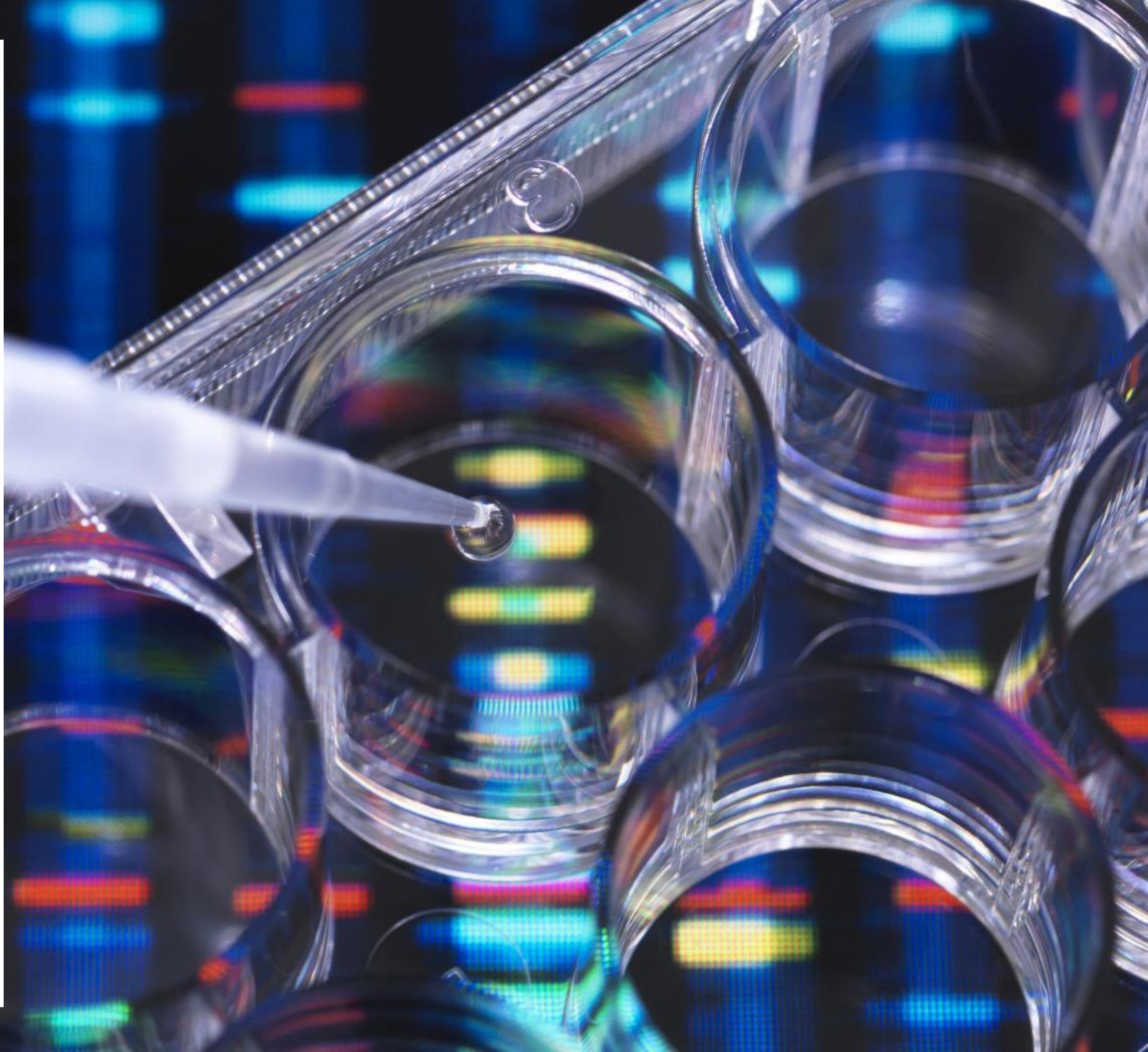
DIGITAL IC DESIGN LAB

AMREEN KAUR

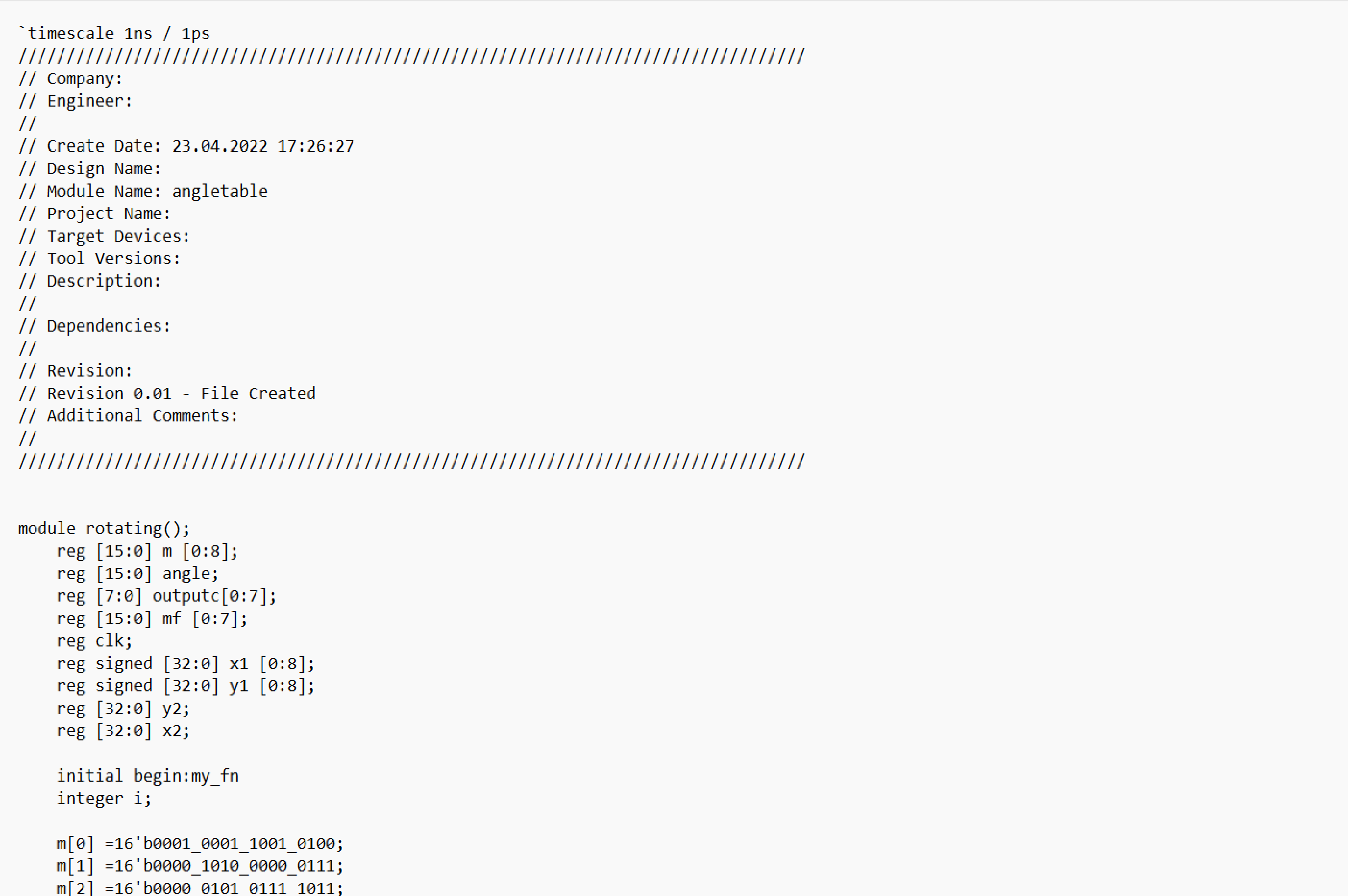
IS21MTECH14002

2D-CORDIC ROTATING MODE-CODE,TEST BENCH,SIMULATIONS AND VALIDATIONS

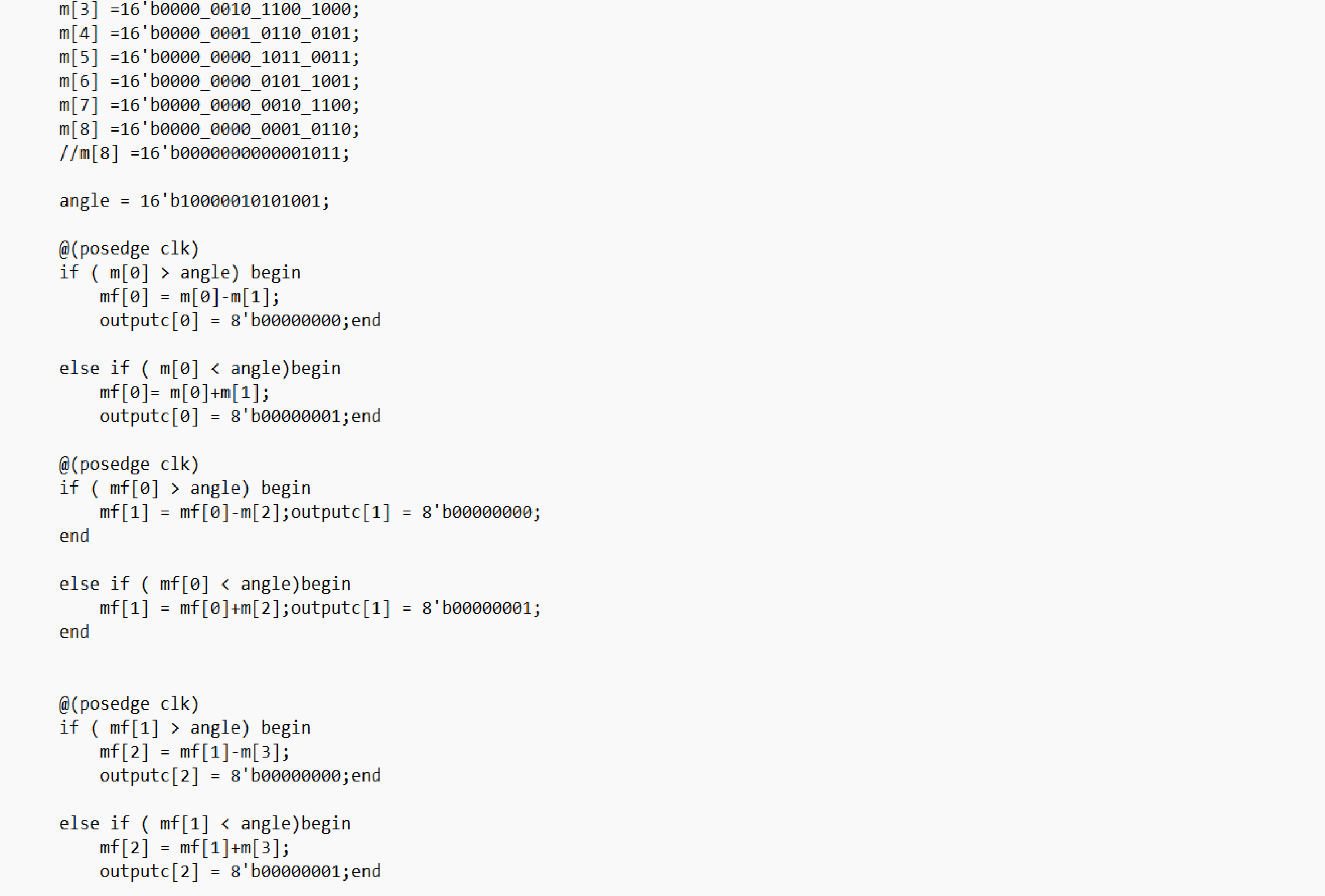


QUESTION STATEMENT-Upload the Code for 2D Rotation Mode CORDIC along with Test Bench, Validation/ simulation