Project 2 - MAC

Nathan Paternoster
Andrew O’Neil-Smith
Garrett Clausen

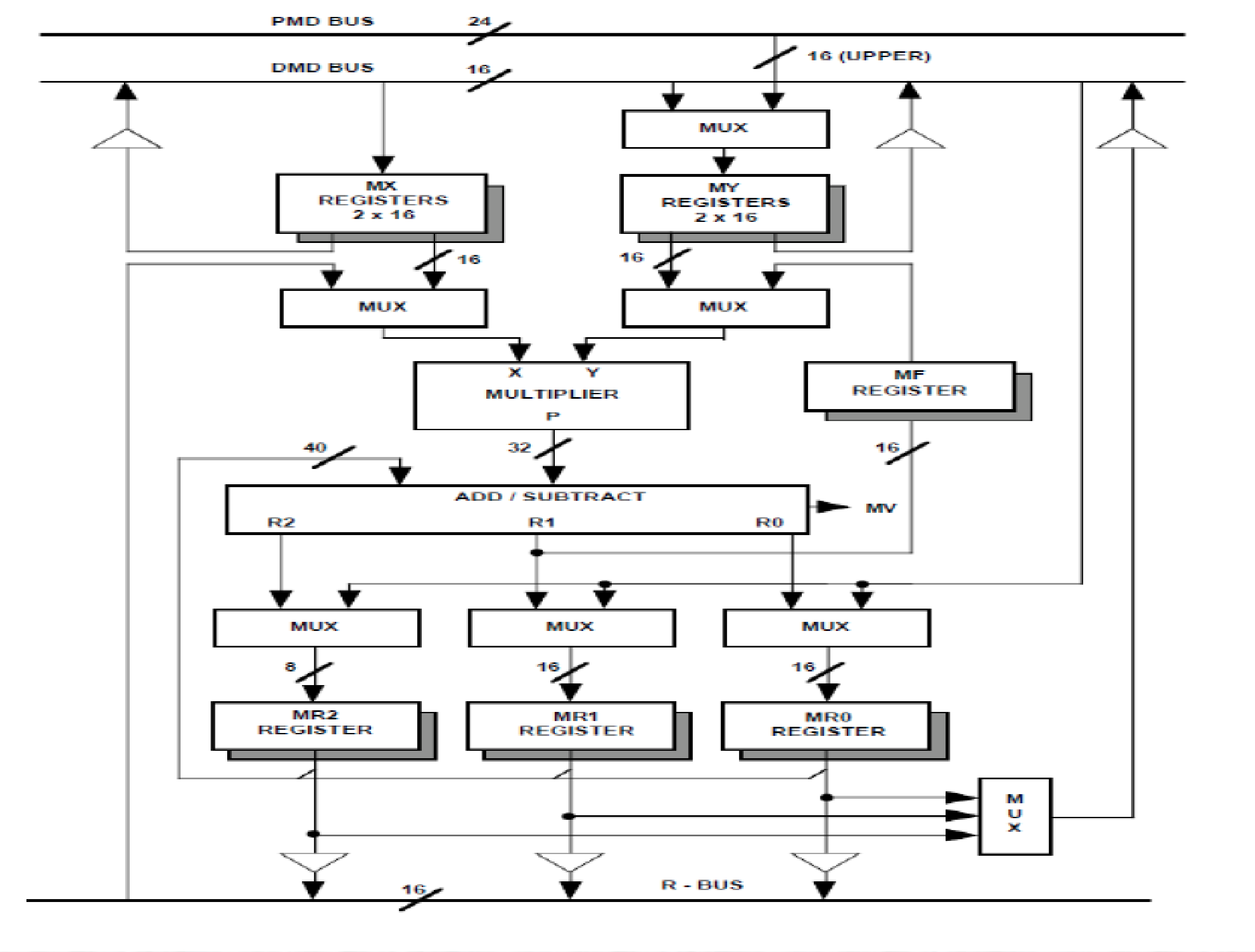
EEN 316
University of Miami
4/16/14

Background - MAC

- Multiplier-Accumulator
- Hardware unit that allows the summation of products
- Performs multiplication faster than the add-and-shift algorithm
- Works by calculating the product of two numbers and adding that to an accumulator
- Also can perform single multiplication and multiply-subtract

