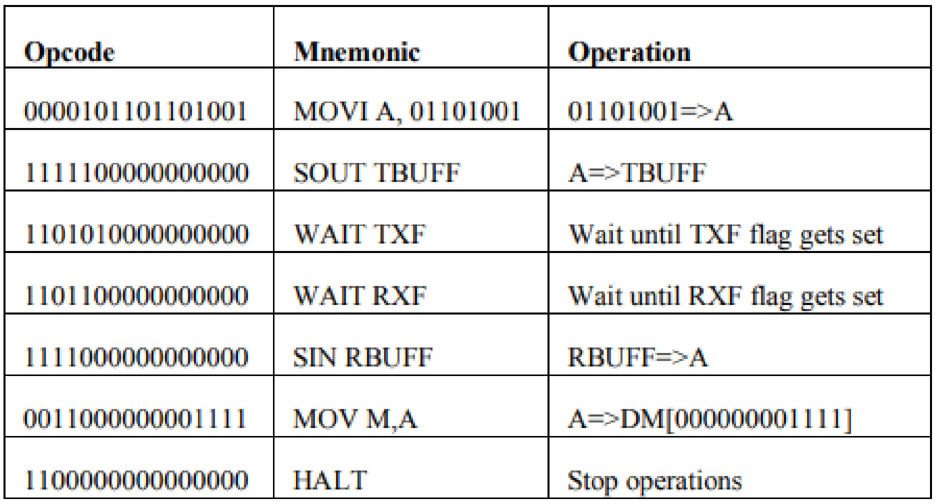
011000xxxxxxdddd	SETCLRF 4bitdata	clear/set flags
011001xxxxxxxxxx	CLRTMRF	clear timer flag
0111xxxxxxxxxddd	EDINTER 3bitdata	to control interrupt's actions, enable and mask
Branching Instructions		
10000aaxxxxxxxxxx aaaaaaaaaaaaaaaa	JUMP add18	Jump to 18-bit address add18
10001aaxxxxxxxxxx aaaaaaaaaaaaaaaa	JZ add18	Jump to 18-bit address add18 if Z=1
10010aaxxxxxxxxxx aaaaaaaaaaaaaaaa	JC add18	Jump to 18-bit address add18 if C=1
10001aaxxxxxxxxxx aaaaaaaaaaaaaaaa	JB add18	Jump to 18-bit address add18 if B=1
10001aaxxxxxxxxxx aaaaaaaaaaaaaaaa	JP add18	Jump to 18-bit address add18 if P=1
Machine Control and I/O Instructions		
110000xxxxxxxxxx	HALT	To stop operations
110001xxxxxxxxxx	RESET	To reset Accumulator, Register set and PC
110010xxxxxxxxxx	SAV PC	Save PC+2 value in PCS register
110011xxxxxxxxxx	RES PC	Restore PC with PCS content
110100xxxxxxxxxx	RETI	Return from interrupt service routine
110101xxxxxxxxxx	WAIT TXF	Wait until TXF flag gets set
110110xxxxxxxxxx	WAIT RXF	Wait until RXF flag gets set
11100xxxxxxxxxxx	IN P0	Accumulator gets Port P0 content
11101xxxxxxxxxxx	OUT P1	Accumulator sends it content to Port P1
11110xxxxxxxxxxx	SIN RBUFF	Accumulator gets RBUFF register content
11111xxxxxxxxxxx	SOUT TBUFF	Accumulator sends it content to TBUFF register