results. Include everything in a PDF File and Upload as a Single file.

CODE FOR CORDIC-ROTATING MODE

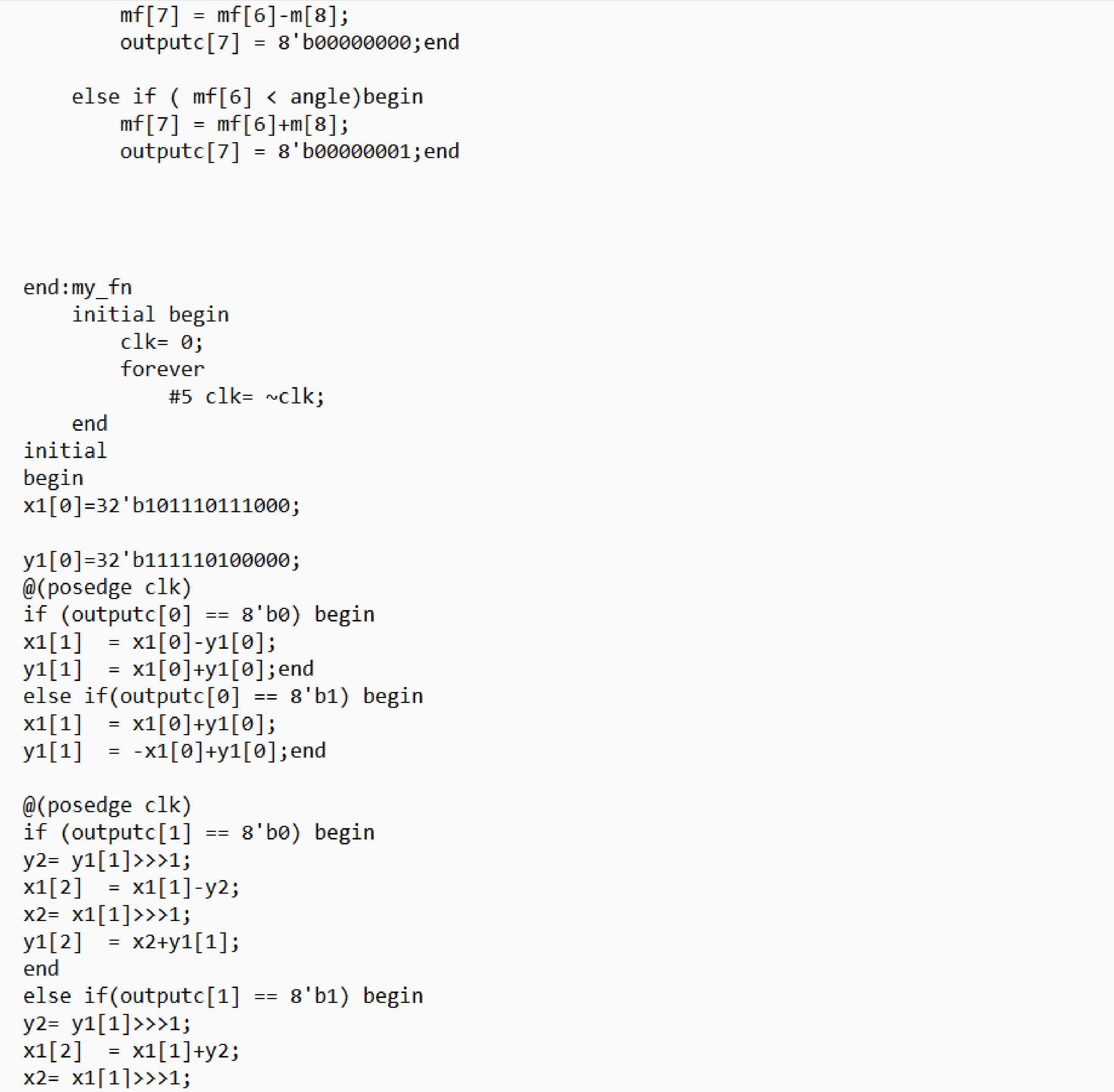


```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 23.04.2022 17:26:27
// Design Name:
// Module Name: angletable
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module rotating();
    reg [15:0] m [0:8];
    reg [15:0] angle;
    reg [7:0] outputc[0:7];
    reg [15:0] mf [0:7];
    reg clk;
    reg signed [32:0] x1 [0:8];
    reg signed [32:0] y1 [0:8];
    reg [32:0] y2;
    reg [32:0] x2;

    initial begin:my_fn
    integer i;

    m[0] =16'b0001_0001_1001_0100;
    m[1] =16'b0000_1010_0000_0111;
    m[2] =16'b0000_0101_0111_1011;
```

