

LAB3 - 16 bits Full Adder using instantiation

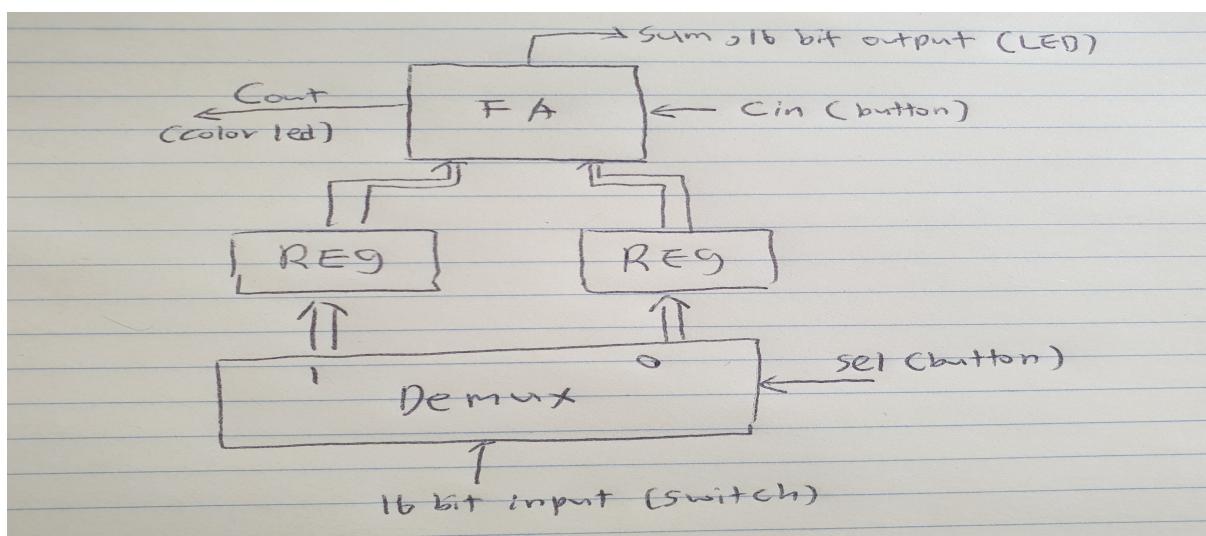
Group A :

- Jihun Choi(Bronco ID#: 011841206)
- Jacky Li(Bronco ID#:014181245)

Prof.Aly

ECE3300L.02 - Summer 2022

This lab3 is about designing a 16 bit Full Adder by instantiation using vivado software and test the results with FPGA board.



First, 1 bit Full Adder was designed by the results of its k-maps.

```
module FA(
    input A,
    input B,
    input CI,
    output wire SUM,
    output wire CO
);

    assign SUM = A ^ B ^ CI; // ^ is xor
    assign CO = (A & B) | (A & CI) | (B & CI);
endmodule
```

And Demultiplexer was designed by using its functionality

```
module Demux(
    input  in,
    input  sel,
    output reg out0,
    output reg out1
);

    always @(in,sel)
    begin
        if(sel)
            out1=in;
        else
            out0=in;
    end
endmodule
```

if sel equals high, input 'in' equals to output 'out1', else input 'in' equals to output 'out0.' Using the idea of instantiation and parameter, module 'top' was created to design n-bit Full Adder connected with n-bit Demultiplexer. For this lab, n equals 16 which is 16 bit FA with Demux

```
module top
#(parameter SIZE = 16)
(
    input [SIZE-1:0] in,
    input sel,
    input CI,
    output [SIZE-1:0] SUM,
    output CO
);

    wire [SIZE-1:0] x;
    wire [SIZE-1:0] y;
    wire [SIZE:0] wire_RCA;
    assign wire_RCA[0] = CI;

    genvar i; //for loop for RTL
    for (i=0; i<SIZE; i=i+1)
        begin
            Demux d1(
                .in(in[i]),
                .sel(sel),
                .out0(x[i]),
                .out1(y[i])
            );
            FA f1 (
                .A(x[i]),
                .B(y[i]),
                .CI(wire_RCA[i]),
                .SUM(SUM[i]),
                .CO(wire_RCA[i+1])
            );
        end
    assign CO = wire_RCA[SIZE];
endmodule
```

After setting up the source file, test bench was created to test if 16 bit Full Adder connected with Demultiplexer functions as it was intended. First, it tests to see if the function of Demultiplexer works by changing the value of the select line. Once it's checked, carry-in and the function of full adder was also checked.

