LAB 2 INSTRUCTIONS

Updated for Spring 2021, using Quartus Prime Lite 17.1

First step: Create Project in Quartus

- open CPE324_Lab2_Assignment.pdf
    - Read the intro, and stop after the pre-lab assignment when it comes to Part I
- open Quartus, and click on the New Project Wizard
    - I used c:\projects\cpe324\lab2 as my working directory, with lab2 as the name of my project and top-level, though these names are somewhat arbitrary
    - Create and empty project, and when it comes to adding files, just hit “next”
    - For DE2-115 board users
        - Use the Device tab
        - select Cyclone IV E in the device family
        - find the following device

Available devices:

Name	Core Voltage	LEs	Total I/Os	GPIOs	Memory Bits	Embedded m
EP4CE115F23I7	1.2V	114480	281	281	3981312	532
EP4CE115F23I8L	1.0V	114480	281	281	3981312	532
EP4CE115F29C7	1.2V	114480	529	529	3981312	532

    - For DE10-Lite board users
        - Use the Board tab
        - Find the following board

Available boards:

	Name	Version	Family	Device	Vendor	LE
	Arrow MAX 10 DECA	0.9	MAX 10	10M50DAF484C6GES	Arrow	49760
	BeMicro MAX 10 FPGA Evaluation Kit	1.0	MAX 10	10M08DAF484C8GES	Arrow	8064
	MAX 10 DE10 - Lite	1.0	MAX 10	10M50DAF484C6GES	Altera	49760

    - Click Finish

Second step: Part I of the lab assignment

- WITHOUT FIRST IMPLEMENTING, READ the Background and Design Capture Assignment sections
- Perform the schematic entry, starting at Fig. 5 and working up to Fig. 2.

- After all the schematic files are created (eight_bit_sub_add), go to PROCESSING, and start COMPILATION (check for errors)
- Open your toplevel Verilog file
    - For DE2-115 users, this will be called lab2.v and you can find it in your Project Navigator tab, whether in Hierarchy view or in Files view
        - Add the following text to create one instance of eight_bit_sub_add

```
// simple I/O - only connecting SW and LEDG pins on DE2
module lab2 (
  input [17:0] SW,
  output [8:0] LEDG
);

// one instance,
eight_bit_sub_add lab2_uut (
 .A (SW[7:0]),
 .B (SW[15:8]),
 .B_CIN (SW[16]),
 .SUB_ADD (SW[17]),
 .D_S (LEDG[7:0]),
 .B_COUT (LEDG[8])
);
```

    - For DE10-Lite users, this will be called DE10_LITE_Golden_Top based on the board files
        - Comment-out all but the following `define lines: ENABLE_KEY, ENABLE_LED, ENABLE_SW, ENABLE_GPIO:

```
//`define ENABLE_ADC_CLOCK
//`define ENABLE_CLOCK1
//`define ENABLE_CLOCK2
//`define ENABLE_SDRAM
//`define ENABLE_HEX0
//`define ENABLE_HEX1
//`define ENABLE_HEX2
//`define ENABLE_HEX3
//`define ENABLE_HEX4
//`define ENABLE_HEX5
`define ENABLE_KEY
`define ENABLE_LED
`define ENABLE_SW
//`define ENABLE_VGA
//`define ENABLE_ACCELEROMETER
//`define ENABLE_ARDUINO
`define ENABLE_GPIO
```

        - Right before the endmodule line, add the following text to create one instance of eight_bit_sub_add
        - Note that the {4{SW[0]}} operation makes 4 copies of the SW[0] input
        - Note that {SW[4:1],{4{SW[0]}}} line concatenates the 4 bits from [4:1] of the SW switch inputs bus to the replicated-4-times SW[0] input
        - Note that we didn't have enough switches on the DE10-Lite to make A and B 16 unique inputs, along with B_CIN and SUB_ADD, so we've made their 4 LSbits either 4'b1111 or 4'b0000, depending on the SW[0] and SW[5] switches

```
eight_bit_sub_add lab2_uut (
 .A ({SW[4:1],{4{SW[0]}}}),
 .B ({SW[9:6],{4{SW[5]}}}),
 .B_CIN (KEY[0]),
 .SUB_ADD (KEY[1]),
 .D_S (LEDR[7:0]),
 .B_COUT (LEDR[8])
);
```