$$a \leftarrow a + (b \times c)$$

Logic Diagram

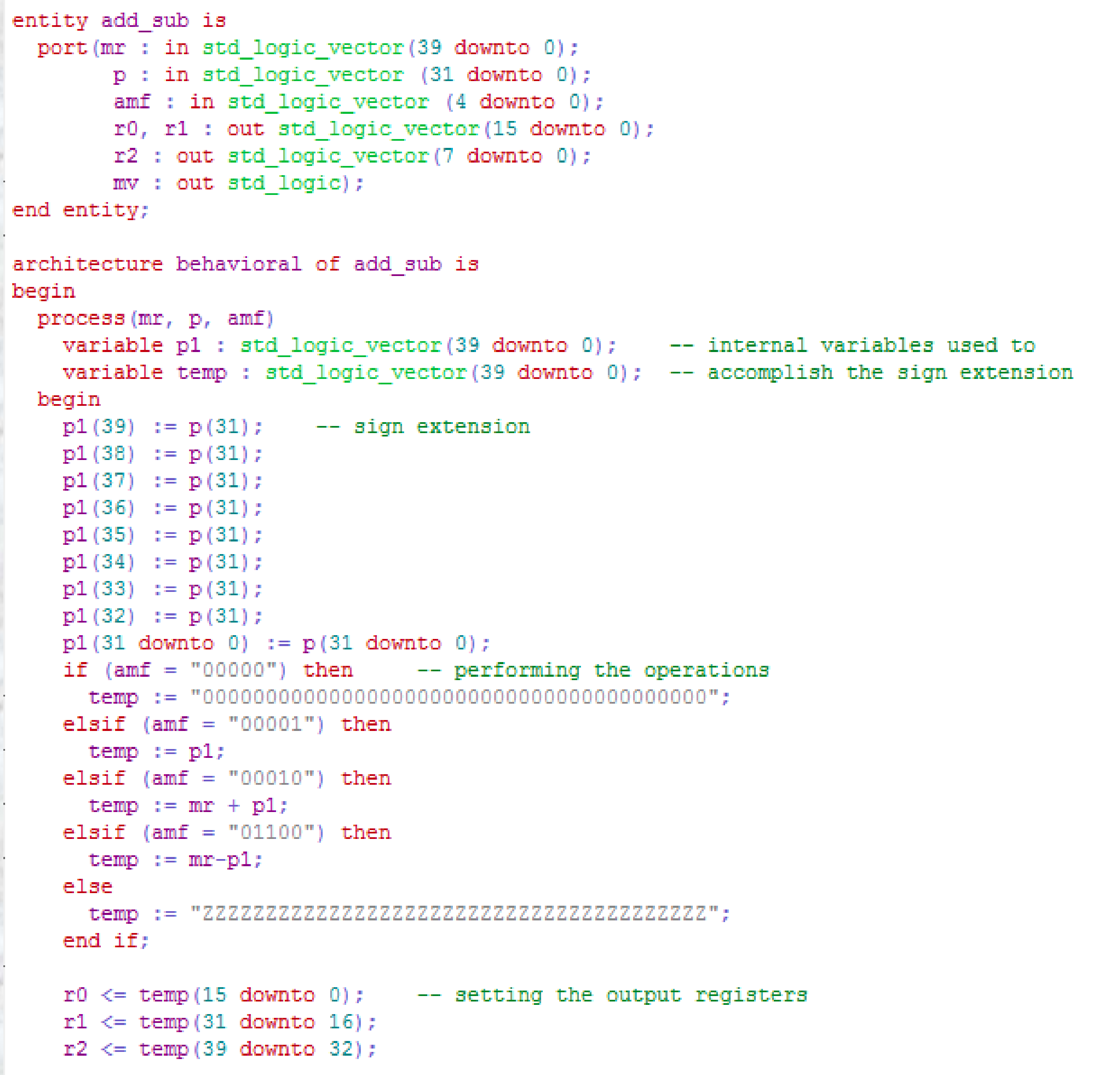


Design Process

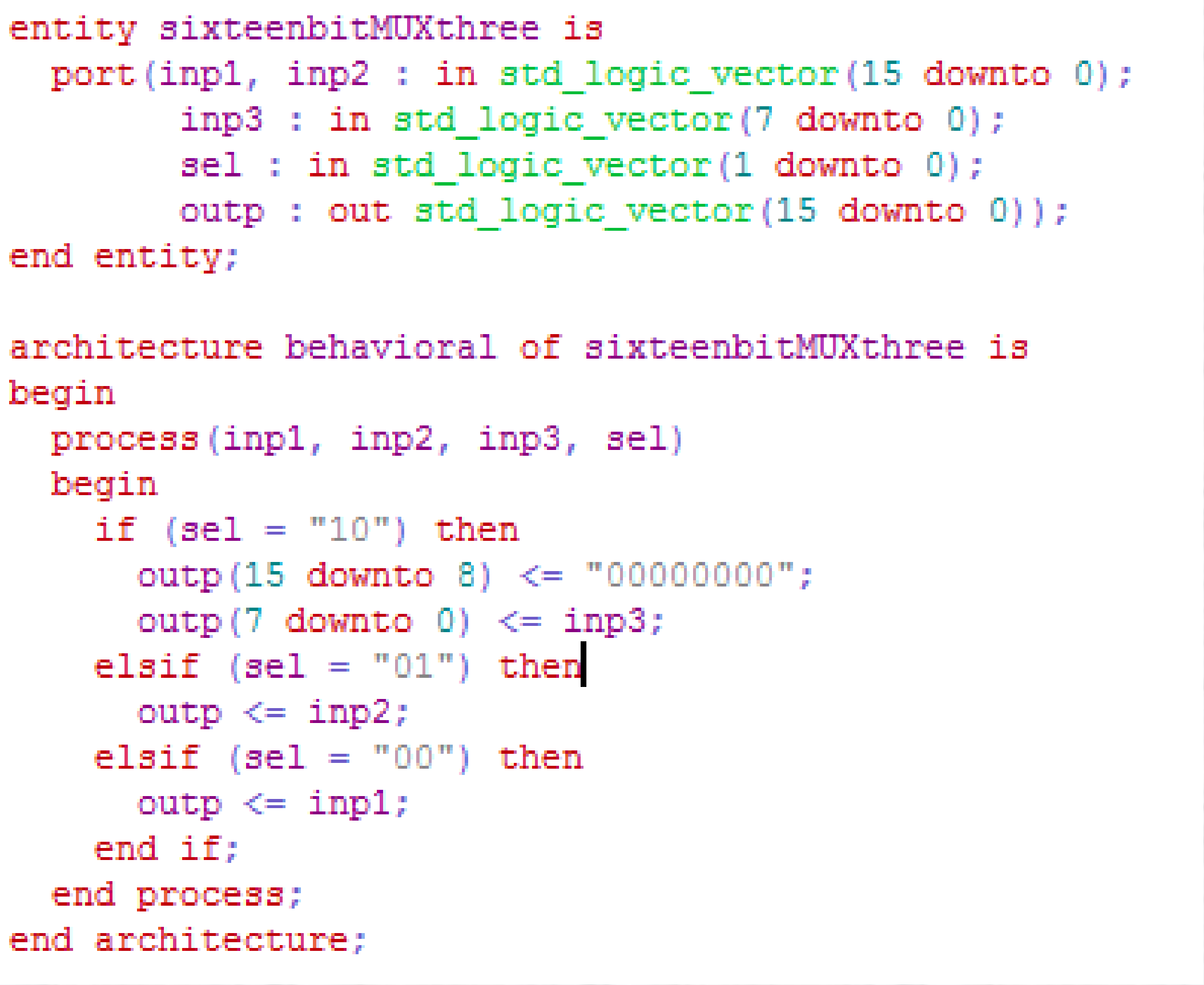
- Design the sub-components of the circuit (behavioral)
  - 16-bit Register, 8-bit Register
  - 16-bit 2-to-1 MUX, 8-bit 2-to-1 MUX, 16-bit 3-to-1 MUX
  - 16-bit Tri-State Buffer
- Design the multiplier and adder (behavioral)
- Code the top-level (structural)
- Write a DO script for simulation

