

Design Sources (3)

- Design_1_wrapper(STRUCTURE) (Design_1_wrapper.vhd) (1)
 - Design_1_i : Design_1 (Design_1.bd) (1)
 - Design_1(STRUCTURE) (Design_1.vhd) (9)
 - c_addsub_0 : Design_1_c_caddsub_0_0 (Design_1_c_caddsub_0_0.xci)
 - divider_1 : Design_1_divider_0_2 (Module Reference Wrapper) (1)
 - mult_gen_0 : Design_1_mult_gen_0_0 (Design_1_mult_gen_0_0.xci)
 - mux_4to1_2 : Design_1_mux_4to1_1_1 (Module Reference Wrapper) (1)
 - xlconcat_0 : Design_1_xlconcat_0_0 (Design_1_xlconcat_0_0.xci)
 - xlconcat_1 : Design_1_xlconcat_0_1 (Design_1_xlconcat_0_1.xci)
 - xlconcat_2 : Design_1_xlconcat_0_2 (Design_1_xlconcat_0_2.xci)
 - xlconstant_0 : Design_1_xlconstant_0_1 (Design_1_xlconstant_0_1.xci)
 - xlconstant_3 : Design_1_xlconstant_2_2 (Design_1_xlconstant_2_2.xci)

UserIP			
AdderAndSubtractor_wrapper_v1_0	Production	Included	xilinx.com:user:AdderAndSubtractor:1.0
Design_1_wrapper_v1_0	Production	Included	xilinx.com:user:Design_1_wrapper:1.0
Divider_v1_0	Production	Included	xilinx.com:module_ref:divider:1.0
divider_v1_0	Production	Included	xilinx.com:module_ref:divider:1.0
mux 4to1 v1_0	Production	Included	xilinx.com:module_ref:mux 4to1:1.0

Adder/Subtractor

Adder/Subtractor (12.0)

Documentation IP Location

IP Symbol Information Component Name c_addsub_0

Show disabled ports

Basic Control

Implement using: Fabric

S = A +/− B

Input Type: Unsigned

Input Width: 8 [1..256] 8 [1..256]

Add Mode: Add Subtract

Output Width: 8 [8..16] 8 [8..16]

Latency Configuration: Manual

Latency: 0 [0..8] 0 [0..8]

Constant Input:

Constant Value (Bin): 00000000

Multiplication

IP Symbol Information Component Name mult_gen_0

Show disabled ports

Basic Output and Control

Multiplication Type:

Parallel Multiplier Constant Coefficient Multiplier

Input Options

P = A · B

A Data Type: Unsigned

B Data Type: Unsigned

A Width: 8 [1..64] Range: 1..64

B Width: 8 [1..64] Range: 1..64

Multiplier Construction: Use LUTs

Optimization Options: Speed Optimized

Area: The multiplier will be optimized to reduce slice logic and overall area.

Speed: The multiplier will be optimized for performance.

OK Cancel

Output Product Range

Use Custom Output Width

Output MSB: 15 [0..127]

Output LSB: 0 [0..15]

Output product width (max, min) = (15,0)

Use Symmetric Rounding

Pipelining and Control Signals

Pipeline Stages: 0 Optimum pipeline stages: 3

Clock Enable Synchronous Clear

Synchronous Controls and Clock Enable(CE) Priority: SCLR Overrides CE

DIVIDER

```
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.ALL;
3 use IEEE.NUMERIC_STD.all;
4
5 entity divider is
6     Port ( A : in STD_LOGIC_VECTOR (7 downto 0);
7             B : in STD_LOGIC_VECTOR (7 downto 0);
8
9             Q: out STD_LOGIC_VECTOR (7 downto 0);
10            R: out STD_LOGIC_VECTOR(7 downto 0)
11         );
12 end divider;
13
14 architecture Behavioral of divider is
15
16 signal tempA:unsigned(7 downto 0);
17 signal tempB:unsigned(7 downto 0);
18 begin
19
20     tempA <= unsigned(A) / unsigned(B);
21     tempB <= unsigned(A) mod unsigned(B);
22     Q <= std_logic_vector(tempA);
23     R <= std_logic_vector(tempB);
24 end Behavioral;
```

