Lab 3 Report

ECE 124

Group 3 Session 201

Nandita Lohani

Puneet Bhullar

Magnitude Comparator Truth Table

Comparison Inputs from 1-Bit Comparators												4-Bit Comparator Outputs		
A3<B3	A3=B3	A3>B3	A2<B2	A2=B2	A2>B2	A1<B1	A1=B1	A1>B1	A0<B0	A0=B0	A0>B0	A<B	A=B	A>B
0	0	1	X	X	X	X	X	X	X	X	X	0	0	1
1	0	0	X	X	X	X	X	X	X	X	X	1	0	0
0	1	0	0	0	1	X	X	X	X	X	X	0	0	1
0	1	0	1	0	0	X	X	X	X	X	X	1	0	0
0	1	0	0	1	0	0	0	1	X	X	X	0	0	1
0	1	0	0	1	0	1	0	0	X	X	X	1	0	0
0	1	0	0	1	0	0	1	0	0	0	1	0	0	1
0	1	0	0	1	0	0	1	0	1	0	0	1	0	0
0	1	0	0	1	0	0	1	0	0	1	0	0	1	0

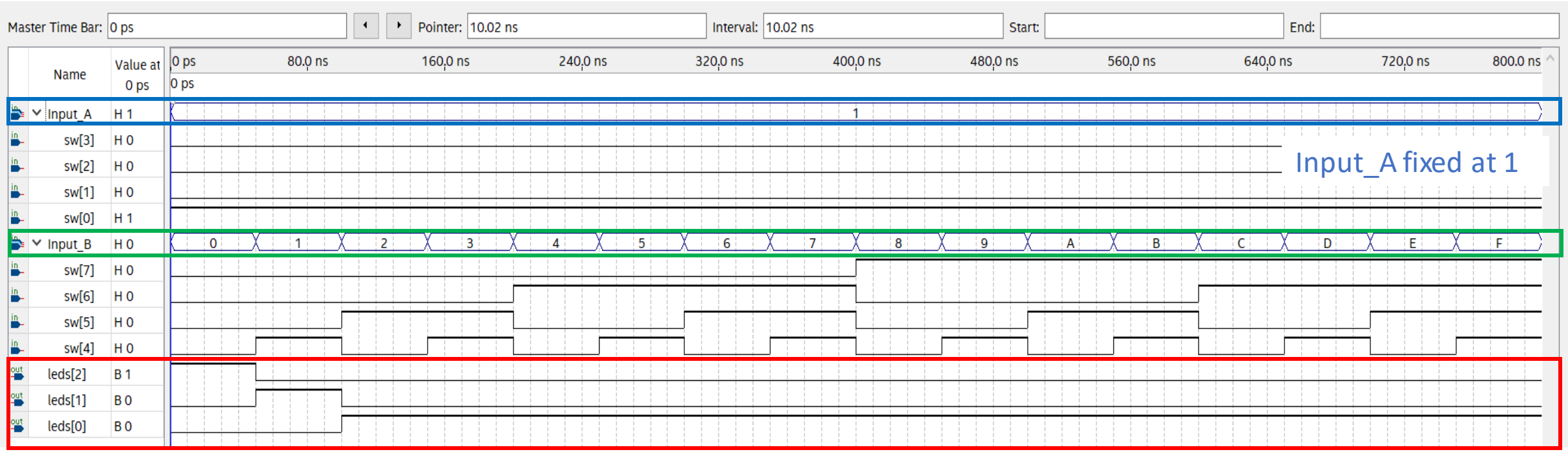
Single Bit Magnitude Comparator (Compx1)

```vhdl
--Author: Group 3, Nandita Lohani, Puneet Bhullar
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

----------------------------------------------------------------------
entity Compx1 is
    port (
            A, B : in std_logic;
            AgreatB, AequalB, AlessB : out std_logic
        );
end Compx1;

-- ********************************************************************************
-- *  Architecture                                                                *
-- ********************************************************************************

architecture Single_Bit_Comparator of Compx1 is

begin

-- boolean logic applied to each output with input operands

AgreatB <= A and (not B);
AequalB <=  A xnor B;
AlessB  <=  (not A) and B;

end architecture Single_Bit_Comparator;
----------------------------------------------------------------
```

4-Bit Magnitude Comparator (Compx4)

