Lab 2 Report

ECE 124

Group 3 Session 201

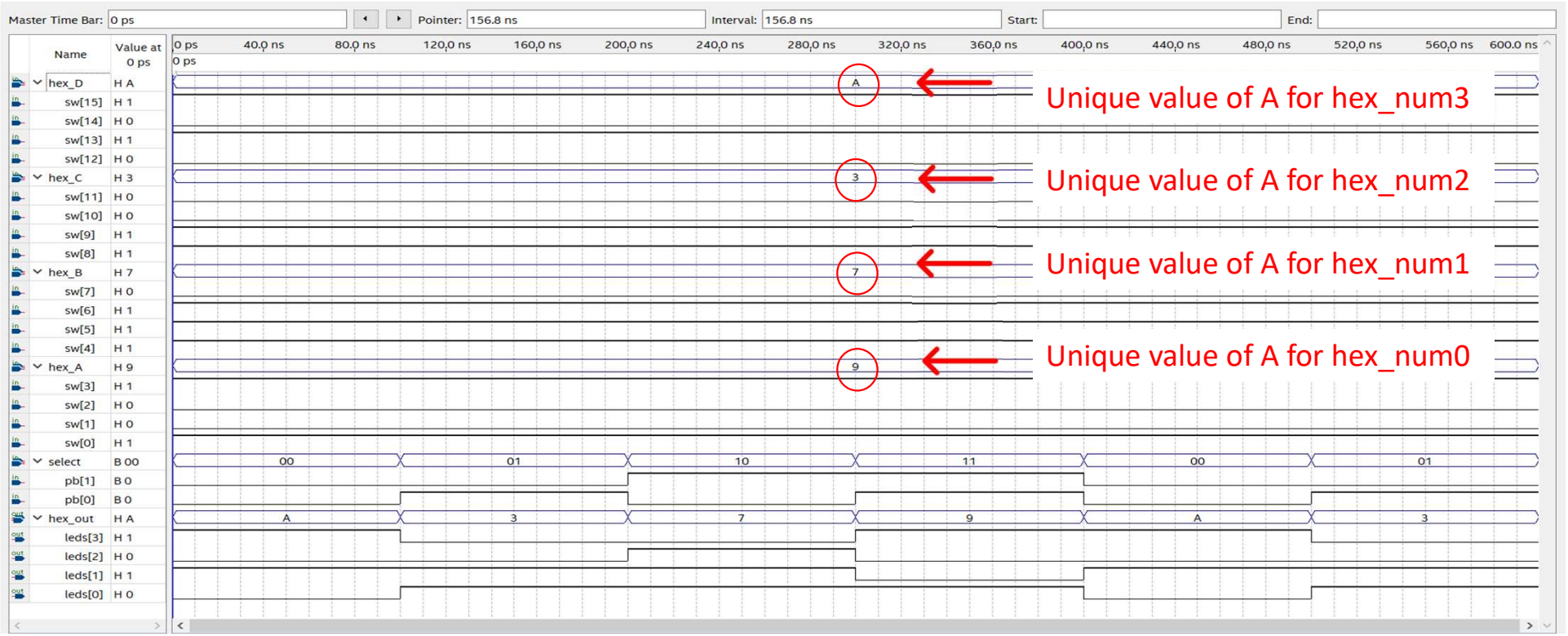
Nandita Lohani

Puneet Bhullar

Hex Multiplexer File

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

entity hex_mux is
   port (
         hex_num3, hex_num2, hex_num1, hex_num0 : in std_logic_vector(3 downto 0);
         mux_select                             : in std_logic_vector(1 downto 0);
         hex_out                                : out std_logic_vector(3 downto 0)
         );

end entity hex_mux;

architecture mux_logic of hex_mux is


begin

--for the multiplexing of four hex input busses
with mux_select(1 downto 0) select
hex_out <= hex_num0 when "00",
           hex_num1 when "01",
           hex_num2 when "10",
           hex_num3 when "11";

end architecture mux_logic;

```

Hex_Mux Simulation



Single-Bit Full Adder File

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

library work;

entity full_adder_1bit is
    port (
          cin, bit_val1, bit_val2    : in std_logic;
          bit_sum                    : out std_logic;
          carry_out_bit              : out std_logic
          );
end full_adder_1bit;

architecture Circuit of full_adder_1bit is

--------- Signals ----------------------------------------------
-----------------------------------------------------------------

signal half_adder_sum, half_adder_carry   : std_logic;

-----------------------------------------------------------------
-----------------------------------------------------------------

begin

half_adder_carry     <= bit_val1 AND bit_val2;

half_adder_sum       <= bit_val1 XOR bit_val2;

bit_sum              <= half_adder_sum XOR cin;

carry_out_bit        <= (half_adder_sum AND cin) OR half_adder_carry;


end;

