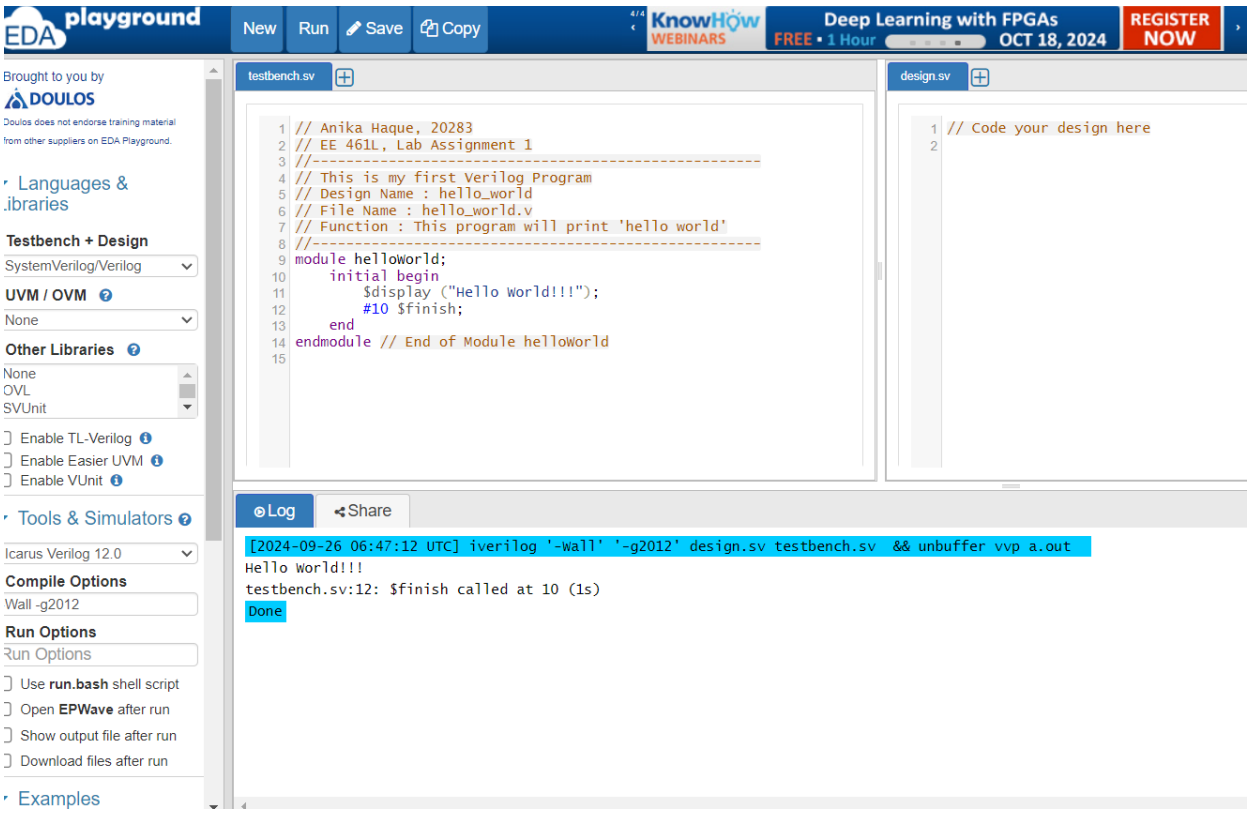
# San Francisco Bay University
## EE461L - Verilog HDL Lab Assignment 1

