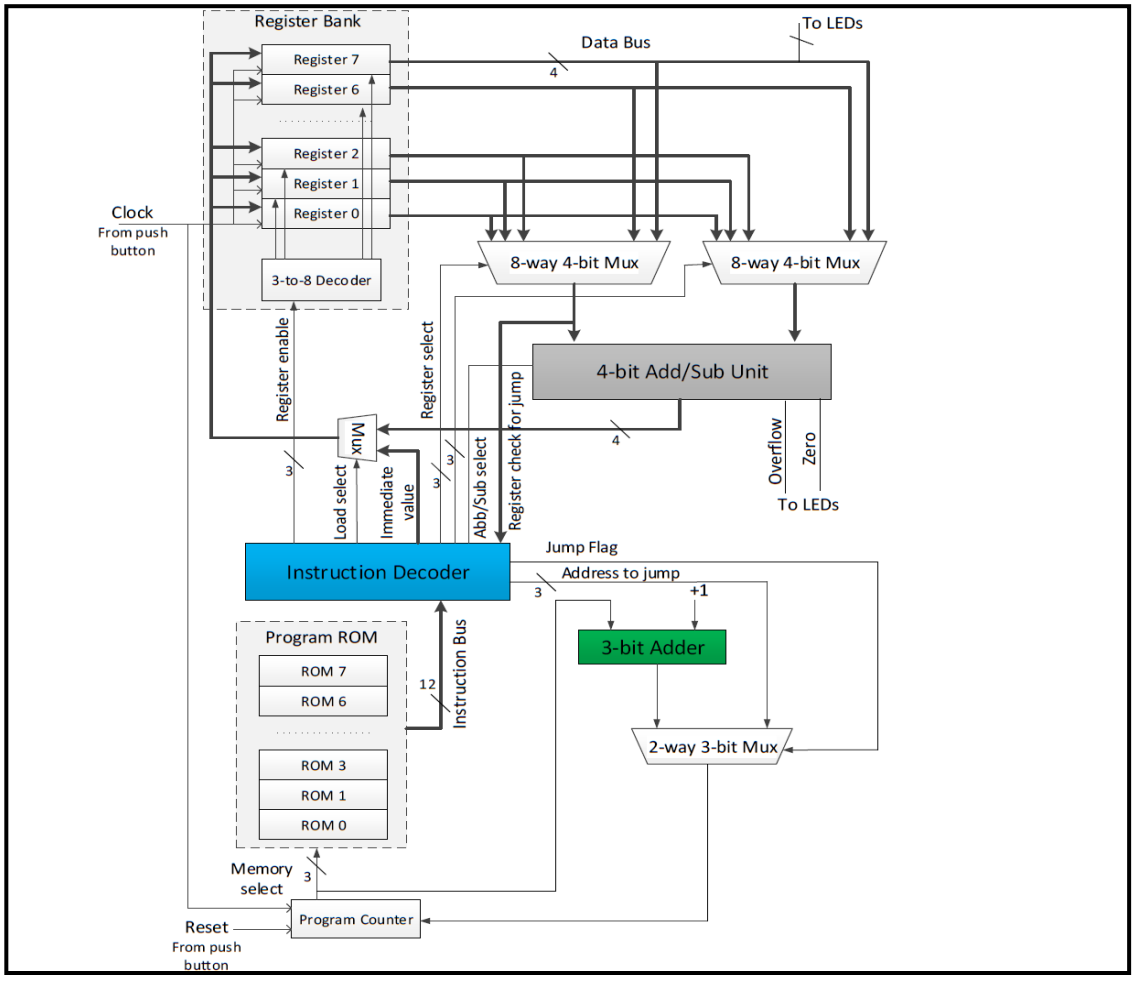
Nano processor Design (Lab 9 & 10 – Report)

Group Members	:	S. Sabesan, P. Loghi, B. Abinayan, T. Nirojan
Index numbers	:	160538L, 160358J, 160007J, 160442L
Group	:	CSE A (Group - L)
Title	:	Nano Processor Design
Date	:	16-11-2017

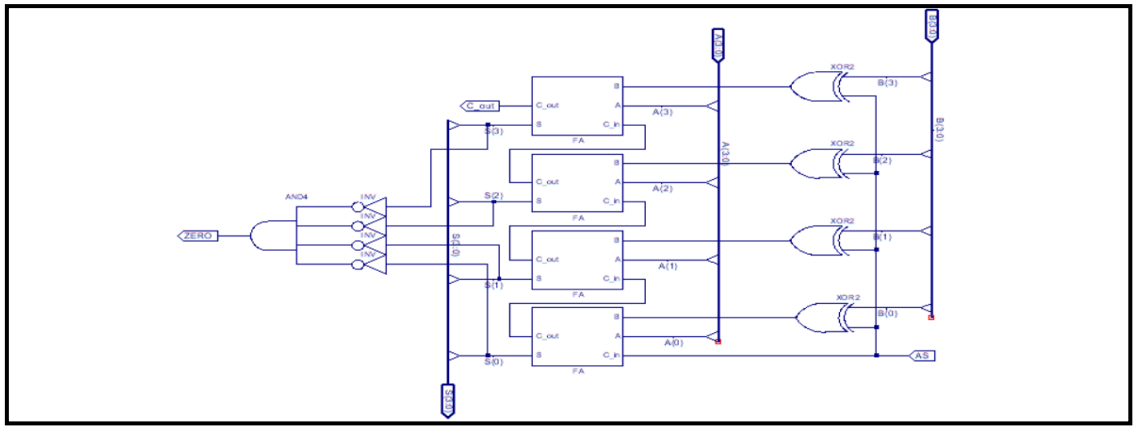
Lab Task

- Design the components which are needed to make Processor.
- Check the liability of the components using ISim simulator.
- Design a 4-bit processor capable of executing 4 instructions by combining the created components. The high-level architecture of the Nano processor is given below.



1. 4-bit Add/Subtract unit

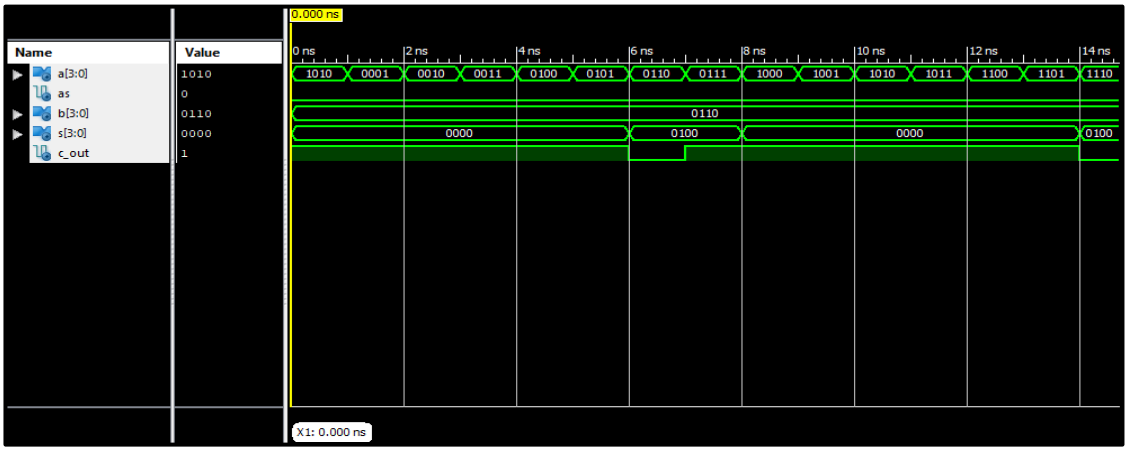
Schematic Diagram



Test bench Code

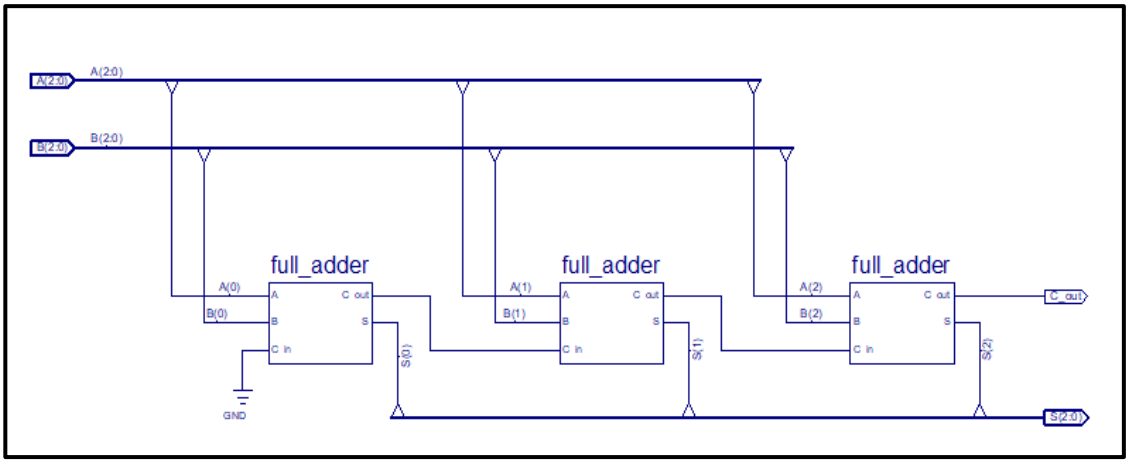
```
48  -- *** Test Bench - User Defined Section ***
49     tb : PROCESS
50     BEGIN
51     AS <= '0' ;
52
53     B<= "0001" ;
54        A <= "0000" ;
55     WAIT FOR 1 NS;
56
57     A <= "0001" ;
58     WAIT FOR 1 NS;
59
60     A <= "0010" ;
61     WAIT FOR 1 NS;
62
63     A <= "0011" ;
64     WAIT FOR 1 NS;
65
66     A <= "0100" ;
67     WAIT FOR 1 NS;
68
69     A <= "0101" ;
70     WAIT FOR 1 NS;
71
72     A <= "0110" ;
73     WAIT FOR 1 NS;
74
75     A <= "0111" ;
76     WAIT FOR 1 NS;
77
78     A <= "1000" ;
79     WAIT FOR 1 NS;
80
81     A <= "1001" ;
82     WAIT FOR 1 NS;
83
84     A <= "1010" ;
85     WAIT FOR 1 NS;
86
87     A <= "1011" ;
88     WAIT FOR 1 NS;
89
90     A <= "1100" ;
91     WAIT FOR 1 NS;
92
93     A <= "1101" ;
94     WAIT FOR 1 NS;
95
96     A <= "1110" ;
97     WAIT FOR 1 NS;
98
99     A <= "1111" ;
100    WAIT FOR 1 NS;
101
102       AS <= '1' ;
103
104    A<= "0001" ;
105       B <= "0000" ;
106    WAIT FOR 1 NS;
107
108    B <= "0001" ;
109    WAIT FOR 1 NS;
110
111    B <= "0010" ;
112    WAIT FOR 1 NS;
113
114    B <= "0011" ;
115    WAIT FOR 1 NS;
116
117    B <= "0100" ;
118    WAIT FOR 1 NS;
119
120    B <= "0101" ;
121    WAIT FOR 1 NS;
122
123    B <= "0110" ;
124    WAIT FOR 1 NS;
125
126    B <= "0111" ;
```