```

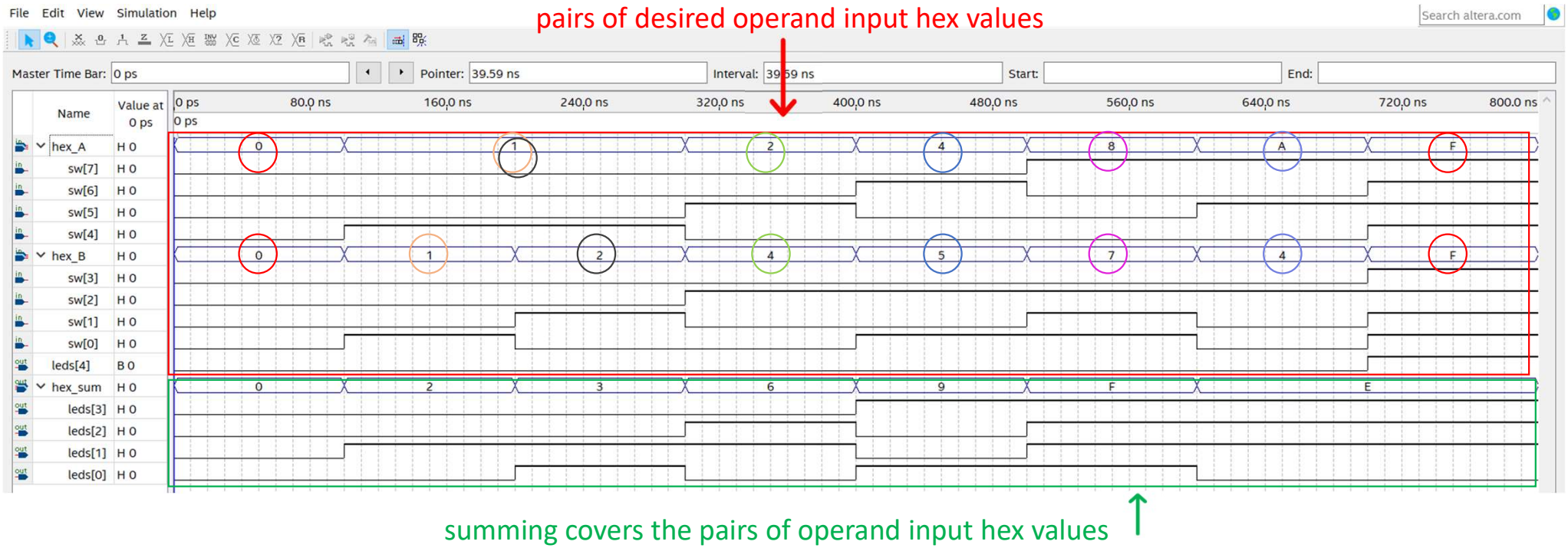
Four Bit Full Adder File

```
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  library work;
7
8  entity full_adder_4bit is
9     port (
10          cin                     : in std_logic;
11          hex_val_A, hex_val_B    : in std_logic_vector(3 downto 0);
12          hex_sum                 : out std_logic_vector(3 downto 0);
13          carry_out               : out std_logic
14          );
15 end full_adder_4bit;
16
17 architecture Circuit of full_adder_4bit is
18
19 ------------------Components -------------------------------------
20 ------------------------------------------------------------------
21
22 component full_adder_1bit
23    port (
24          cin, bit_val1, bit_val2    : in std_logic;
25          bit_sum                    : out std_logic;
26          carry_out_bit              : out std_logic
27          );
28 end component;
29
30 ------------------------------------------------------------------
31
32 ---------------------- Signals -----------------------------------
33 ------------------------------------------------------------------
34 -- group of 4 logic signals with the group type defined as std_logic_vector(MSB downto LSB)
35    signal cout : std_logic_vector(3 downto 0);
36
37 ------------------------------------------------------------------
```