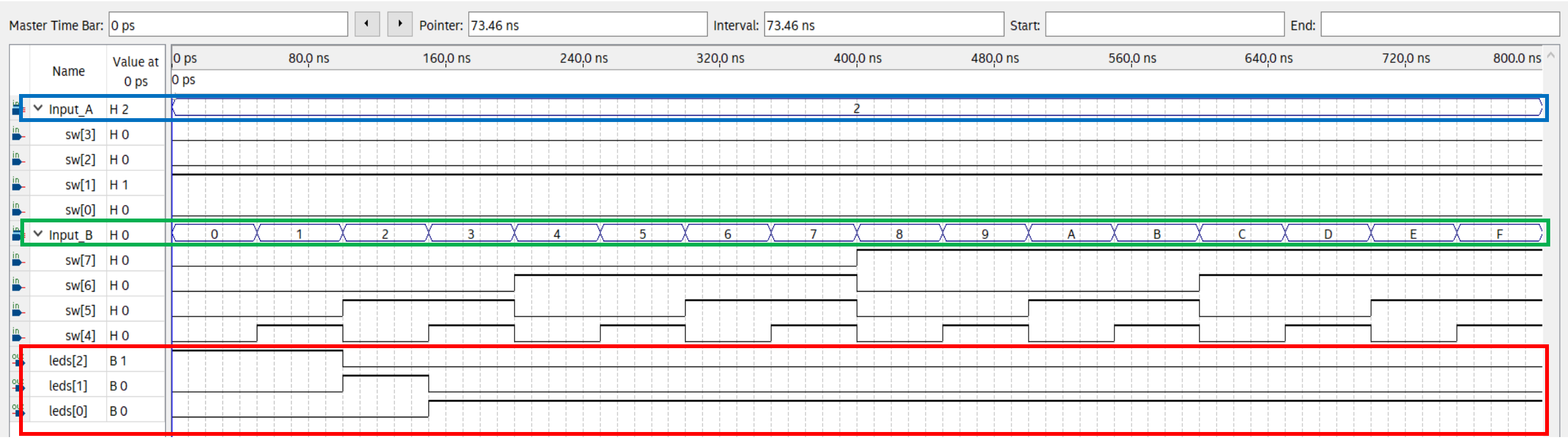
```vhdl
--Author: Group 3, Nandita Lohani, Puneet Bhullar
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
----------------------------------------------------------------------------
entity Compx4 is
    port (
            A, B : in std_logic_vector(3 downto 0);
            AGTB_out, AEQB_out, ALTB_out : out std_logic
        );
end Compx4;
                                                                *
-- ******************************************************************************
architecture Four_Bit_Comparator of Compx4 is

component Compx1
    port (
          A, B : in std_logic;
          AgreatB, AequalB, AlessB : out std_logic
          );
end component;

-------------------------------------------------------------------
signal AGTB, AEQB, ALTB : std_logic_vector (3 downto 0);
begin

-- Outputs expressed as boolean equations of operand inputs

AGTB_out <= ((AGTB(3)) or
              (AEQB(3) and AGTB(2)) or
              (AEQB(3) and AEQB(2) and AGTB(1)) or
              (AEQB(3) and AEQB(2) and AEQB(1) and AGTB(0))
          );

AEQB_out <= (AEQB(3) and AEQB(2) and AEQB(1) and AEQB(0));


ALTB_out <= ((ALTB(3)) or
              (AEQB(3) and ALTB(2)) or
              (AEQB(3) and AEQB(2) and ALTB(1)) or
              (AEQB(3) and AEQB(2) and AEQB(1) and ALTB(0))
          );

------instantiation of four single bit comparators----------------------------

inst1: Compx1 port map (A(3), B(3), AGTB(3), AEQB(3), ALTB(3));
inst2: Compx1 port map (A(2), B(2), AGTB(2), AEQB(2), ALTB(2));
inst3: Compx1 port map (A(1), B(1), AGTB(1), AEQB(1), ALTB(1));
inst4: Compx1 port map (A(0), B(0), AGTB(0), AEQB(0), ALTB(0));

end architecture Four_Bit_Comparator;
-------------------------------------------------------------------
```

4-Bit Magnitude Comparator Simulation 1



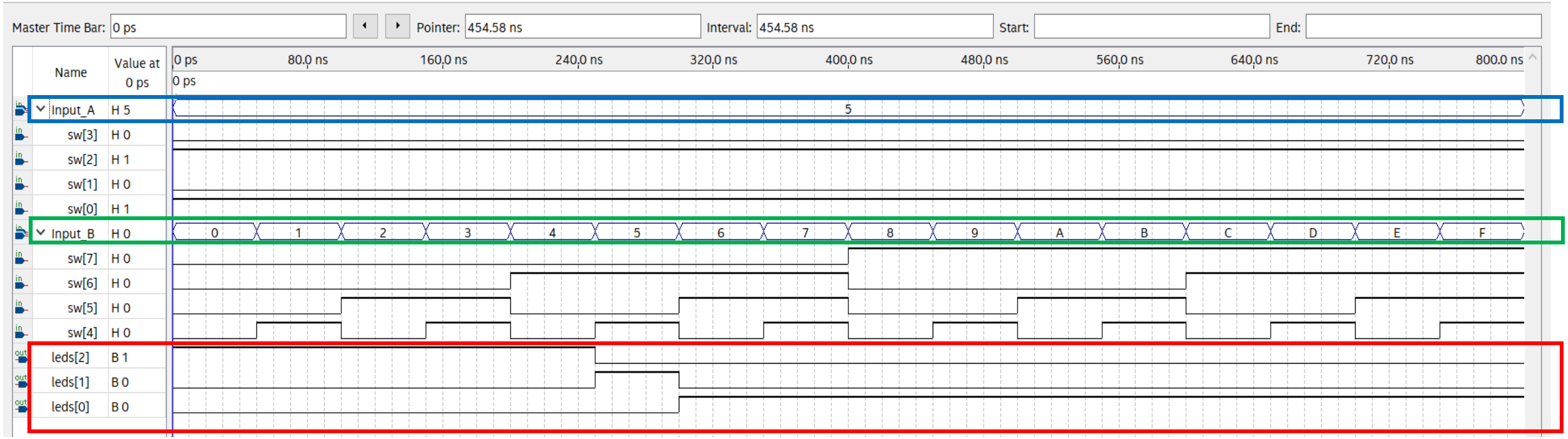
Each output is activated at some point during the simulation with correct functionality.

