DDCO Assignment

Week 2
MUX 2:1, MUX 4:1 and Ripple Carry Adder

Name : Renita Kurian

SRN : PES1UG20CS331

Roll No. : 27

Section : F

1. MUX 2:1

```
module MUX2(input wire A,B,S, output wire Y);
assign Y=(S==0)?A:B;
endmodule
```

Testbench file –

```
module TB;
reg A,B,S;
wire X;
initial
begin
$dumpfile("dump.vcd");
$dumpvars(0,TB);
end
MUX2 newMUX(.A(A), .B(B), .S(S), .Y(X));
initial
begin
S = 1'b0;
A = 1'b0;
B = 1'b0;
#5
A = 1'b0;
B = 1'b1;
#5
A = 1'b1;
B = 1'b0;
#5
A = 1'b1;
B = 1'b1;
#5
S = 1'b0;
A = 1'b0;
B = 1'b0;
#5
A = 1'b0;
B = 1'b1;
#5
A = 1'b1;
B = 1'b0;
#5
A = 1'b1;
B = 1'b1;
end
endmodule
```

Output –

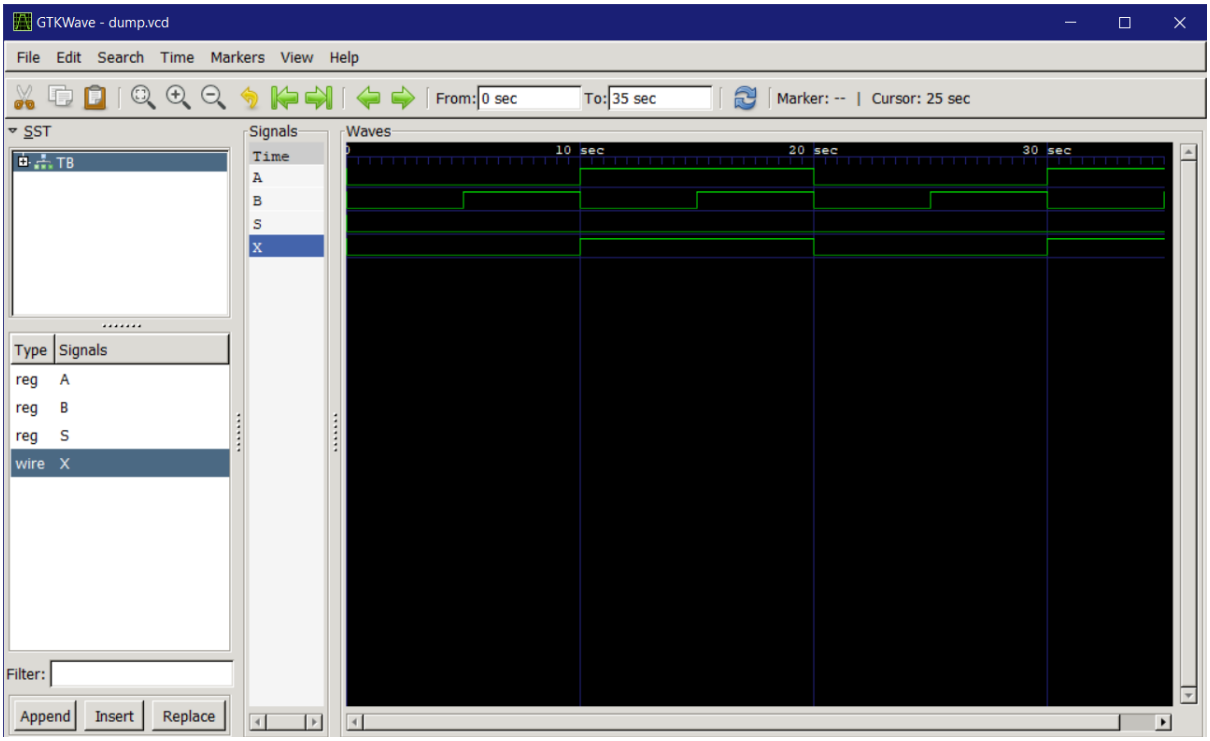
```
C:\PES1UG20CS331\iverilog\bin>iverilog -o m2 MUX2.v TB_MUX2.v

C:\PES1UG20CS331\iverilog\bin>vvp m2
VCD info: dumpfile dump.vcd opened for output.

C:\PES1UG20CS331\iverilog\bin>gtkwave dump.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[35] end time.
WM Destroy
```