## Week#1 Introduction to Verilog and Combinational & Sequential Logic Design

## Example "Hello World" & Simulation

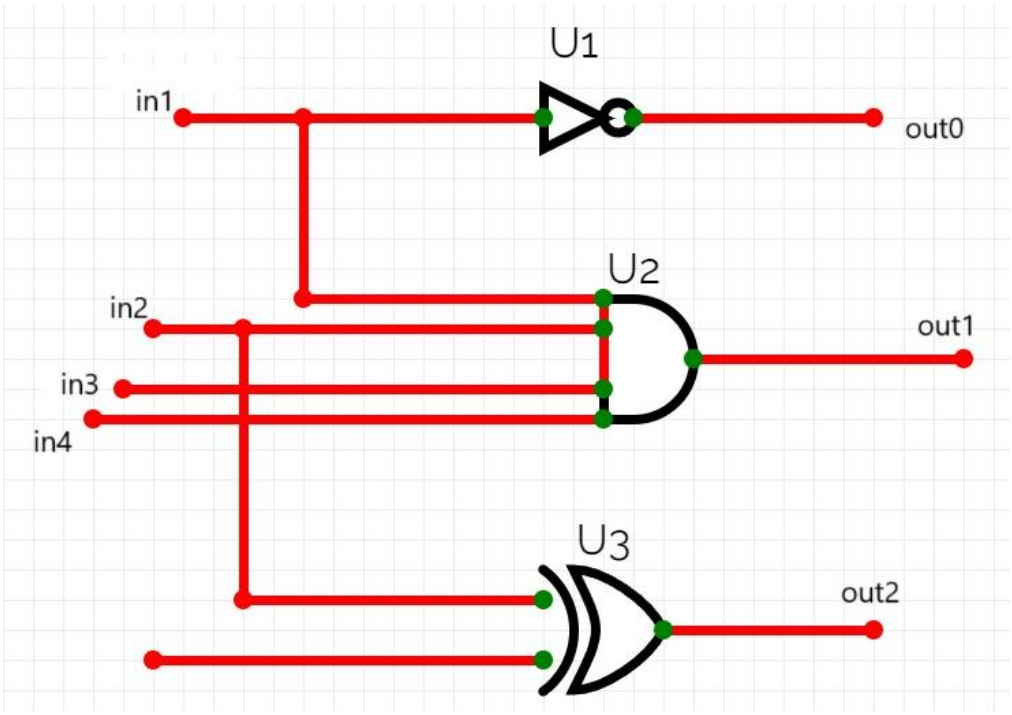
- Directly run the module `helloWorld` in the online compiler



## Based on the module *gates*, draw the digital circuit schematic.

```
module gates();
 wire out0;
 wire out1;
 wire out2;
 reg  in1,in2,in3,in4;

 not U1(out0,in1);
 and U2(out1,in1,in2,in3,in4);
 xor U3(out2,in1,in2,in3);
endmodule
```



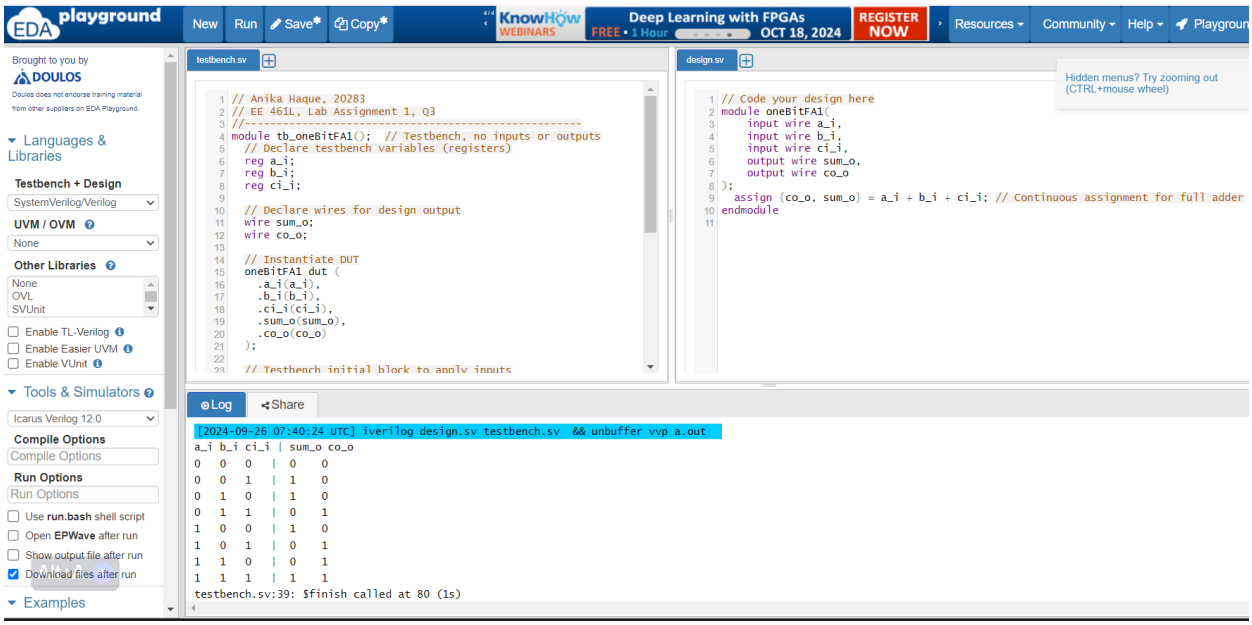
# Combinational & Sequential Logic in RTL