4-Bit Magnitude Comparator Simulation 2



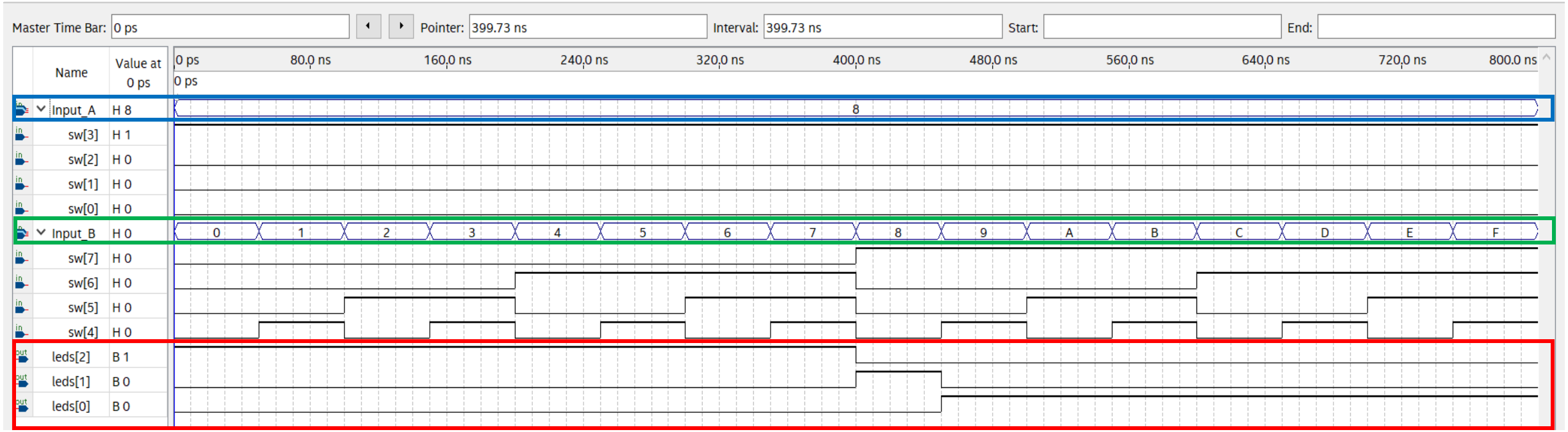
Each output is activated at some point during the simulation with correct functionality.

4-Bit Magnitude Comparator Simulation 3



Each output is activated at some point during the simulation with correct functionality.

4-Bit Magnitude Comparator Simulation 4



Each output is activated at some point during the simulation with correct functionality.

MUX File

```
1  --Author: Group 3, Nandita Lohani, Puneet Bhullar
2  library ieee;
3  use ieee.std_logic_1164.all;
4  library work;
5  ----------------------------------------------------------------------------
6  entity MUX is
7     port (
8           desired_temp, vacation_temp          : in std_logic_vector(3 downto 0);
9           vacation_mode                        : in std_logic;
10          mux_temp                             : out std_logic_vector(3 downto 0)
11          );
12
13 end entity MUX;
14 ----------------------------------------------------------------------------
15
16 architecture mux_logic of MUX is
17
18 begin
19
20 --for the multiplexing of two mux input busses
21 with vacation_mode select
22 mux_temp <= desired_temp when '0',
23             vacation_temp when '1';
24
25 end architecture mux_logic;
26
27
```

SevenSegment File

```
1  --Author: Group 3, Nandita Lohani, Puneet Bhullar
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.numeric_std.all;
5
6  ----------------------------------------------------------------------------
7
8  entity SevenSegment is port (
9
10    hex      :  in  std_logic_vector(3 downto 0);   -- The 4 bit data to be displayed
11
12    sevenseg :  out std_logic_vector(6 downto 0)    -- 7-bit outputs to a 7-segment
13 );
14 end SevenSegment;
15
16 architecture decode_function1 of SevenSegment is
17
18 --
19 -- The following statements convert a 4-bit input, called dataIn to a pattern of 7 bits.
20 -- The segment turns on when it is '1' otherwise '0'.
21 --
22 begin
23    with hex select            --GFEDCBA        3210      -- data in
24    sevenseg(6 downto 0);     <="0111111" when "0000",    -- [0]
25                                "0000110" when "0001",    -- [1]
26                                "1011011" when "0010",    -- [2]      +---- a -----+
27                                "1001111" when "0011",    -- [3]      |            |
28                                "1100110" when "0100",    -- [4]      |            |
29                                "1101101" when "0101",    -- [5]      f            b
30                                "1111101" when "0110",    -- [6]      |            |
31                                "0000111" when "0111",    -- [7]      |            |
32                                "1111111" when "1000",    -- [8]      +---- g -----+
33                                "1101111" when "1001",    -- [9]      |            |
34                                "1110111" when "1010",    -- [A]      |            |
35                                "1111100" when "1011",    -- [b]      e            c
36                                "1011000" when "1100",    -- [c]      |            |
37                                "1011110" when "1101",    -- [d]      |            |
38                                "1111001" when "1110",    -- [E]      +---- d -----+
39                                "1110001" when "1111",    -- [F]
40                                "0000000" when others;    -- [ ]
41
42 end architecture decode_function1;
43 ----------------------------------------------------------------------------
44
```

