Inside Digital Design Accompany Lab Manual

Simulation Prototyping Synthesis and Post Synthesis

Name-_____

Roll Number-_____

Total/Obtained Marks-_____

Instructor Signature-_____

Lab Number	Total Marks	Marks Obtained	Signatures of TA
Lab-1			
Lab-2			
Lab-3			
Lab-4			
Lab-5			
Lab-6			
Lab-7			
Lab-8			
Lab-9			
Lab-10			
Lab-11			
Lab-12			
Lab-13			
Lab-14			
Lab-15			
Lab-16			

Lab Equipment Required

Hardware

1. FPGA Board, Preferably Spartan 3 and 2
2. Oscilloscopes, Data Timing Generator, Logic Analyzer

Software

1. Xilinx ISE
2. Model Sim, Mentor Graphics
3. Synopsys, Synopsys Inc.
4. X Power for Power Analysis
5. Chip Scope Pro for Real Time On Chip Debugging
6. Timing Analyzer

Contents

Lab-8: Design of UART system using PC and FPGA

Design of UART transmitter on FPGA and Receiving Bits on HyperTerminal, Using .NET as Transmitter and Designing Receiver on FPGA, Integrating UART Receiver and Transmitter

Lab-9: Design of Complete Processor with Compiler on PC

Integrating RISC and UART, Sending Data from .NET based compiler and receiving it on FPGA, Executing Instruction on FPGA and sending it to .NET Application on PC using UART Transmitter

Lab-10: Design of Veterbi and Reed Solomon

Simulating Veterbi System and Reed Solomon Encoder, Decoder, Integrating Interleaver with Veterbi.

Lab-11: Line Encoding Techniques Simulation and Implementation

Getting Data from Voice Input from User, Line Encoding the data and sending it to PC using UART

Lab-12: Design of Complete Communication System-1

Transmitting Data from the PC .Net Application to FPGA, Receiving Data on FPGA using UART receiver, Encoding the Data in the FPGA Encoding it using Communication Transmitter, Receiving the Data on .NET application form FPGA.

Lab-13: Design of Complete Communication System-2

The Received Data in Lab 12 is transmitter again with some noise on FPGA, FPGA Behaves like Receiver now and Recovering Data from the received data, Received data is send to .NET Application

Lab-14: Digital Signal Processing Application

FIR and IIR Filters Design, Wallace Tree Multiplier, Conditional Sum Adder, Cordic Algorithm

Lab-15: Digital Signal Processing and Image Processing Application-1

FFT, Decimator and Interpolator, Canny Edge Detection

Lab-16: USB and PCI Bus Design

Controllers for USB and PCI Bus

Lab-17: Post Synthesis (Optional)

ASIC Chip Designing using Synopsys

This Page is intentionally Left Blank

Combinational Logic, Model Sim

Lab-1

Adder Design, Decoder, Multiplexers, Comparator

Question-1: Design the 4-Bit Ripple carry Adder Circuit. Partition your design using down up approach. Design a half adder circuit, seeing the truth table and the diagram in Figure 1-1.

x	y	C	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

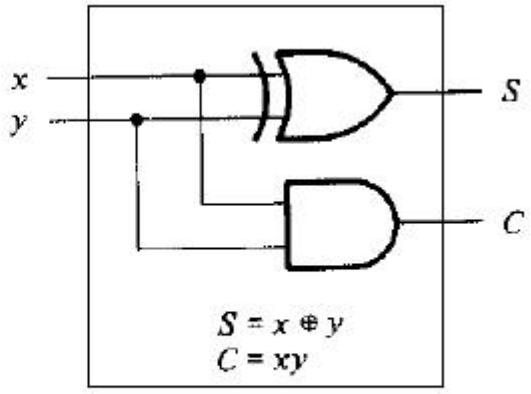


Figure 1-1: The half adder circuit and truth table

Using the half adder design, make full adder design, seeing diagram in figure 1-2.