Four Bit Full Adder File Cont.

```
38  ----------------------------------------------------------------------
39  ------- Instances for the Full_Adder_4bit design ------------------
40  begin
41
42  adder0: full_adder_1bit port map (cin, hex_val_A(0), hex_val_B(0),hex_sum(0), cout(0));
43
44  adder1: full_adder_1bit port map (cout(0), hex_val_A(1), hex_val_B(1), hex_sum(1), cout(1));
45
46  adder2: full_adder_1bit port map (cout(1), hex_val_A(2), hex_val_B(2), hex_sum(2), cout(2));
47
48  adder3: full_adder_1bit port map (cout(2), hex_val_A(3), hex_val_B(3), hex_sum(3), cout(3));
49
50  carry_out <= cout(3);
51
52  end circuit;
53
```

Four-Bit Full Adder Simulation

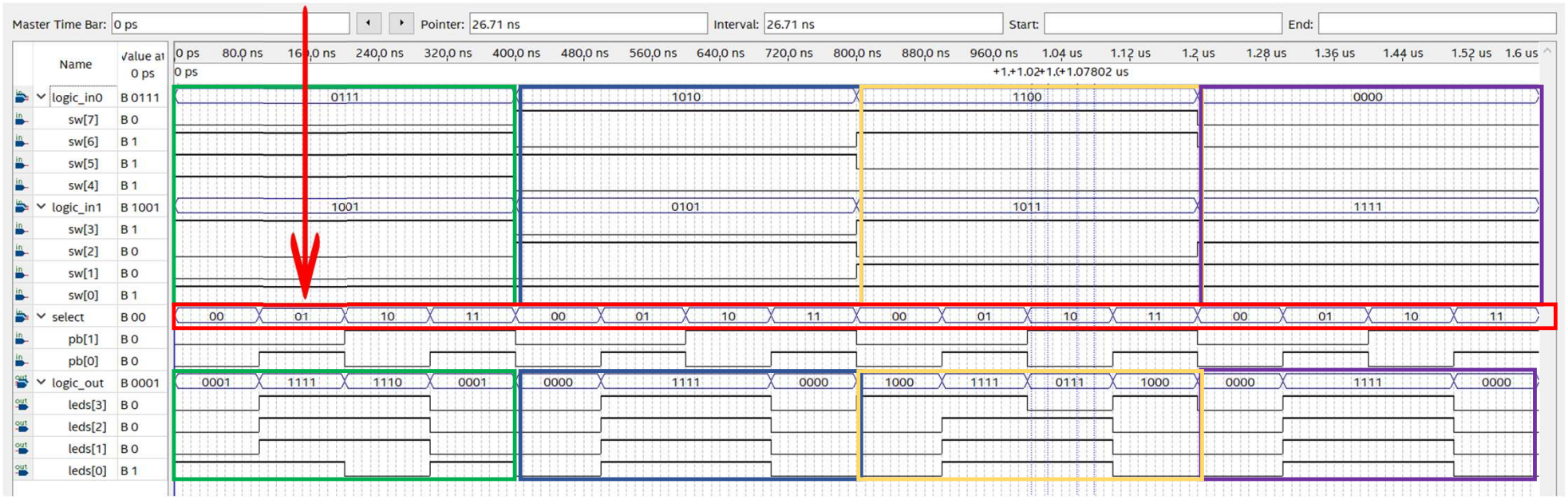


Logical Processor File

```
1  --Author: Group 3, Nandita Lohani, Puneet Bhullar
2  library ieee;
3  use ieee.std_logic_1164.all;
4  library work;
5
6  entity logic_proc is
7     port (
8           logic_in0,logic_in1           : in std_logic_vector(3 downto 0);
9           logic_select                  : in std_logic_vector(1 downto 0);
10          logic_out                     : out std_logic_vector(3 downto 0)
11          );
12
13 end entity logic_proc;
14
15 architecture logic_proc_logic of logic_proc is
16
17
18 begin
19
20 --for the multiplexing of two logic input busses
21 with logic_select(1 downto 0) select
22 logic_out <= (logic_in0 AND logic_in1)   when "00",
23               (logic_in0 OR logic_in1)    when "01",
24               (logic_in0 XOR logic_in1)   when "10",
25               (logic_in0 XNOR logic_in1)  when "11";
26
27 end architecture logic_proc_logic;
28
29
```