Segment7_mux File

```
38  --Author: Group 3, Nandita Lohani, Puneet Bhullar
39  library IEEE;
40  use IEEE.std_logic_1164.all;
41  use IEEE.numeric_std.all;
42
43  --   Entity
44  -- ----------------------------------------------------------------------------
45
46  entity segment7_mux is
47      port (
48              clk         : in  std_logic := '0';
49              DIN2        : in  std_logic_vector(6 downto 0);
50              DIN1        : in  std_logic_vector(6 downto 0);
51              DOUT        : out std_logic_vector(6 downto 0);
52              DIG2        : out std_logic;
53              DIG1        : out std_logic
54          );
55  end entity segment7_mux;
56
57  -- ----------------------------------------------------------------------------
58  architecture syn of segment7_mux is
59
60      signal toggle        : std_logic;
61      signal DOUT_TEMP     : std_logic_vector(6 downto 0);
62
63  begin
64      ------------------------------------------
65      -- Register File
66      ------------------------------------------
67      clk_proc:process(CLK)
68      variable COUNT       :unsigned(10 downto 0) := "00000000000";
69      begin
70          if (rising_edge(CLK)) then
71              COUNT := COUNT + 1;
72          else
73              COUNT := COUNT;
74          end if;
75      toggle <= COUNT(10);
76      end process clk_proc;
77      DIG1 <= NOT toggle;
78      DIG2 <= toggle;
79
80      DOUT_TEMP(0) <= (DIN2(0)) WHEN (toggle = '1')   ELSE (DIN1(0));
81      DOUT_TEMP(1) <= (DIN2(1)) WHEN (toggle = '1')   ELSE (DIN1(1));
82      DOUT_TEMP(2) <= (DIN2(2)) WHEN (toggle = '1')   ELSE (DIN1(2));
83      DOUT_TEMP(3) <= (DIN2(3)) WHEN (toggle = '1')   ELSE (DIN1(3));
84      DOUT_TEMP(4) <= (DIN2(4)) WHEN (toggle = '1')   ELSE (DIN1(4));
85      DOUT_TEMP(5) <= (DIN2(5)) WHEN (toggle = '1')   ELSE (DIN1(5));
86      DOUT_TEMP(6) <= (DIN2(6)) WHEN (toggle = '1')   ELSE (DIN1(6));
87  -- DOUT_TEMP(7) <= (DIN2(7)) WHEN (toggle = '1')   ELSE (DIN1(7));
```

Segment7_mux File Cont.

```
89      DOUT(0) <= '0' WHEN (DOUT_TEMP(0) = '0')  ELSE '1';
90      DOUT(1) <= '0' WHEN (DOUT_TEMP(1) = '0')  ELSE 'Z'; --open drain
91      DOUT(2) <= '0' WHEN (DOUT_TEMP(2) = '0')  ELSE '1';
92      DOUT(3) <= '0' WHEN (DOUT_TEMP(3) = '0')  ELSE '1';
93      DOUT(4) <= '0' WHEN (DOUT_TEMP(4) = '0')  ELSE '1';
94      DOUT(5) <= '0' WHEN (DOUT_TEMP(5) = '0')  ELSE 'Z'; --open drain
95      DOUT(6) <= '0' WHEN (DOUT_TEMP(6) = '0')  ELSE 'Z'; --open drain
96  -- DOUT(7) <= '0' WHEN (DOUT_TEMP(7) = '0')  ELSE '1';
97
98  end architecture syn;
```

Inverter File

```
1  --Author: Group 3, Nandita Lohani, Puneet Bhullar
2  library ieee;
3  use ieee.std_logic_1164.all;
4  library work;
5  ----------------------------------------------------------------------------
6  entity Inverter is
7     port (
8           pb_n0, pb_n1, pb_n2, pb_n3                    : in std_logic;
9           pb_n0out, pb_n1out, pb_n2out, pb_n3out        : out std_logic
10          );
11
12 end entity Inverter;
13 ----------------------------------------------------------------------------
14 architecture inverter_logic of Inverter is
15
16 begin
17
18 --- Ouputs with correct boolean equations using input operands
19
20 pb_n0out <= not pb_n0;
21 pb_n1out <= not pb_n1;
22 pb_n2out <= not pb_n2;
23 pb_n3out <= not pb_n3;
24
25 end architecture inverter_logic;
26
27
```