x	y	z	C	S
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

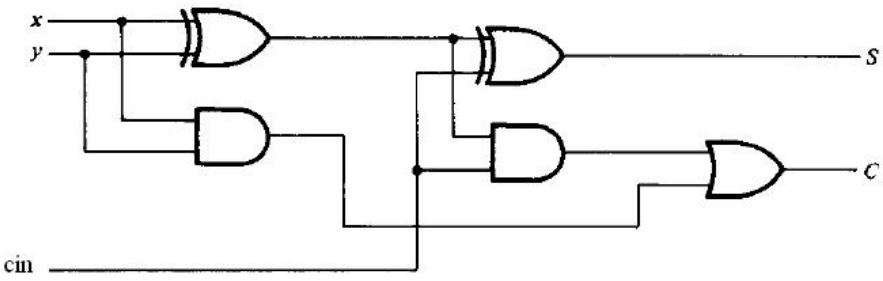


Figure 1-2: Full adder Circuit and truth table

Finally integrate the full adder to make 4-bit ripple carry adder shown in figure 1-3.
(Paste your results above the waveform 1-1, 1-2 and 1-3 given below).

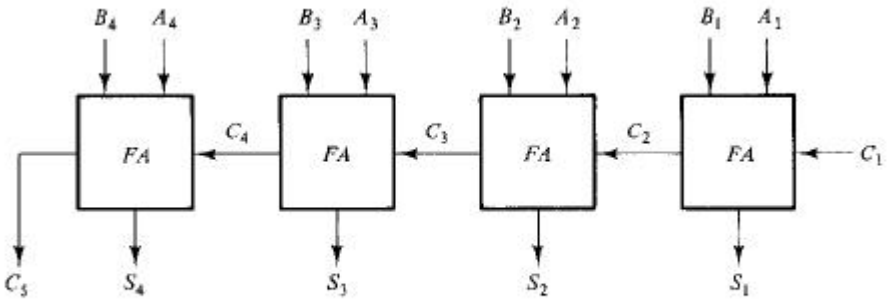


Figure 1-3: 4-Bit Ripple Carry Adder

Modify the 4-bit adder so that it can work as adder-subtractor circuit. The M decides whether to add or subtract, in case M is 1 the circuit take two complement by inverting all the bits and add one to the least significant position. In other way it behaves like a normal adder the illustration is given in Figure 1-4 and the waveform is given in waveform 1-4.

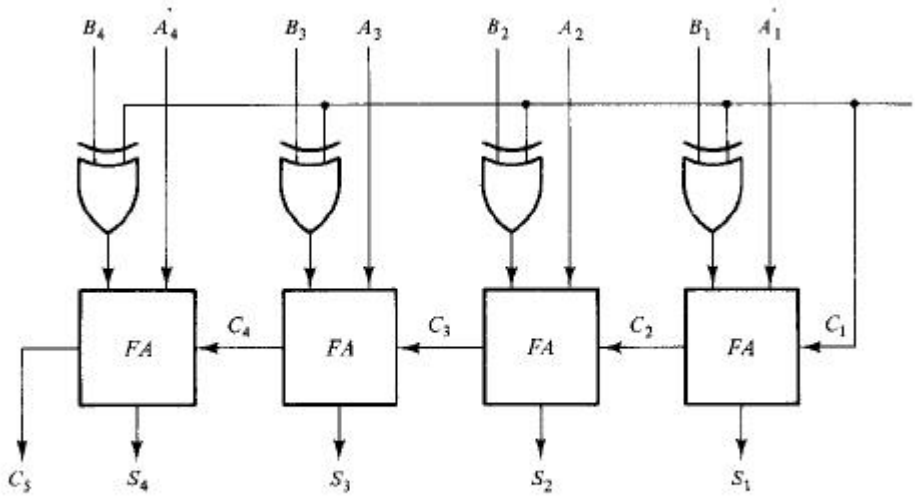


Figure 1-4: Adder Subtractor Circuit

Question-2: Write a code for 4*1 Multiplex, one of the four inputs is passed to the output, as shown in figure 1-5. This requires selection line of 2 bit. Paste the results below waveform number 1-5.