- Create the testbench first, and then simulate the design modules *oneBitFA1*, *oneBitFA2*, *DFFSynch*, and *DFFAsynch*. *Note: Ignore creating the testbenches for module *DFFSynch*, and *DFFAsynch*

## Combinational Logic in RTL

1. "continuous assignment" in `oneBitFA1.v`

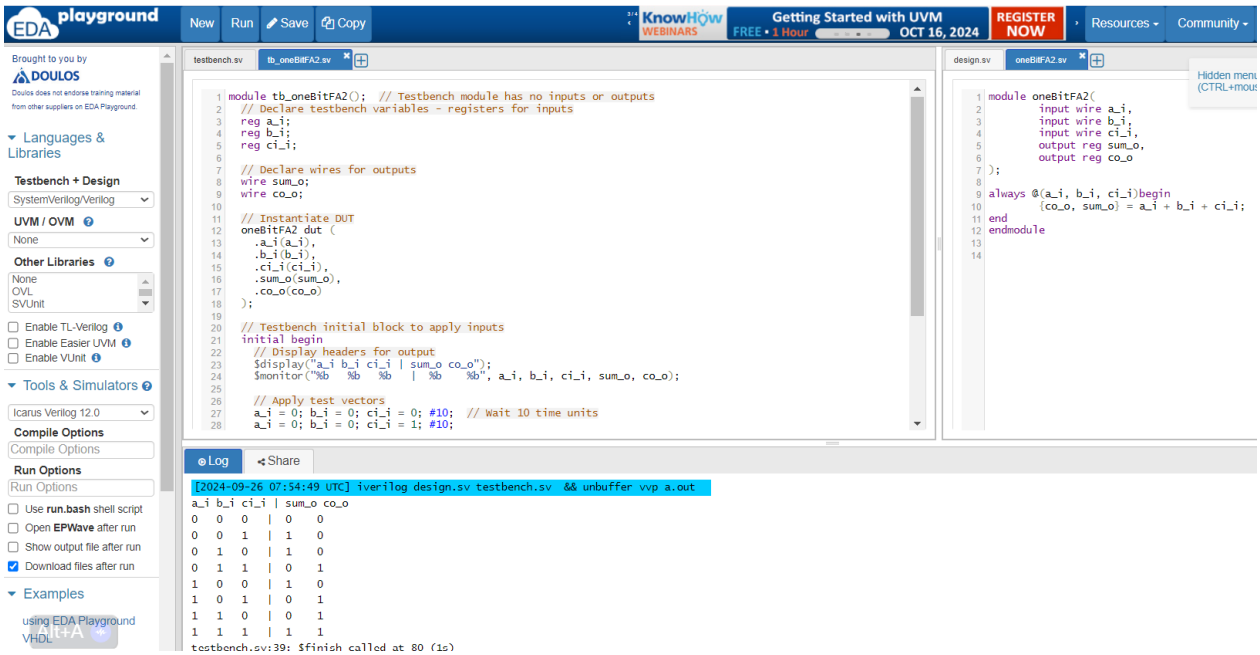
```
module oneBitFA1(
                input wire a_i,
                input wire b_i,
                input wire ci_i,
                output wire sum_o,
                output wire co_o
);
assign   {co_o,sum_o} = a_i + b_i + ci_i;
endmodule
```