4TO1 MULTIPLEXER

```
Q | ☰ | ← | → | ✎ | ☐ | ☱ | X | // | ☰ | ? |
1 library IEEE;
2 use IEEE.STD_LOGIC_1164.all;
3
4 entity mux_4to1 is
5 port(
6
7     A,B,C,D : in STD_LOGIC_VECTOR(15 downto 0);
8     S0,S1: in STD_LOGIC;
9     Z: out STD_LOGIC_VECTOR(15 downto 0)
10    );
11 end mux_4to1;
12
13 architecture bhv of mux_4to1 is
14 begin
15 process (A,B,C,D,S0,S1) is
16 begin
17     if (S0 ='0' and S1 = '0') then
18         Z <= A;
19     elsif (S0 ='1' and S1 = '0') then
20         Z <= B;
21     elsif (S0 ='0' and S1 = '1') then
22         Z <= C;
23     else
24         Z <= D;
25     end if;
26
27 end process;
28 end bhv;
```

Design Wrapper

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
library UNISIM;
use UNISIM.VCOMPONENTS.ALL;
entity Design_1_wrapper is
  port (
    A : in STD_LOGIC_VECTOR ( 7 downto 0 );
    ADDORSUB : in STD_LOGIC;
    B : in STD_LOGIC_VECTOR ( 7 downto 0 );
    Result : out STD_LOGIC_VECTOR ( 15 downto 0 );
    S0 : in STD_LOGIC;
    S1 : in STD_LOGIC
  );
end Design_1_wrapper;