Timing Diagram

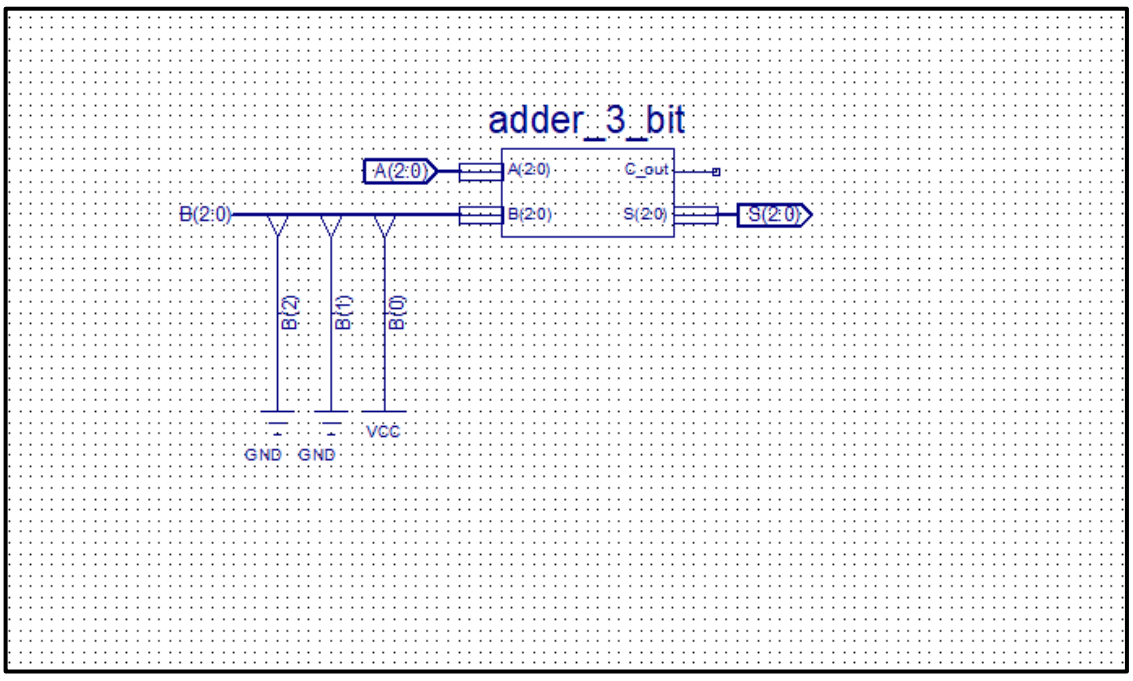


2. 3-bit Adder

Schematic Diagram



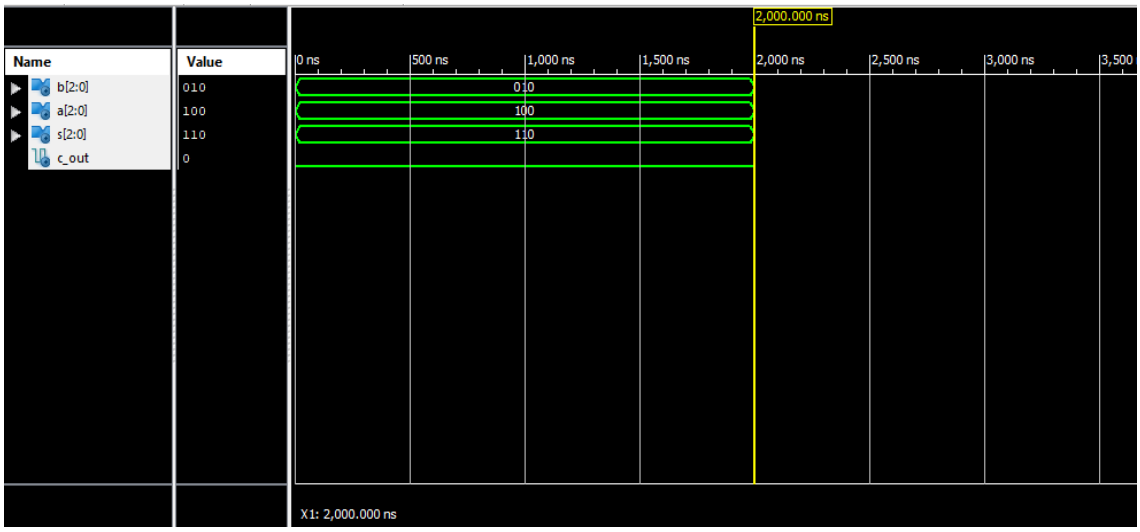
Incrementor (Build using 3-bit adder)



Test bench Code

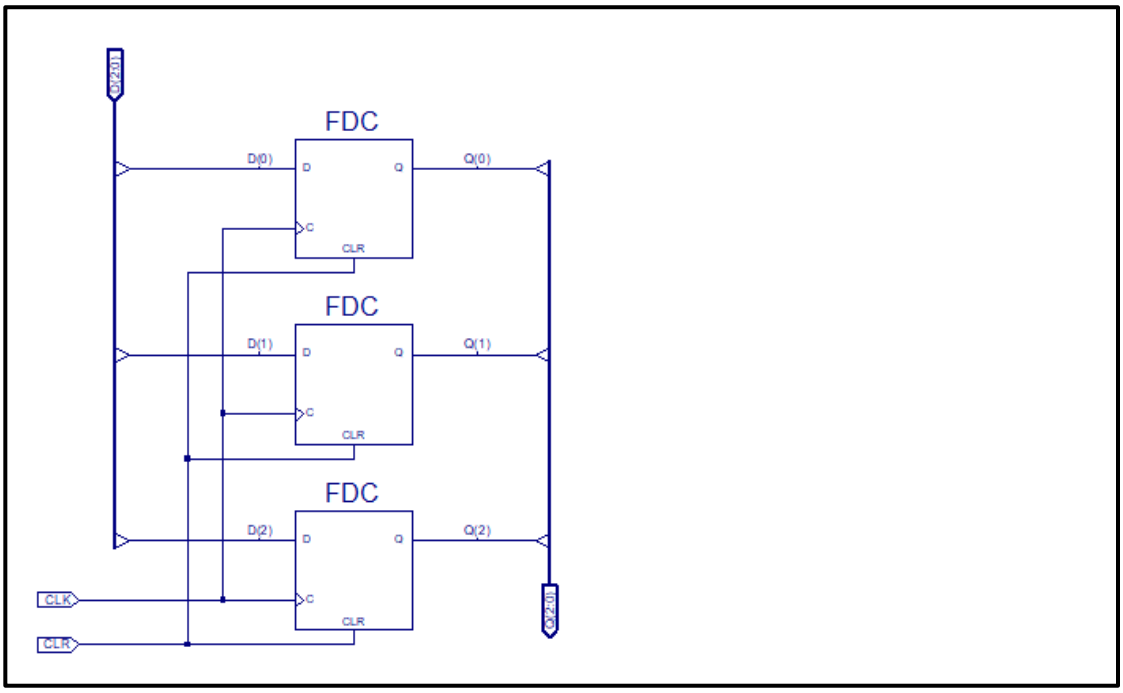
```
42  -- *** Test Bench - User Defined Section ***
43     tb : PROCESS
44     BEGIN
45     A <= "001" ;
46     WAIT FOR 1 NS;
47
48     A <= "010" ;
49     WAIT FOR 1 NS;
50
51     A <= "011" ;
52     WAIT FOR 1 NS;
53
54     A <= "100" ;
55     WAIT FOR 1 NS;
56
57     A <= "101" ;
58     WAIT FOR 1 NS;
59
60     A <= "110" ;
61     WAIT FOR 1 NS;
62
63     A <= "111" ;
64     WAIT FOR 1 NS;
65
66     A <= "000" ;
67     WAIT FOR 1 NS;
68
69        WAIT; -- will wait forever
```

