|1BM20AI008 | ALAKA BALAJI VEMBAR |

|1BM20AI011 | ANUSHKA SHIVKUMAR |

Round Robin Arbiter implemented in Verilog

testbench.v

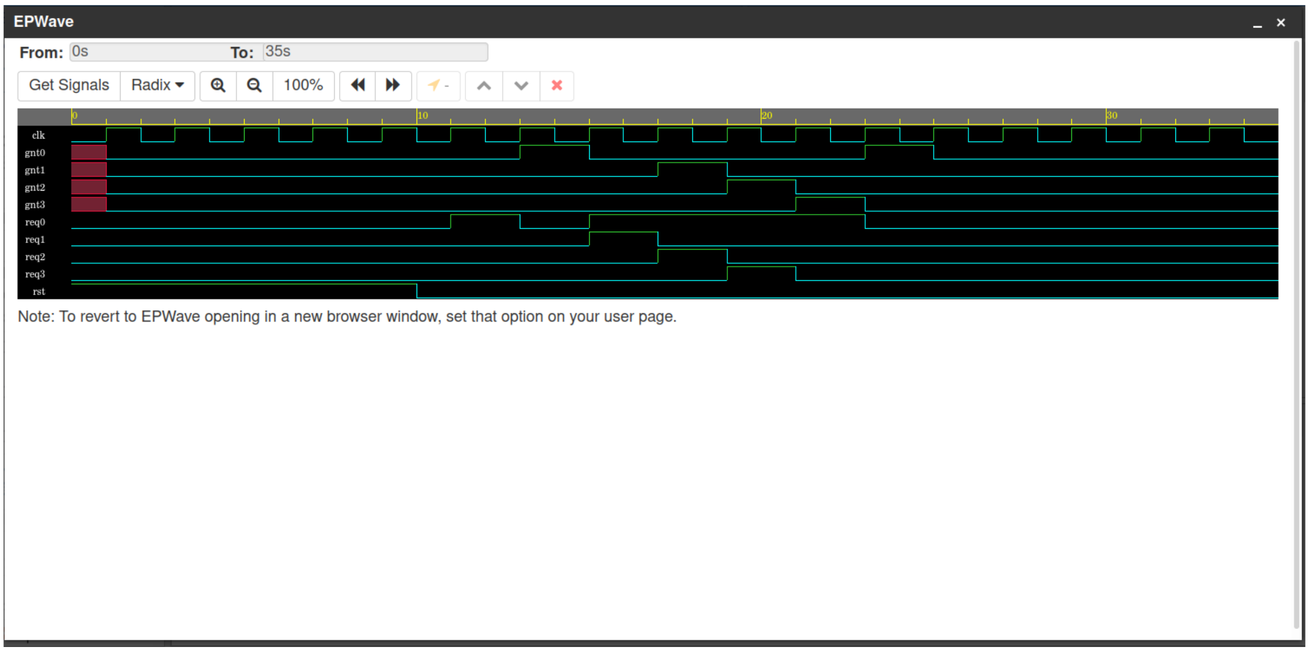
```
module top ();

reg                clk;
reg                rst;
reg                req3;
reg                req2;
reg                req1;
reg                req0;
wire               gnt3;
wire               gnt2;
wire               gnt1;
wire               gnt0;

// Clock generator
always #1 clk = ~clk;

initial begin
  $dumpfile ("arbiter.vcd");
  $dumpvars();
  clk = 0;
  rst = 1;
  req0 = 0;
  req1 = 0;
  req2 = 0;
  req3 = 0;
  #10 rst = 0;
  repeat (1) @ (posedge clk);
  req0 <= 1;
  repeat (1) @ (posedge clk);
  req0 <= 0;
  repeat (1) @ (posedge clk);
  req0 <= 1;
  req1 <= 1;
  repeat (1) @ (posedge clk);
  req2 <= 1;
  req1 <= 0;
  repeat (1) @ (posedge clk);
  req3 <= 1;
  req2 <= 0;
  repeat (1) @ (posedge clk);
  req3 <= 0;
  repeat (1) @ (posedge clk);
  req0 <= 0;
  repeat (1) @ (posedge clk)
  #10 $finish;
end

// Connect the DUT
arbiter U (
 clk,
 rst,
 req3,
 req2,
 req1,
 req0,
 gnt3,
 gnt2,
 gnt1,
 gnt0
);

endmodule
```