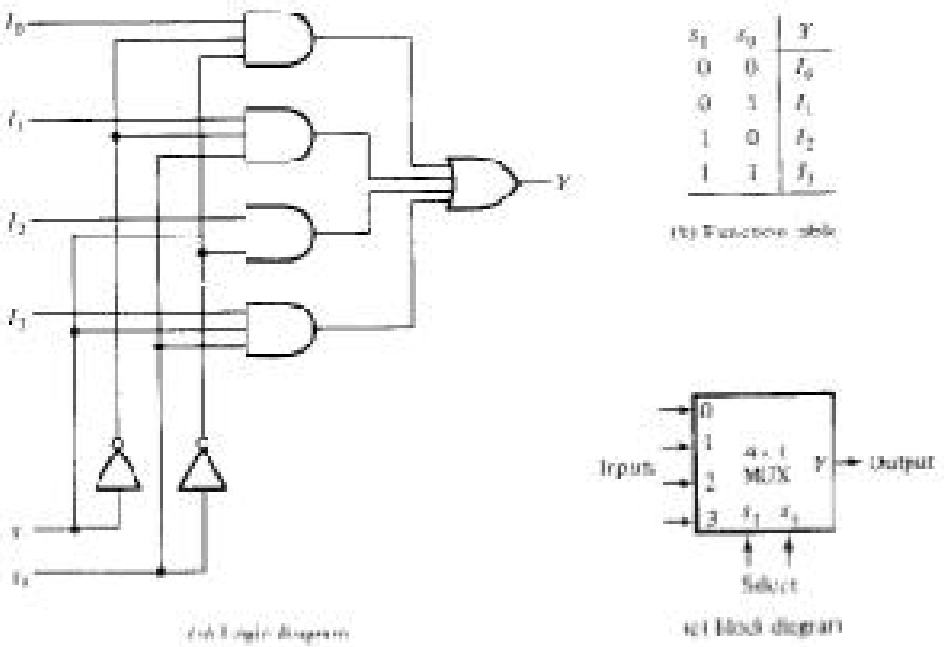


Figure 1-5: 4x1 Multiplexer a-) Logical b-) Functional Table and c-) Block Diagram.

Modify the above design so that it can be a quadruple 2x1 Multiplexer, the design is shown in figure 1-6. Paste the results below waveform number 1-5.

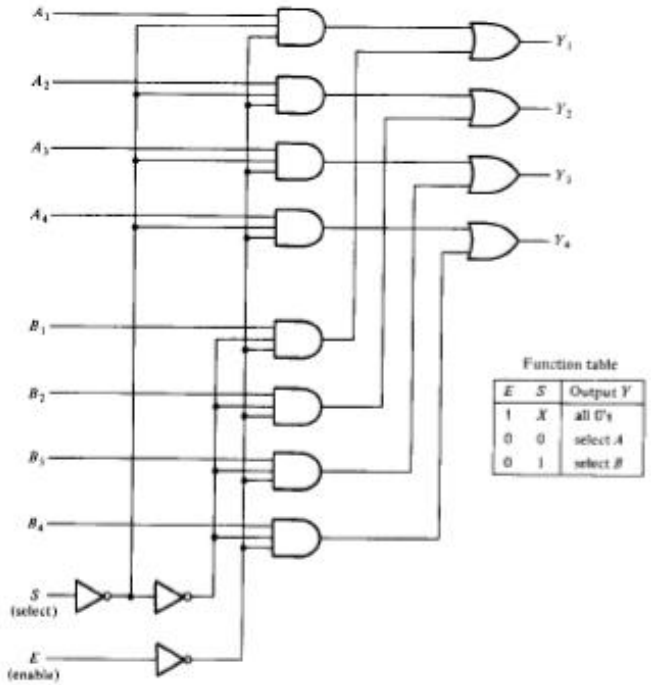
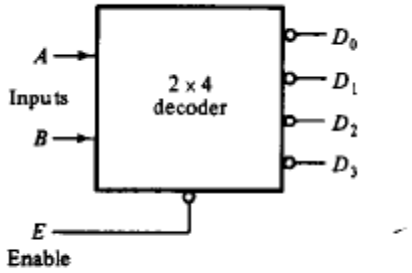


Figure 1-6: Quadruple 2x1 Multiplex

Question-3: Design the Encoder given in Figure 1-7. The waveform is given in waveform 1-6.