- Must account for the third output of the adder being only 8-bits

Adder

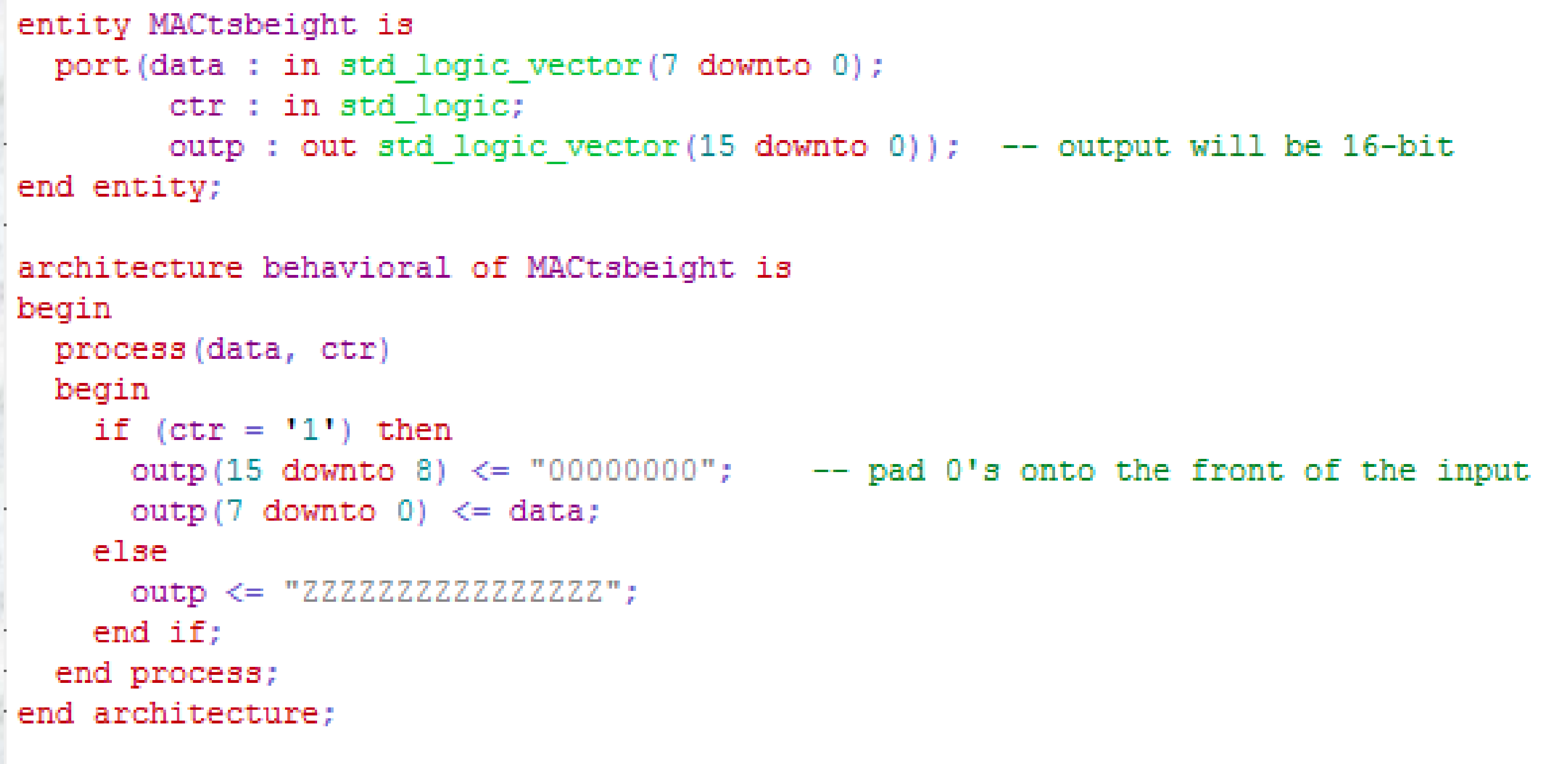


```
entity add_sub is
  port(mr : in std_logic_vector(39 downto 0);
        p : in std_logic_vector (31 downto 0);
        amf : in std_logic_vector (4 downto 0);
        r0, r1 : out std_logic_vector(15 downto 0);
        r2 : out std_logic_vector(7 downto 0);
        mv : out std_logic);
end entity;

architecture behavioral of add_sub is
begin
  process(mr, p, amf)
    variable p1 : std_logic_vector(39 downto 0);    -- internal variables used to
    variable temp : std_logic_vector(39 downto 0);  -- accomplish the sign extension
  begin
    p1(39) := p(31);    -- sign extension
    p1(38) := p(31);
    p1(37) := p(31);
    p1(36) := p(31);
    p1(35) := p(31);
    p1(34) := p(31);
    p1(33) := p(31);
    p1(32) := p(31);
    p1(31 downto 0) := p(31 downto 0);
    if (amf = "00000") then     -- performing the operations
      temp := "0000000000000000000000000000000000000000";
    elsif (amf = "00001") then
      temp := p1;
    elsif (amf = "00010") then
      temp := mr + p1;
    elsif (amf = "01100") then
      temp := mr-p1;
    else
      temp := "ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ";
    end if;

    r0 <= temp(15 downto 0);    -- setting the output registers
    r1 <= temp(31 downto 16);
    r2 <= temp(39 downto 32);
```



```
entity sixteenbitMUXthree is
  port(inp1, inp2 : in std_logic_vector(15 downto 0);
        inp3 : in std_logic_vector(7 downto 0);
        sel : in std_logic_vector(1 downto 0);
        outp : out std_logic_vector(15 downto 0));
end entity;

architecture behavioral of sixteenbitMUXthree is
begin
  process(inp1, inp2, inp3, sel)
  begin
    if (sel = "10") then
      outp(15 downto 8) <= "00000000";
      outp(7 downto 0) <= inp3;
    elsif (sel = "01") then
      outp <= inp2;
    elsif (sel = "00") then
      outp <= inp1;
    end if;
  end process;
end architecture;
```