architecture STRUCTURE of Design_1_wrapper is
  component Design_1 is
    port (
      Result : out STD_LOGIC_VECTOR ( 15 downto 0 );
      A : in STD_LOGIC_VECTOR ( 7 downto 0 );
      B : in STD_LOGIC_VECTOR ( 7 downto 0 );
      ADDORSUB : in STD_LOGIC;
      S0 : in STD_LOGIC;
      S1 : in STD_LOGIC
    );
  end component Design_1;
  begin
    Design_1_i: component Design_1
      port map (
        A(7 downto 0) => A(7 downto 0),
        ADDORSUB => ADDORSUB,
        B(7 downto 0) => B(7 downto 0),
        Result(15 downto 0) => Result(15 downto 0),
        S0 => S0,
        S1 => S1
      );
  end;
end STRUCTURE;
```

Testbench

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity test_Design_1 is
end test_Design_1;

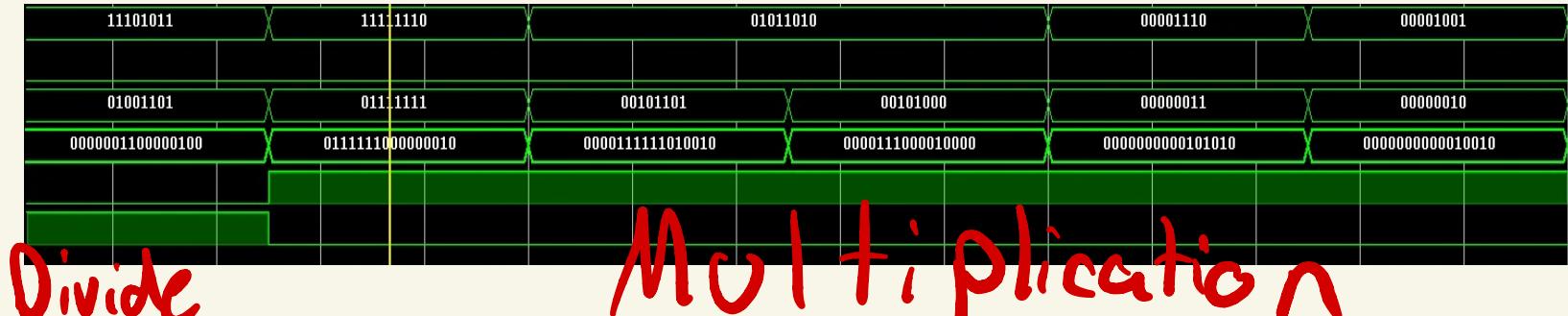
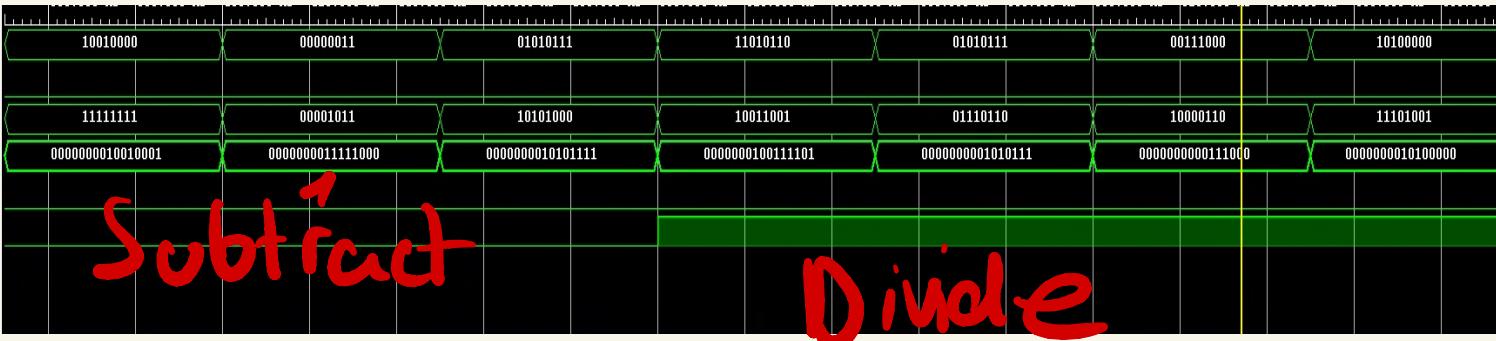
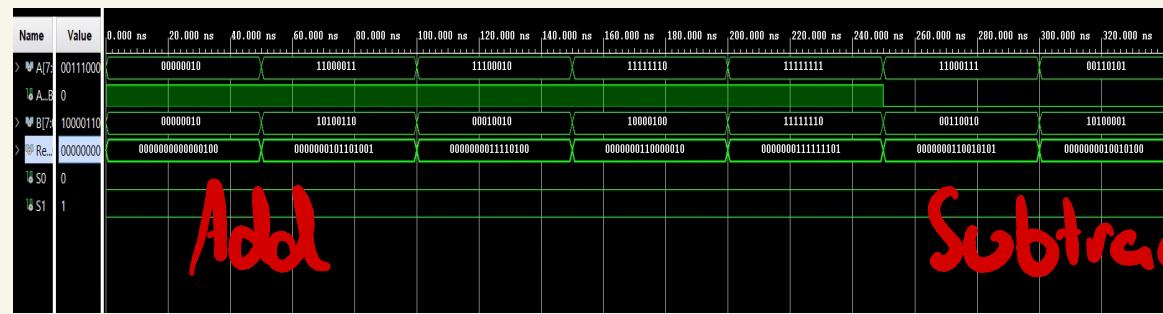
architecture TB of test_Design_1 is
  component Design_1 is
    port (
      A : in STD_LOGIC_VECTOR (7 downto 0);
      ADDORSUB : in STD_LOGIC;
      B : in STD_LOGIC_VECTOR (7 downto 0);
      Result : out STD_LOGIC_VECTOR (15 downto 0);
      S0 : in STD_LOGIC;
      S1 : in STD_LOGIC
    );
  end component Design_1;
  begin
    Design_1_i: component Design_1
      port map (
        A => A,
        ADDORSUB => ADDORSUB,
        B => B,
        Result => Result,
        S0 => S0,
        S1 => S1
      );
    stimulus: process
    begin
      ADDORSUB <= '1'; --ADD
      A <= "00000010";
      B <= "00000010";
    end;
  end;
```

testbench cont.