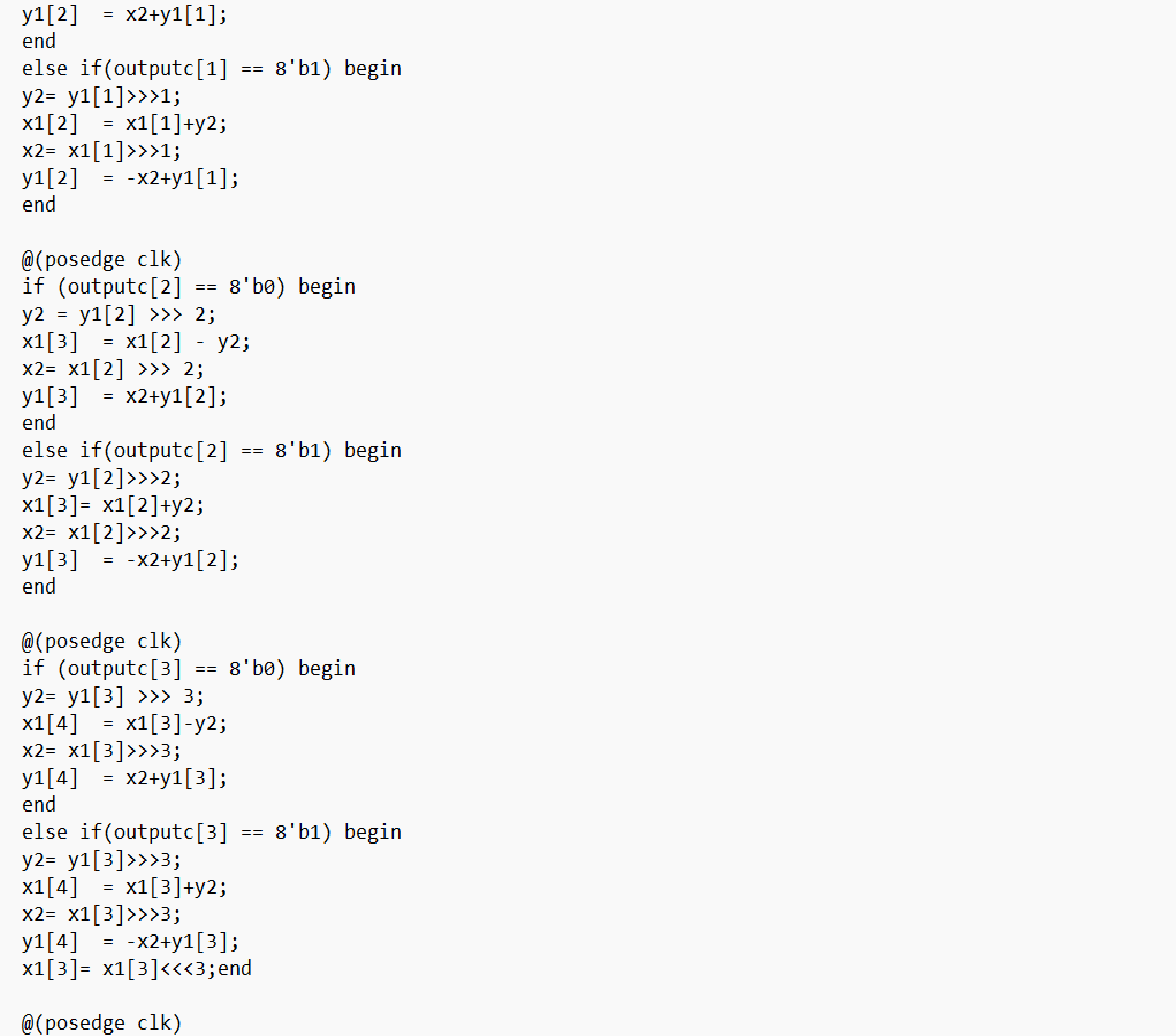


```
m[3] =16'b0000_0010_1100_1000;
m[4] =16'b0000_0001_0110_0101;
m[5] =16'b0000_0000_1011_0011;
m[6] =16'b0000_0000_0101_1001;
m[7] =16'b0000_0000_0010_1100;
m[8] =16'b0000_0000_0001_0110;
//m[8] =16'b0000000000001011;

angle = 16'b10000010101001;

@(posedge clk)
if ( m[0] > angle) begin
    mf[0] = m[0]-m[1];
    outputc[0] = 8'b00000000;end

else if ( m[0] < angle)begin
    mf[0]= m[0]+m[1];
    outputc[0] = 8'b00000001;end

@(posedge clk)
if ( mf[0] > angle) begin
    mf[1] = mf[0]-m[2];outputc[1] = 8'b00000000;
end

else if ( mf[0] < angle)begin
    mf[1] = mf[0]+m[2];outputc[1] = 8'b00000001;
end


@(posedge clk)
if ( mf[1] > angle) begin
    mf[2] = mf[1]-m[3];
    outputc[2] = 8'b00000000;end

else if ( mf[1] < angle)begin
    mf[2] = mf[1]+m[3];
    outputc[2] = 8'b00000001;end
```