(a) Decoder with enable

E	A	B	D_0	D_1	D_2	D_3
1	X	X	1	1	1	1
0	0	0	0	1	1	1
0	0	1	1	0	1	1
0	1	0	1	1	0	1
0	1	1	1	1	1	0

(b) Truth table

Figure 1-7: Block Diagram and Truth Table for Encoder

Question-4: Design a 16 Bit Encoder with 2 inputs A and B. There are three outputs A equal to B, A less than B and A greater than B. The waveform of the Design is Given in 1-6.

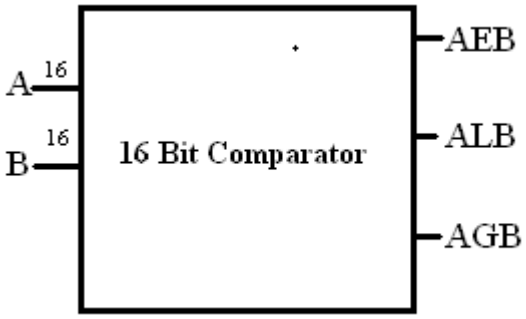


Figure 1-8: Block Diagram for 16 Bit Comparator

Sequential Logic, Model Sim

Lab-2

Flip-flop (D, Toggle, JK), Counters, Shift Register, Linear Feed Back shift Registers, Stop Watch with Seven Segment

Question 1-(a): Write Code and Stimulus for D Flip Flop. The required Circuit is given in Figure below 2-1?

E	Q	D	CLK	Q(t+1)
1	x	x	↑	D
1	x	x	↓	Q
0	x	x	↑↓	Q

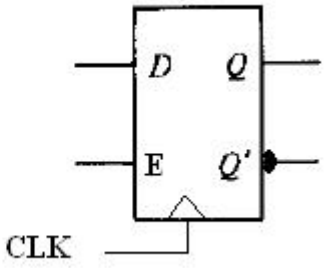


Figure 2-1: D Flip Flop and its Truth Table

Question 1-(b): Write Code and Stimulus for T Flip Flop by Calling D Flip Flop and Without Calling D Flip Flop. The required Circuit is given in Figure 2-2 below?

Q	J	K	Q (t + 1)
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

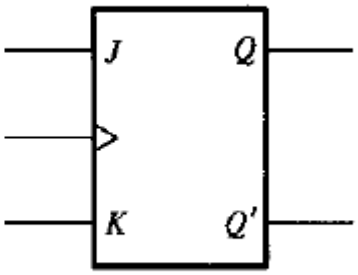


Figure 2-2: JK Flip Flop and its Truth Table

Question 1-(c): Write Code and Stimulus for JK Flip Flop by Calling D Flip Flop and Without Calling D Flip Flop. The required Circuit is given in Figure below 2-3?

Q	T	$Q(t+1)$
0	0	0
0	1	1
1	0	1
1	1	0

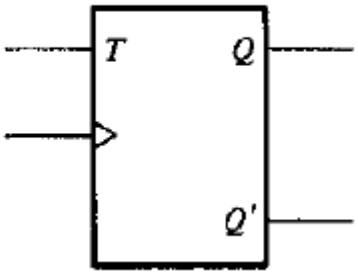
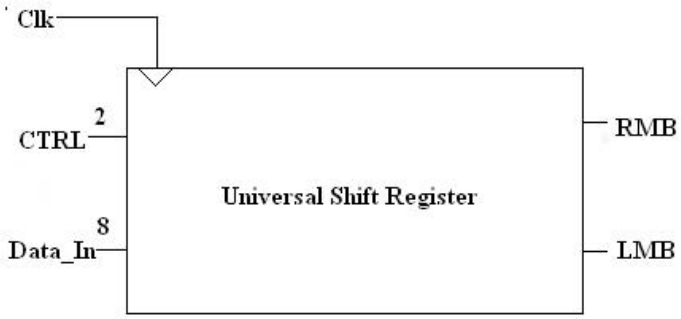


Figure 2-3: T Flip Flop and its Truth Table

Question 2: Write Code and Stimulus for the universal Shift Register. Which is capable to perform following operations? The required Circuit is given in Figure 2-4 below?