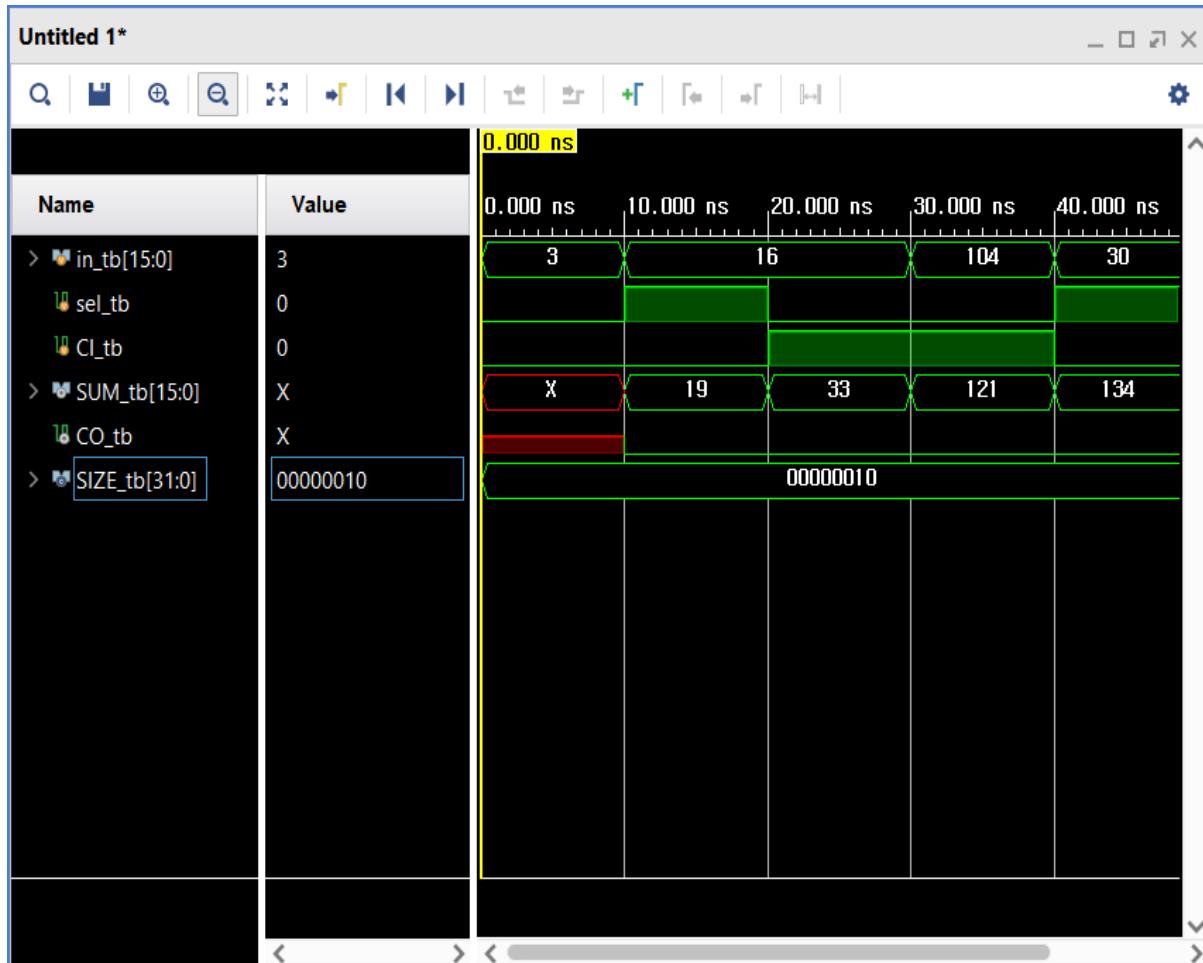
```

module top_testbench
  #(parameter SIZE_tb=16)(
    );
  reg [SIZE_tb-1:0] in_tb;
  reg sel_tb;
  reg CI_tb;
  wire [SIZE_tb-1:0] SUM_tb;
  wire CO_tb;

  //We can change the size by changing number inside .SIZE().
  top #(.SIZE(SIZE_tb)) Test1 (
    .in(in_tb),
    .sel(sel_tb),
    .CI(CI_tb),
    .SUM(SUM_tb),
    .CO(CO_tb)
  );
  initial
    begin
      CI_tb = 1'b0;
      in_tb = 16'd3;
      sel_tb = 1'b0;
      #10
      in_tb = 16'd16;
      sel_tb = 1'b1;
      #10
      CI_tb = 1'b1;
      in_tb = 16'd16;
      sel_tb = 1'b0;
      #10
      in_tb=16'd104;
      sel_tb = 1'b0;
      #10
      CI_tb = 1'b0;
      in_tb=16'd30;
      sel_tb = 1'b1;
      #10
      $finish;
    end
  endmodule