2. MUX 4:1

```
module MUX4 (input wire [0:3] i, input wire j1, j0, output wire o);
wire x,y;
MUX2 M1(i[0],i[1],j1,x);
MUX2 M2(i[2],i[3],j1,y);
MUX2 M3(x,y,j0,o);
endmodule
```

Testbench file –

```
module TB;
reg [0:3]ii;
reg s0;
reg s1;
wire yy;
initial
begin
$dumpfile("dump.vcd");
$dumpvars(0, TB);
end
MUX4 newMUX(.i(ii), .j0(s0),.j1(s1),.o(yy));
initial
begin
ii = 4'b0001;
s0=1'b0;
s1=1'b0;
#5
ii = 4'b1000;
#5
ii = 4'b0001;
s0=1'b1;
s1=1'b1;
#5
ii = 4'b0000;
s0=1'b1;
s1=1'b0;
end
endmodule
```

Output –

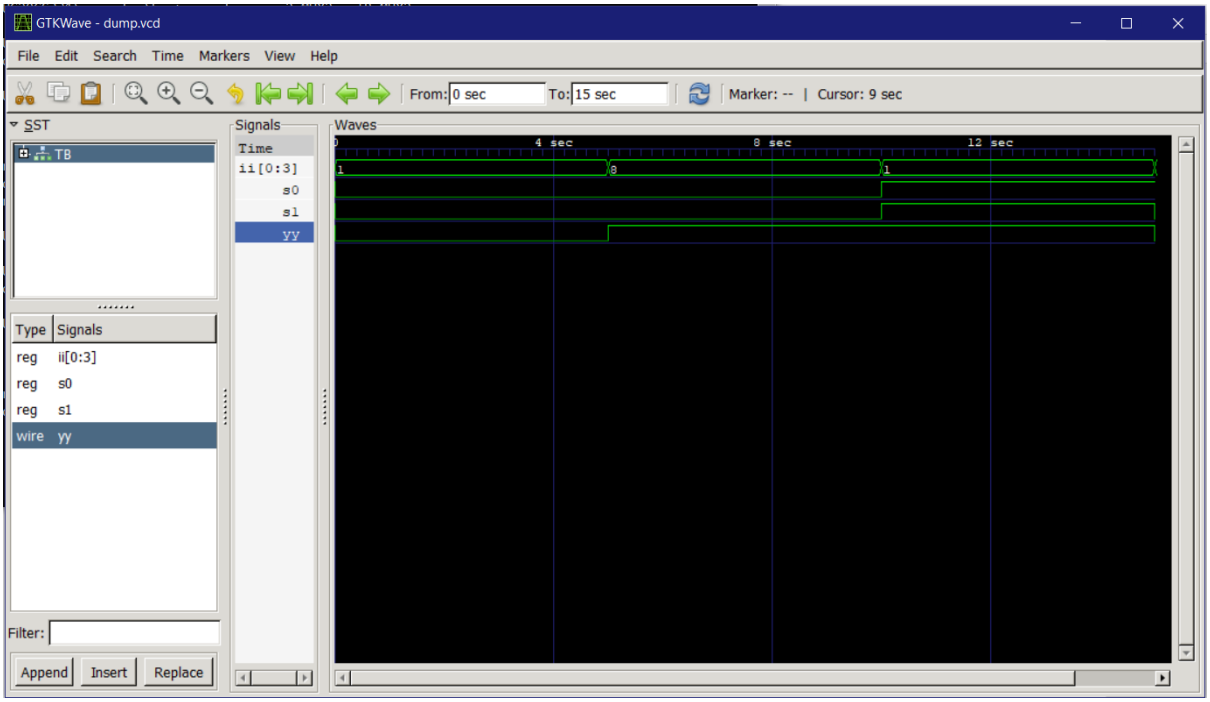
```
C:\PES1UG20CS331\iverilog\bin>iverilog -o m4 MUX2.v MUX4.v TB_MUX4.v

C:\PES1UG20CS331\iverilog\bin>vvp m4
VCD info: dumpfile dump.vcd opened for output.

C:\PES1UG20CS331\iverilog\bin>gtkwave dump.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[15] end time.
WM Destroy
```