HVAC File

```
1  --Author: Group 3, Nandita Lohani, Puneet Bhullar
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.numeric_std.all;
5  --*************************************************************************
6  -- clk input is the LogicalStep 50MHz clock input
7  entity HVAC is
8
9     port
10    (
11       HVAC_SIM             : in boolean;
12       clk                  : in std_logic;
13       run_n                : in std_logic;
14       increase, decrease   : in std_logic;
15       temp                 : out std_logic_vector (3 downto 0)
16    );
17
18 end entity;
19 --*************************************************************************
20 architecture rtl of HVAC is
21
22 signal clk_2hz          : std_logic;
23 signal HVAC_clock       : std_logic;
24 signal digital_counter  : std_logic_vector(23 downto 0);
25
26 --*************************************************************************
27 begin
28
29 -- clk_divider process generates a 2Hz Clck from the 50 Mhz clk
30
31 clk_divider: process (clk)
32    variable   counter      :  unsigned(23 downto 0);
33
34    begin
35 -- Synchronously update counter
36       if (rising_edge(clk))  then
37             counter :=  counter + 1;
38       end if;
39       digital_counter <= std_logic_vector(counter);
40
41    end process;
42
```

HVAC File Cont.

```
42
43  clk_2hz <= digital_counter(23);
44
45  clk_mux: process (HVAC_SIM)
46      begin
47          if (HVAC_SIM)  then
48                  HVAC_clock<=  clk;
49          else
50                  HVAC_clock<=  clk_2hz;
51          end if;
52      end process;
53
54
55  counter: process (HVAC_clock)
56      variable   cnt    :  unsigned(3 downto 0) := "0111";
57      begin
58
59      -- Synchronously update counter
60          if ((rising_edge(HVAC_clock)) AND (run_n ='0')) then
61
62              if ((increase = '1') AND (cnt /= 15))then
63                  -- Increment only if not at maxvalue
64                  cnt := cnt + 1;
65              elsif ((decrease = '1') AND (cnt /= 0)) then
66                  -- Decrement only if not at minvalue
67                  cnt := cnt - 1;
68              end if;
69
70          end if;
71
72      -- Output the current count
73          temp <= std_logic_vector(cnt);
74      end process;
75
76  end rtl;
```

TESTER File

```
1  --Author: Group 3, Nandita Lohani, Puneet Bhullar
2  library ieee;
3  use ieee.std_logic_1164.all;
4  ----------------------------------------------------------------------
5
6  entity Tester is port (
7     MC_TESTMODE          : in  std_logic;
8     I1EQI2,I1GTI2,I1LTI2 : in  std_logic;
9     input1               : in  std_logic_vector(3 downto 0);
10    input2               : in  std_logic_vector(3 downto 0);
11    TEST_PASS            : out std_logic
12    );
13 end Tester;
14 ----------------------------------------------------------------------
15
16 architecture Test_ckt of Tester is
17
18 signal EQ_PASS, GT_PASS, LT_PASS : std_logic;
19
20 begin
21 Tester1:
22 PROCESS (MC_TESTMODE, input1, input2, I1EQI2, I1GTI2, I1LTI2) is
23
24 begin
25
26       IF ((input1 = input2) AND (I1EQI2 = '1')) THEN
27       EQ_PASS <= '1';
28       GT_PASS <= '0';
29       LT_PASS <= '0';
30
31       ELSIF ((input1 > input2) AND (I1GTI2 = '1')) THEN
32       GT_PASS <= '1';
33       EQ_PASS <= '0';
34       LT_PASS <= '0';
35
36       ELSIF ((input1 < input2) AND (I1LTI2 = '1')) THEN
37       LT_PASS <= '1';
38       EQ_PASS <= '0';
39       GT_PASS <= '0';
40
41       ELSE
42
43       EQ_PASS <= '0';
44       GT_PASS <= '0';
45       LT_PASS <= '0';
46
47       END IF;
48       TEST_PASS <= MC_TESTMODE AND (EQ_PASS OR GT_PASS OR LT_PASS);
49
50 end process;
51
52 end;
```

Energy Monitor Control File

```
1  --Author: Group 3, Nandita Lohani, Puneet Bhullar
2  library ieee;
3  use ieee.std_logic_1164.all;
4  use ieee.numeric_std.all;
5
6  ----------------------------------------------------------------------------
7  entity Energy_Monitor is
8     port (
9           AGTB, AEQB, ALTB, vacation_mode, MC_test, window_open, door_open : in std_logic;
10          furnace, at_temp, AC, blower, window, door, vacation: out std_logic;
11          run_n, increase, decrease   : out std_logic
12         );
13 end Energy_Monitor;
14
15 -- ****************************************************************************
16 -- *  Architecture                                                            *
17 -- ****************************************************************************
18
19 architecture Energy_Monitoring_Controller of Energy_Monitor is
20
21 signal activate: std_logic;
22
23 ------------------------------------------------------------------------------------------------
24 begin
25
26 energymonitor1:
27
28    process (AGTB, AEQB, ALTB, vacation_mode, MC_test, window_open, door_open) is
29
30    begin
31       --- Ouputs expressed as correct boolean expressions with input operands and signals
32
33       furnace <= ((AGTB) and (not MC_test) and (not window_open) and (not door_open));
34       at_temp <= AEQB;
35       AC <= ((ALTB) and (not MC_test) and (not window_open) and (not door_open));
36       blower <= ((AGTB or ALTB) and (not MC_test) and (not window_open) and (not door_open));
37       window <= window_open;
38       door <= door_open;
39       vacation <= vacation_mode;
40
41       decrease <= ALTB;
42       increase <= AGTB;
43       run_n <= ((AEQB) or (window_open) or (door_open) or (MC_test));
44
45 end process;
46
47 end architecture Energy_Monitoring_Controller;
48 ----------------------------------------------------------------------
```