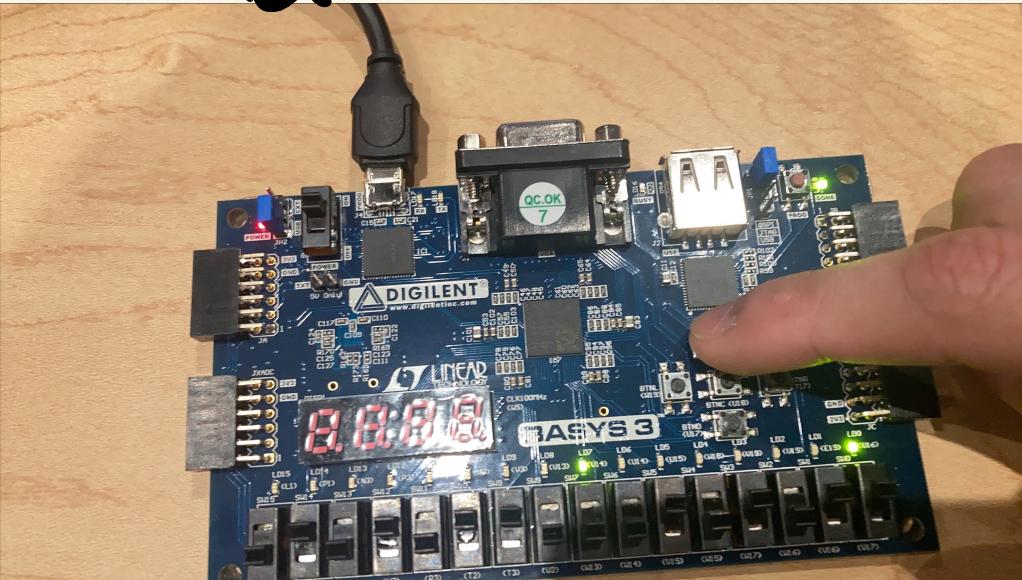
```
13    S1 <= '0';
14    wait for 50ns;
15    ADDORSUB <= '1'; --ADD
16    A <= "110000011";
17    B <= "101000110";
18    SO <= '0';
19    S1 <= '0';
20    wait for 50ns;
21    ADDORSUB <= '1'; --ADD
22    A <= "111000010";
23    B <= "000100010";
24    SO <= '0';
25    S1 <= '0';
26    wait for 50ns;
27    ADDORSUB <= '1'; --ADD
28    A <= "111100010";
29    B <= "100000100";
30    SO <= '0';
31    S1 <= '0';
32    wait for 50ns;
33    ADDORSUB <= '1'; --ADD
34    A <= "111111111";
35    B <= "111111110";
36    SO <= '0';
37    S1 <= '0';
38    wait for 50ns;
39
40    ADDORSUB <= '0'; --SUB
41    A <= "110000111";
42    B <= "001100010";
43    SO <= '0';
44    S1 <= '0';
45    wait for 50ns;
46    ADDORSUB <= '0'; --SUB
47    A <= "001101010";
48    B <= "101000001";
49    SO <= '0';
50    S1 <= '0';
51    wait for 50ns;
52    ADDORSUB <= '0'; --SUB
53    A <= "100100000";
54
55    B <= "111111111";
56    SO <= '0';
57    S1 <= '0';
58    wait for 50ns;
59    ADDORSUB <= '0'; --SUB
60    A <= "000000011";
61    B <= "000001011";
62    SO <= '0';
63    S1 <= '0';
64    wait for 50ns;
65    ADDORSUB <= '0'; --SUB
66    A <= "01010111";
67    B <= "10101000";
68    SO <= '0';
69    S1 <= '0';
70    wait for 50ns;
71
72    A <= "11010110"; --MULTIPLY
73    B <= "100110001";
74    SO <= '0';
75    S1 <= '1';
76    wait for 50ns;
77    A <= "01010111"; --MULTIPLY
78    B <= "01110110";
79    SO <= '0';
80    S1 <= '1';
81    wait for 50ns;
82    A <= "001110000"; --MULTIPLY
83    B <= "10000110";
84    SO <= '0';
85    S1 <= '1';
86    wait for 50ns;
87    A <= "10100000"; --MULTIPLY
88    B <= "111010001";
89    SO <= '0';
90    S1 <= '1';
91    wait for 50ns;
92    A <= "11101011"; --MULTIPLY
93    B <= "01001101";
94    SO <= '0';
```