3. Ripple Carry Adder

```
module RCA(input wire [3:0] a, b, input wire cin, output wire
wire [2:0]c;
fulladd f1(a[0],b[0],cin,sum[0],c[0]);
fulladd f2(a[1],b[1],c[0],sum[1],c[1]);
fulladd f3(a[2],b[2],c[1],sum[2],c[2]);
fulladd f4(a[3],b[3],c[2],sum[3],cout);
endmodule
```

Full adder module –

```
module fulladd(input wire a, b, cin, output wire sum, cout);
wire [4:0] t;
    xor1 x0(a, b, t[0]);
    xor1 x1(t[0], cin, sum);

    and1 a0(a, b, t[1]);
    and1 a1(a, cin, t[2]);
    and1 a2(b, cin, t[3]);

    or1 o0(t[1], t[2], t[4]);
    or1 o1(t[3], t[4], cout);
endmodule
```

Gates module –

```
module and1(input wire A,B, output wire Y);
assign Y = A & B;
endmodule

module or1(input wire A,B, output wire Y);
assign Y = A | B;
endmodule

module not1(input wire A, output wire Y);
assign Y=~A;
endmodule

module nand1(input wire A,B, output wire Y);
wire x;
and1 a1(A,B,x);
not1 a2(x,Y);
endmodule


module nor1(input wire A,B, output wire Y);
wire x;
or1 a1(A,B,x);
not1 a2(x,Y);
endmodule


module xor1(input wire A,B, output wire C);
wire x,y,p,q;
not1 n1(A,x);
not1 n2(B,y);
and1 a1(B,x,p);
and1 a2(A,y,q);
or1 o1(p,q,C);
endmodule
```

Testbench file –

```
`timescale 1 ns / 100 ps
`define TESTVECS 8

module tb;
  reg clk, reset;
  reg [3:0] i0, i1; reg cin;
  wire [3:0] o; wire cout;
  reg [8:0] test_vecs [0:(`TESTVECS-1)];
  integer i;
  initial begin $dumpfile("dump.vcd");
 $dumpvars(0,tb);
  end
  initial begin reset = 1'b1; #12.5 reset = 1'b0; end
  initial clk = 1'b0; always #5 clk =~ clk;
  initial begin
    test_vecs[0] = 9'b000000010;
    test_vecs[1] = 9'b000100010;
    test_vecs[2] = 9'b011100010;
    test_vecs[3] = 9'b000001110;
    test_vecs[4] = 9'b011001111;
    test_vecs[5] = 9'b001110011;
    test_vecs[6] = 9'b111100011;
    test_vecs[7] = 9'b011101110;
  end
  initial {i0, i1, cin, i} = 0;
  RCA u0 (i0, i1, cin, o, cout);
  initial begin
    #6 for(i=0;i<`TESTVECS;i=i+1)
      begin #10 {i0, i1, cin}=test_vecs[i]; end
    #100 $finish;
  end
endmodule
```