Timing Diagram



3. 3-bit Program Counter

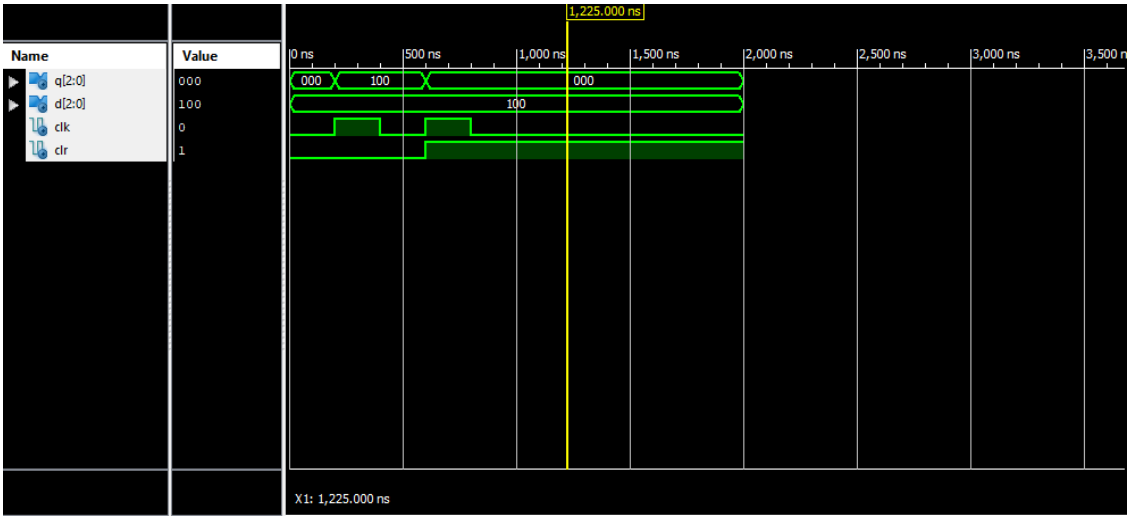
Schematic Diagram



Test bench Code

```
37
38    UUT: program_counter PORT MAP(
39       Q => Q,
40       D => D,
41       CLK => CLK,
42       CLR => CLR
43    );
44
45  -- *** Test Bench - User Defined Section ***
46    tb : PROCESS
47    BEGIN
48       D(2) <= '1'; D(1) <= '0'; D(0) <= '0';
49       CLK <= '0';
50       CLR <= '0';
51       WAIT FOR 200 ns;
52       CLK <= '1';
53       WAIT FOR 200 ns;
54       CLK <= '0';
55       WAIT FOR 200 ns;
56       CLR <= '1';
57       CLK <= '1';
58       WAIT FOR 200 ns;
59       CLK <= '0';
60
61       WAIT; -- will wait forever
62    END PROCESS;
63  -- *** End Test Bench - User Defined Section ***
64
```

