Assignment 4

Due date: November 15, 2017

Group No: 5

Students:
Lalit Bhat: 01671833
Priya Parikh: 01679645

Objectives:

The objective of this assignment is to learn to build a 4-bit CPU, which executes several RTL instructions, and to learn to design and learn the importance of a controller in a CPU circuit.

List Of components:

Part description	Qty
Proto board PB-105T	1
7489 RAM	1
17DIP8SS 8-Switch DIP Switch sets	2
08TIE524 Dual 7 Segment Display	1
74LS181 ALU	1
74LS247 BCD to 7 Segment Display	2
74LS04 Hex Invertor	2
74LS00 NAND gate	2
74LS74 Dual positive edge triggered D-flipflop	1
74LS174 Hex D-type Flip-flop	2
74LS569 PC	1
5V Power supply	1
1kΩ Resistor	16
330Ω Resistor	2
1kΩ Resistor	1
Cable, Banana with Alligator Clips	2
Jumper wires	
Wire cutter	2
Logic Probe	1

Experimental Approach:

1. We first studied the data sheets of the shift registers, hex D-type FF, ALU and started planning what modes we need to hard code our ALU and other registers for the RTL instructions to be executed as required.

2. Made a rough outline of the circuit on paper.

3. The most challenging part of this assignment was designing the controller and the clock control for some of the registers, as this section of the project was responsible for the efficient policing of the circuit control.

4. We designed the circuit to execute the circuits with both 1 and 2 clock cycles.

5. Constructed this circuit on logic works and simulated it to find and fix any bugs.

6. After simulating the circuit with all the instructions successfully, the hardware circuit was built on a Proto-Board.

7. In this circuit we used pullup resistors for the dip switches to get logical 1 and 0 at the input of the registers.

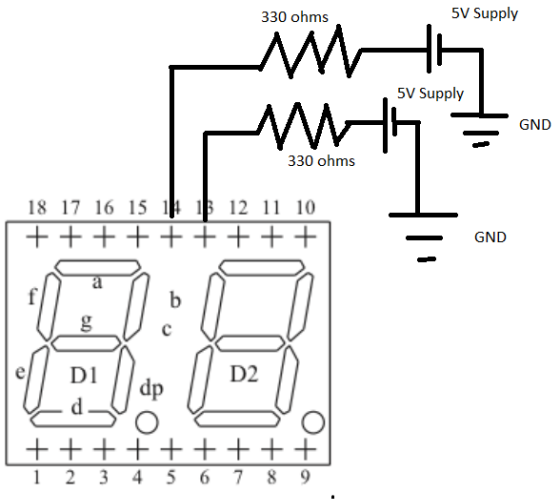
8. Instead of using the function generator to give us our clock pulses we built our own clock circuit using a toggle switch and a nand-gate chip.

9. We implemented out hardware and after several debugging steps we could get the circuit to the RTL instructions as required.

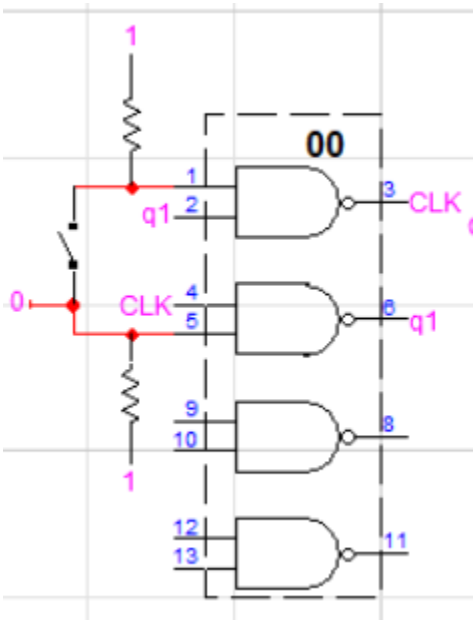
Logic works schematics:

7-Segment display

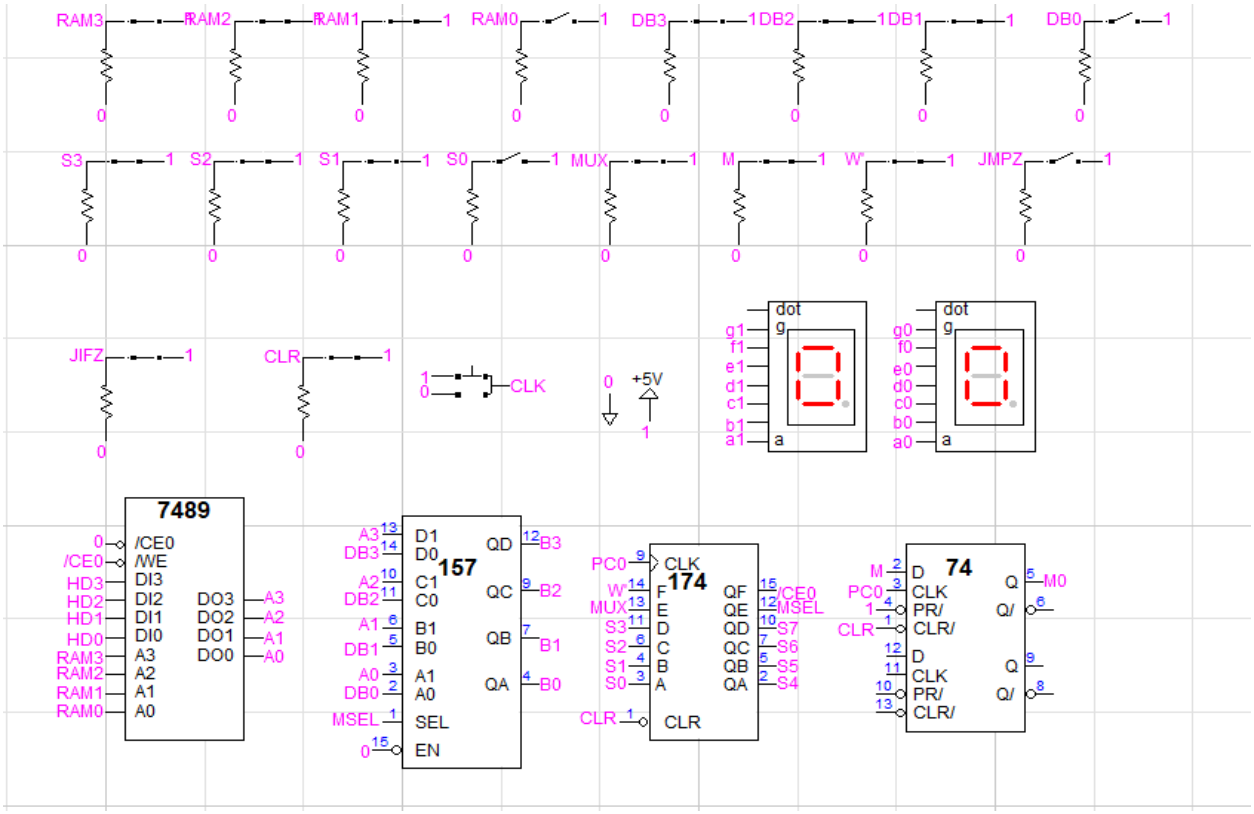
Pin No.	Assignment	Assignment
1	Cathode e1	Anode e1
2	Cathode d1	Anode d1
3	Cathode c1	Anode c1
4	Cathode dp1	Anode dp1
5	Cathode e2	Anode e2
6	Cathode d2	Anode d2
7	Cathode g2	Anode g2
8	Cathode c2	Anode c2
9	Cathode dp2	Anode dp2
10	Cathode b2	Anode b2
11	Cathode a2	Anode a2
12	Cathode f2	Anode f2
13	Common Anode D2	Common Cathode D2
14	Common Anode D1	Common Cathode D1
15	Cathode b1	Anode b1
16	Cathode a1	Anode a1
17	Cathode g1	Anode g1
18	Cathode f1	Anode f1