```

And simulation was performed,



First #10 nanoseconds: Looking at the result we can see that when Cl(carry in) = 0, in(input) = 3, sel(select line = 0), there is no sum result since no value was assigned to out0.

Second #10 nanoseconds: in(input) = 16, sel(select line = 1), the result of sum comes out to be $3(\text{out0}) + 16(\text{out1}) = 19(\text{sum})$.

Third #10 nanoseconds: Cl(carry_in) = 1, in(input) = 16, sel(select line = 0), the result of sum comes out to be $16(\text{out0}) + 16(\text{out1}) + 1(\text{Cl}) = 33(\text{sum})$.

Fourth #10 nanoseconds: Cl(carry_in) = 1, in(input) = 104, sel(select line = 0), the result of sum comes out to be $104(\text{out0}) + 16(\text{out1}) + 1(\text{Cl}) = 121(\text{sum})$.

Final #10 nanoseconds: Cl(carry_in) = 0, in(input) = 30, sel(select line = 1), the result of sum comes out to be $104(\text{out0}) + 30(\text{out1}) + 0(\text{Cl}) = 134(\text{sum})$.

After checking simulation results, 16 bit Full Adder connected with

Demultiplexer is functioning correctly. xdc file was modified to properly work and the result was tested on FPGA.

```
##Switches

set_property -dict { PACKAGE_PIN J15 10STANDARD LVCMS33 } [get_ports { in[0] }]; #10_L24N_T3_RS0_15 Sch=sw[0]
set_property -dict { PACKAGE_PIN L16 10STANDARD LVCMS33 } [get_ports { in[1] }]; #10_L3N_T0_DQS_EMCCLK_14 Sch=sw[1]
set_property -dict { PACKAGE_PIN M13 10STANDARD LVCMS33 } [get_ports { in[2] }]; #10_L6N_T0_D08_VREF_14 Sch=sw[2]
set_property -dict { PACKAGE_PIN R15 10STANDARD LVCMS33 } [get_ports { in[3] }]; #10_L13N_T2_MRCC_14 Sch=sw[3]
set_property -dict { PACKAGE_PIN R17 10STANDARD LVCMS33 } [get_ports { in[4] }]; #10_L12N_T1_MRCC_14 Sch=sw[4]
set_property -dict { PACKAGE_PIN T18 10STANDARD LVCMS33 } [get_ports { in[5] }]; #10_L7N_T1_D10_14 Sch=sw[5]
set_property -dict { PACKAGE_PIN U18 10STANDARD LVCMS33 } [get_ports { in[6] }]; #10_L17N_T2_A13_D29_14 Sch=sw[6]
set_property -dict { PACKAGE_PIN R13 10STANDARD LVCMS33 } [get_ports { in[7] }]; #10_L5N_T0_D07_14 Sch=sw[7]
set_property -dict { PACKAGE_PIN T8 10STANDARD LVCMS18 } [get_ports { in[8] }]; #10_L24N_T3_34 Sch=sw[8]
set_property -dict { PACKAGE_PIN U8 10STANDARD LVCMS18 } [get_ports { in[9] }]; #10_25_34 Sch=sw[9]
set_property -dict { PACKAGE_PIN R16 10STANDARD LVCMS33 } [get_ports { in[10] }]; #10_L15P_T2_D05_RDWR_B_14 Sch=sw[10]
set_property -dict { PACKAGE_PIN T13 10STANDARD LVCMS33 } [get_ports { in[11] }]; #10_L23P_T3_A03_D19_14 Sch=sw[11]
set_property -dict { PACKAGE_PIN H6 10STANDARD LVCMS33 } [get_ports { in[12] }]; #10_L24P_T3_35 Sch=sw[12]
set_property -dict { PACKAGE_PIN U12 10STANDARD LVCMS33 } [get_ports { in[13] }]; #10_L20P_T3_A08_D24_14 Sch=sw[13]
set_property -dict { PACKAGE_PIN U11 10STANDARD LVCMS33 } [get_ports { in[14] }]; #10_L19N_T8_A09_D25_VREF_14 Sch=sw[14]
set_property -dict { PACKAGE_PIN V10 10STANDARD LVCMS33 } [get_ports { in[15] }]; #10_L21P_T3_DQS_14 Sch=sw[15]

## LEDs

set_property -dict { PACKAGE_PIN H17 10STANDARD LVCMS33 } [get_ports { SUM[0] }]; #10_L18P_T2_A24_15 Sch=led[0]