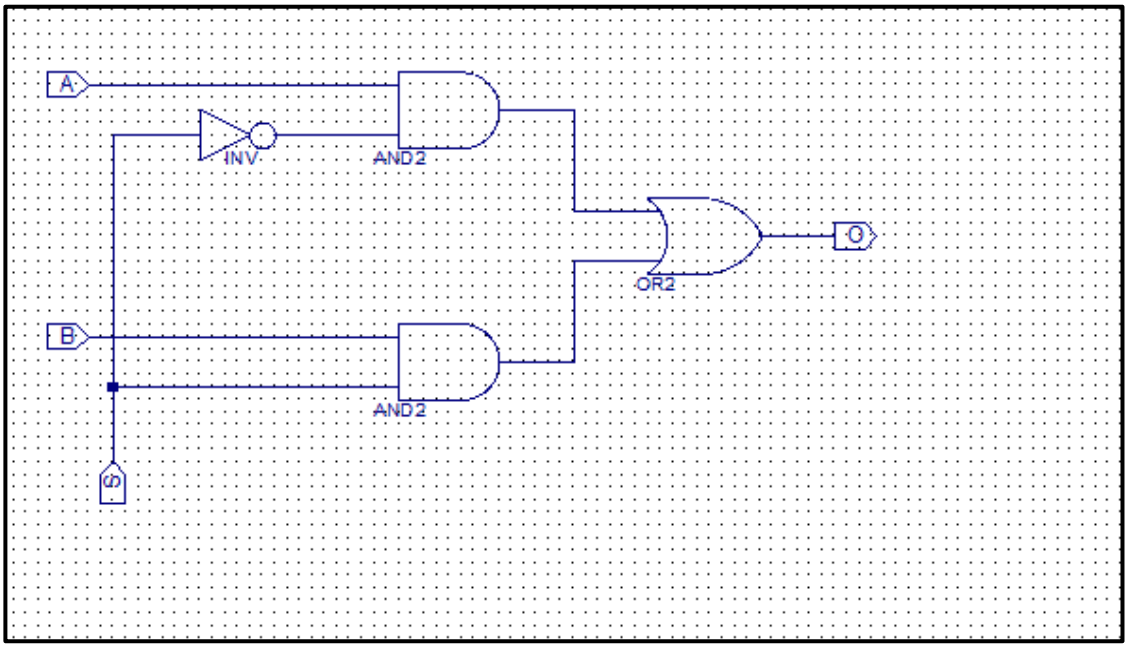
Timing Diagram



4. K-way b-bit Multiplexer

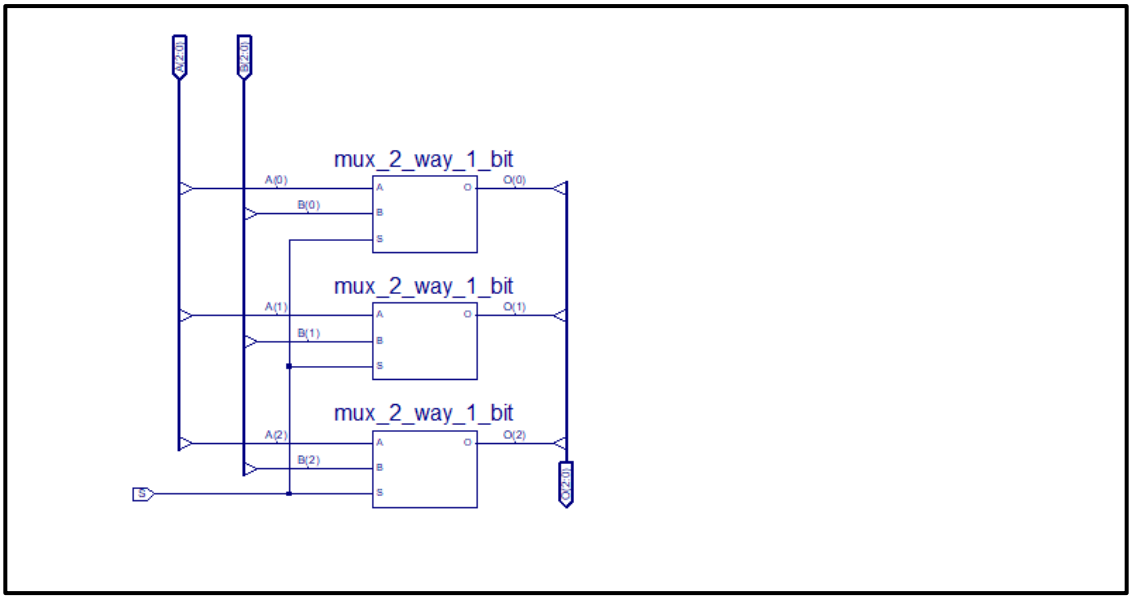
i. 2-way 1-bit Multiplexer

Schematic Diagram



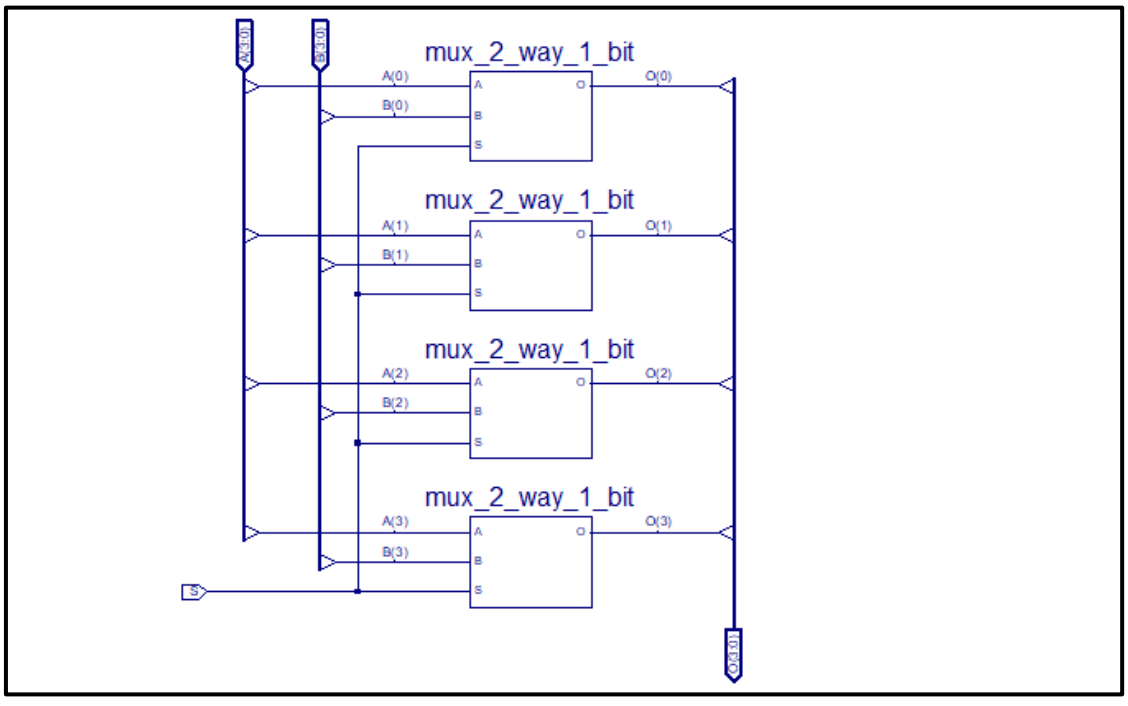
ii. 2-way 3-bit Multiplexer

Schematic Diagram



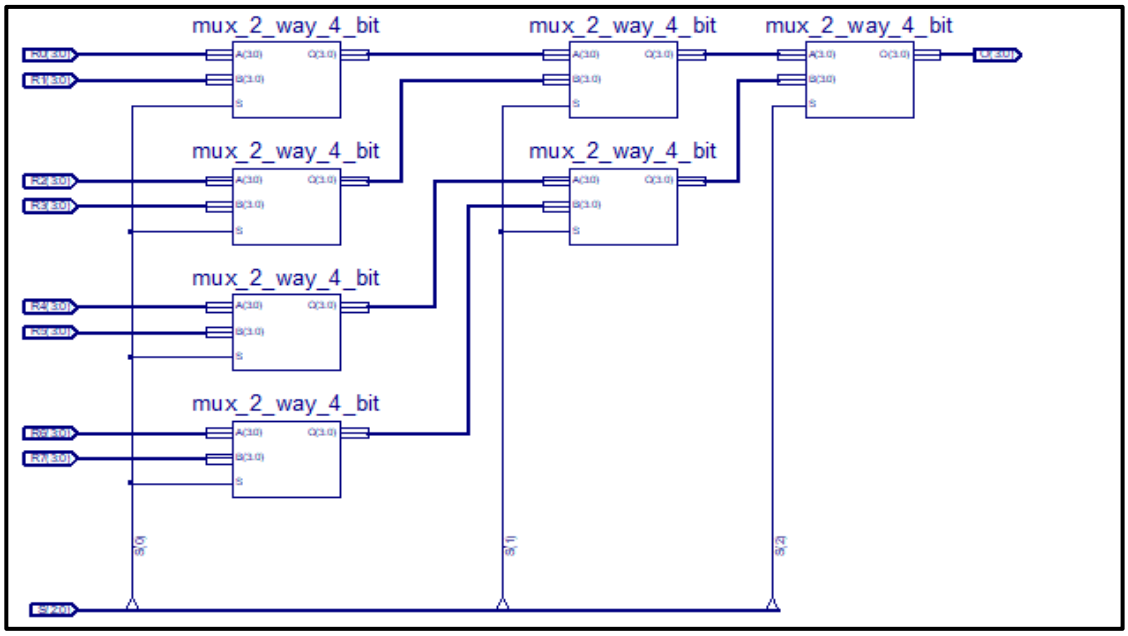
iii. 2-way 4-bit Multiplexer

Schematic Diagram



iv. 8-way 4-bit Multiplexer

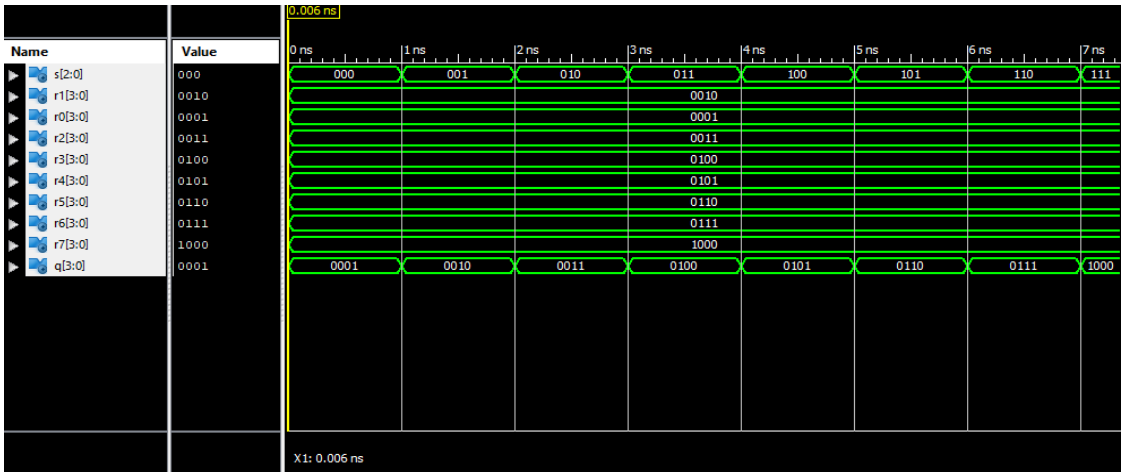
Schematic Diagram



Test bench Code

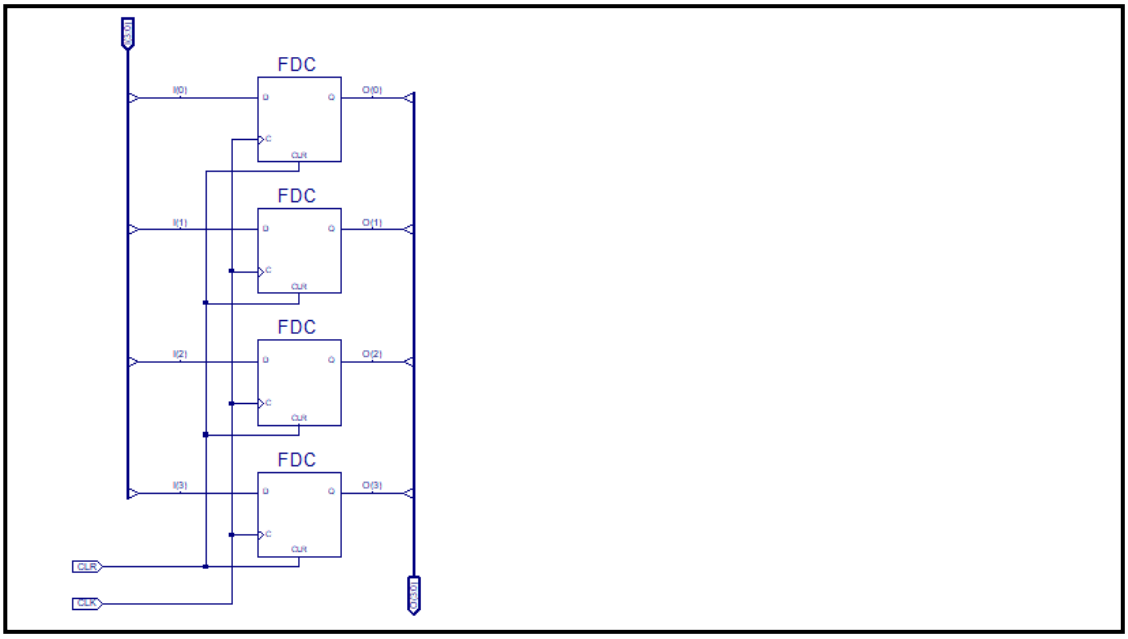
```
63  -- *** Test Bench - User Defined Section ***
64     tb : PROCESS
65     BEGIN
66
67     R0 <= "0001" ;
68     R1 <= "0010" ;
69     R2 <= "0011" ;
70     R3 <= "0100" ;
71     R4 <= "0101" ;
72     R5 <= "0110" ;
73     R6 <= "0111" ;
74     R7 <= "1000" ;
75
76     S <="000";
77     WAIT FOR 1 NS;
78
79     S <="001";
80     WAIT FOR 1 NS;
81
82     S <="010";
83     WAIT FOR 1 NS;
84
85     S <="011";
86     WAIT FOR 1 NS;
87
88     S <="100";
89     WAIT FOR 1 NS;
90
91     S <="101";
92     WAIT FOR 1 NS;
93
94     S <="110";
95     WAIT FOR 1 NS;
96     S <="111";
97     WAIT FOR 1 NS;
98     S <="000";
99
100    R0 <= "0000" ;
101    R1 <= "0000" ;
102    R2 <= "0000" ;
103    R3 <= "0000" ;
104    R4 <= "0000" ;
105    R5 <= "0000" ;
106    R6 <= "0000" ;
107    R7 <= "0000" ;
108
109    WAIT FOR 1 NS;
110       WAIT; -- will wait forever
111    END PROCESS;
112 -- *** End Test Bench - User Defined Section ***
113
114 END;
```

Timing Diagram

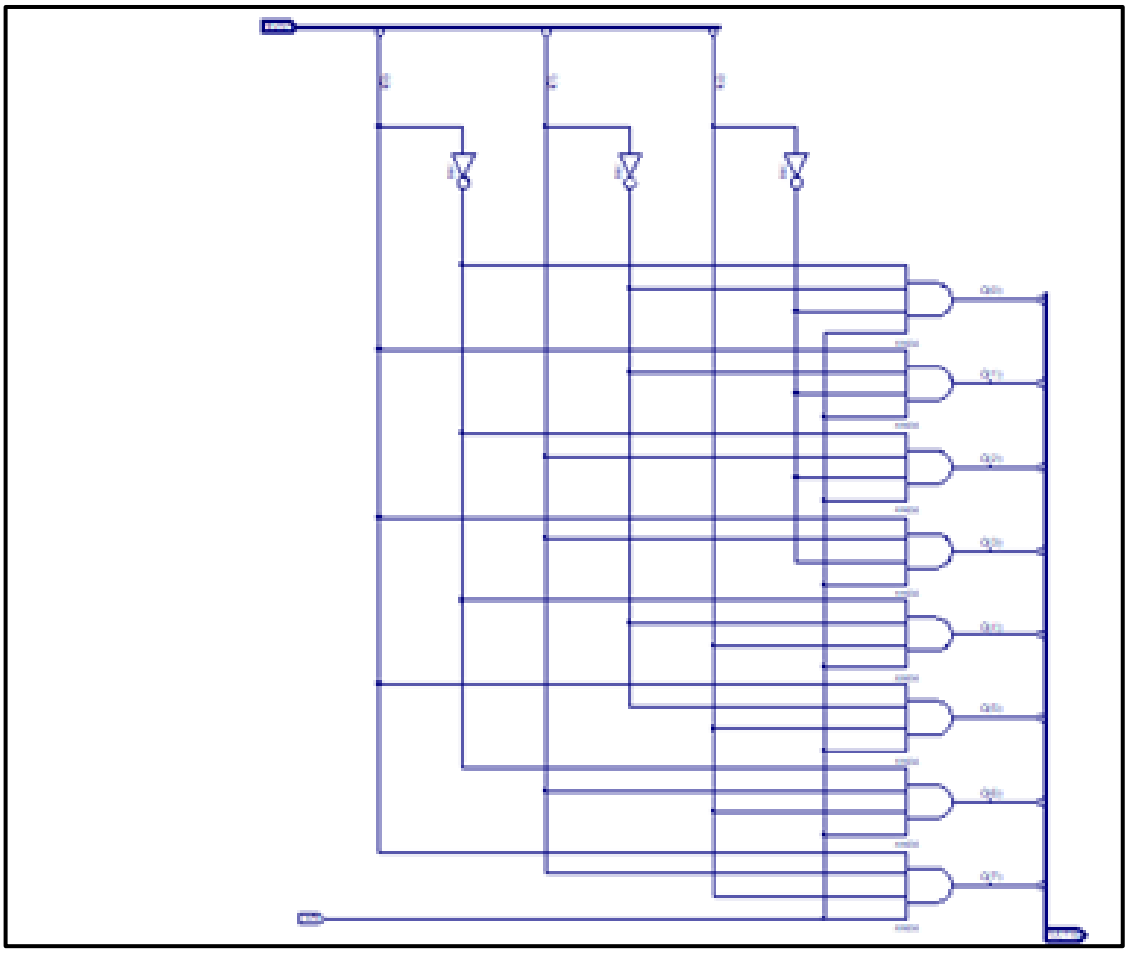


5. Register Bank

4-bit Register (Used in Register Bank)