Output –

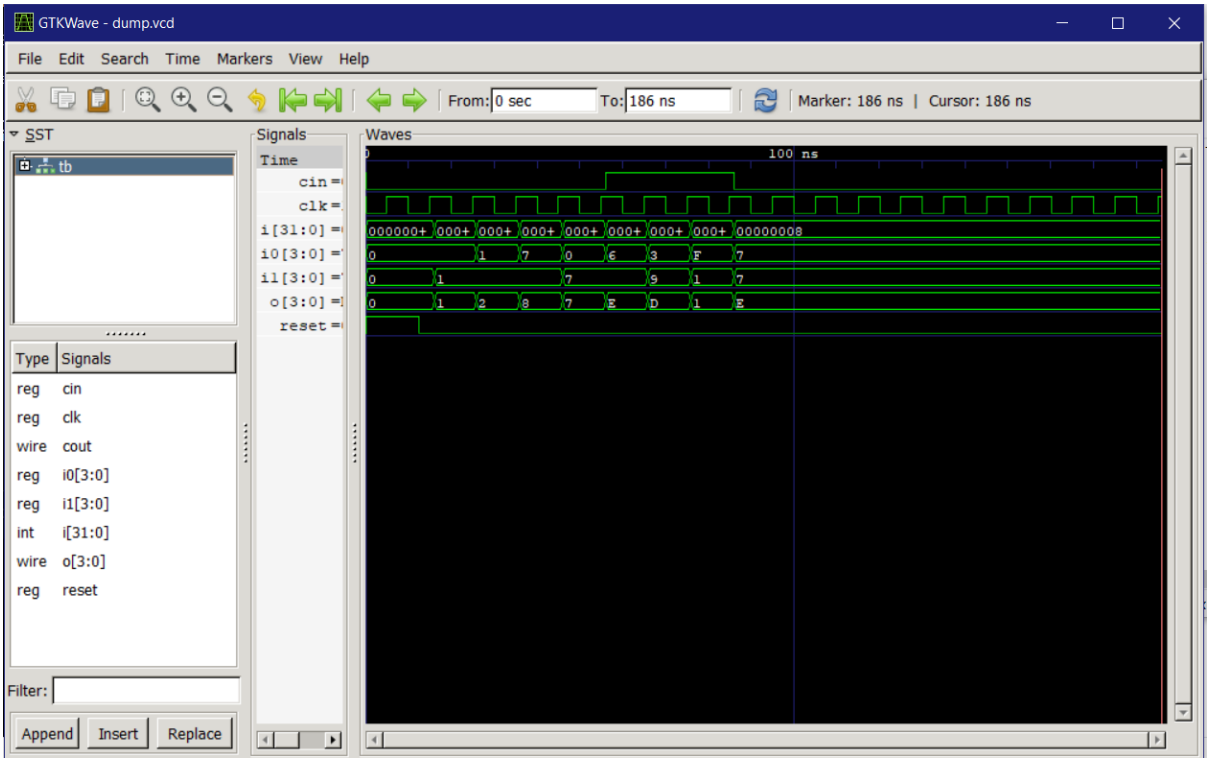
```
C:\PES1UG20CS331\iverilog\bin>iverilog -o rca gates1.v fulladd.v RCA.v TB_RCA.v

C:\PES1UG20CS331\iverilog\bin>vvp rca
VCD info: dumpfile dump.vcd opened for output.

C:\PES1UG20CS331\iverilog\bin>gtkwave dump.vcd

GTKWave Analyzer v3.3.48 (w)1999-2013 BSI

[0] start time.
[186000] end time.
WM Destroy
```


GTKWave - dump.vcd
File Edit Search Time Markers View Help
From: 0 sec
To: 186 ns
Marker: 186 ns | Cursor: 186 ns
SST
tb
Type Signals
reg cin
reg clk
wire cout
reg i0[3:0]
reg i1[3:0]
int i[31:0]
wire o[3:0]
reg reset
Filter:
Append Insert Replace
Signals
Time
cin
clk
i[31:0]
i0[3:0]
i1[3:0]
o[3:0]
reset
Waves
100 ns