Ex1

module msd (

input [4:0] i,

output reg [3:0] o

);

always@(\*) begin

if(i>=0 && i<10)

o=i;

if(i>=10 && i<20)

o=1;

if(i>=20 && i<30)

o=2;

if(i>=30 && i<32)

o=3;

end

endmodule

module msd\_tb;

reg [4:0] i;

wire [3:0] o;

msd msd\_i (.i(i), .o(o));

integer k;

initial begin

$display("Time\ti\t\to");

$monitor("%0t\t%b(%2d)\t%b(%0d)", $time, i, i, o, o);

i = 0;

for (k = 1; k < 32; k = k + 1)

#10 i = k;

end

endmodule

Ex2

module div3 (

input [3:0] i,

output reg [2:0] o

);

always@(\*) begin

o=i/3;

end

endmodule

module div3\_tb;

reg [3:0] i;

wire [2:0] o;

div3 div3\_i (.i(i), .o(o));

integer k;

initial begin

$display("Time\ti\t\to");

$monitor("%0t\t%b(%2d)\t%b(%0d)", $time, i, i, o, o);

i = 0;

for (k = 1; k < 16; k = k + 1)

#10 i = k;

end

endmodule

Ex3

module cnt1s (

input [5:0] i,

output reg [2:0] o

);

integer j;

always@(\*) begin

o=0;

for(j=0;j<6;j=j+1)

begin

if(i[j]!=0)

o=o+1;

end

end

endmodule

module cnt1s\_tb;

reg [5:0] i;

wire [2:0] o;

cnt1s cnt1s\_i (.i(i), .o(o));

integer k;

initial begin

$display("Time\ti\t\to");

$monitor("%0t\t%b(%2d)\t%b(%0d)", $time, i, i, o, o);

i = 0;

for (k = 1; k < 64; k = k + 1)

#10 i = k;

end

endmodule

Ex4

module seq3b (

input [3:0] i,

output reg o

);

integer bit1=0;

integer bit2=0;

integer j=0;

integer ct=1;

integer ok=0;

always@(\*) begin

bit1=i[0];

ct=1;

ok=0;

for(j=1;j<4;j=j+1) begin

bit2=i[j];

if(bit1==bit2) begin

ct=ct+1;

if(ct==3)

ok=1;

end

else

ct=1;

bit1=bit2;

end

if(ct>=3 || ok==1)

o=1;

else

o=0;

end

endmodule

module seq3b\_tb;

reg [3:0] i;

wire o;

seq3b seq3b\_i (.i(i), .o(o));

integer k;

initial begin

$display("Time\ti\t\to");

$monitor("%0t\t%b(%2d)\t%b", $time, i, i, o);

i = 0;

for (k = 1; k < 16; k = k + 1)

#10 i = k;

end

endmodule

Ex5

module mul5bcd (

input [3:0] i,

output reg [3:0] d, u

);

integer x=0;

always@(\*) begin

x=i\*5;

d=x/10;

u=x%10;

end

endmodule

module mul5bcd\_tb;

reg [3:0] i;

wire [3:0] d, u;

mul5bcd mul5bcd\_i (.i(i), .d(d), .u(u));

integer k;

initial begin

$display("Time\ti\t\td\t\tu");

$monitor("%0t\t%b(%4d)\t%b(%4d)\t%b(%4d)", $time, i, i, d, d, u, u);

i = 0;

for (k = 1; k < 10; k = k + 1)

#10 i = k;

end

endmodule

Ex6

module text2nibble (

input [7:0] i,

output reg [3:0] o

);

always@(\*) begin

if(i>=8'b00110000 && i<=8'b00111001)

o=i-8'b00110000;

else

o=4'b1111;

end

endmodule

module text2nibble\_tb;

reg [7:0] i;

wire [3:0] o;

text2nibble text2nibble\_i (.i(i), .o(o));

integer k;

initial begin

$display("Time\ti\ti\_chr\to");

$monitor("%0t\t%b\t%c\t%b(%d)", $time, i, i, o, o);

i = 0;

for (k = 1; k < 256; k = k + 1)

#10 i = k;

end

endmodule

Ex7

module r4b (

input clk, rst\_b, ld, sh, sh\_in,

input [3:0] d,

output reg [3:0] q

);

always@(\*) begin

if(rst\_b==0)

q=4'b0000;

else

if(ld==1)

q=d;

else

if(sh==1)

q=(q>>1)|sh\_in<<3;

end

endmodule

module r4b\_tb;

reg clk, rst\_b, ld, sh, sh\_in;

reg [3:0] d;

wire [3:0] q;

r4b r4b\_i (.clk(clk), .rst\_b(rst\_b), .ld(ld), .sh(sh), .sh\_in(sh\_in), .d(d), .q(q));

initial begin

{clk, rst\_b} = 0;

#5 rst\_b = 1;

#45 clk = 1;

repeat (40)

#50 clk = ~clk;

end

integer k, l;

initial begin

$display("Time\top\td\tsh\_in\tq");

{d, sh\_in} = 0; {ld, sh} = 0;

for (k = 0; k < 32; k = k + 1) begin

$display("%0t\t%s\t%b\t%b\t%b", $time, (ld) ? "LOAD" : (sh) ? "SHIFT" : "NO\_OP", d, sh\_in, q);

#100 l = $urandom; {d, sh\_in} = l[6:2]; {ld, sh} = l[1:0] % 3;

end

$display("%0t\t%s\t%b\t%b\t%b", $time, (ld) ? "LOAD" : (sh) ? "SHIFT" : "NO\_OP", d, sh\_in, q);

end

endmodule