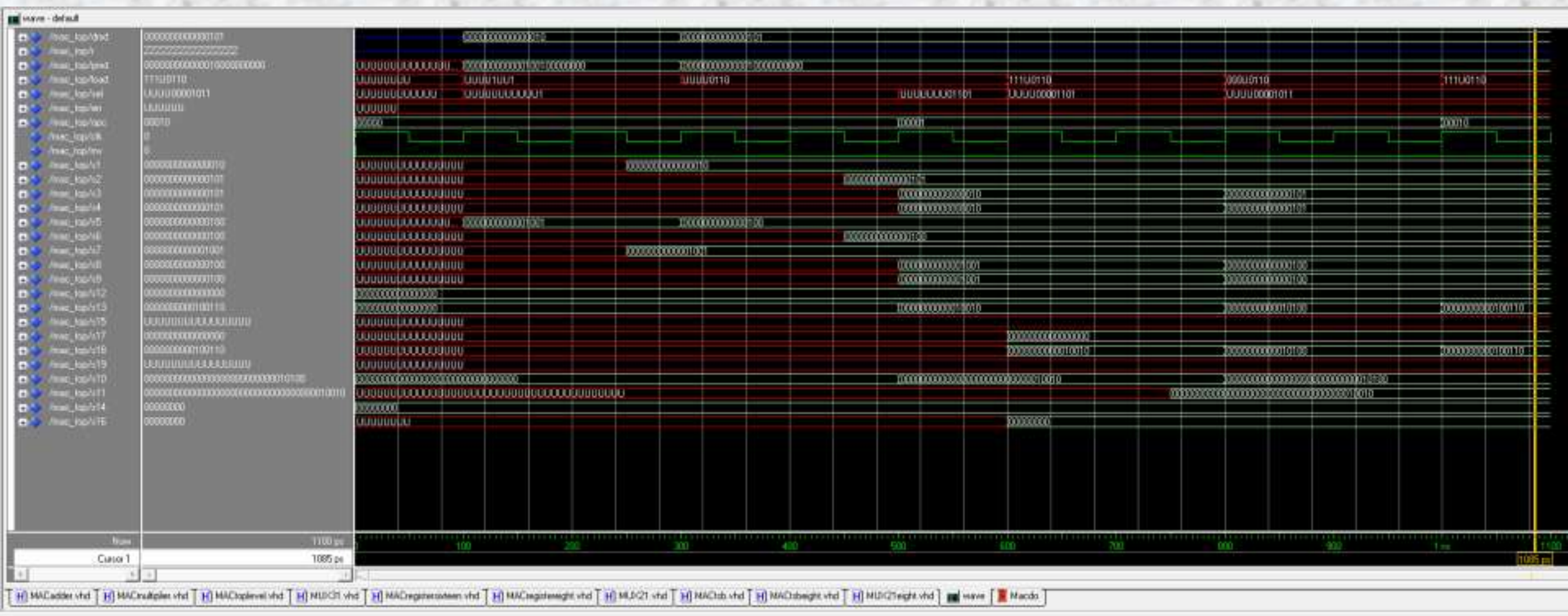
3-to-1 MUX



```
entity MACtsbeight is
  port(data : in std_logic_vector(7 downto 0);
        ctr : in std_logic;
        outp : out std_logic_vector(15 downto 0));  -- output will be 16-bit
end entity;

architecture behavioral of MACtsbeight is
begin
  process(data, ctr)
  begin
    if (ctr = '1') then
      outp(15 downto 8) <= "00000000";    -- pad 0's onto the front of the input
      outp(7 downto 0) <= data;
    else
      outp <= "ZZZZZZZZZZZZZZZZ";
    end if;
  end process;
end architecture;
```