module.v

```
module arbiter (
  clk,
  rst,
  req3,
  req2,
  req1,
  req0,
  gnt3,
  gnt2,
  gnt1,
  gnt0
);
// PORT DECLARATION
input           clk;    //CLOCK
input           rst;    //RESET
input           req3;   //REQUEST SIGNALS
input           req2;
input           req1;
input           req0;
output          gnt3;   //GRANT SIGNALS
output          gnt2;
output          gnt1;
output          gnt0;

//INTERNAL REGISTERS
wire    [1:0]   gnt       ;
wire            comreq    ;
wire            beg       ;   // BEGIN SIGNAL
wire   [1:0]    lgnt      ;   // LATCHED ENCODED GRANT
wire            lcomreq   ;   // BUS STATUS
reg             lgnt0     ;   // LATCHED GRANTS
reg             lgnt1     ;
reg             lgnt2     ;
reg             lgnt3     ;
reg             mask_enable   ;
reg             lmask0    ;
reg             lmask1    ;
reg             ledge    ;

//--//

always @ (posedge clk)

  if (rst)  //if reset is true
  begin
    lgnt0 <= 0;
    lgnt1 <= 0;
    lgnt2 <= 0;
    lgnt3 <= 0;
  end

  else
    begin
  lgnt0 <=(~lcomreq & ~lmask1 & ~lmask0 & ~req3 & ~req2 & ~req1 & req0)
        | (~lcomreq & ~lmask1 &  lmask0 & ~req3 & ~req2 &  req0)
        | (~lcomreq &  lmask1 & ~lmask0 & ~req3 &  req0)
        | (~lcomreq &  lmask1 &  lmask0 & req0   )
        | ( lcomreq & lgnt0 );

  lgnt1 <=(~lcomreq & ~lmask1 & ~lmask0 &  req1)
        | (~lcomreq & ~lmask1 &  lmask0 & ~req3 & ~req2 &  req1 & ~req0)
        | (~lcomreq &  lmask1 & ~lmask0 & ~req3 &  req1 & ~req0)
        | (~lcomreq &  lmask1 &  lmask0 &  req1 & ~req0)
        | ( lcomreq &  lgnt1);

  lgnt2 <=(~lcomreq & ~lmask1 & ~lmask0 &  req2  & ~req1)
        | (~lcomreq & ~lmask1 &  lmask0 &  req2)
        | (~lcomreq &  lmask1 & ~lmask0 & ~req3 &  req2  & ~req1 & ~req0)
        | (~lcomreq &  lmask1 &  lmask0 &  req2 & ~req1 & ~req0)
        | ( lcomreq &  lgnt2);
  lgnt3 <=(~lcomreq & ~lmask1 & ~lmask0 & req3  & ~req2 & ~req1)
        | (~lcomreq & ~lmask1 &  lmask0 & req3  & ~req2)
        | (~lcomreq &  lmask1 & ~lmask0 & req3)
        | (~lcomreq &  lmask1 &  lmask0 & req3  & ~req2 & ~req1 & ~req0)
        | ( lcomreq & lgnt3);
end

 //BEGIN SIGNAL

  assign beg = (req3 | req2 | req1 | req0) & ~lcomreq;


 // comreq logic (BUS STATUS)

assign lcomreq = ( req3 & lgnt3 )
                | ( req2 & lgnt2 )
                | ( req1 & lgnt1 )
                | ( req0 & lgnt0 );


// Encoder logic

assign  lgnt =  {(lgnt3 | lgnt2),(lgnt3 | lgnt1)};


// lmask register.

always @ (posedge clk )
if( rst )
  begin
    lmask1 <= 0;
    lmask0 <= 0;
  end
 else if(mask_enable)
    begin
      lmask1 <= lgnt[1];
      lmask0 <= lgnt[0];
    end
 else
    begin
      lmask1 <= lmask1;
      lmask0 <= lmask0;
    end


assign comreq = lcomreq;
assign gnt    = lgnt;
```

```
// Drive the outputs

assign gnt3    = lgnt3;
assign gnt2    = lgnt2;
assign gnt1    = lgnt1;
assign gnt0    = lgnt0;

endmodule
```

Output Waveform Obtained



Note: To revert to EPWave opening in a new browser window, set that option on your user page.