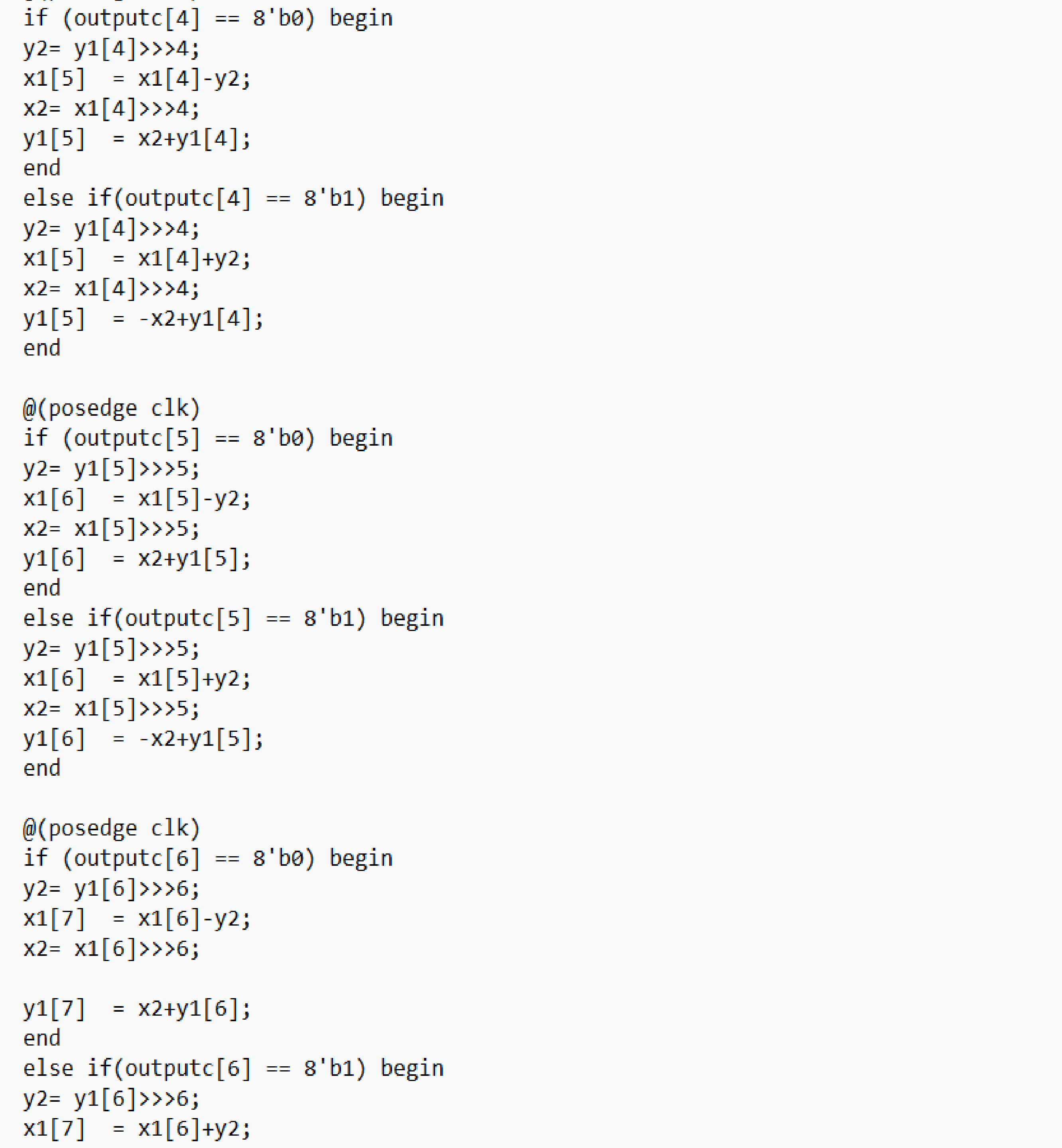
```
@(posedge clk)
if ( mf[2] > angle) begin
    mf[3] = mf[2]-m[4];
    outputc[3] = 8'b00000000;end

else if ( mf[2] < angle)begin
    mf[3] = mf[2]+m[4];
    outputc[3] = 8'b00000001;end

@(posedge clk)
if ( mf[3] > angle) begin
    mf[4] = mf[3]-m[5];
    outputc[4] = 8'b00000000;end

else if ( mf[3] < angle)begin
    mf[4] = mf[3]+m[5];
    outputc[4] = 8'b00000001;end

@(posedge clk)
if ( mf[4] > angle) begin
    mf[5] = mf[4]-m[6];
    outputc[5] = 8'b00000000;end

else if ( mf[4] < angle)begin
    mf[5] = mf[4]+m[6];
    outputc[5] = 8'b00000001;end

@(posedge clk)
if ( mf[5] > angle) begin
    mf[6] = mf[5]-m[7];
    outputc[6] = 8'b00000000;end

else if ( mf[5] < angle)begin
    mf[6] = mf[5]+m[7];
    outputc[6] = 8'b00000001;end

@(posedge clk)
if ( mf[6] > angle) begin
```