Logical Processor Simulation

Logic Selector Inputs



Leds results match the outputs via the logic selector inputs for the four unique operand values

Mux_Out 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_out is
7      port (
8            mux_num0, mux_num1            : in std_logic_vector(4 downto 0);
9            muxout_select                 : in std_logic;
10           mux_out                       : out std_logic_vector(4 downto 0)
11           );
12
13  end entity mux_out;
14
15  architecture muxout_logic of mux_out is
16
17  begin
18
19  --for the multiplexing of two mux input busses
20  with muxout_select select
21  mux_out <= mux_num0 when '0',
22             mux_num1 when '1';
23
24  end architecture muxout_logic;
25
26
```

Logical Step Top

```
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   library work;
6
7   entity LogicalStep_Lab2_top is port (
8      pb             : in  std_logic_vector(6 downto 0);     -- push buttons used for data input selection/operation control
9      sw             : in  std_logic_vector(15 downto 0);    -- The switch inputs used for data inputs
10     leds           : out std_logic_vector(5 downto 0)      --  leds for outputs
11
12  );
13  end LogicalStep_Lab2_top;
14
15  architecture Circuit of LogicalStep_Lab2_top is
16
17  -- Components to be Used ---
18  ------------------------------------------------------------------
19  component hex_mux
20     port (
21           hex_num3, hex_num2, hex_num1, hex_num0 : in std_logic_vector(3 downto 0);
22           mux_select                             : in std_logic_vector(1 downto 0);
23           hex_out                                : out std_logic_vector(3 downto 0)
24           );
25
26  end component;
27
28  component logic_proc
29     port (
30           logic_in0,logic_in1           : in std_logic_vector(3 downto 0);
31           logic_select                  : in std_logic_vector(1 downto 0);
32           logic_out                     : out std_logic_vector(3 downto 0)
33           );
34
35  end component;
36
```

Logical Step Top Cont.