Logical Step Top

```
1  --Author: Group 3, Nandita Lohani, Puneet Bhullar
2  library ieee;
3  use ieee.std_logic_1164.all;
4  ---------------------------------------------------------------------------------
5  entity LogicalStep_Lab3_top is port (
6     clkin_50    : in  std_logic;
7     pb_n        : in  std_logic_vector(3 downto 0);
8     sw          : in  std_logic_vector(7 downto 0);
9
10    ------------------------------------------------------
11    --HVAC_temp : out std_logic_vector(3 downto 0); -- used for simulations only. Comment out for FPGA download compiles.
12    ------------------------------------------------------
13
14    leds        : out std_logic_vector(7 downto 0);
15    seg7_data   : out std_logic_vector(6 downto 0); -- 7-bit outputs to a 7-segment
16    seg7_char1  : out std_logic;                    -- seg7 digit1 selector
17    seg7_char2  : out std_logic                     -- seg7 digit2 selector
18
19 );
20 end LogicalStep_Lab3_top;
21
22 architecture design of LogicalStep_Lab3_top is
23 --
24 -- Provided Project Components Used
25 ------------------------------------------------------------------------
26 component SevenSegment  port (
27    hex      :  in  std_logic_vector(3 downto 0);   -- The 4 bit data to be displayed
28    sevenseg :  out std_logic_vector(6 downto 0)    -- 7-bit outputs to a 7-segment
29 );
30 end component SevenSegment;
31
32 component segment7_mux port (
33           clk        : in  std_logic := '0';
34           DIN2       : in  std_logic_vector(6 downto 0);
35           DIN1       : in  std_logic_vector(6 downto 0);
36           DOUT       : out std_logic_vector(6 downto 0);
37           DIG2       : out std_logic;
38           DIG1       : out std_logic
39        );
40 end component segment7_mux;
41 ------------------------------------------------------------------------
42 component Tester port (
43       MC_TESTMODE          : in  std_logic;
44       I1EQI2,I1GTI2,I1LTI2 : in  std_logic;
45       input1               : in  std_logic_vector(3 downto 0);
46       input2               : in  std_logic_vector(3 downto 0);
47       TEST_PASS            : out std_logic
48       );
49    end component;
```

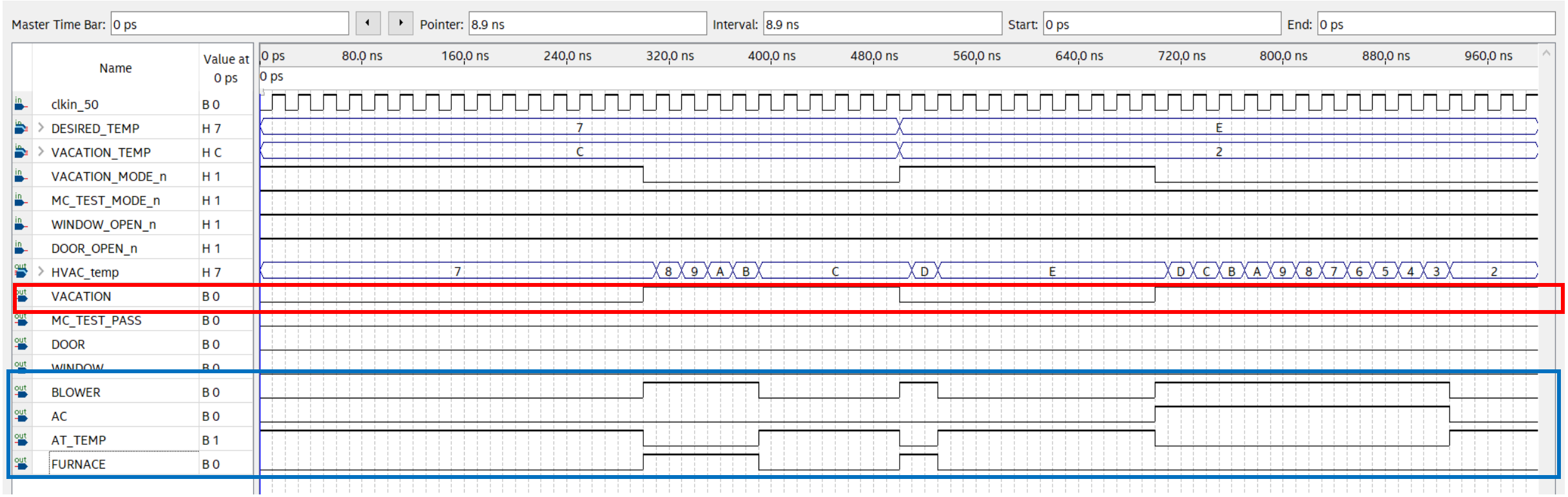
Logical Step Top Cont.

```
50  ------------------------------------------------------------------
51  component HVAC    port (
52        HVAC_SIM             : in boolean;
53        clk                  : in std_logic;
54        run_n                : in std_logic;
55        increase, decrease   : in std_logic;
56        temp                 : out std_logic_vector (3 downto 0)
57        );
58     end component;
59  ------------------------------------------------------------------
60  component Compx4 port (
61           A, B : in std_logic_vector(3 downto 0);
62           AGTB_out, AEQB_out, ALTB_out : out std_logic
63           );
64  end component;
65  ------------------------------------------------------------------------------
66  component MUX port (
67           vacation_mode                        : in std_logic;
68           desired_temp, vacation_temp          : in std_logic_vector(3 downto 0);
69           mux_temp                             : out std_logic_vector(3 downto 0)
70           );
71
72  end component;
73  ------------------------------------------------------------------------------
74  component Inverter port (
75           pb_n0, pb_n1, pb_n2, pb_n3                          : in std_logic;
76           pb_n0out, pb_n1out, pb_n2out, pb_n3out              : out std_logic
77           );
78
79  end component;
80  ------------------------------------------------------------------------------------------
81  component Energy_Monitor port (
82           AGTB, AEQB, ALTB, vacation_mode, MC_test, window_open, door_open : in std_logic;
83           furnace, at_temp, AC, blower, window, door, vacation: out std_logic;
84           run_n, increase, decrease    : out std_logic
85          );
86  end component;
87
88  ------------------------------------------------------------------
89  constant HVAC_SIM : boolean := FALSE; -- set to FALSE when compiling for FPGA download to LogicalStep board
90  ------------------------------------------------------------------
```