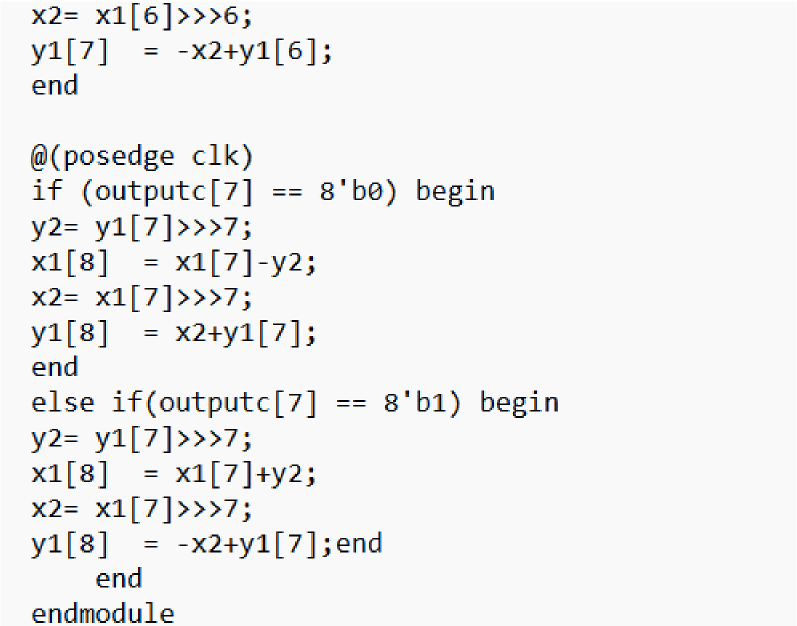
```
        mf[7] = mf[6]-m[8];
        outputc[7] = 8'b00000000;end

    else if ( mf[6] < angle)begin
        mf[7] = mf[6]+m[8];
        outputc[7] = 8'b00000001;end




end:my_fn
    initial begin
        clk= 0;
        forever
            #5 clk= ~clk;
    end
initial
begin
x1[0]=32'b101110111000;

y1[0]=32'b111110100000;
@(posedge clk)
if (outputc[0] == 8'b0) begin
x1[1]  = x1[0]-y1[0];
y1[1]  = x1[0]+y1[0];end
else if(outputc[0] == 8'b1) begin
x1[1]  = x1[0]+y1[0];
y1[1]  = -x1[0]+y1[0];end

@(posedge clk)
if (outputc[1] == 8'b0) begin
y2= y1[1]>>>1;
x1[2]  = x1[1]-y2;
x2= x1[1]>>>1;
y1[2]  = x2+y1[1];
end
else if(outputc[1] == 8'b1) begin
y2= y1[1]>>>1;
x1[2]  = x1[1]+y2;
x2= x1[1]>>>1;
```



```
y1[2]  = x2+y1[1];
end
else if(outputc[1] == 8'b1) begin
y2= y1[1]>>>1;
x1[2]  = x1[1]+y2;
x2= x1[1]>>>1;
y1[2]  = -x2+y1[1];
end

@(posedge clk)
if (outputc[2] == 8'b0) begin
y2 = y1[2] >>> 2;
x1[3]  = x1[2] - y2;
x2= x1[2] >>> 2;
y1[3]  = x2+y1[2];
end
else if(outputc[2] == 8'b1) begin
y2= y1[2]>>>2;
x1[3]= x1[2]+y2;
x2= x1[2]>>>2;
y1[3]  = -x2+y1[2];
end

@(posedge clk)
if (outputc[3] == 8'b0) begin
y2= y1[3] >>> 3;
x1[4]  = x1[3]-y2;
x2= x1[3]>>>3;
y1[4]  = x2+y1[3];
end
else if(outputc[3] == 8'b1) begin
y2= y1[3]>>>3;
x1[4]  = x1[3]+y2;
x2= x1[3]>>>3;
y1[4]  = -x2+y1[3];
x1[3]= x1[3]<<<3;end

@(posedge clk)
```