Constraint

```
2 ## To use it in a project:  
3 ## - uncomment the lines corresponding to used pins  
4 ## - rename the used ports (in each line, after get_ports) according to the top level s  
5 set_property SEVERITY {Warning} [get_drc_checks NSTD-1]  
6 set_property SEVERITY {Warning} [get_drc_checks UCIO-1]  
7 ## Clock signal  
8 #set_property PACKAGE_PIN W5 [get_ports clk]  
9 #set_property IOSTANDARD LVCMOS33 [get_ports clk]  
0 #create_clock -add -name sys_clk_pin -period 10.00 -waveform {0 5} [get_ports clk]  
1  
2 ## Switches  
3 set_property PACKAGE_PIN V17 [get_ports {A[0]}]  
4 set_property IOSTANDARD LVCMOS33 [get_ports {A[0]}]  
5 set_property PACKAGE_PIN V16 [get_ports {A[1]}]  
6 set_property IOSTANDARD LVCMOS33 [get_ports {A[1]}]  
7 set_property PACKAGE_PIN W16 [get_ports {A[2]}]  
8 set_property IOSTANDARD LVCMOS33 [get_ports {A[2]}]  
9 set_property PACKAGE_PIN W17 [get_ports {A[3]}]  
0 set_property IOSTANDARD LVCMOS33 [get_ports {A[3]}]  
1 set_property PACKAGE_PIN W15 [get_ports {A[4]}]  
2 set_property IOSTANDARD LVCMOS33 [get_ports {A[4]}]  
3 set_property PACKAGE_PIN V15 [get_ports {A[5]}]  
4 set_property IOSTANDARD LVCMOS33 [get_ports {A[5]}]  
5 set_property PACKAGE_PIN W14 [get_ports {A[6]}]  
6 set_property IOSTANDARD LVCMOS33 [get_ports {A[6]}]  
7 set_property PACKAGE_PIN W13 [get_ports {A[7]}]  
8 set_property IOSTANDARD LVCMOS33 [get_ports {A[7]}]  
9 set_property PACKAGE_PIN V2 [get_ports {B[0]}]  
0 set_property IOSTANDARD LVCMOS33 [get_ports {B[0]}]  
1 set_property PACKAGE_PIN T3 [get_ports {B[1]}]  
2 set_property IOSTANDARD LVCMOS33 [get_ports {B[1]}]  
3 set_property PACKAGE_PIN T2 [get_ports {B[2]}]  
4 set_property IOSTANDARD LVCMOS33 [get_ports {B[2]}]  
5 set_property PACKAGE_PIN R3 [get_ports {B[3]}]  
6 set_property IOSTANDARD LVCMOS33 [get_ports {B[3]}]  
7 set_property PACKAGE_PIN W2 [get_ports {B[4]}]  
8 set_property IOSTANDARD LVCMOS33 [get_ports {B[4]}]  
9 set_property PACKAGE_PIN U1 [get_ports {B[5]}]  
0 set_property IOSTANDARD LVCMOS33 [get_ports {B[5]}]  
1 set_property PACKAGE_PIN T1 [get_ports {B[6]}]  
set_property PACKAGE_PIN R2 [get_ports {B[7]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {B[7]}]  
  
## LEDs  
set_property PACKAGE_PIN U16 [get_ports {Result[0]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[0]}]  
set_property PACKAGE_PIN E19 [get_ports {Result[1]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[1]}]  
set_property PACKAGE_PIN U19 [get_ports {Result[2]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[2]}]  
set_property PACKAGE_PIN V19 [get_ports {Result[3]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[3]}]  
set_property PACKAGE_PIN W18 [get_ports {Result[4]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[4]}]  
set_property PACKAGE_PIN U15 [get_ports {Result[5]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[5]}]  
set_property PACKAGE_PIN U14 [get_ports {Result[6]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[6]}]  
set_property PACKAGE_PIN V14 [get_ports {Result[7]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[7]}]  
set_property PACKAGE_PIN V13 [get_ports {Result[8]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[8]}]  
set_property PACKAGE_PIN V3 [get_ports {Result[9]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[9]}]  
set_property PACKAGE_PIN W3 [get_ports {Result[10]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[10]}]  
set_property PACKAGE_PIN U3 [get_ports {Result[11]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[11]}]  
set_property PACKAGE_PIN F3 [get_ports {Result[12]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[12]}]  
set_property PACKAGE_PIN N3 [get_ports {Result[13]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[13]}]  
set_property PACKAGE_PIN P1 [get_ports {Result[14]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[14]}]  
set_property PACKAGE_PIN L1 [get_ports {Result[15]}]  
set_property IOSTANDARD LVCMOS33 [get_ports {Result[15]}]  
# set_property IOSTANDARD LVCMOS33 [get_ports btnc]  
set_property PACKAGE_PIN T18 [get_ports ADDRSUB]  
set_property IOSTANDARD LVCMOS33 [get_ports ADDRSUB]  
set_property PACKAGE_PIN W19 [get_ports S0]  
set_property IOSTANDARD LVCMOS33 [get_ports S0]  
set_property PACKAGE_PIN T17 [get_ports S1]  
set_property IOSTANDARD LVCMOS33 [get_ports S1]  
set_property CONFIG_VOLTAGE 3.3 [current_design]  
set_property CFGBVS VCCO [current_design]
```



Add



Result = "000000000100000001"

I have three buttons that selects the operation.