set_property -dict { PACKAGE_PIN K15 10STANDARD LVCMS33 } [get_ports { SUM[1] }]; #10_L24P_T3_RS1_15 Sch=led[1]
set_property -dict { PACKAGE_PIN J13 10STANDARD LVCMS33 } [get_ports { SUM[2] }]; #10_L17N_T2_A25_15 Sch=led[2]
set_property -dict { PACKAGE_PIN N14 10STANDARD LVCMS33 } [get_ports { SUM[3] }]; #10_L8P_T1_D11_14 Sch=led[3]
set_property -dict { PACKAGE_PIN R18 10STANDARD LVCMS33 } [get_ports { SUM[4] }]; #10_L7P_T1_D09_14 Sch=led[4]
set_property -dict { PACKAGE_PIN V17 10STANDARD LVCMS33 } [get_ports { SUM[5] }]; #10_L18N_T2_A11_D27_14 Sch=led[5]
set_property -dict { PACKAGE_PIN U17 10STANDARD LVCMS33 } [get_ports { SUM[6] }]; #10_L17P_T2_A14_D30_14 Sch=led[6]
set_property -dict { PACKAGE_PIN U16 10STANDARD LVCMS33 } [get_ports { SUM[7] }]; #10_L18P_T2_A12_D28_14 Sch=led[7]
set_property -dict { PACKAGE_PIN V16 10STANDARD LVCMS33 } [get_ports { SUM[8] }]; #10_L16N_T2_A15_D31_14 Sch=led[8]
set_property -dict { PACKAGE_PIN T15 10STANDARD LVCMS33 } [get_ports { SUM[9] }]; #10_L14N_T2_SRCC_14 Sch=led[9]
set_property -dict { PACKAGE_PIN U14 10STANDARD LVCMS33 } [get_ports { SUM[10] }]; #10_L22P_T3_A05_D21_14 Sch=led[10]
set_property -dict { PACKAGE_PIN T16 10STANDARD LVCMS33 } [get_ports { SUM[11] }]; #10_L15N_T2_DQS_DOUT_CS0_B_14 Sch=led[11]
set_property -dict { PACKAGE_PIN V15 10STANDARD LVCMS33 } [get_ports { SUM[12] }]; #10_L16P_T2_CSI_B_14 Sch=led[12]
set_property -dict { PACKAGE_PIN V14 10STANDARD LVCMS33 } [get_ports { SUM[13] }]; #10_L22N_T3_A04_D20_14 Sch=led[13]
set_property -dict { PACKAGE_PIN V12 10STANDARD LVCMS33 } [get_ports { SUM[14] }]; #10_L20N_T3_A07_D23_14 Sch=led[14]
set_property -dict { PACKAGE_PIN V11 10STANDARD LVCMS33 } [get_ports { SUM[15] }]; #10_L21N_T3_DQS_A06_D22_14 Sch=led[15]

set_property -dict { PACKAGE_PIN R12 10STANDARD LVCMS33 } [get_ports { CO }]; #10_L5P_T0_D06_14 Sch=led16_b
#set_property -dict { PACKAGE_PIN M16 10STANDARD LVCMS33 } [get_ports { LED16_G }]; #10_L10P_T1_D14_14 Sch=led16_g
#set_property -dict { PACKAGE_PIN N15 10STANDARD LVCMS33 } [get_ports { LED16_R }]; #10_L11P_T1_SRCC_14 Sch=led16_r
#set_property -dict { PACKAGE_PIN G14 10STANDARD LVCMS33 } [get_ports { LED17_B }]; #10_L15N_T2_DQS_ADV_B_15 Sch=led17_b
#set_property -dict { PACKAGE_PIN R11 10STANDARD LVCMS33 } [get_ports { LED17_G }]; #10_0_14 Sch=led17_g
#set_property -dict { PACKAGE_PIN N16 10STANDARD LVCMS33 } [get_ports { LED17_R }]; #10_L11N_T1_SRCC_14 Sch=led17_r

##Buttons

#set_property -dict { PACKAGE_PIN C12 10STANDARD LVCMS33 } [get_ports { CPU_RESETN }]; #10_L3P_T0_DQS_AD1P_15 Sch=cpu_resetn

set_property -dict { PACKAGE_PIN N17 10STANDARD LVCMS33 } [get_ports { CI }]; #10_L9P_T1_DQS_14 Sch=btnc
#set_property -dict { PACKAGE_PIN M18 10STANDARD LVCMS33 } [get_ports { BTNU }]; #10_L4N_T0_D05_14 Sch=btnd
set_property -dict { PACKAGE_PIN P17 10STANDARD LVCMS33 } [get_ports { sel }]; #10_L12P_T1_MRCC_14 Sch=btln
#set_property -dict { PACKAGE_PIN M17 10STANDARD LVCMS33 } [get_ports { BTNR }]; #10_L10N_T1_D15_14 Sch=btnr
#set_property -dict { PACKAGE_PIN P18 10STANDARD LVCMS33 } [get_ports { BTND }]; #10_L9N_T1_DQS_D13_14 Sch=btnd
```