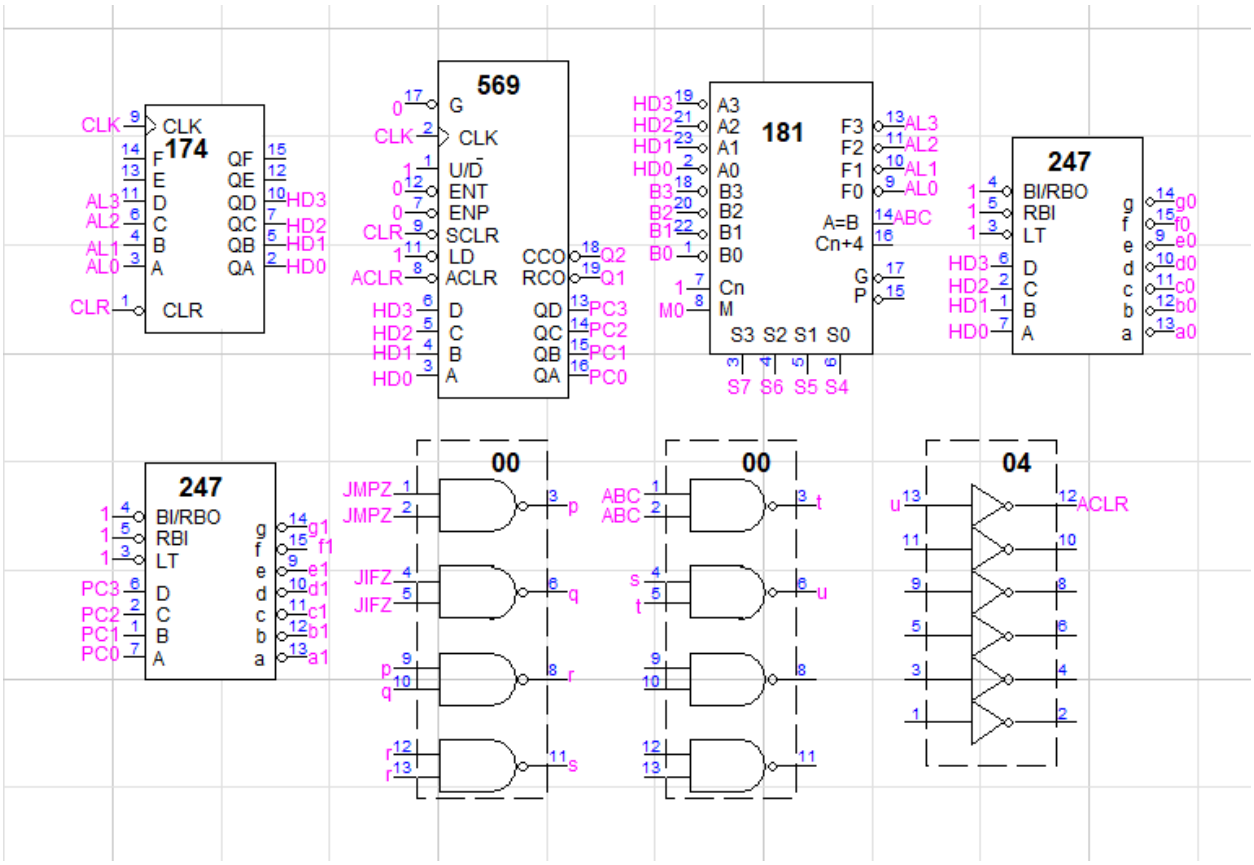


Circuit for De-bouncing switch



Schematics of the circuit




174
CLK
CLR
AL3
AL2
AL1
AL0
HD3
HD2
HD1
HD0
569
ACLR
PC3
PC2
PC1
PC0
181
A=B
Cn+4
S3 S2 S1 S0
247
BI/RBO
RBI
LT
00
JMPZ
JIFZ
04
ABC

Results:

The TA asked us to execute several RTL instructions in series and to note our observations of the clock and the program counter.

We were able to perform each of these instructions successfully.

Group 5

A Run the Following Program

Instruction	ACC Disp.	PC Disp. Fixed and counting	Notes
LDI 9	9	1	
STORE 5	9	2	
LDI 4	4	3	
LOAD 5	9	4	
LDI 2	2	5	
ANDI 4	0	6	
JIFN	0	7	
SUB 5	6	8	Did not count correct
INV	9	9	
JMPZ	9	0	
LDI 15	F	1	
JIFN	F	0	

!!! PUT THESE RESULTS IN THE REPORT!!!

Conclusion:

We successfully designed, simulated and made a hardware realization of the CPU, and learnt how to design the CPU controller which is responsible for policing all the data and address transfer logic of the CPU.

FTQs:

Perform the following instruction code and explain the Result.

1. [Priya Parikh]

Byte 0					Byte 1	
M	MUX	W’	Cn	S	DB	RAM
1	1	1	0	0000	0000	0000

Ans:

M	1	Shows mode control input high
MUX	1	Indicate RAM is being used
W’	1	Read operation
Cn	0	Carry input is don’t care for the given operation
S	0000	Instruction execution A’
DB	0000	Value in ACC
RAM	0000	Register Location / address

2. Lalit Bhat

Byte 0					Byte 1	
M	MUX	W’	Cn	S	DB	RAM
1	1	0	1	0000	0001	0100

According to my circuit when MUX=1 it selects the data from the ram.

The OP code M=1 and S= 0000 means no operation means the ALU will have to invert the value at the input A (which in my case the accumulator value us loaded to A)

The RAM is addressed to the location 0100, and the write is enabled i.e. W’=1

When these modes are set on my circuit the values currently on the accumulator will be inverted and shown on the display and that value will get stored to the RAM address 0100.