Assignment 2

Ashmitha Murali

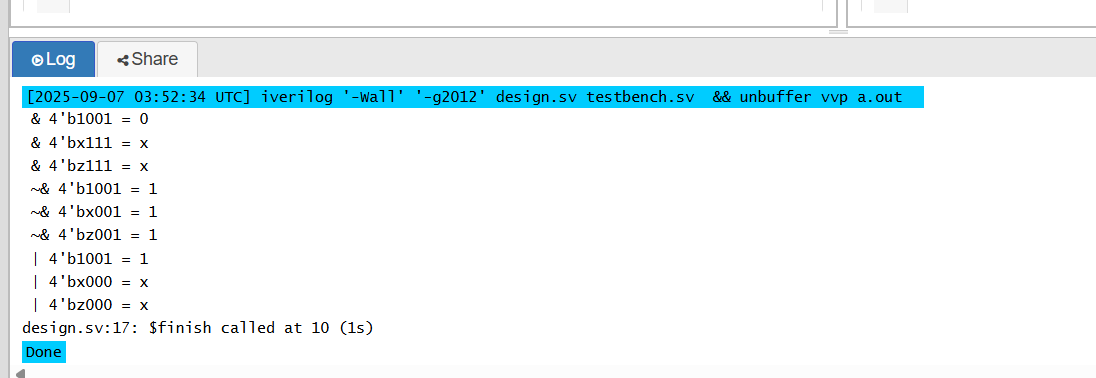
Prof. Alex Yang

Verilog HDL Lab E461L

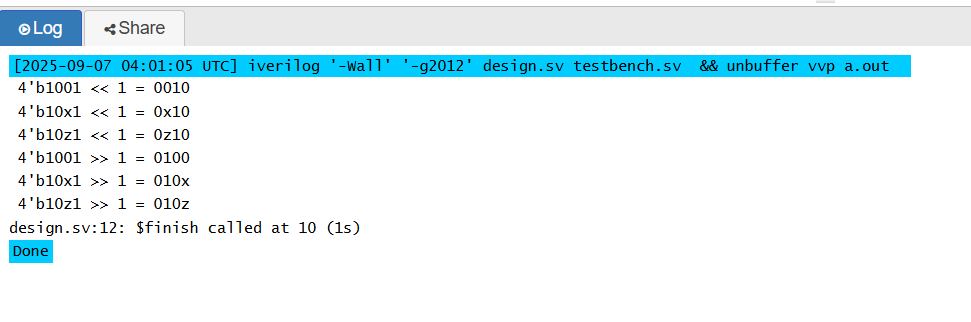
8 September 2025

1. Run module ReductionOperators, ShiftOperators, Concatenation Operator and  
   ReplicationOperator

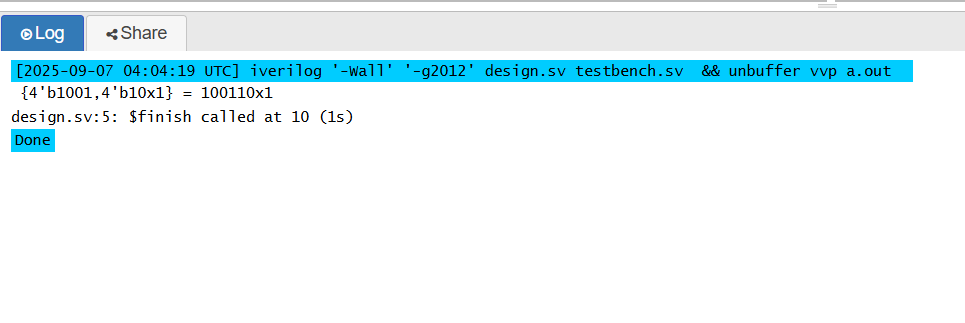
* **Reduction Operators**



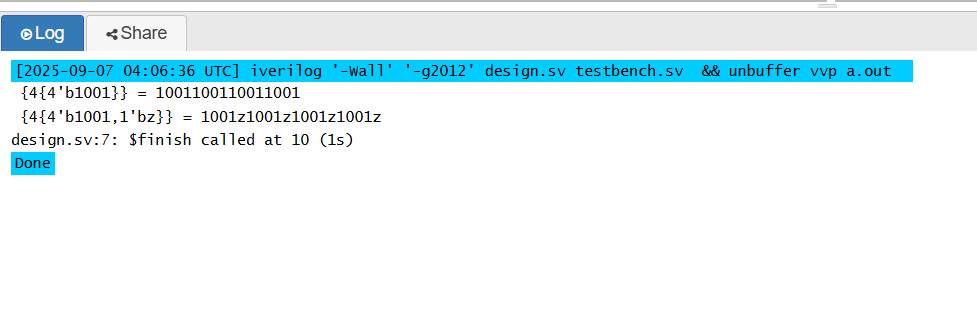
* **Shift Operators**



* **Concatenation Operator**



* **Replication Operator**



1. Complete the module addbit first, and then run module adder\_hier and its  
   testbench.

* addbit module

module addbit(

r1,

r2,

ci,

sum,

carry\_op

);

input r1;

input r2;

input ci;

output sum;

output carry\_op;

wire r1;

wire r2 ;

wire ci ;