Ctrl Value	Operation
00	Parallel Load
01	Shift Right
10	Shift Left
11	Hold Value

Figure 2-4: Universal Shift Register and its Table

Question 3: Write a code and Stimulus for the generic Linear Feed back Shift register. The output is feed back to the MSB unconditional all other feed backs are conditional to Tap Coefficients. Define any Coefficients for your Self. The required Circuit is given in Figure 2-5 below.

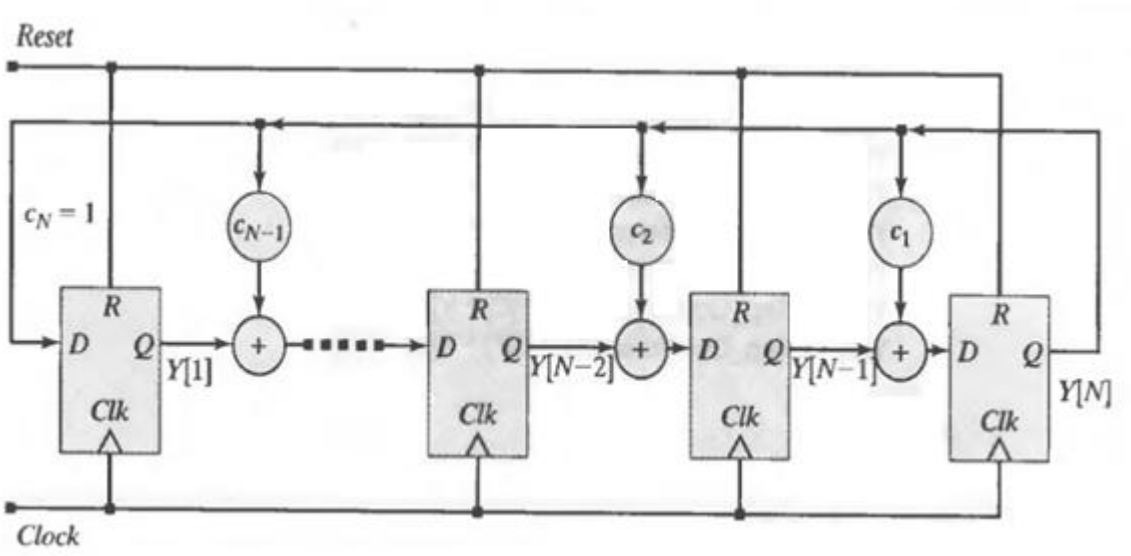


Figure 2-5: Generic Linear Feed Back Shift Register

Question 4-(a): Write a code for a Simple BCD Counter Which can count Up and Down from 0~9 and 9~0 respectively. The up down sequence can be switched using UD Control i.e. if the value of UD is 1 the Counter will count from 0~9 and back and if it is 0 it will count from 9~0 and back. The required Circuit is given in Figure 2-6 a below.

