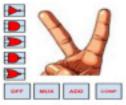


### LUCID VLSI

Learning Unlimited. Coding InDelible.

```
module memory rtl
(clk, reset, wr, rd, addr, wdata, rdata, response);
//Synchronous write read memory
parameter reg [15:0] ADDR WIDTH=8;
parameter reg [15:0] DATA WIDTH=32;
parameter reg [15:0] MEM SIZE=16;
      clk, reset;
input
input wr; // for write wr=1;
input rd; // for read rd=1;
input [ADDR WIDTH-1:0] addr;
                                     raja@lucidvlsi.com
input [DATA WIDTH-1:0] wdata;
output [DATA WIDTH-1:0] rdata;
                                Mobile: 994 995 4576
output response;
        [DATA WIDTH-1:0] rdata;
        [DATA WIDTH-1:0] mem [MEM SIZE];
req
reg
        [DATA WIDTH-1:0] data out;
reg response ; //Provides response to master on successful
write
reg out enable; //controls when to pass read data on rdata
//if rd=0 rdata should be in high impedance state
//if rd=1 rdata should be content of memory with given
address
assign rdata = out_enable \ ? data_out D'bz; UCIDVLS
//asynchronous reset and synchronous write always @(posedge clk or posedge reset)
                                     Development Program
  if (reset) begin ()
        for(int i=0;i<MEM SIZE;i++)
            mem[i] \leftarrow 'b0;
     end
   else if(wr ) begin
            mem[addr] <= wdata ;
             response <=1'b1;
            end
      else response <=1'b0;
end//end of write
```



# LUCID VLSI

Learning Unlimited. Coding InDelible.

```
//Synchronous Read
always @(posedge clk)
if (reset) begin
            data out <= 'b0;
            out enable<=1'b0;
            end
else
    begin
     if (rd==1) begin
          data out <= mem[addr[3:0]] ;
                               raja@lucidvlsi.com
          out enable <= 1'b1;
                           Mobile: 994 995 4576
         out_enable <=1'b0;
end//end_of_read
endmodule/****************
File Name:
         ./2x1Mux/dut.sv
Author : Raja Bandi
Organization: Lucid VLSI
Email ID: raja@lucidvlsi.com
Phone No: +91 994 995 4576
module mux (y,s,i1,i0);
input s, i1, i0;
                www.youtube.com/LucidVLSI
output y;
always @(s, i1, i0)
```

Veriegy; VHDL. SystemVerilog. UVM. FPGA.

Studen //always @(s,i1,i0)
//always @(s or i1 or i0) Faculty Development Program
//always @(\*)
//always\_comb
begin
if (s==0) y = 1'b1;
else y = i1;
end

endmodule

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



File Name: ./2x1Mux/run.do

Email ID: raja@lucidvlsi.com Phone No: +91 994 995 4576

Organization: Lucid VLSI

Author : Raja Bandi

vlib work vdel -all vlib work

vlog tb.sv +acc

vsim work.tb add wave -r \*

run -all

vlog dut.sv +acc

1010

1101

Author : Raja Bandi \*\*\*\*\*\*\*\*\*\*

module tb: reg s, i1, i0; logic y; reg Yexp; integer errcnt=0;

integer matched count=0;

raja@lucidvlsi.com Mobile: 994 995 4576

Unlimited. Coding InDelible.

www.youtube.com/LucidVLSI

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* File Name: /./2x1Mux/tb.sv emVerilog, UVM, FPGA.

Organization: Lucid VLSI ulty Development Program Studen Email ID: raja@lucidvlsi.com Phone No: +91 994 995 4576



# LUCID VLSI

Learning Unlimited. Coding InDelible.

```
mux mux inst
( //port by name
.i0(i0),
.y(y),
.i1(i1),
.s (s)
reg [3:0] mem[1:8];
integer j;
int golden [$];
//Queue
                                  raja@lucidvlsi.com
//push front
                             Mobile: 994 995 4576
initial
begin
    $readmemb("mem.txt",mem);
    repeat (200000)
         begin
              {s,i1,i0}=$random;
             #1;
             generate_golden_result;
             compare; //Lin32 executes 2 lakhs
                             ited. Coding InDelible.
end
                 www.youtube.com/LucidVLSI
task generate golden result;
begin.
                        stemVerilog, UVM, FPGA.
    for (j=1; j<=8; j=j+1)
         begin
            if (mem[j][3:1]=={s,i1,i0}) Yexp=mem[j][0];
         end
end
endtask
task compare;
begin
         if (Yexp != y) //checking
```

begin

errcnt=errcnt+1;