Figure 5.1: INSTRUCTION SET

CHAPTER 6

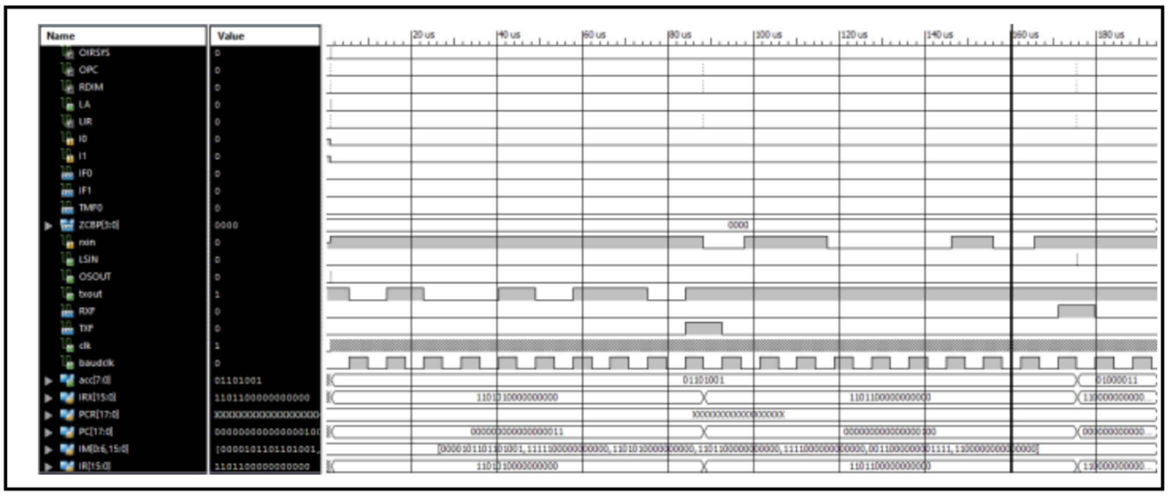
SIMULATION RESULTS

The simulation results have been performed by using ISim. The Fig. 6.2 shows the simulation results of the proposed 8-bit RISC processor with pipeline architecture for a simple seveninstruction program in TABLE II. The TABLE III shows the device utilization of the Xilinx Spartan 3E Starter Board FPGA. Simulation results in Fig. 6.1 shows that serial-in port rxin and serial-out port txout are 1 (high signal) in idle state and transmission/reception starts with 0 (low signal). In the Accumulator is represented as acc, 4-bit flag register is represented as ZCBP, baud rate for serial communication is represented by baudclk and IF0, IF1 and TMF0 represents the interrupt flags. Control Unit will generate 59 control signals and 4 clocks while simulating the program in TABLE II, but for convenience, by Fig. 6.2 a part of total simulation is shown. TABLE IV shows the Xilinx Power Estimator (XPE)-11.1 device summary report for the proposed processor carried out for the Spartan-3E Starter Board FPGA.



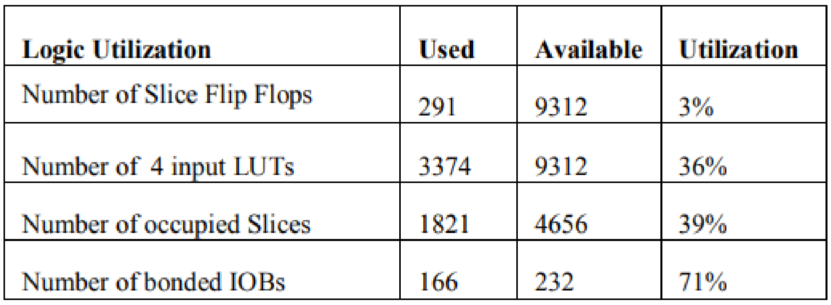
Opcode	Mnemonic	Operation
0000101101101001	MOVI A, 01101001	01101001=>A
1111100000000000	SOUT TBUFF	A=>TBUFF
1101010000000000	WAIT TXF	Wait until TXF flag gets set
1101100000000000	WAIT RXF	Wait until RXF flag gets set
1111000000000000	SIN RBUFF	RBUFF=>A
0011000000001111	MOV M,A	A=>DM[000000001111]
1100000000000000	HALT	Stop operations

Figure 6.1: SAMPLE SEVEN INSTRUCTION PROGRAM FOR SIMULATION



Name	Value
OIRSYS	0
OPC	0
RDIM	0
LA	0
LIR	0
I0	0
I1	0
IF0	0
IF1	0
TMF0	0
ZCBP[3:0]	0000
rxin	0
LSIN	0
OSOUT	0
txout	1
RXF	0
TXF	0
clk	1
baudclk	0
acc[7:0]	01101001
IRX[15:0]	1101100000000000
PCR[17:0]	XXXXXXXXXXXXXXXXXX
PC[17:0]	000000000000000100
IM[0:6,15:0]	[0000101101101001,
IR[15:0]	1101100000000000

Figure 6.2: SIMULATION RESULTS OF PROPOSED PROCESSOR



Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	291	9312	3%
Number of 4 input LUTs	3374	9312	36%
Number of occupied Slices	1821	4656	39%
Number of bonded IOBs	166	232	71%

Figure 6.3: XILINX POWER ESTIMATOR (XPE)-11.1 DEVICE SUMMARY REPORT OF SPARTAN-3E FPGA FOR THE PROPOSED PROCESSOR

CHAPTER 7

CONCLUSION

Pipelined 8-bit RISC processor using Verilog HDL is designed and implemented on Xilinx Spartan 3E Starter Board FPGA at 25MHz. ISim is used to perform the simulation results. The whole 34 instructions in instruction set are verified individually and collectively.

Clock-gating technique provides a way to reduce power in the proposed RISC processor design. Pipelining technique yields better performance as the processor executes one instruction on every clock cycle. Serial communication using UART protocol is successfully carried out at a baud rate of 115200.

REFERENCES

[1] Yogesh K. Chauhan. “Generation of PWM using verilog In FPGA”. In: 2016.