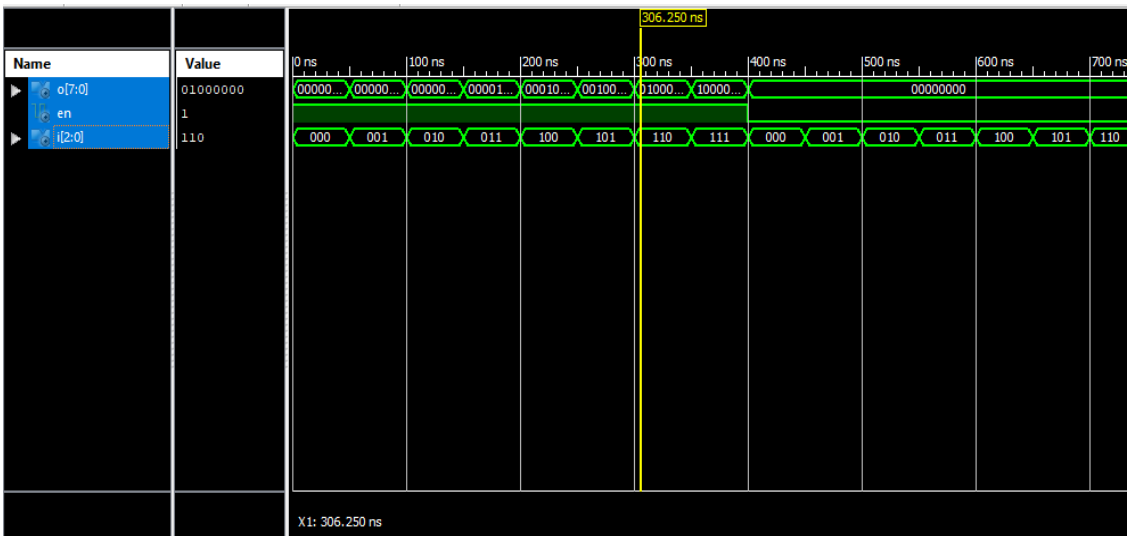
3-8 Decoder (Used in Register Bank)



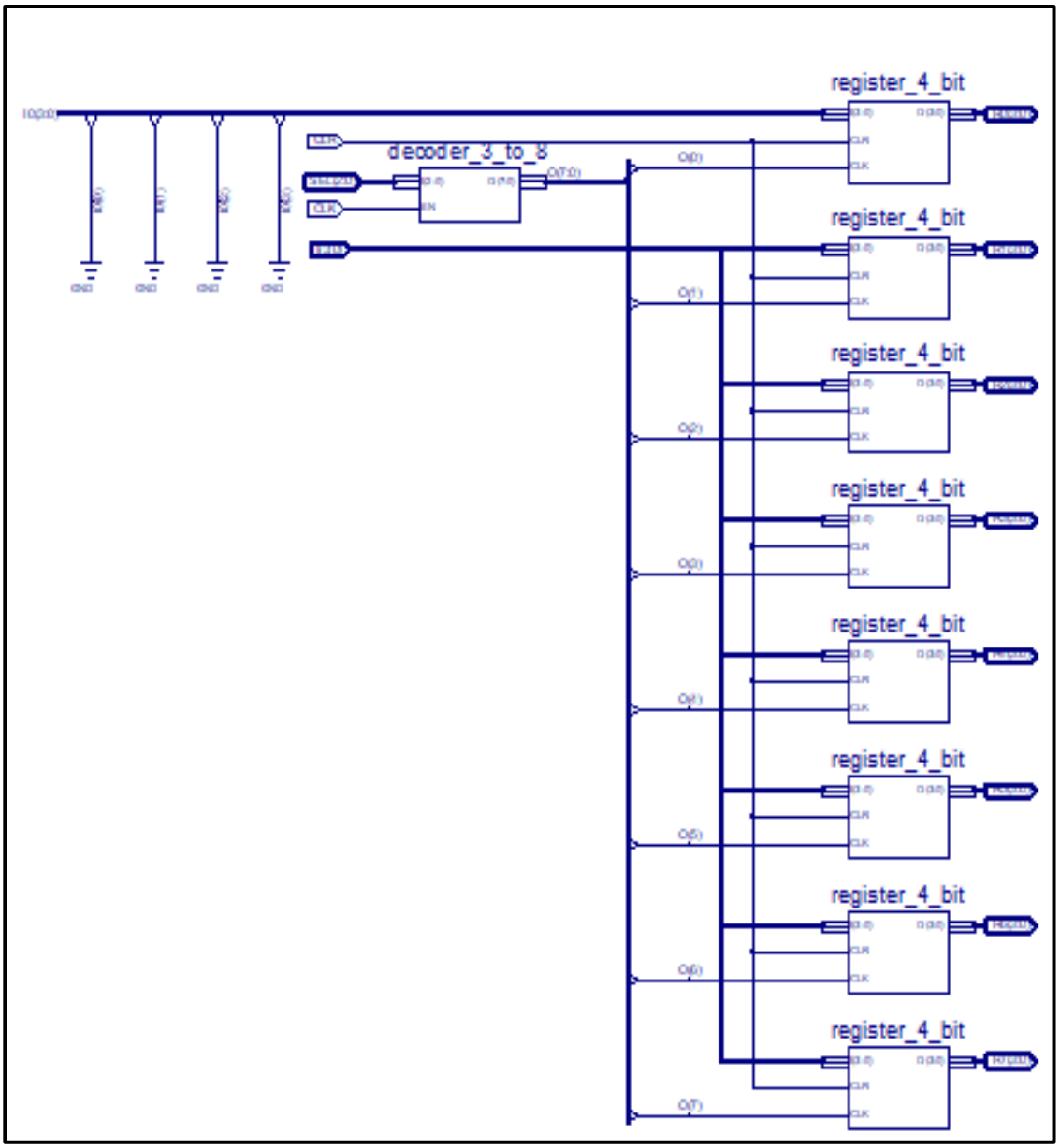
Test bench Code for 3-8 Decoder

```
70    tb : PROCESS
71    BEGIN
72    EN <= '1' ;
73
74    I2 <= '0' ;
75    I1 <= '0' ;
76    I0 <= '0' ;
77    WAIT FOR 1 NS;
78    I0 <= '1' ;
79    WAIT FOR 1 NS;
80    I1 <= '1' ;
81    I0 <= '0' ;
82    WAIT FOR 1 NS;
83    I0 <= '1' ;
84    WAIT FOR 1 NS;
85    I2 <= '1' ;
86    I1 <= '0' ;
87    I0 <= '0' ;
88    WAIT FOR 1 NS;
89    I0 <= '1' ;
90    WAIT FOR 1 NS;
91    I1 <= '1' ;
92    I0 <= '0' ;
93    WAIT FOR 1 NS;
94    I0 <= '1' ;
95    WAIT FOR 1 NS;
96       WAIT; -- will wait forever
97    END PROCESS;
```

Timing Diagram



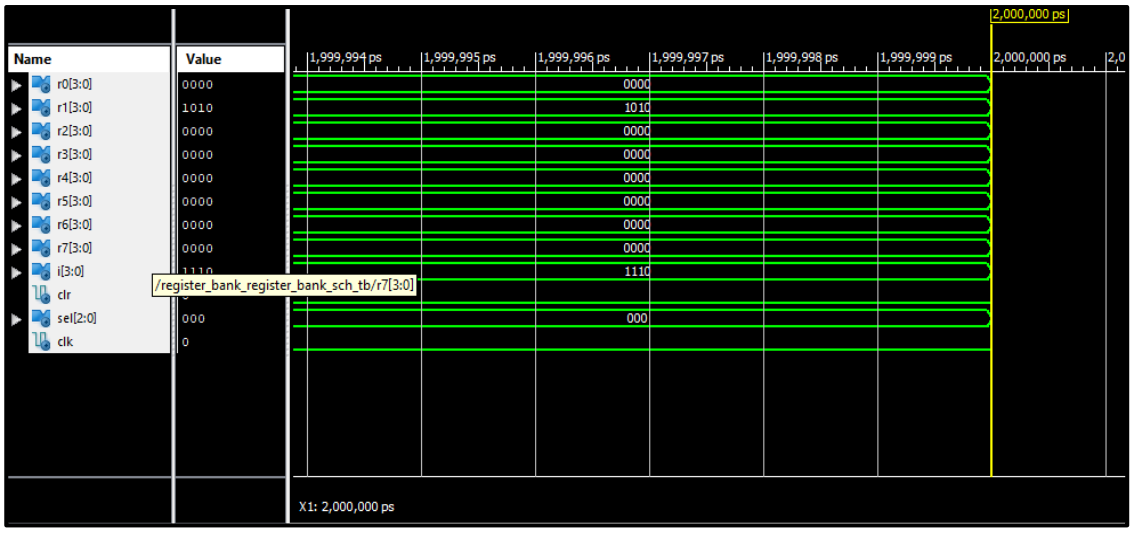
Schematic Diagram



Test bench Code for Register Bank

```
64          CLR => CLR,
65          SEL => SEL,
66          CLK => CLK
67       );
68
69    -- *** Test Bench - User Defined Section ***
70       tb : PROCESS
71       BEGIN
72          I(3) <= '1'; I(2) <= '0'; I(1) <= '1'; I(0) <= '0';
73          CLR <= '0';
74          CLK <= '0';
75          SEL(2) <= '0'; SEL(1) <= '0'; SEL(0) <= '1';
76          WAIT FOR 50 ns;
77          CLK <= '1';
78          WAIT FOR 50 ns;
79          CLK <= '0';
80          I(3) <= '1'; I(2) <= '1'; I(1) <= '1'; I(0) <= '0';
81          SEL(2) <= '0'; SEL(1) <= '0'; SEL(0) <= '0';
82          WAIT FOR 50 ns;
83          CLK <= '1';
84          WAIT FOR 50 ns;
85          CLK <= '0';
86          WAIT; -- will wait forever
87       END PROCESS;
88    -- *** End Test Bench - User Defined Section ***
89
90    END;
91
```