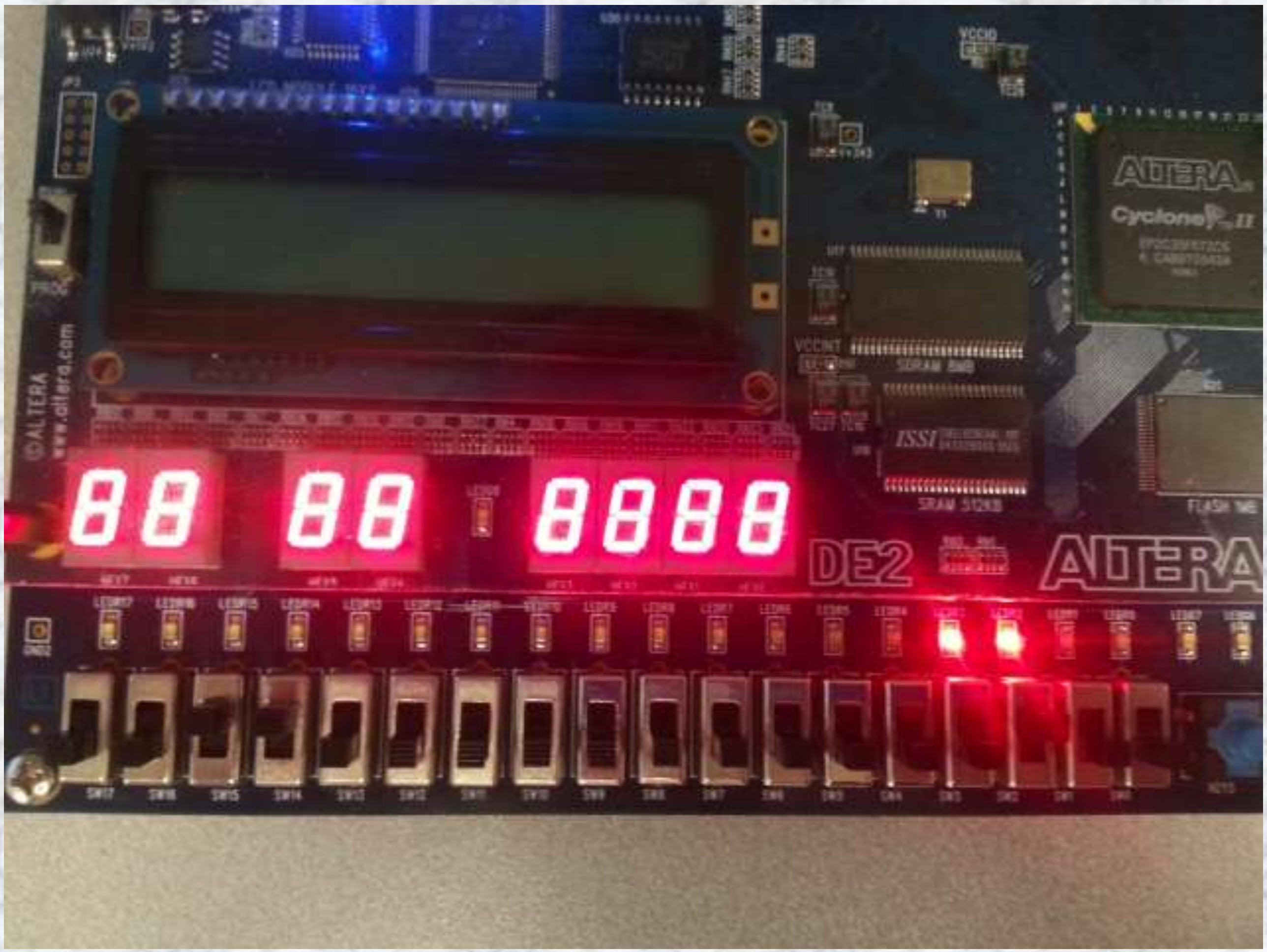
8-bit Tri-State Buffer

ModelSim Simulation



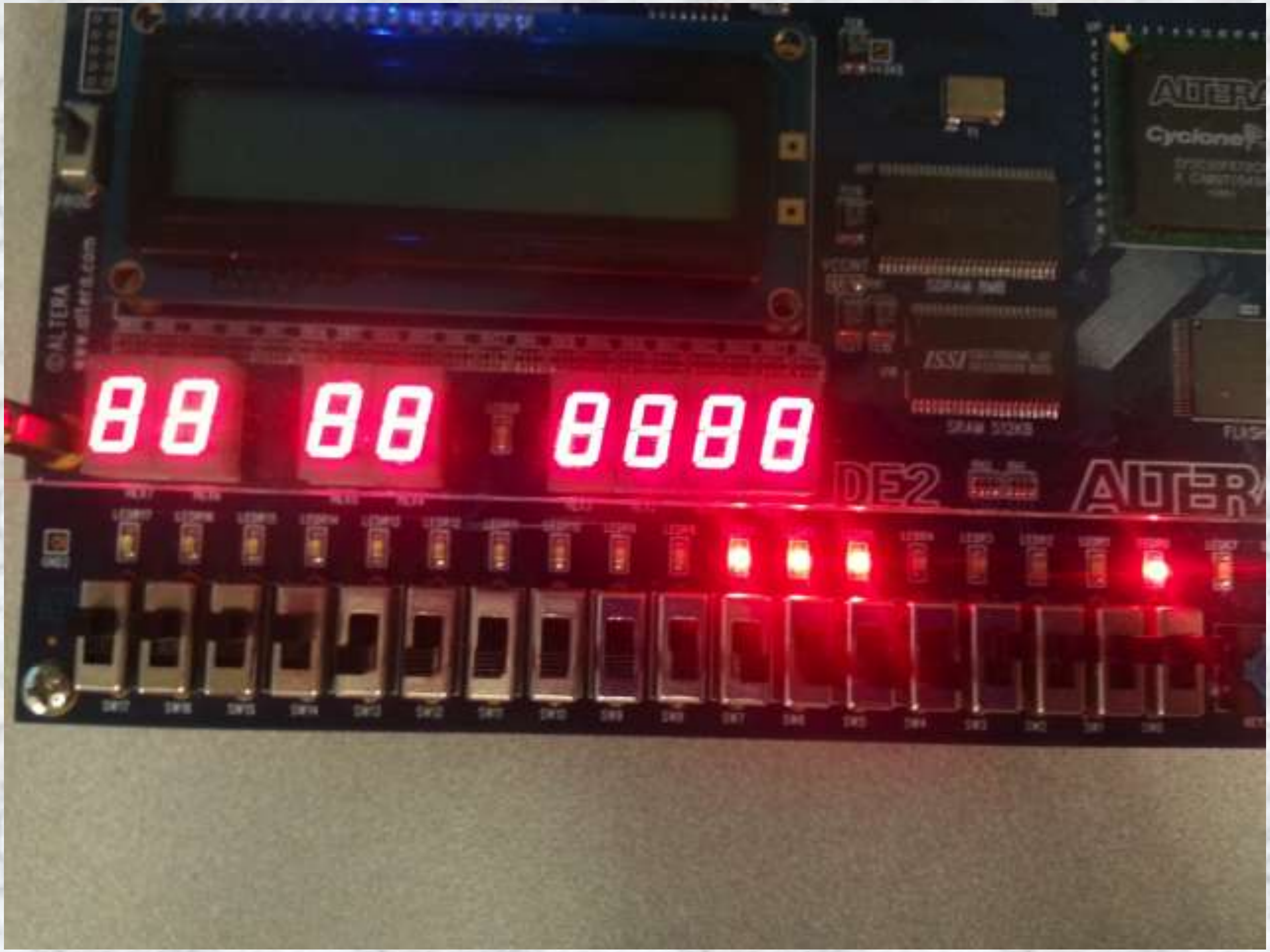
```
wave - default
/mac_top/dout     0000000000000101
/mac_top/h        ZZZZZZZZZZZZZZZZZ
/mac_top/pred     000000000000010000000000
/mac_top/load     1111U0110
/mac_top/sel      UUUU00001011
/mac_top/en       UUUUUUU
/mac_top/opc      00010
/mac_top/clk      0
/mac_top/rw       0
/mac_top/s1       0000000000000010
/mac_top/s2       0000000000000101
/mac_top/s3       0000000000000101
/mac_top/s4       0000000000000101
/mac_top/s5       0000000000000100
/mac_top/s6       0000000000000100
/mac_top/s7       0000000000001001
/mac_top/s8       0000000000000100
/mac_top/s9       0000000000000100
/mac_top/s12      0000000000000000
/mac_top/s13      0000000000100110
/mac_top/s15      UUUUUUUUUUUUUUUU
/mac_top/s17      0000000000000000
/mac_top/s18      0000000000100110
/mac_top/s19      UUUUUUUUUUUUUUUU
/mac_top/s10      00000000000000000000000000010100
/mac_top/s11      0000000000000000000000000000000000000010010
/mac_top/s14      00000000
/mac_top/s16      00000000

Now        1100 ps
Cursor 1   1085 ps
```