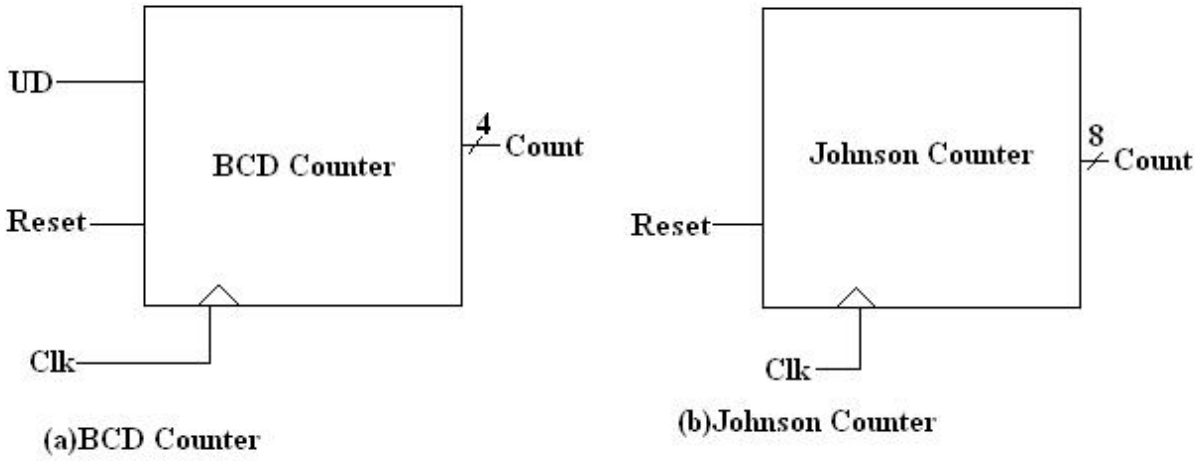
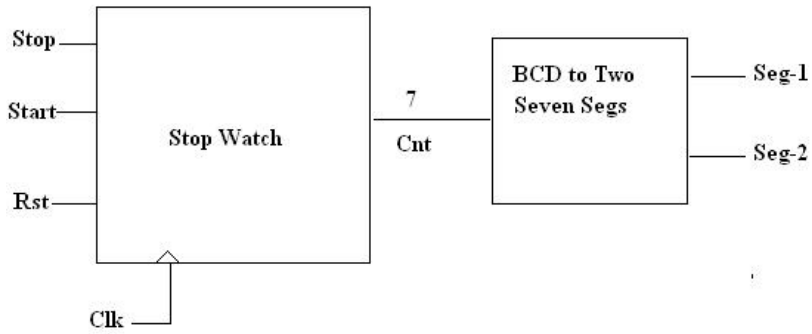


Figure 2-6: BCD Counter with 4 bit output and Up down Control and Johnson Counter. Both can be reset

Question 4-(b): Write Verilog code and Stimulus for the Johnson counter which can count in the sequence 2,4,8,16,32,64,128,256,128,64,32,16,8,4,2. Consider an 8 Bit register in which

one is present at the LSB. Each time the value is shift once left (if Little Endean is followed) until the one reaches the MSB and then this is shifting process is reversed until it reaches LSB and So Forth. The required Circuit is given in Figure 2-6 b above.

Question 5: Write code and Stimulus for the Stop watch Circuit with Seven Segment Display. The Stop Watch starts its operation with the start control. Once Triggered it will count from 0~99, It can be reset at any time to zero using rst control. The stop watch can be paused in its Operation using stop control; it will resume its halted counted by using start again. The 7 bit count is passed to BCD to seven segment module which generated the 7 bit Segment code to display on two seven Segments. The required Circuit is given in Figure 2-7 below.




```
BLANK  = 7'b111_1111
ZERO   = 7'b000_0001
ONE    = 7'b100_1111
TWO    = 7'b001_0010
THREE  = 7'b000_0110
FOUR   = 7'b100_1100
FIVE   = 7'b010_0100
SIX    = 7'b010_0000
SEVEN  = 7'b000_1111
EIGHT  = 7'b000_0000
NINE   = 7'b000_0100
```

Figure 2-7: Stop watch on Left and Seven Segment Codes on Right Hand Side

Simulation of ASMD Design, RISC Lab-3

RISC Processor Simulation, Control Unit, Memory and Data path in Model Sim

Question-1: Design the control unit of the RISC Processor [1]. The ASMD Chart of the RISC Processor is given in the Figure 3-3. Write the Test Bench of the Control Unit, Pass different values of the instruction and study the states of the Control Unit. Architecture, Instruction Set and Instruction format are given in Figure 3-1, Figure 3-2a and Figure 3-2b.