```
36  
37  component full_adder_4bit
38      port (
39            cin                   : in std_logic;
40            hex_val_A, hex_val_B  : in std_logic_vector(3 downto 0);
41            hex_sum               : out std_logic_vector(3 downto 0);
42            carry_out             : out std_logic
43            );
44  
45  end component;
46  
47  
48  component mux_out
49      port (
50            mux_num0, mux_num1            : in std_logic_vector(4 downto 0);
51            muxout_select                 : in std_logic;
52            mux_out                       : out std_logic_vector(4 downto 0)
53            );
54  
55  end component;
56  --
57  ------------ Signals ------------------------------------------------------------------------------------
58  ---------------------------------------------------------------------------------------------------------
59  -- groups of logic signals with each group defined as std_logic_vector(MSB downto LSB)
60      signal hex_A, hex_B, hex_C, hex_D                                       : std_logic_vector(3 downto 0);
61  
62  --- some selector nets;
63      signal hex_mux_sel0, hex_mux_sel1, logic_proc_sel                       : std_logic_vector(1 downto 0);
64      signal hex_mux_out0, hex_mux_out1, logic_proc_out, adder_bit_out        : std_logic_vector (3 downto 0);
65  
66  --- some concatenation nets
67      signal muxout_in0, muxout_in1                                           : std_logic_vector(4 downto 0);
68  
69      signal muxout_sel                                                       : std_logic;
70      signal carry_out                                                        : std_logic;
```

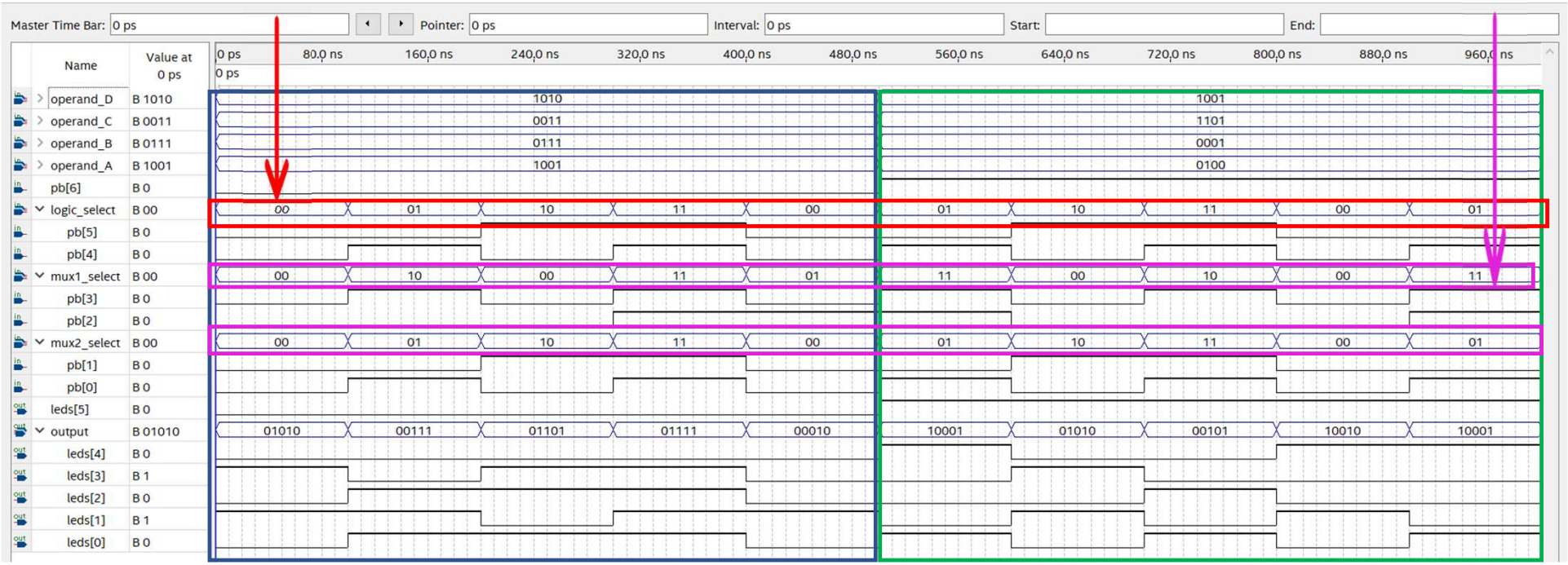
Logical Step Top Cont.

```
72  ----------------------------------------------------------------------------------------------------------------------
73  begin
74
75  -- assign (connect) one end of each input group (bus) to sepecific switch inputs
76  hex_A <= sw(3 downto 0);
77  hex_B <= sw(7 downto 4);
78  hex_C <= sw(11 downto 8);
79  hex_D <= sw(15 downto 12);
80
81
82  -- assign 4 of the pb inputs to drive mux selection port
83  hex_mux_sel0   <= pb(1 downto 0);
84  hex_mux_sel1   <= pb(3 downto 2);
85  logic_proc_sel <= pb(5 downto 4);
86  muxout_sel     <= pb(6);
87
88  --combine singles through concatenation for the mux_out multiplexer inputs
89  muxout_in0     <= '0' & logic_proc_out;
90  muxout_in1     <= carry_out & adder_bit_out;
91  ------------------------------------------------------------------------
92  ------------------------------------------------------------------------
93  --instances for hex_mux, logic processor, full adder 4 bit and mux_out component
94  inst0: hex_mux port map (hex_A, hex_B, hex_C, hex_D, hex_mux_sel0, hex_mux_out0);
95  inst1: hex_mux port map (hex_A, hex_B, hex_C, hex_D, hex_mux_sel1, hex_mux_out1);
96
97  inst2: logic_proc port map (hex_mux_out0, hex_mux_out1, logic_proc_sel, logic_proc_out);
98  inst3: full_adder_4bit port map ('0', hex_mux_out0, hex_mux_out1, adder_bit_out, carry_out);
99
100 inst4: mux_out port map (muxout_in0, muxout_in1, muxout_sel, leds(4 downto 0));
101 leds(5) <= pb(6);
102
103
104 end Circuit;
105
106
```

Arithmetic Logic Unit Simulation

logic_select chooses any operator

mux_select chooses any operator



Logic Processor result is selected when pb(6) and p(5) is on

ADDER result is selected when pb(6) and p(5) is on