- For DE2-115 users, we need to create the pin location assignments (i.e. tell Quartus which pins are connected to the switches, and which ones are connected to the LEDS)
    - There are 2 alternatives for doing this. One uses the GUI called "Pin Planner" and the other allows you to copy and paste it into your project settings file. You only need to do one or the other:
        - Copy and paste option:
            - Go to File > Open, and change the file filter to All Files (*.*)
            - Select lab2.qsf (or whatever you chose to name the project)
            - Insert the following lines anywhere (somewhere after the set_global_assignment instructions is recommended)

```
set_location_assignment PIN_AB26 -to SW[17]
set_location_assignment PIN_AD26 -to SW[16]
set_location_assignment PIN_AC26 -to SW[15]
set_location_assignment PIN_AB27 -to SW[14]
set_location_assignment PIN_AD27 -to SW[13]
set_location_assignment PIN_AC27 -to SW[12]
set_location_assignment PIN_AC28 -to SW[11]
set_location_assignment PIN_AB28 -to SW[10]
set_location_assignment PIN_AA22 -to SW[9]
set_location_assignment PIN_AA23 -to SW[8]
set_location_assignment PIN_AA24 -to SW[7]
set_location_assignment PIN_AB23 -to SW[6]
set_location_assignment PIN_AB24 -to SW[5]
set_location_assignment PIN_AC24 -to SW[4]
set_location_assignment PIN_AB25 -to SW[3]
set_location_assignment PIN_AC25 -to SW[2]
set_location_assignment PIN_Y24 -to SW[1]
set_location_assignment PIN_Y23 -to SW[0]
set_location_assignment PIN_F17 -to LEDG[8]
set_location_assignment PIN_G21 -to LEDG[7]
set_location_assignment PIN_G22 -to LEDG[6]
set_location_assignment PIN_G20 -to LEDG[5]
set_location_assignment PIN_H21 -to LEDG[4]
set_location_assignment PIN_E24 -to LEDG[3]
set_location_assignment PIN_E25 -to LEDG[2]
set_location_assignment PIN_E22 -to LEDG[1]
set_location_assignment PIN_E21 -to LEDG[0]
```

        - Pin Planner option:
            - Go to Assignments > Pin Planner, and select every location cell, filling in the above info (or the table on page 4 of the lab assignment pdf)
- COMPILE again.
- To download your file, go to TOOLS, PROGRAMMER.

- Select HARDWARE SETUP and choose the USB Blaster
    - You'll need to have your driver up-to-date for this
- You should see your device, if not hit "Auto Detect"
- You should see the lab2.sof (or <project name>.sof) file
    - If not, double click the file generated after selecting E115, go to OUTPUT files, select your top-level module.
- Click START
- After this step downloading is complete. Now check all the conditions mentioned in the lab manual.
    - For DE10-Lite users, keep in mind the following as you perform your demonstration
        - The KEY0 and KEY1 pushbuttons are connected to B_CIN and SUB_ADD inputs. When you press KEY0 down, B_CIN is 0 but if you aren't pressing it the value is 1. When you press KEY1 down, SUB_ADD is 0 (ADD), but if you aren't pressing it the value is 1 (SUBTRACT).
        - Again, A[7:0] is connected to {SW[4],SW[3],SW[2],SW[1],SW[0], SW[0], SW[0], SW[0]} ({a,b,c} is Verilog syntax for concatenation of a, b, and c from left to right). B is similarly connected, repeating SW[5] 4 times in the LSBits

Third step: Part II of the lab assignment

- Repeat the project creation, except changing project name
- Go to File, New, and select Verilog HDL
- Copy/Paste the Verilog code from the Lab2 Assignment PDF into your top-level file (same top-level file as you added Verilog to in the prior lab)
- Repeat the steps you used for pin assignment and for editing the top-level file to create and connect one instance of eight_bit_add_sub
- Repeat the steps you used for compilation and downloading