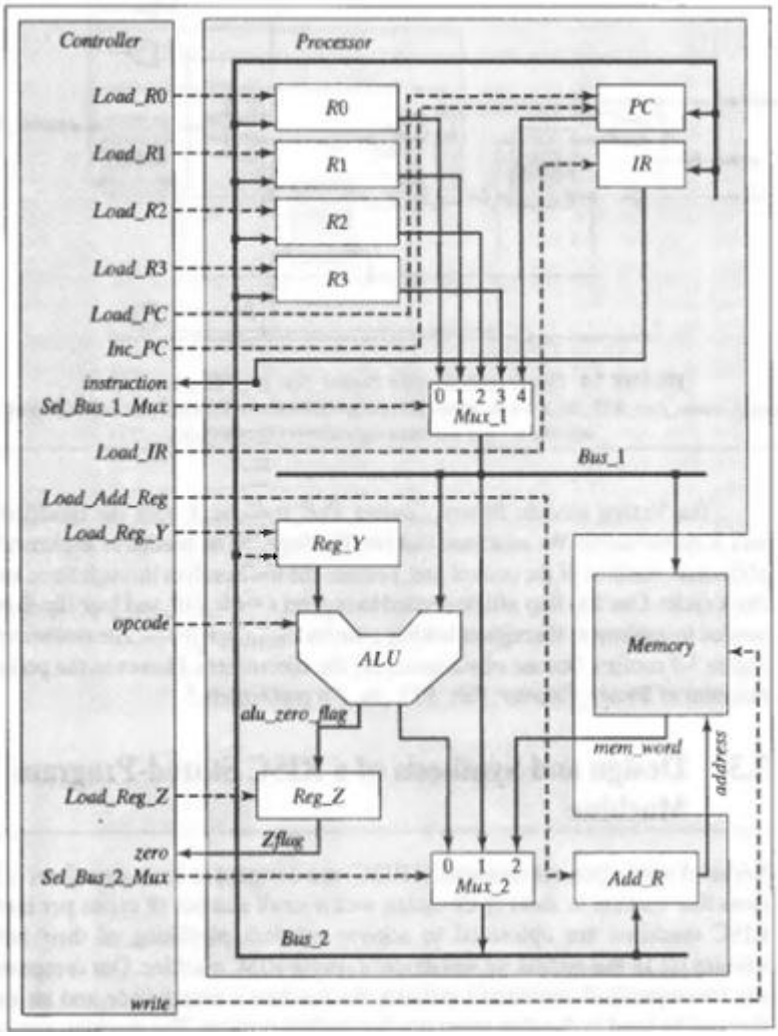


Figure 3-1: Architecture of the RISC Processor

Instr	Instruction Word			Action
	opcode	src	dest	
NOP	0000	??	??	none
ADD	0001	src	dest	dest <= src + dest
SUB	0010	src	dest	dest <= dest − src
AND	0011	src	dest	dest <= src && dest
NOT	0100	src	dest	dest <= ~ src
RD*	0101	??	dest	dest <= memory [Add_R]
WR*	0110	src	??	memory[Add_R] < = src
BR*	0111	??	??	PC <= memory[Add_R]
BRZ*	1000	??	??	PC <= memory [Add_R]
HALT	1111	??	??	Halts execution until reset

*Requires a second word of data; ? denotes a don't-care.

Figure 3-2a: Instruction Set of the Processor

opcode				source		destination	
0	0	1	0	0	1	1	0

One Byte Instruction Format

opcode				source		destination	
0	1	1	0	1	0	don't care	don't care
address							
0	0	0	1	1	1	0	1