Timing Diagram



6. ROM

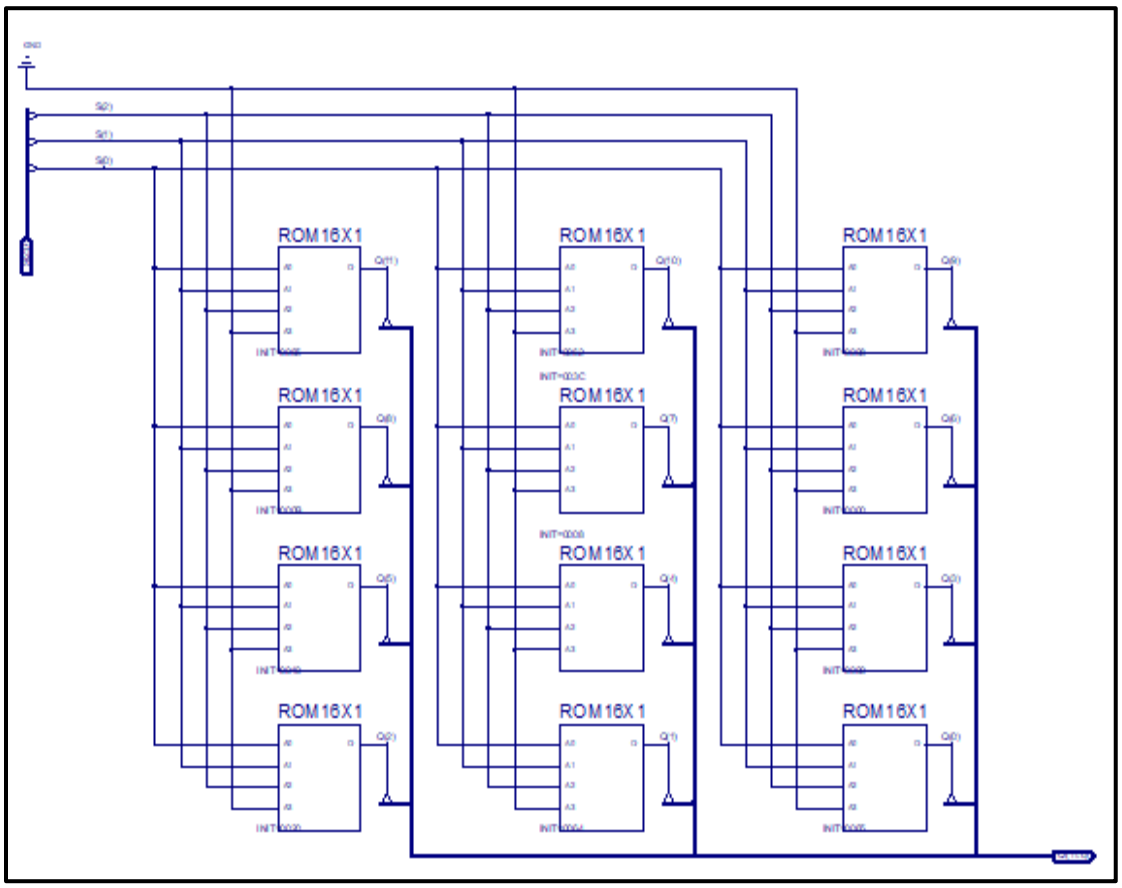
Assembly Program

```
0=>  MOVI   R2, 1;      <-  R2=1;
1=>  NEG    R2;         <-  R2= -1;
2=>  MOVI   R1, 3;      <-  R1 = 3;
3=>  ADD    R7, R1;     <-  R7= R7+R1;
4=>  ADD    R1, R2;     <-  R1 = R1 + R2    i.e.  R2 = R2 - 1;
5=>  JZR    R1, 7;      <-  If R1 = 0, GOTO  7.
6=>  JZR    R0, 3;      <-  If R0 = 0, GOTO  3. (LOOP CONTROL)
7=>  ADD    R0, R0;     <-  R0= R0+R0
```

Look Up Table

Input			Output											
S2	S1	S0	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
0	0	0	1	0	0	1	0	0	0	0	0	0	0	1
0	0	1	0	1	0	1	0	0	0	0	0	0	0	0
0	1	0	1	0	0	0	1	0	0	0	0	0	1	1
0	1	1	0	0	1	1	1	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	1	0	0	0	0	0
1	0	1	1	1	0	0	1	0	0	0	0	1	1	1
1	1	0	1	1	0	0	0	0	0	0	0	0	1	1
1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
			0065	0062	0008	000B	003C	0000	0010	0008	0000	0020	0064	0065

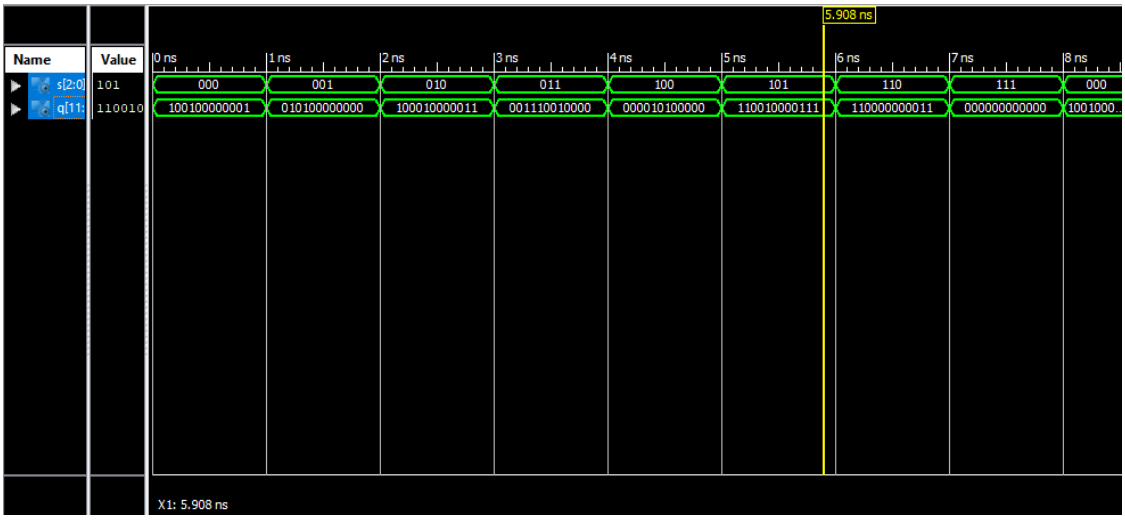
Schematic Diagram



Test bench Code

```
40    tb : PROCESS
41    BEGIN
42    S <= "000" ;
43    WAIT FOR 1 NS ;
44
45    S <= "001" ;
46    WAIT FOR 1 NS ;
47
48    S <= "010" ;
49    WAIT FOR 1 NS ;
50
51    S <= "011" ;
52    WAIT FOR 1 NS ;
53
54    S <= "100" ;
55    WAIT FOR 1 NS ;
56
57
58    S <= "101" ;
59    WAIT FOR 1 NS ;
60
61
62    S <= "110" ;
63    WAIT FOR 1 NS ;
64
65
66    S <= "111" ;
67    WAIT FOR 1 NS ;
68    S <= "000" ;
69    WAIT FOR 1 NS ;
70
71       WAIT; -- will wait forever
72    END PROCESS;
73 -- *** End Test Bench - User Defined Section ***
74
75 END;
76
```

