1. **RAJA**

**Design a 64-bit arithmetic shift register, with synchronous load. The shifter can shift both left and right, and by 1 or 8 times. Write Test bench to verify it.**

**(Hint: use a 2 bit register to decide the shifting direction and the number of time shifting has to be done i.e 1 or 8 times).**

**RTL:**

**module alu(input [63:0] ld,input [1:0]shift, input reset, clk, load, output reg [63:0]q );**

**parameter leftb = 2'b00, rightb = 2'b01, leftBy = 2'b10, rightBy = 2'b11;**

**always@(posedge clk) begin**

**if (reset) begin**

**q <= 64'd0;**

**end**

**else if(load) begin**

**q <= ld;**

**end**

**else begin**

**case (shift)**

**leftb : q[63:0] <= {q[62:0], 1'b0};**

**rightb: q[63:0] <= {1'b0,q[63:1]};**

**leftBy : q[63:0] <= {q[55:0], 8'b0};**

**rightBy: q[63:0] <= {8'b0,q[63:8]};**

**default: q[63:0] <= q[63:0];**

**endcase**

**end**

**end**

**endmodule**

**Testbench:**

**module testbench();**

**reg clk,reset,load;**

**reg [1:0] shift;**

**reg [63:0] ld;**

**wire [63:0] q;**

**integer i;**

**alu dut(ld,shift,reset,clk,load,q);**

**initial begin**

**clk = 1'b1;**

**forever #5 clk = ~clk;**

**end**

**task loading();**

**begin**

**load = 1'b1;**

**for (i = 0; i<64; i = i+1)**

**begin**

**ld[i] = {$random}%2;**

**end**

**#10;**

**load = 1'b0;**

**end**

**endtask**

**initial begin**

**$monitor("@time = %4d, load = %b, shift = %b, q = %h",$time,load,shift,q);**

**reset = 1'b1;**

**#10;**

**reset = 1'b0;**

**loading;**

**shift = 2'd0;**

**#20;**

**shift = 2'd1;**

**#20;**

**shift = 2'd2;**

**#20;**

**shift = 2'd3;**

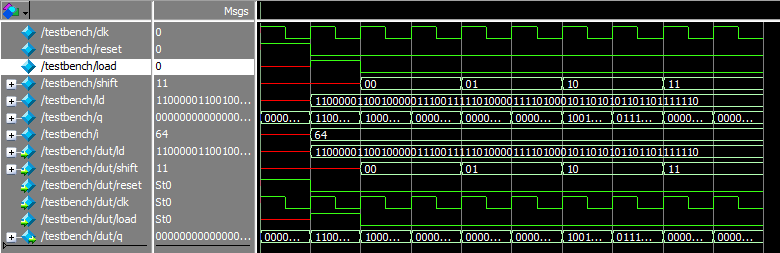
**#20;**

**$finish;**

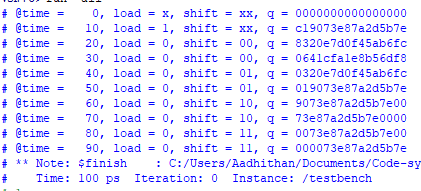
**end**

**endmodule**

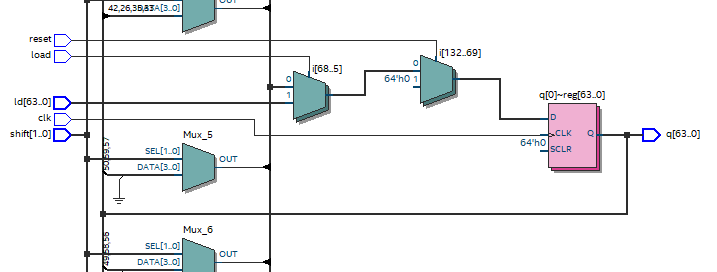
**Wave:**

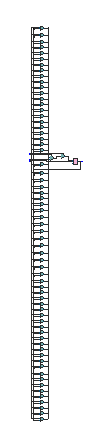
****

**Output:**

****

**RTL output:**

****

****