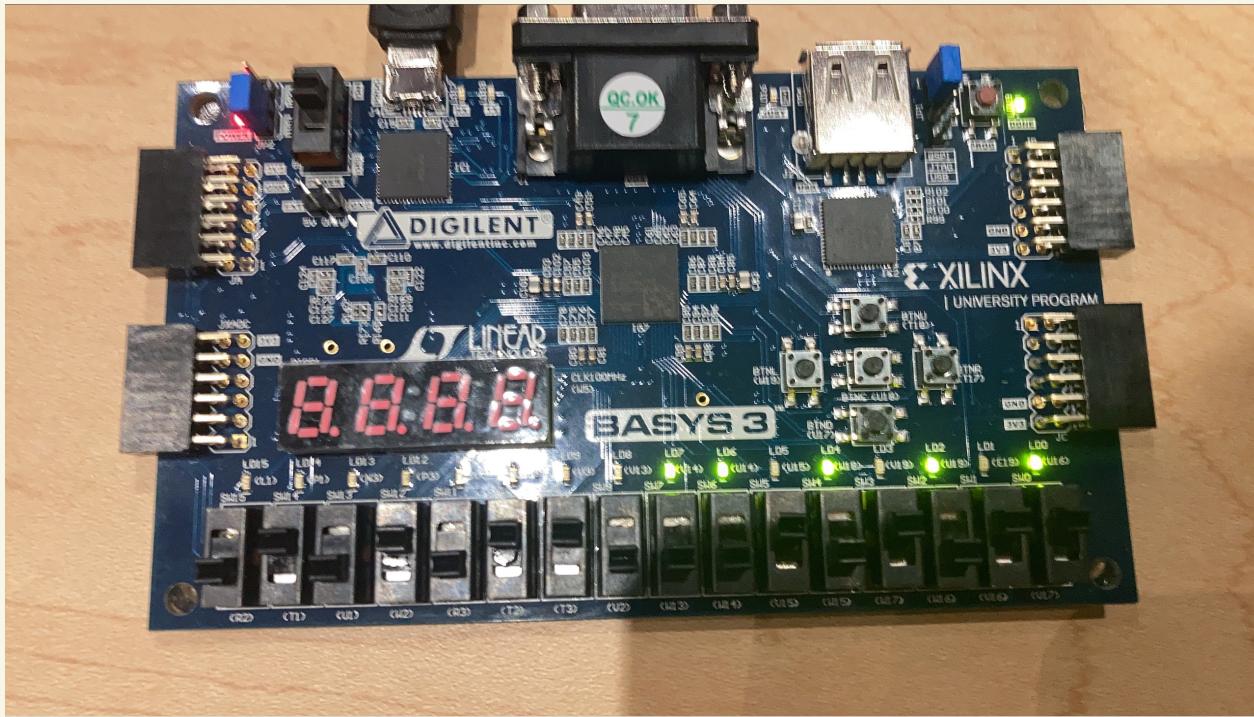
When $S_0 = '0'$, $S_1 = '0'$, then **ADD**