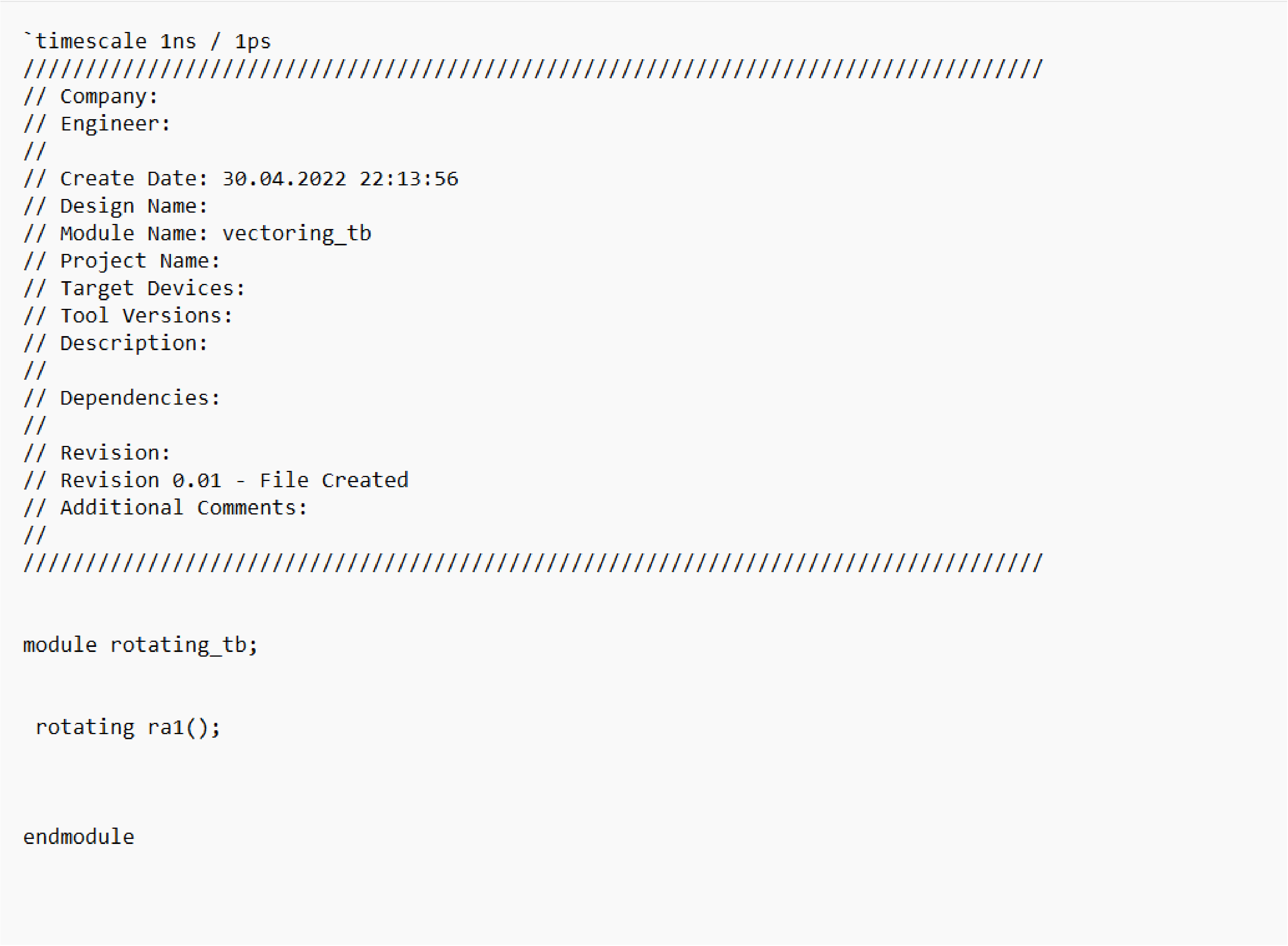


```
if (outputc[4] == 8'b0) begin
y2= y1[4]>>>4;
x1[5]  = x1[4]-y2;
x2= x1[4]>>>4;
y1[5]  = x2+y1[4];
end
else if(outputc[4] == 8'b1) begin
y2= y1[4]>>>4;
x1[5]  = x1[4]+y2;
x2= x1[4]>>>4;
y1[5]  = -x2+y1[4];
end

@(posedge clk)
if (outputc[5] == 8'b0) begin
y2= y1[5]>>>5;
x1[6]  = x1[5]-y2;
x2= x1[5]>>>5;
y1[6]  = x2+y1[5];
end
else if(outputc[5] == 8'b1) begin
y2= y1[5]>>>5;
x1[6]  = x1[5]+y2;
x2= x1[5]>>>5;
y1[6]  = -x2+y1[5];
end

@(posedge clk)
if (outputc[6] == 8'b0) begin
y2= y1[6]>>>6;
x1[7]  = x1[6]-y2;
x2= x1[6]>>>6;

y1[7]  = x2+y1[6];
end
else if(outputc[6] == 8'b1) begin
y2= y1[6]>>>6;
x1[7]  = x1[6]+y2;
```



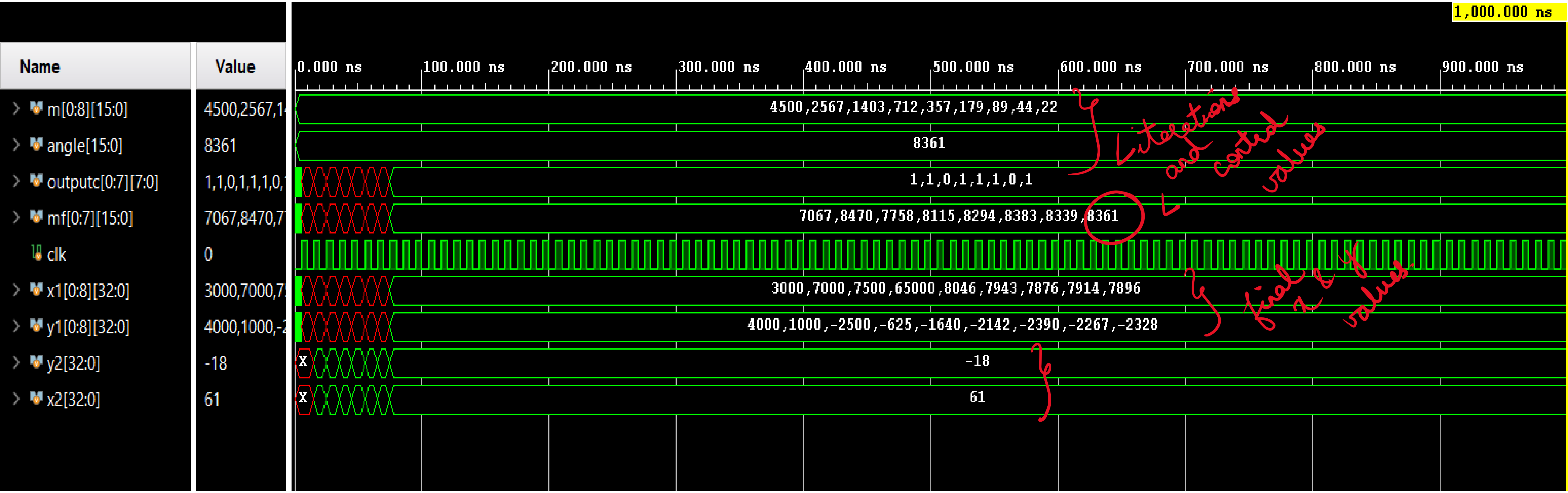
```
x2= x1[6]>>>6;
y1[7]  = -x2+y1[6];
end

@(posedge clk)
if (outputc[7] == 8'b0) begin
y2= y1[7]>>>7;
x1[8]  = x1[7]-y2;
x2= x1[7]>>>7;
y1[8]  = x2+y1[7];
end
else if(outputc[7] == 8'b1) begin
y2= y1[7]>>>7;
x1[8]  = x1[7]+y2;
x2= x1[7]>>>7;
y1[8]  = -x2+y1[7];end
    end
endmodule
```