Logical Step Top Cont.

```
 91
 92  -- global clock
 93  signal clk_in                       : std_logic;
 94  signal mux_temp, current_temp       : std_logic_vector(3 downto 0);
 95  signal mt_7seg, ct_7seg             : std_logic_vector(6 downto 0);
 96
 97
 98  --additional signals-----------------------------------------------------------------------
 99  signal desired_temp                                     : std_logic_vector (3 downto 0);
100  signal vacation_temp                                    : std_logic_vector (3 downto 0);
101  signal vacation_mode, MC_test_mode_signal, window_open, door_open : std_logic;
102  signal AGTB_signal, AEQB_signal, ALTB_signal            : std_logic;
103  signal MC_test_pass                                     : std_logic;
104  signal decrease, increase, run_n                        : std_logic;
105
106  ---------------------------------------------------------------------
107  begin -- Here the circuit begins
108  clk_in <= clkin_50;  --hook up the clock input
109
110  -- temp inputs hook-up to internal busses.
111  desired_temp <= sw(3 downto 0);
112  vacation_temp <= sw(7 downto 4);
113
114  --------------------------------------------------------------------------
115  --instantiation of all components with the appropriate singls and outputs
116  --------------------------------------------------------------------------
117  inst1: sevensegment port map (mux_temp, mt_7seg);
118  inst2: sevensegment port map (current_temp, ct_7seg);
119
120  inst3: segment7_mux port map (clk_in, mt_7seg, ct_7seg, seg7_data, seg7_char2, seg7_char1);
121  inst4: Tester port map(MC_test_mode_signal, AEQB_signal, AGTB_signal, ALTB_signal, desired_temp, current_temp,leds(6));
122
123
124  inst5: HVAC port map (HVAC_SIM, clk_in, run_n, increase, decrease, current_temp );
125  inst6: Compx4 port map (mux_temp, current_temp, AGTB_signal, AEQB_signal, ALTB_signal);
126
127  inst7: MUX port map (vacation_mode, desired_temp, vacation_temp, mux_temp);
128
129  inst8: Inverter port map(pb_n(0), pb_n(1), pb_n(2), pb_n(3), door_open, window_open, MC_test_mode_signal, vacation_mode);
130
131  inst9: Energy_Monitor port map (AGTB_signal, AEQB_signal, ALTB_signal, vacation_mode, MC_test_mode_signal, window_open, door_open,
132                                  leds(0), leds(1),leds(2),leds(3),leds(4),leds(5),leds(7), run_n, increase, decrease);
133  --
134  ---------------------------------------------------------------------
135  --HVAC_temp <= current_temp;  -- used for simulations only. Comment out for FPGA download compiles.
136  ---------------------------------------------------------------------
137  end design;
```