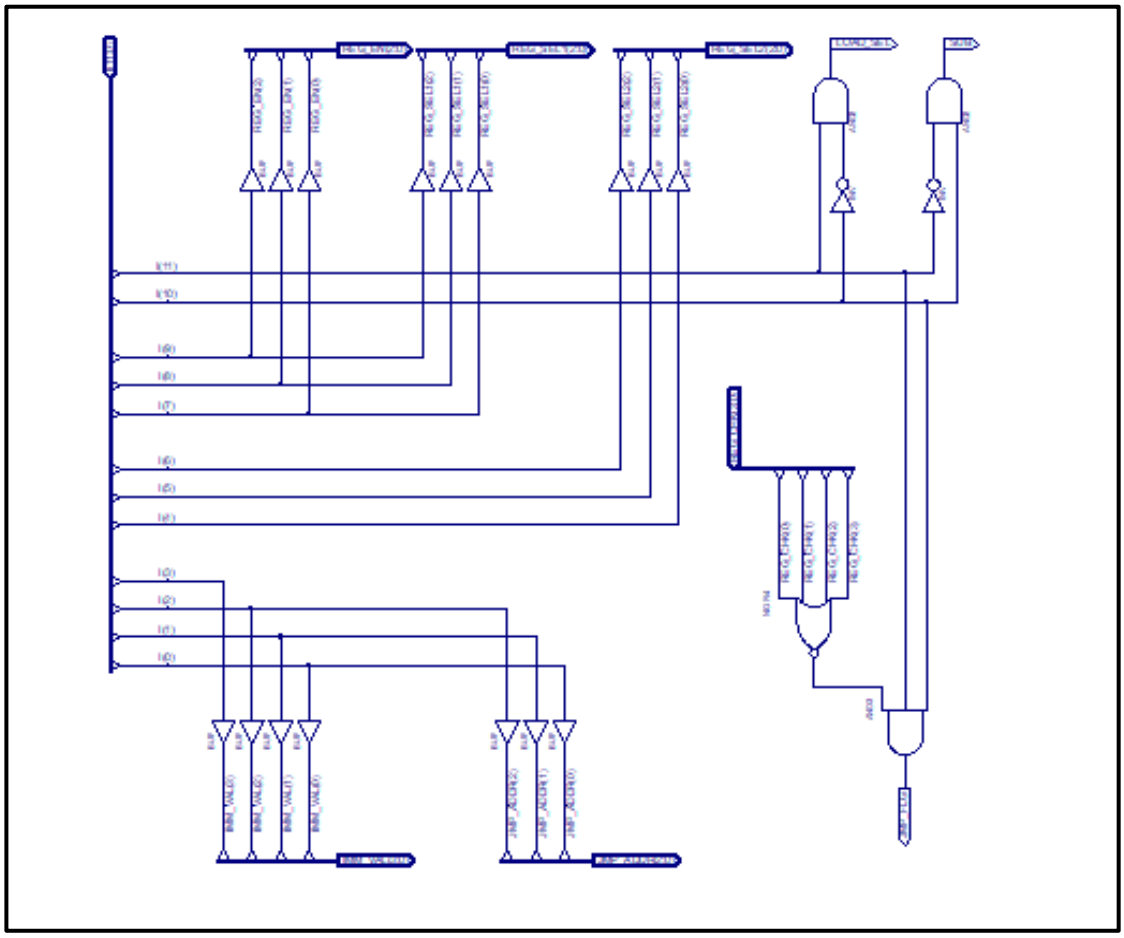
Timing Diagram



7. Instruction Decoder

Operation	I11	I10	Decoder	LSB	Regreen	Mux_A	Mux_B	LOD	SUB	JMP
ADD	0	0	Y0	0	I(9:7)	I(9:7)	I(6:4)	0	0	0
MOVI	0	1	Y1	I(3:0)	I(9:7)	0	0	1	0	0
NEG	1	0	Y2	0	I(9:7)	0	I(9:7)	0	1	0
JZR	1	1	Y3	I(2:0)	0	I(9:7)	0	0	0	1

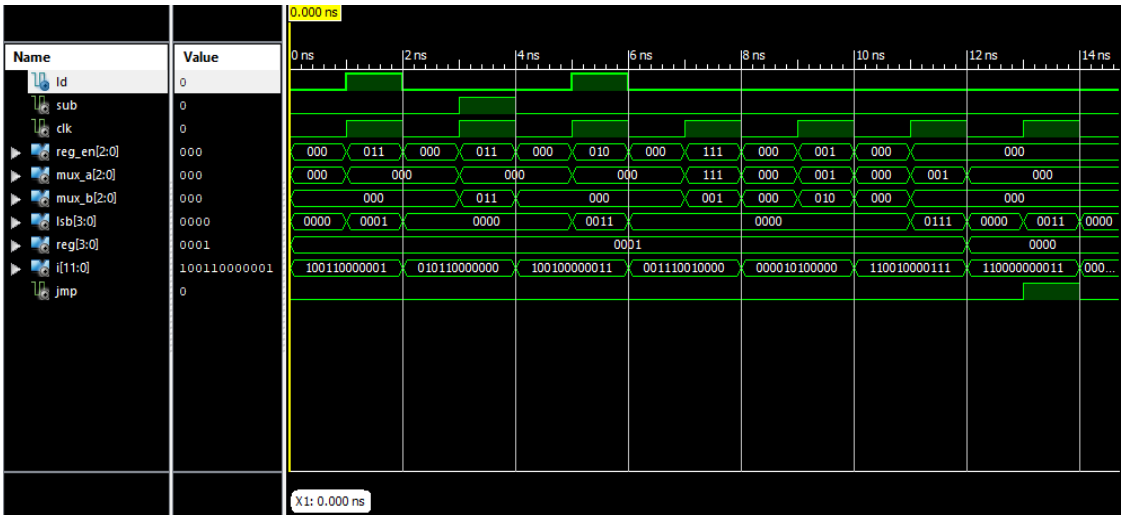
Schematic Diagram



Test bench Code

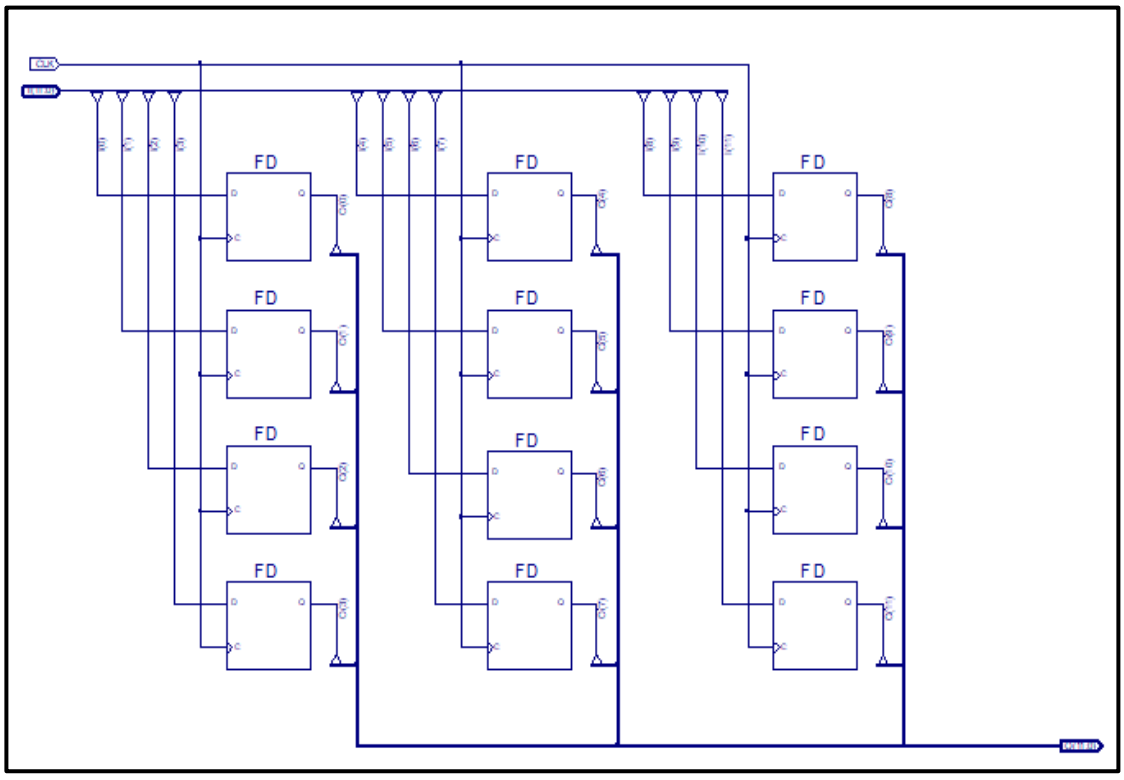
```
64    tb : PROCESS
65    BEGIN
66    REG <= "0001" ;
67
68    I <= "100110000001" ;
69    CLK <= '0' ;
70    WAIT FOR 1 NS;
71
72    CLK <= '1' ;
73    WAIT FOR 1 NS;
74
75    I <= "0101100000000" ;
76    CLK <= '0' ;
77    WAIT FOR 1 NS;
78
79    CLK <= '1' ;
80    WAIT FOR 1 NS;
81
82    I <= "100100000011" ;
83    CLK <= '0' ;
84    WAIT FOR 1 NS;
85
86    CLK <= '1' ;
87    WAIT FOR 1 NS;
88
89    I <= "000010100000" ;
90    CLK <= '0' ;
91    WAIT FOR 1 NS;
92
93    CLK <= '1' ;
94    WAIT FOR 1 NS;
95
96    I <= "000100110000" ;
97    CLK <= '0' ;
98    WAIT FOR 1 NS;
99
100   CLK <= '1' ;
101   WAIT FOR 1 NS;
102
103   I <= "1101000001111" ;
104   CLK <= '0' ;
105   WAIT FOR 1 NS;
106
107   CLK <= '1' ;
108   WAIT FOR 1 NS;
109   REG <= "0000" ;
110
111      I <= "1100000000011" ;
112   CLK <= '0' ;
113   WAIT FOR 1 NS;
114
115   CLK <= '1' ;
116   WAIT FOR 1 NS;
117
118   I <= "001110010000" ;
119   CLK <= '0' ;
```