DE2 Board – Multiplier Simulation



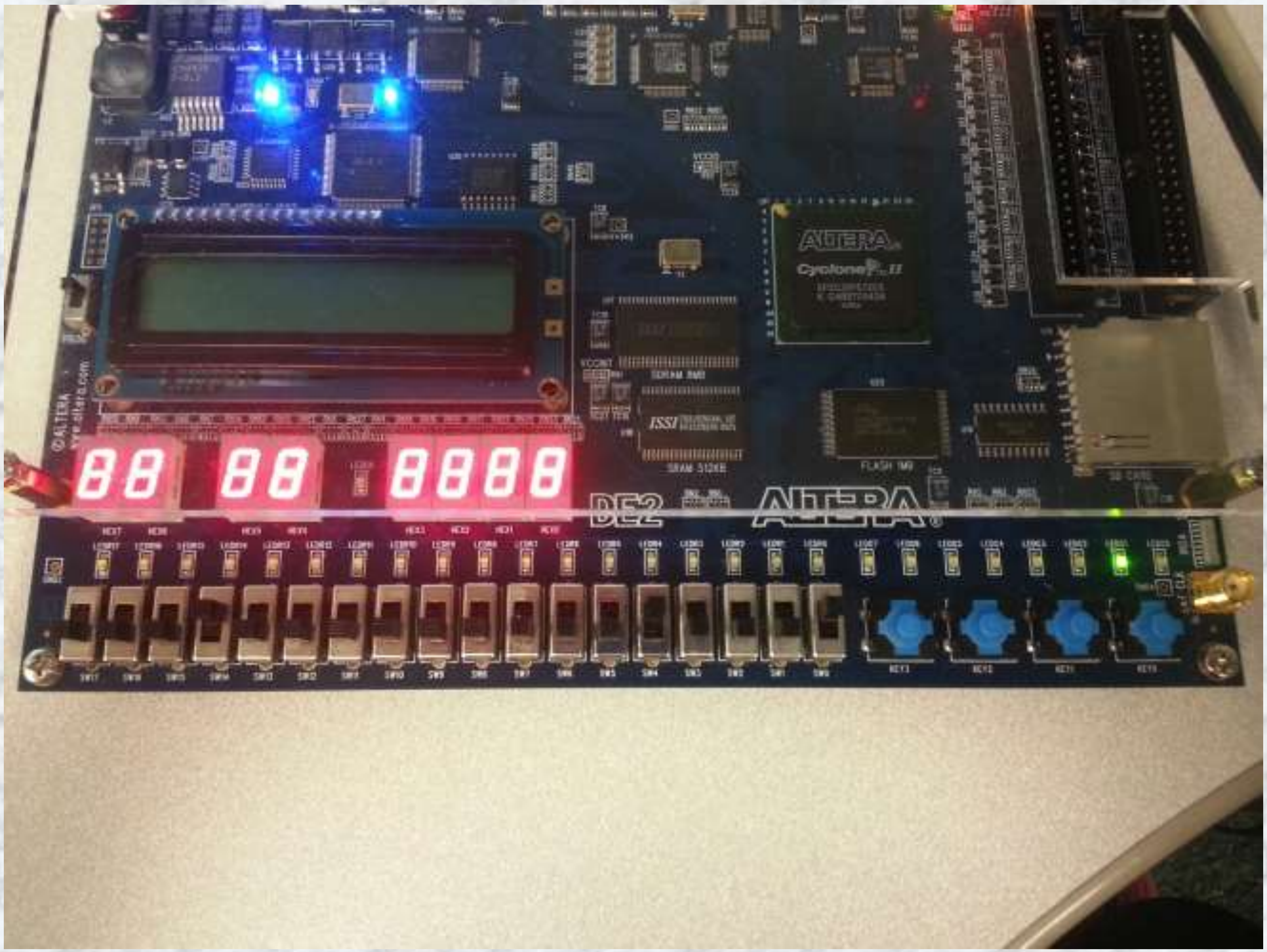
4x3 = 12

DE2 Board – Multiplier Simulation



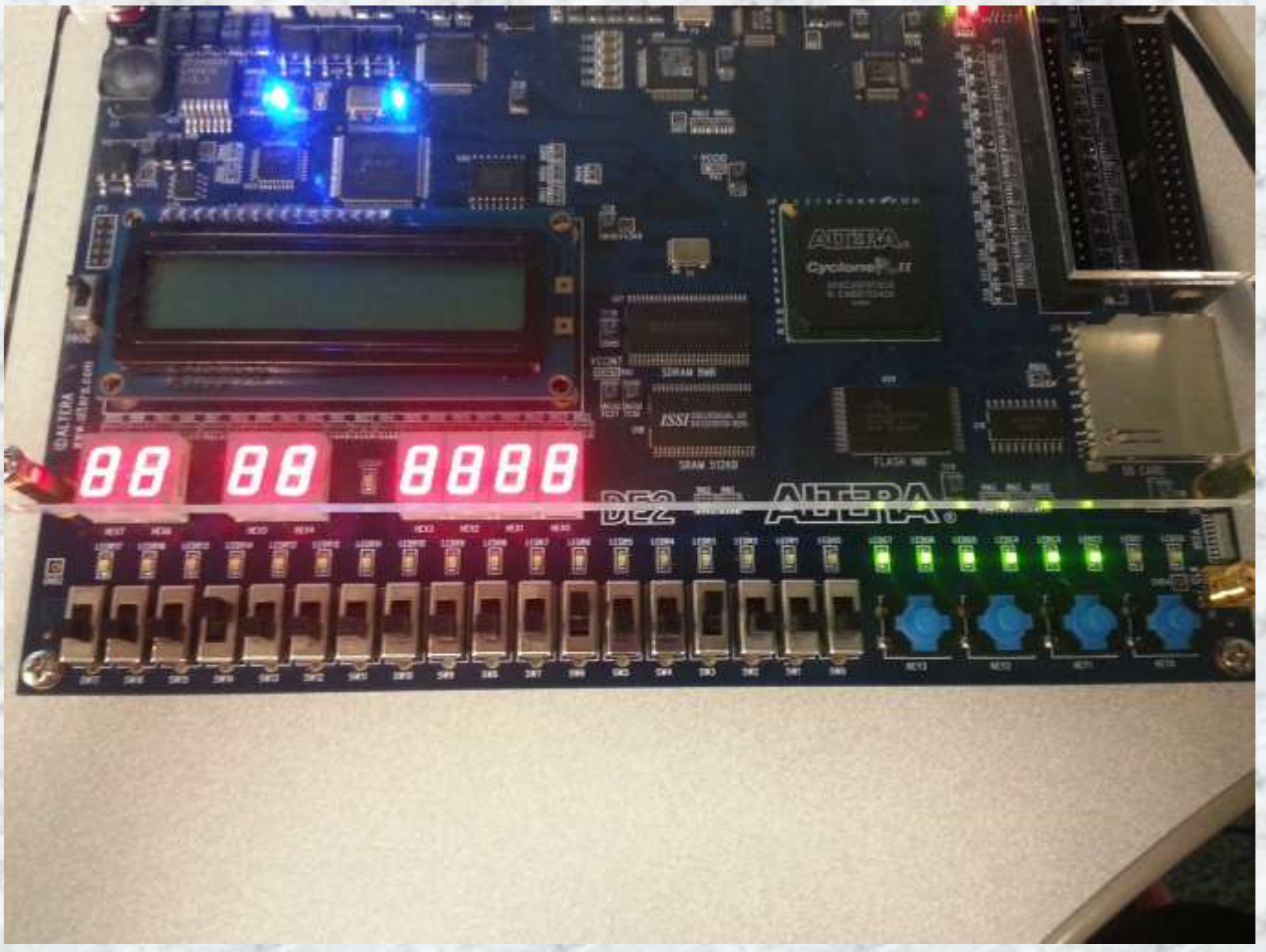