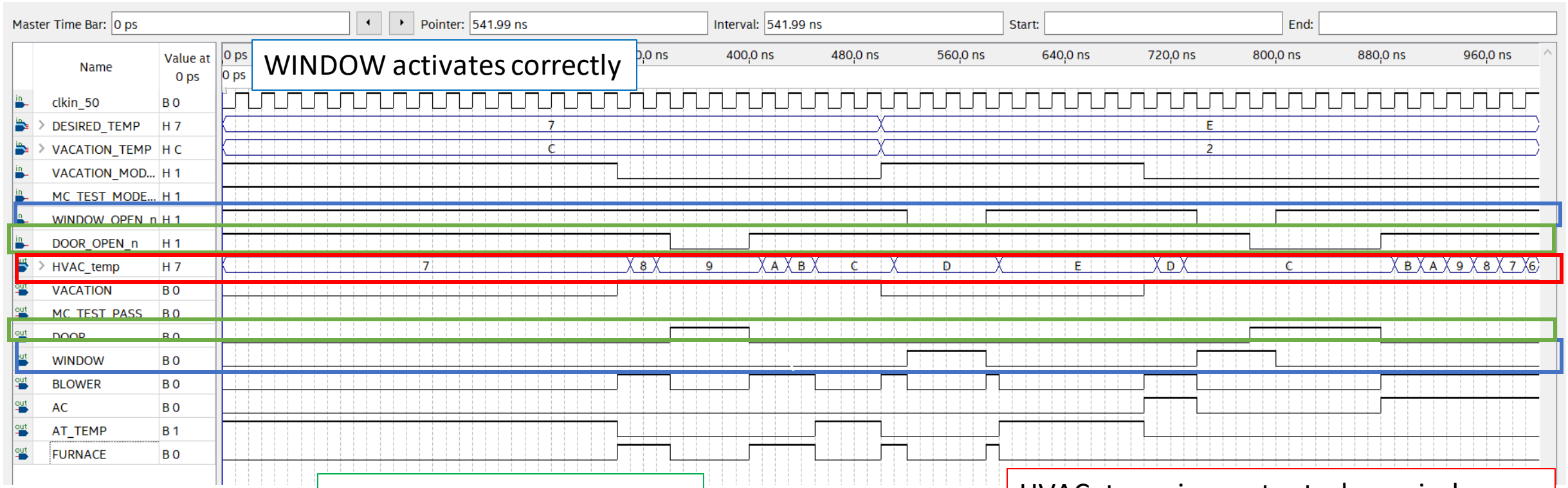
Energy Monitor Function Simulation 1



HVAC_temp changes to match the mux temp during vacation AND normal mode

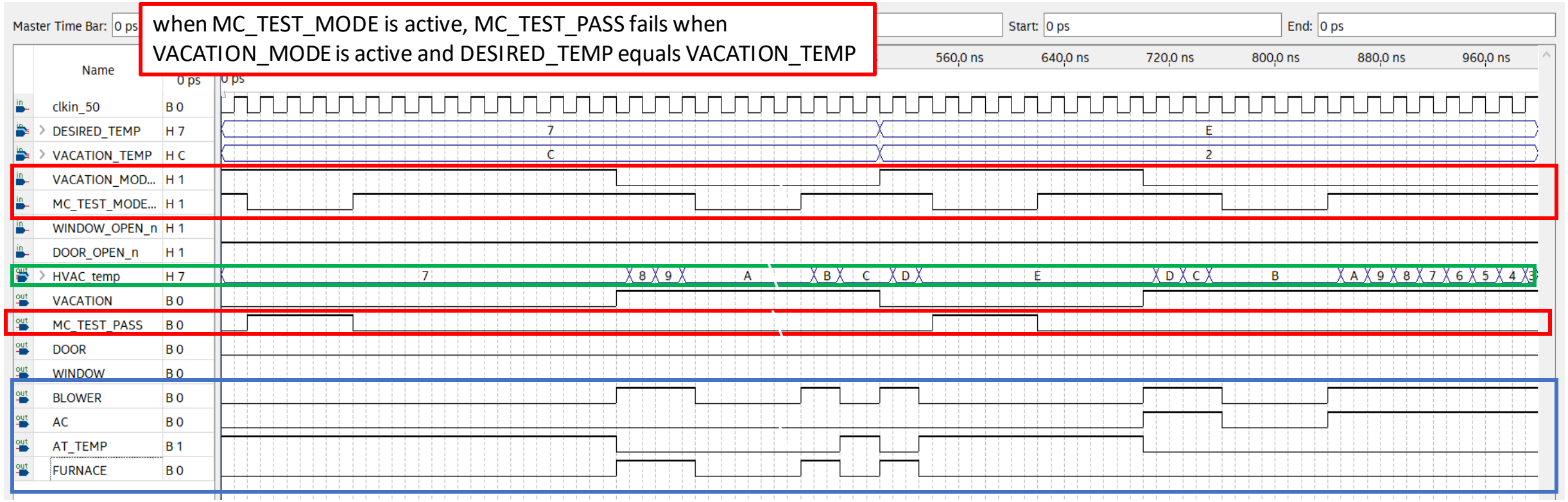
furnace, AC, at_temp, blower function correctly in accordance with mux temp and current temp

Energy Monitor Function Simulation 2



Energy Monitor Function Simulation 3

when MC_TEST_MODE is active, MC_TEST_PASS fails when VACATION_MODE is active and DESIRED_TEMP equals VACATION_TEMP



when MC_TEST_MODE is active, one of AC, AT_TEMP, FURNACE goes on during normal mode AND BLOWER is off

HVAC_TEMP stops changing when MC_TEST_MODE is active