Timing Diagram



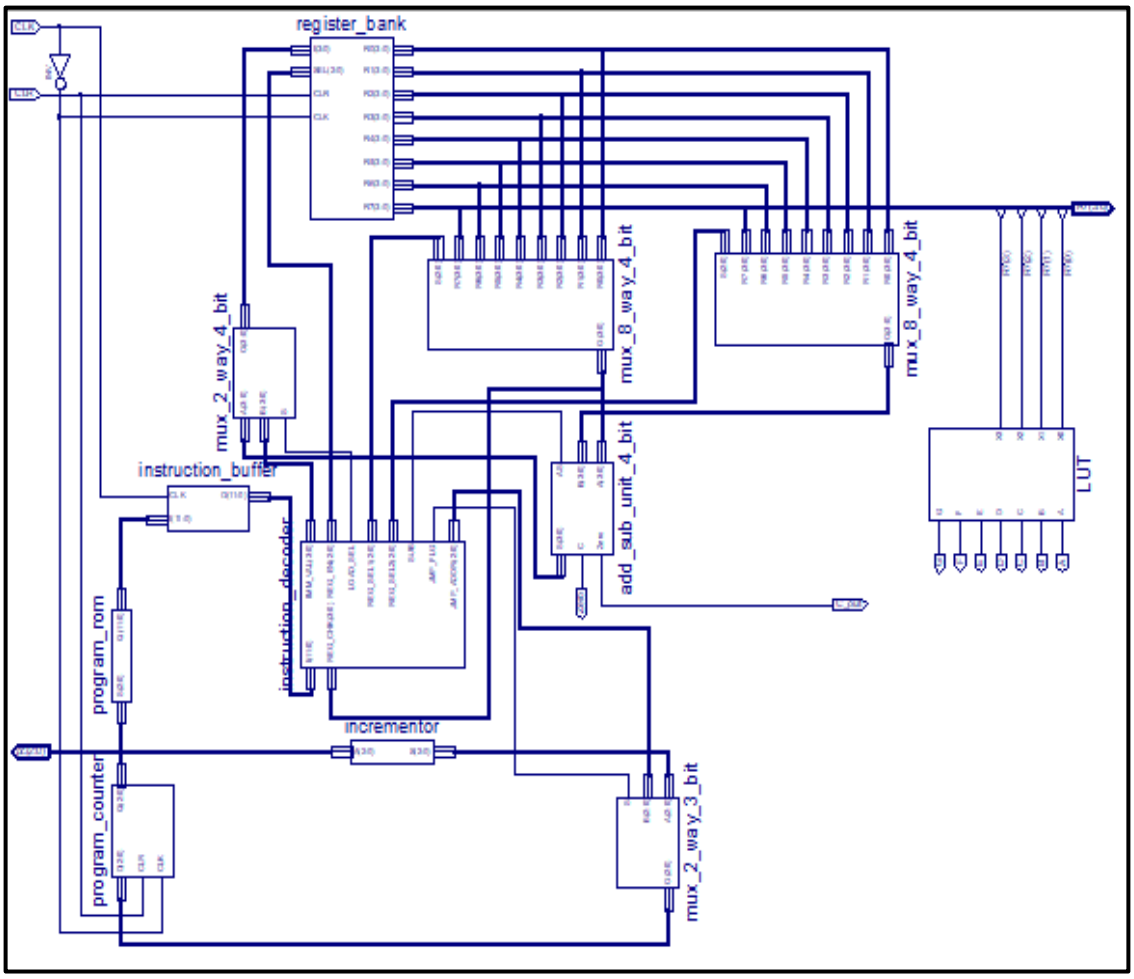
8. Instruction Buffer

Schematic Diagram



9. Nano Processor

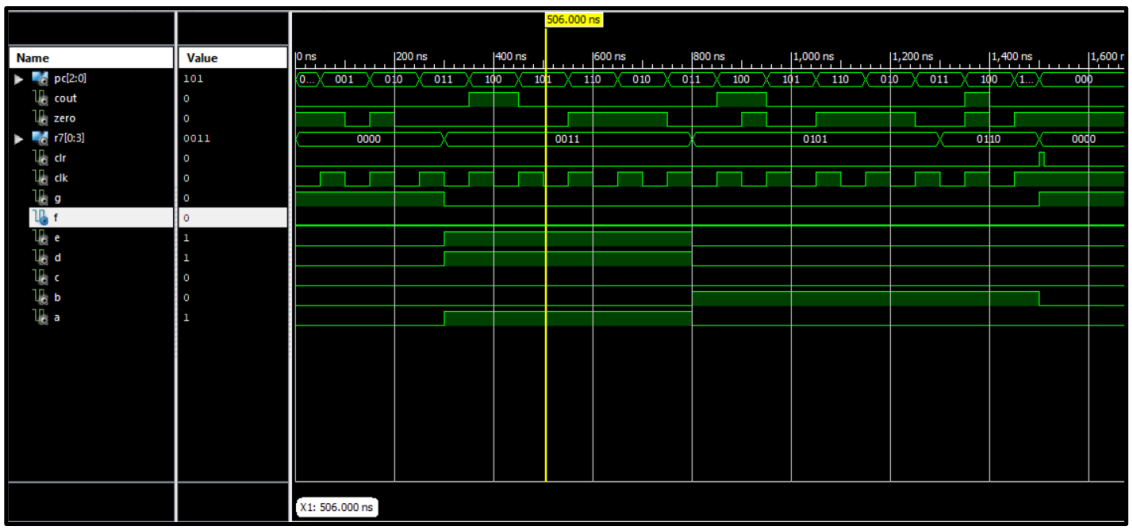
Schematic Diagram



Test bench Code

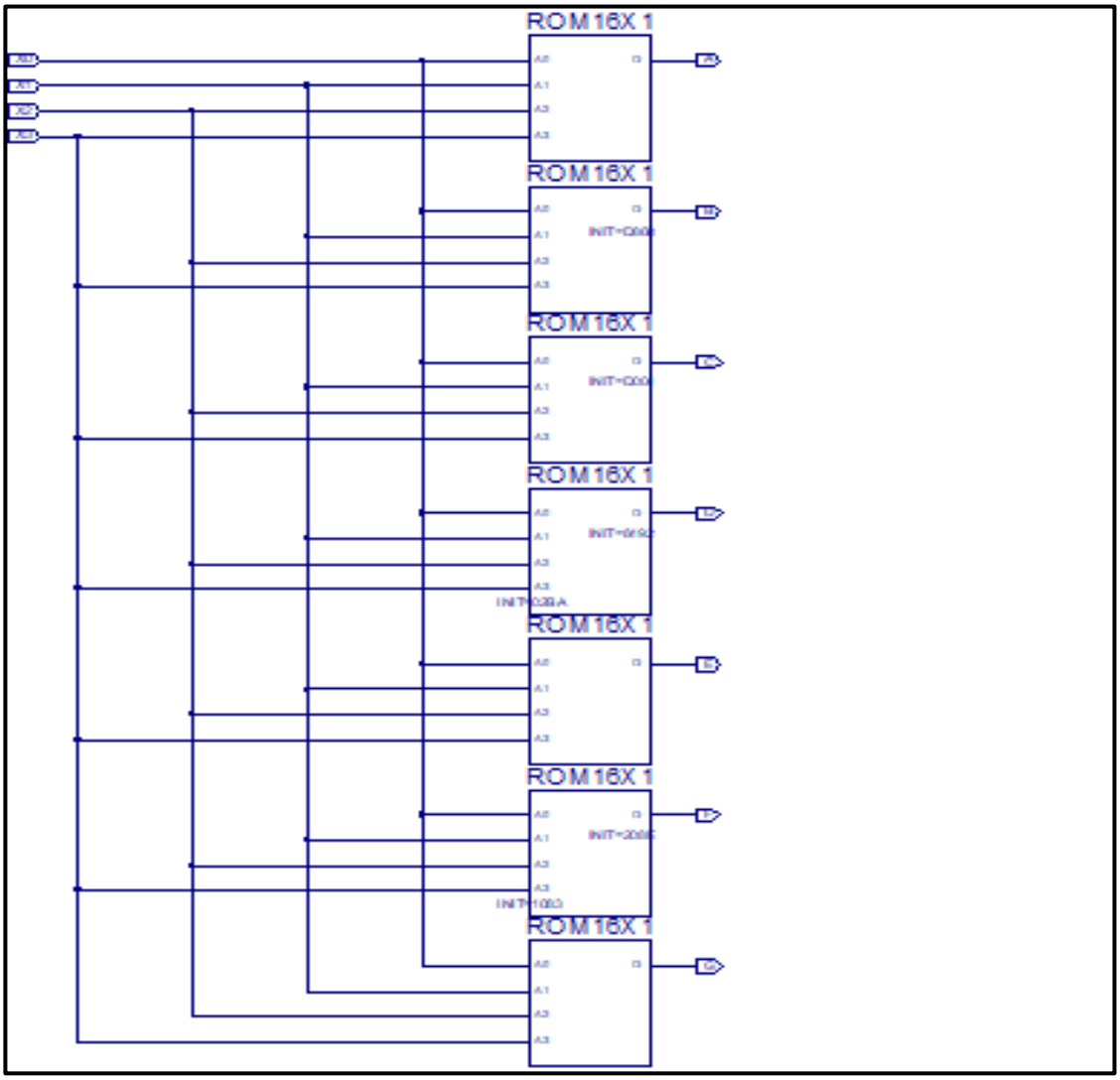
```
46      tb : PROCESS
47      BEGIN
48          CLK <= '0';
49          CLK <= '0';
50          WAIT FOR 50 ns;
51          CLK <= '1';
52          WAIT FOR 50 ns;
53
54          CLK <= '0';
55          WAIT FOR 50 ns;
56          CLK <= '1';
57          WAIT FOR 50 ns;
58
59          CLK <= '0';
60          WAIT FOR 50 ns;
61          CLK <= '1';
62          WAIT FOR 50 ns;
63
64          CLK <= '0';
65          WAIT FOR 50 ns;
66          CLK <= '1';
67          WAIT FOR 50 ns;
68
69          CLK <= '0';
70          WAIT FOR 50 ns;
71          CLK <= '1';
72          WAIT FOR 50 ns;
73
```

Timing Diagram



10. Display the Results in 7-Segment

Schematic Diagram



11. Conclusion

From this lab we were able to:

- design and develop a 4-bit arithmetic unit that can add and subtract integers.
- decode instructions and activate necessary components on the processor.
- design and develop K-way b-bit multiplexers.
- develop and use busses.
- verify their functionality via simulation and on the development board.

***

Date of submission : - 24-11-2017