2. “always block” in `oneBitFA2.v`

```
module oneBitFA2(
            input wire a_i,
            input wire b_i,
            input wire ci_i,
            output reg sum_o,
            output reg co_o
);

always @(a_i, b_i, ci_i)begin
            {co_o, sum_o} = a_i + b_i + ci_i;
end
endmodule
```



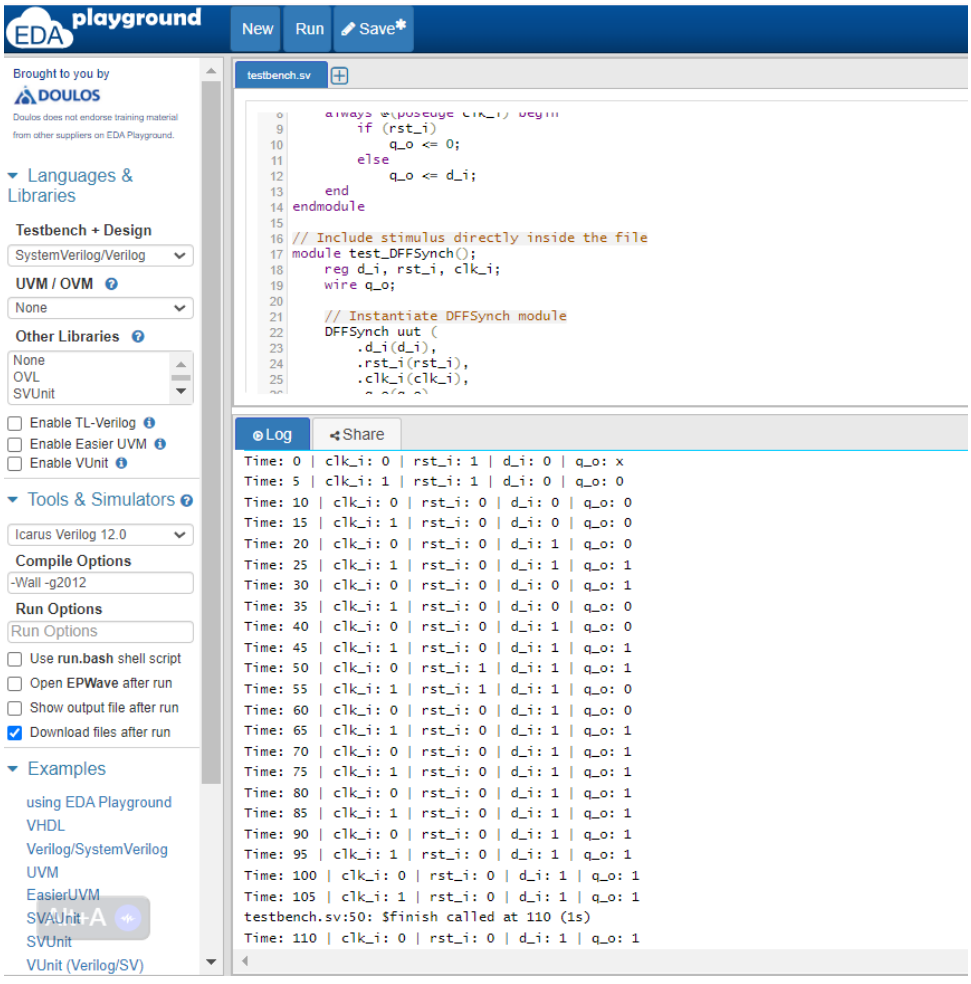
# Sequential Logic in RTL

1. Synchronous DFF:

```
module DFFSynch(
                d_i,
                rst_i,
                clk_i,
                q_o
);
        input d_i,rst_i,clk_i;
        output q_o;

        reg q_o;

      always @(posedge clk_i)begin
              if(rst_i) q_o <= 0;
              else q_o <= d_i;
        end
endmodule
```



2. Asynchronous DFF:

```
module DFFAsynch(
            d_i,
            rst_i,
            clk_i,
            q_o
);
      input d_i,rst_i,clk_i;
      output q_o;

      reg q_o;

      always @(posedge clk_i or posedge rst_i)begin
            if(rst_i) q_o <= 0;
            else q_o <= d_i;
      end
endmodule
```