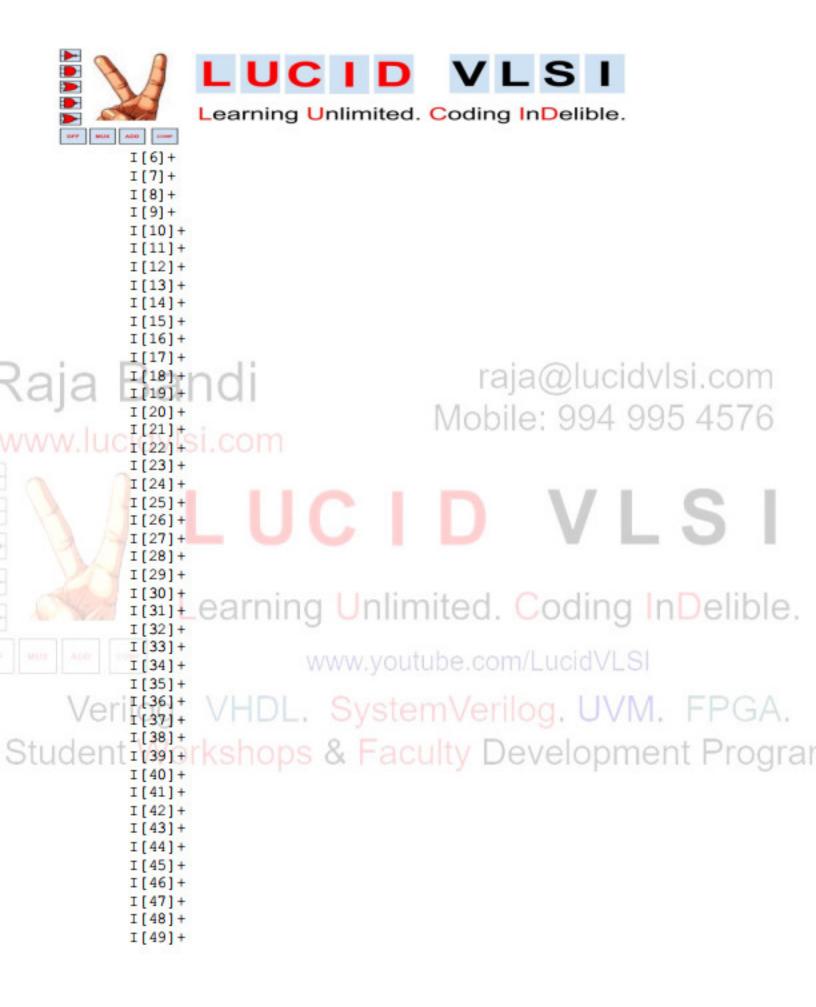


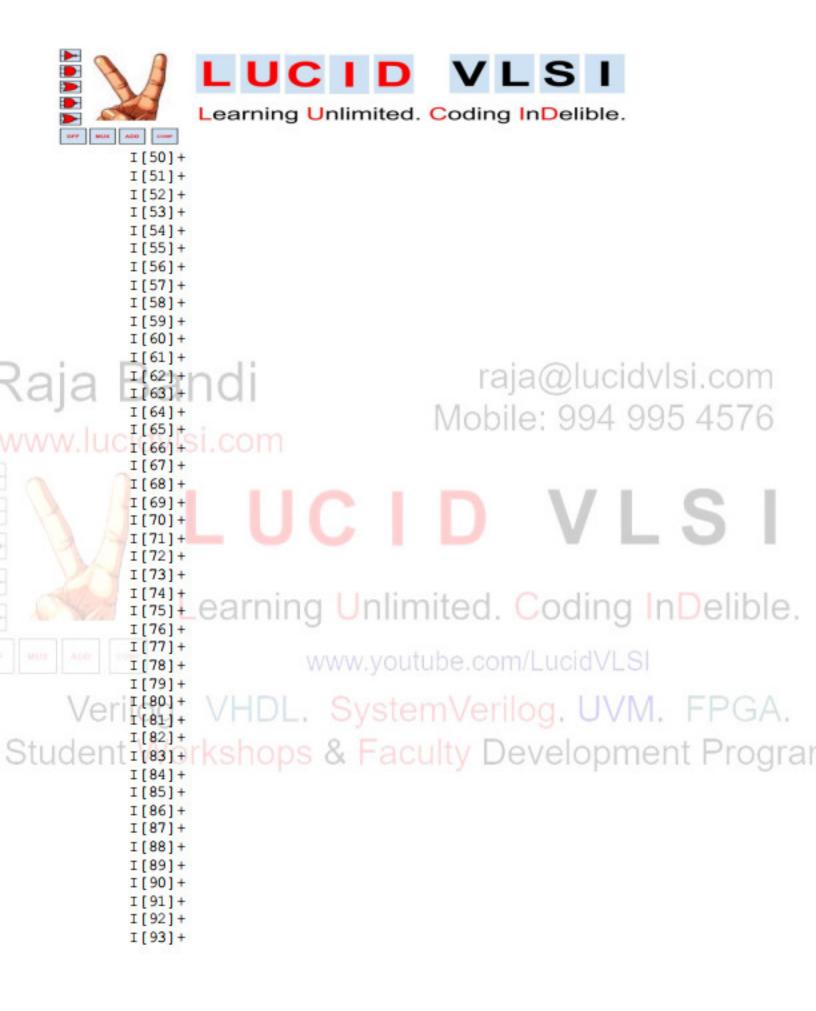
I[5]+

### LUCID VLSI

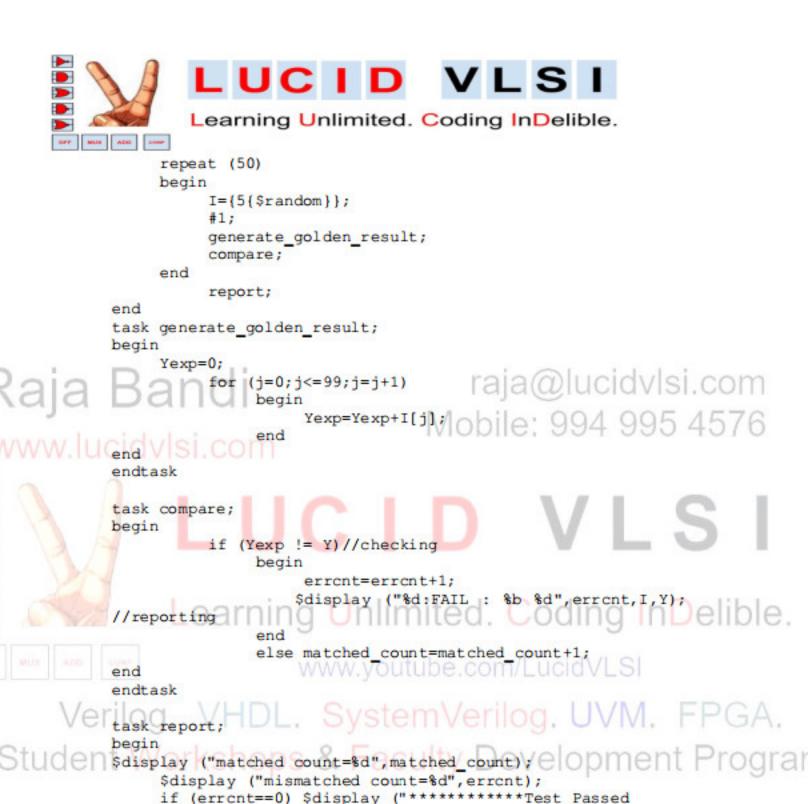
Learning Unlimited. Coding InDelible.

```
$display
 ("%d:FAIL: %b%b%b %b",errcnt,s,i1,i0,y);
                   $finish://
              //reporting
              end
              else matched count=matched count+1;
end
endtask
final
begin
     $display ("I am in the final Block");
    $display ("Please go to LIC office and get money");/ S
                               Mobile: 994 995 4576
task report;
begin
$display ("matched count=%d", matched_count);
     $display ("mismatched count=%d", errcnt);
     if (errcnt==0) $display ("*********Test Passed
     else $display ("* *Test FAILED : FIX THE BUGS : PRAY
GOD *******");
end
endtask
endmodulearning Unlimited. Coding In Delible.
File Name: ./Counting1s_100bits/counting1s.v/C/U/LS
Author : Raja Bandi
Organization: Lucid VLSI
Email ID: raja@lucidvlsi.com
                                      ilog. UVM. FPGA.
Phone No: +91 994 995 4576
                                     Development Progran
**********
module countingls (I,Y);//37800 37800
output wire [6:0] Y;
input [99:0] I;
assign Y =
 I[0]+
 I[1]+
 I[2] +
 I[3]+
 I[4] +
```









else \$display ("\*\*Test FAILED : FIX THE BUGS : PRAY

\*\*\*\*\*\*\*\*\*\*\*\*\*

GOD \*\*\*\*\*\*");

end endtask endmodule