when $S_0 = '0'$, $S_1 = '1'$, then **SUBTRACT**

when $S_0 = '1'$, $S_1 = '0'$, then **DIVIDE**

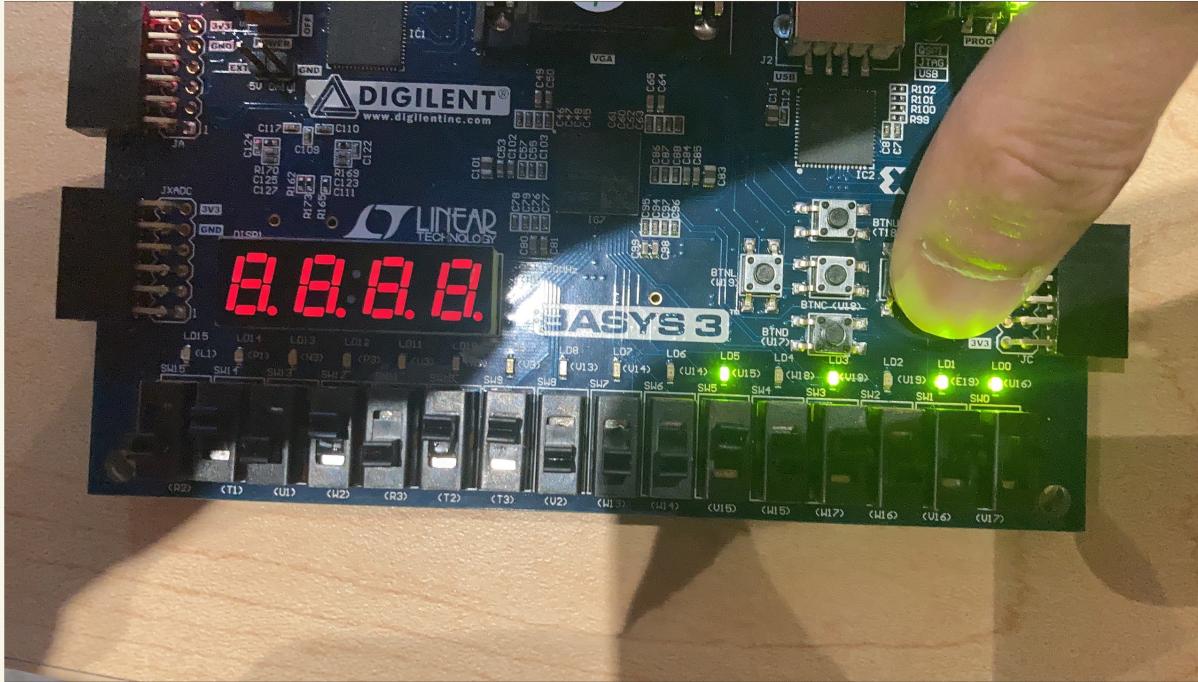
when $S_0 = '1'$, $S_1 = '1'$, then **MULTPLY**

Subtract



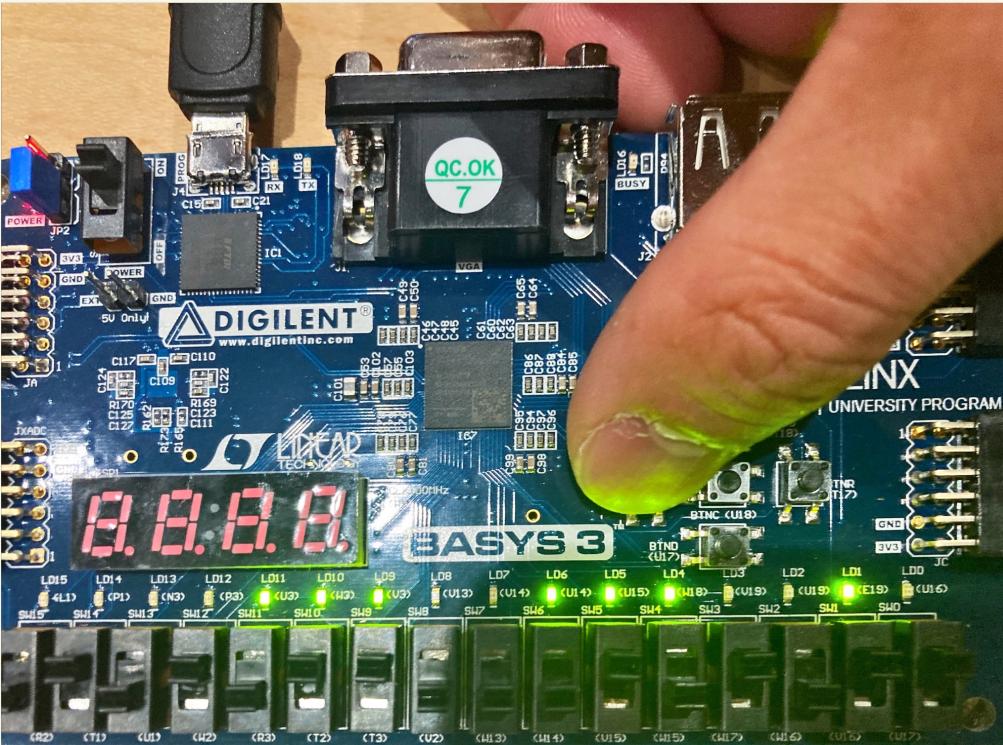
Result = "0000000011010101"

DIVIDE



Result = "000000000000101011"

MULTIPLX



Result = "00001100110010"