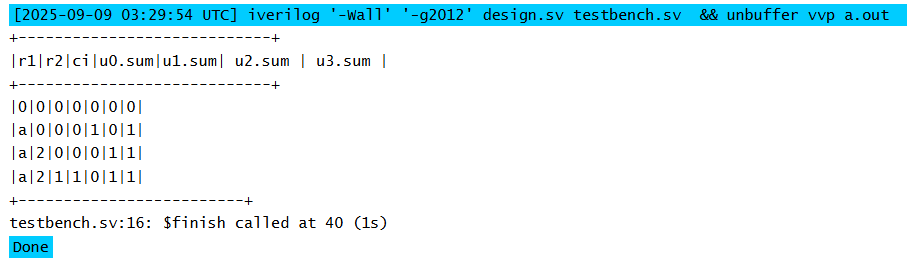
reg sum ;

reg carry\_op ;

assign {carry\_op,sum} = r1+r2+ci;

endmodule

**Output result:**



3. Based on the module adder\_hier, draw the circuit functional block diagram

One bit adder

U3

One bit adder

U2

One bit adder

U1

One bit adder

U0

**C3**

**C2**

**C1**

**Sum[0]**

**Sum[1]**

**Sum[2]**

**Sum[3]**

**C\_out**

**r1[0]**

**r2[0]**

**c\_in**

**c1**

**r1[1]**

**r2[1]**

**c2**

**r1[2]**

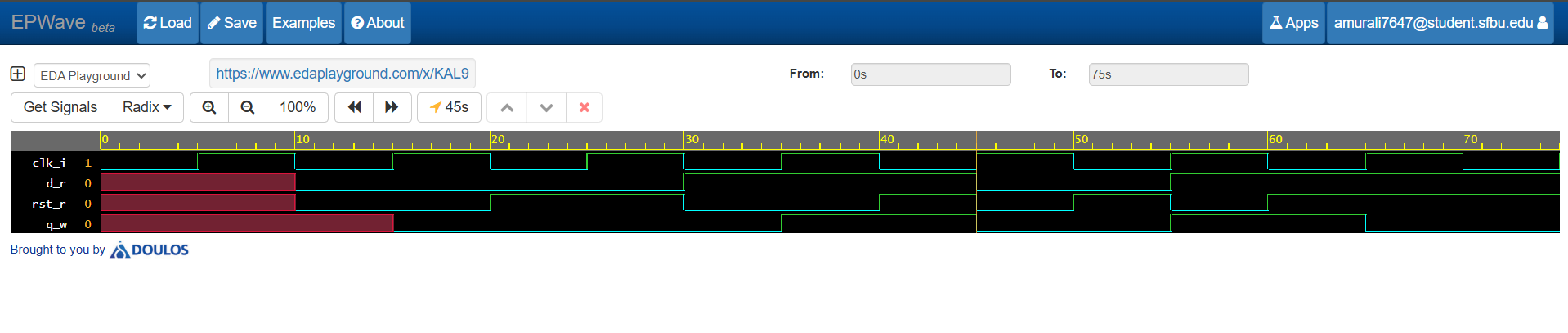
**r2[2]**

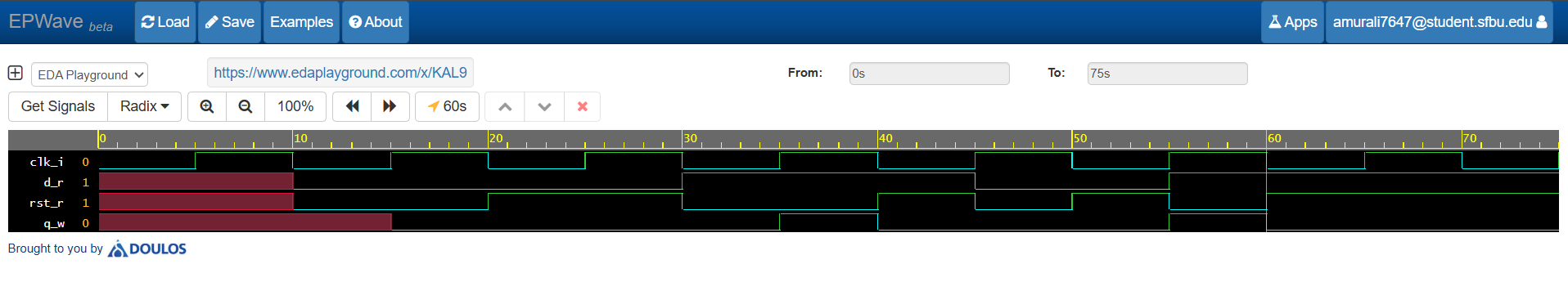
**c3**

**r1[3]**

**r2[3]**

4. Complete the testbenches for the modules DFFSynch, and DFFAsynch in Lab#1

* **DFF Sync**
* **DFF Async**



**Testbench for DFF:**

module tb\_DFFSync;

//Input to design from TB

reg d\_r;

reg rst\_r;

reg clk\_i;

//Output signals from design to TB

wire q\_w;

//Instantiate the design

DFFAsynch DFFSynch2 (

.d\_i(d\_r),

.rst\_i(rst\_r),

.clk\_i(clk\_i),

.q\_o(q\_w)

);

initial begin

//Test cases

#10 d\_r = 0; rst\_r = 0;

#10 d\_r = 0; rst\_r = 1;

#10 d\_r = 1; rst\_r = 0;

#10 d\_r = 1; rst\_r = 1;

#5 d\_r = 0; rst\_r = 0;

#5 d\_r= 0; rst\_r = 1;

#5 d\_r= 1; rst\_r = 0;

#5 d\_r = 1; rst\_r = 1;

end

initial begin

$dumpfile("dump.vcd");

$dumpvars(1); //dump all variables in testbench & design

end

initial begin

clk\_i = 0;

repeat(15) #5 clk\_i = ~clk\_i;

end

endmodule