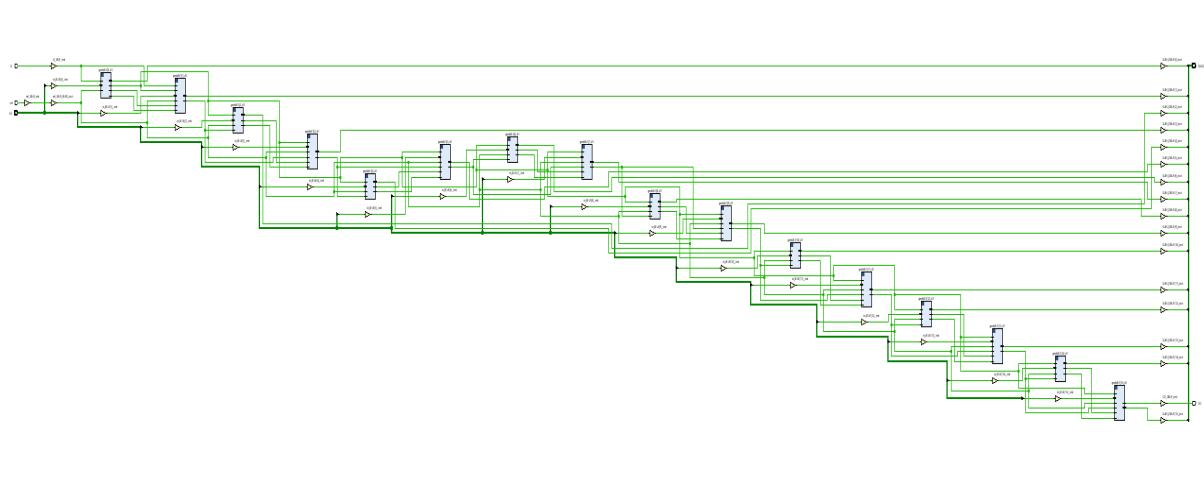
Video link:

<https://youtu.be/9Fy410ECTSY>

or

https://drive.google.com/file/d/1ZTngZoCENQ0sNyndfHRjW6HmGi_g3Yh4/view?usp=sharing

Elaborated Design:



Utilization:

Utilization							
Hierarchy	Name	Slice LUTs (63400)	Slice Registers (126800)	Slice (15850)	LUT as Logic (63400)	Bonded IOB (210)	BUFGCTRL (32)
Summary	top	16	32	28	16	35	1
Slice Logic	genblk1[0].d1 (Demux)	1	2	3	1	0	0
Slice LUTs (<1%)	genblk1[1].d1 (Demux_6)	1	2	2	1	0	0
Slice Registers (<1%)	genblk1[2].d1 (Demux_7)	1	2	2	1	0	0
Register as Latch (<1%)	genblk1[3].d1 (Demux_8)	1	2	3	1	0	0
Slice Logic Distribution	genblk1[4].d1 (Demux_9)	1	2	2	1	0	0
Slice (<1%)	genblk1[5].d1 (Demux_10)	1	2	3	1	0	0
SLICEL	genblk1[6].d1 (Demux_11)	1	2	2	1	0	0
Slice Registers (<1%)	genblk1[7].d1 (Demux_12)	1	2	3	1	0	0
Register driven from ou	genblk1[8].d1 (Demux_13)	1	2	2	1	0	0
LUT in front of the	genblk1[9].d1 (Demux_14)	1	2	2	1	0	0
LUT in front of the	genblk1[10].d1 (Demux_0)	1	2	3	1	0	0
LUT as Logic (<1%)	genblk1[11].d1 (Demux_1)	1	2	2	1	0	0
using O5 and O6<>LUT	genblk1[12].d1 (Demux_2)	1	2	2	1	0	0
using O6 output only<	genblk1[13].d1 (Demux_3)	1	2	2	1	0	0
Memory	genblk1[14].d1 (Demux_4)	1	2	2	1	0	0
DSP	genblk1[15].d1 (Demux_5)	1	2	3	1	0	0
IO and GT Specific							

Power report:

Power analysis from Implemented netlist. Activity derived from constraints files, simulation files or vectorless analysis.

Total On-Chip Power: **16.429 W (Junction temp exceeded!)**

Design Power Budget: **Not Specified**

Power Budget Margin: **N/A**

Junction Temperature: **100.0°C**

Thermal Margin: **-15.0°C (-3.1 W)**

Effective θ_{JA} : **4.6°C/W**

Power supplied to off-chip devices: **0 W**

Confidence level: **Low**

[Launch Power Constraint Advisor](#) to find and fix invalid switching activity

On-Chip Power