15x15 = 225

DE2 Board – Adder Simulation



1+1 = 2

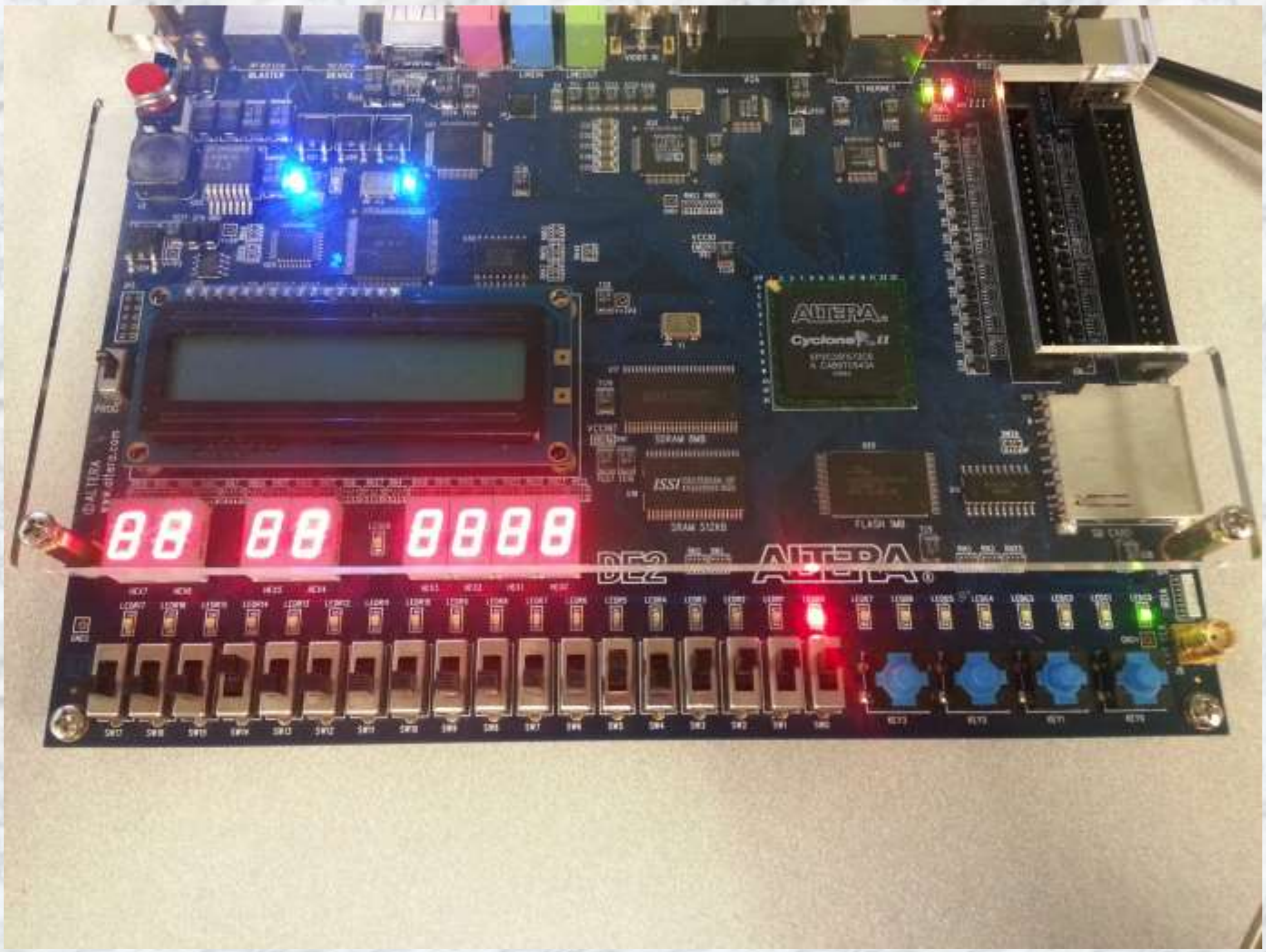
DE2 Board – Adder Simulation



8+4 = 12