Two Byte Instruction Format

Figure 3-2b: Instruction Format

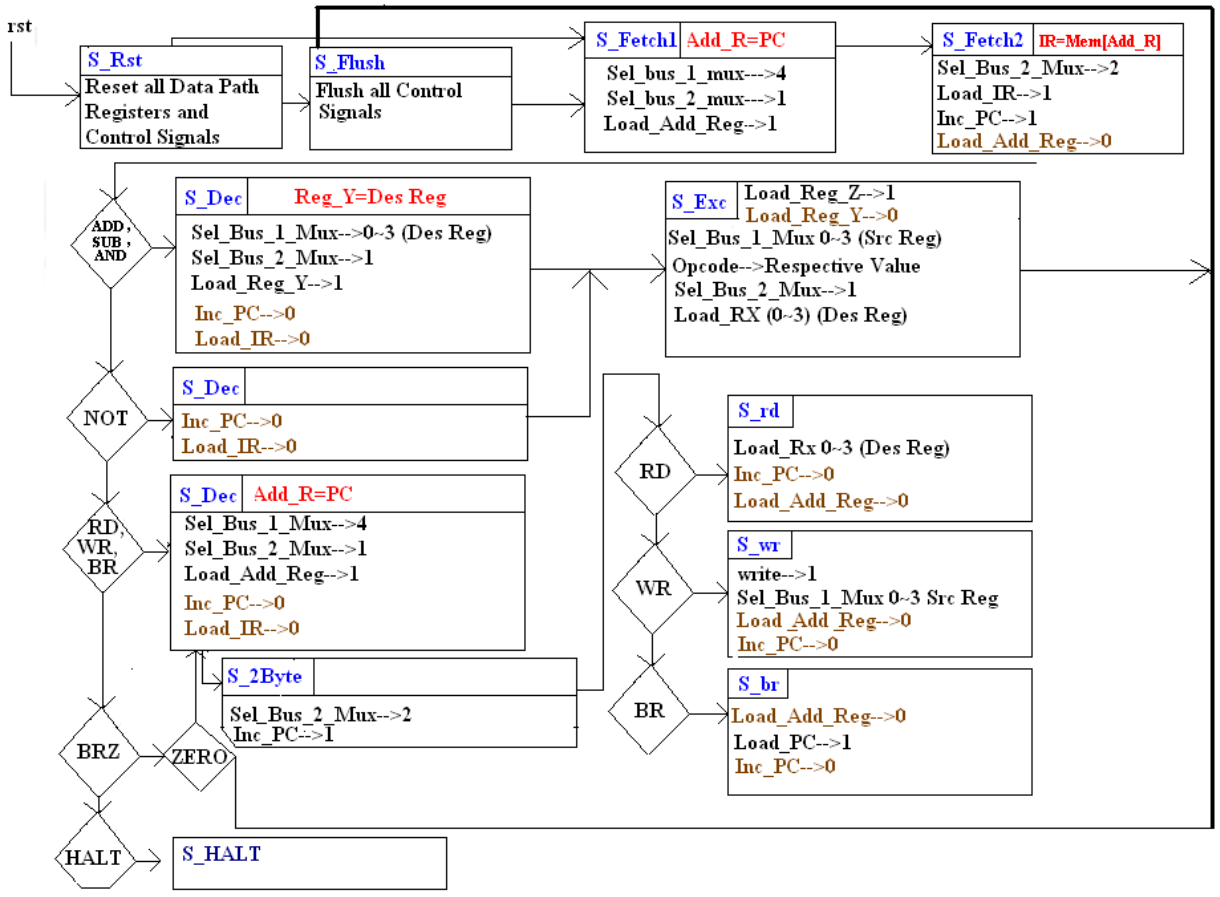


Figure 3-3: ASMD Chart of Control Unit with the States Mentioned in Blue Assertion in Black and Dissertation of Previous State in Brown.

Question-2: Design the Data Path of the RISC Processor given in the Question-1. Also Write the Complete Test Bench for the Design. Follow the Process Step by Step by writing the code of ALU, Register File and Memory. Write Separate Test Bench and Waveform for each Module and then the Complete Waveform for the Data Path.

Question-3: Integrate Data Path and Control Unit, Write Test Bench for the Top Level Module RISC SPM. The Top Level Module Instantiates both Control Unit and Data path? Write a Code to Calculate the Factorial of the Number in the Test Bench of the Top Level module. Store Number 5 as a variable in the Data Memory Partition of the Memory.

*This Lab is Triple Weight age and each Question Contains 15 Marks.