*Sign Extension

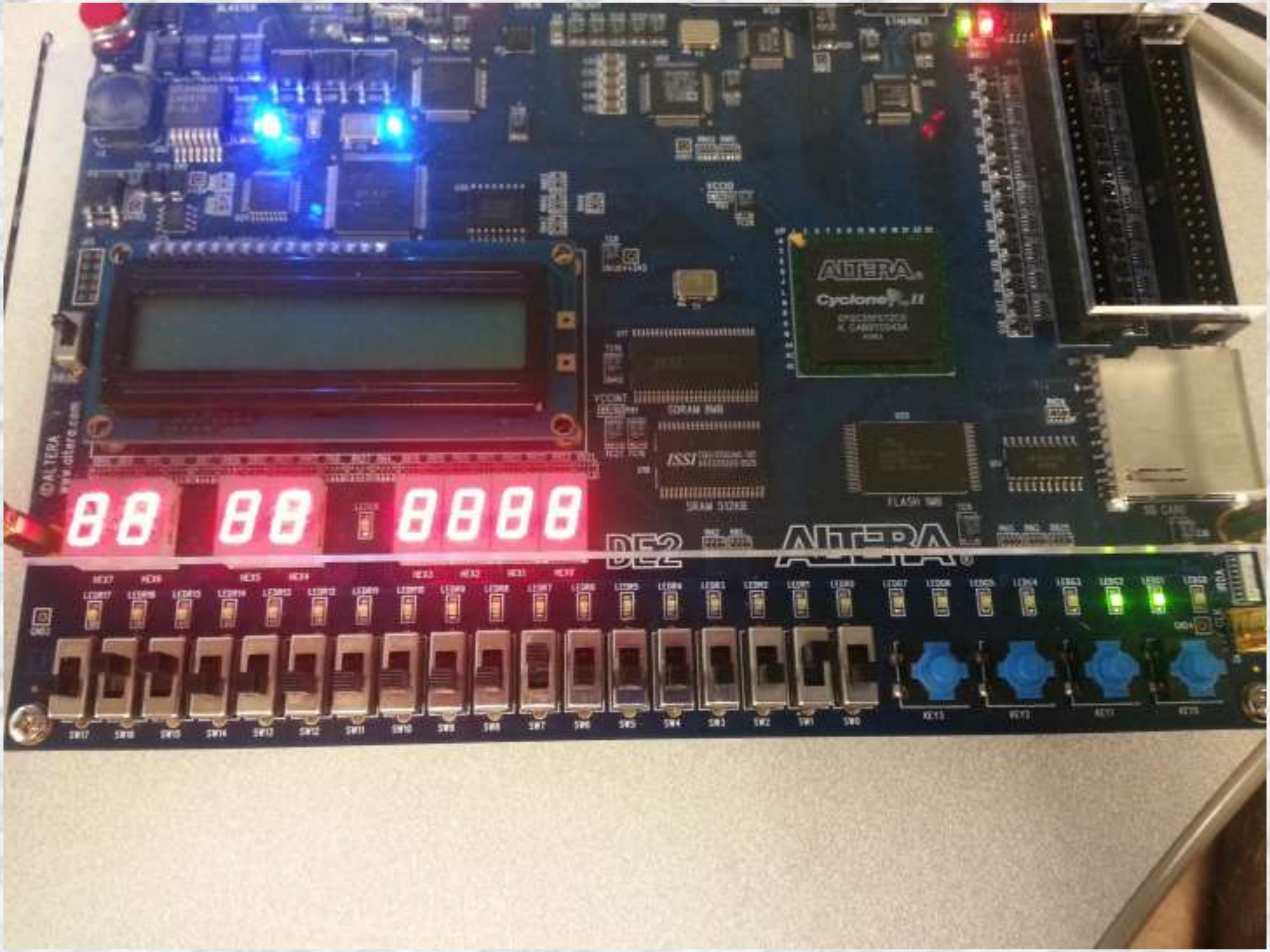
DE2 Board – Adder Simulation



15+2 = 17

*Overflow set

DE2 Board – Adder Simulation



8-2 = 6