TESTBENCH



```
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 30.04.2022 22:13:56
// Design Name:
// Module Name: vectoring_tb
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module rotating_tb;


 rotating ra1();



endmodule
```

OUTPUT SIMULATIONS IN DECIMAL

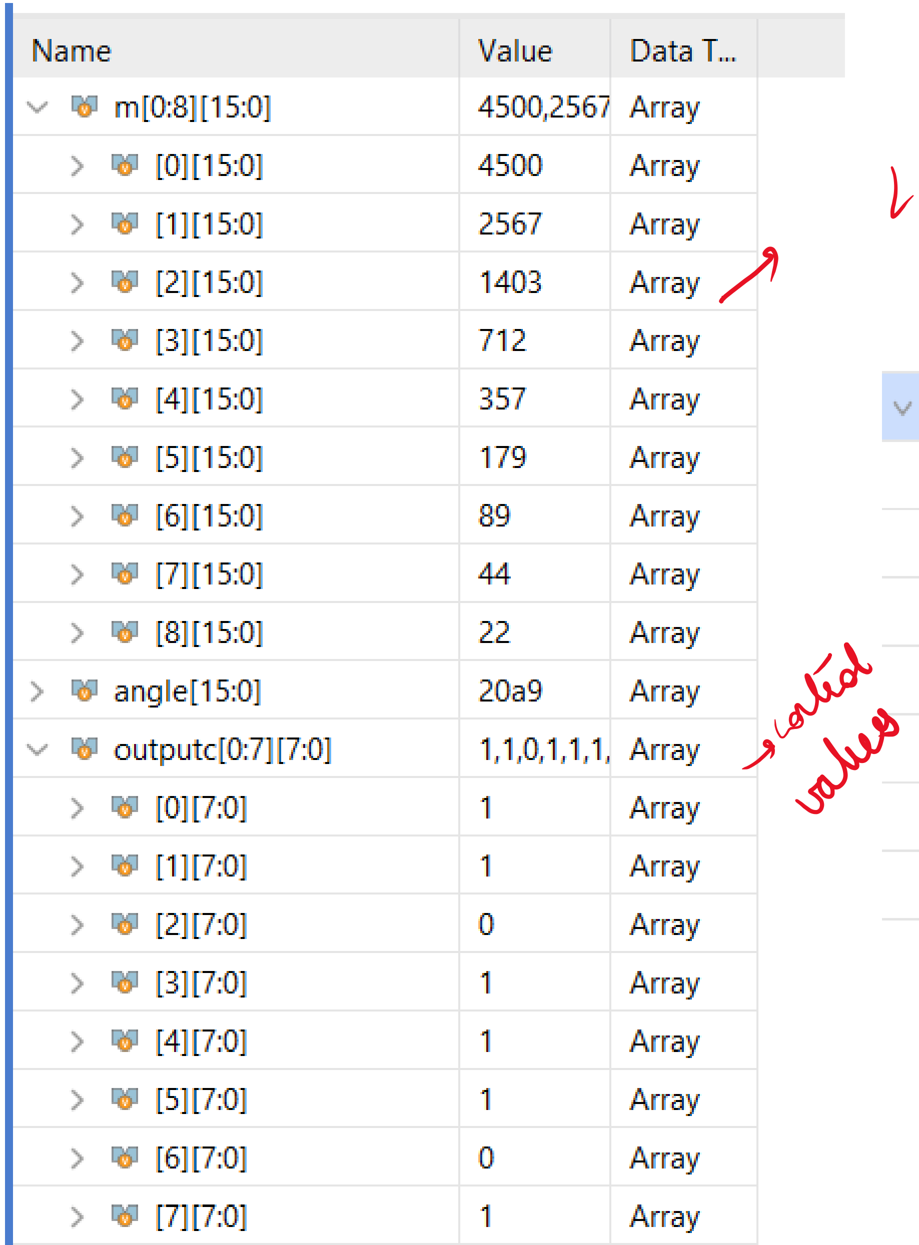


```
                                                                                                   1,000.000 ns
Name               Value          0.000 ns  100.000 ns  200.000 ns  300.000 ns  400.000 ns  500.000 ns  600.000 ns  700.000 ns  800.000 ns  900.000 ns
> m[0:8][15:0]     4500,2567,1         4500,2567,1403,712,357,179,89,44,22
> angle[15:0]      8361                8361
> outputc[0:7][7:0] 1,1,0,1,1,1,0,1    1,1,0,1,1,1,0,1
> mf[0:7][15:0]    7067,8470,7         7067,8470,7758,8115,8294,8383,8339,8361
  clk              0
> x1[0:8][32:0]    3000,7000,7         3000,7000,7500,65000,8046,7943,7876,7914,7896
> y1[0:8][32:0]    4000,1000,-2        4000,1000,-2500,-625,-1640,-2142,-2390,-2267,-2328
> y2[32:0]         -18            X    -18
> x2[32:0]         61             X    61
```

iterations and control values

final x & y values.

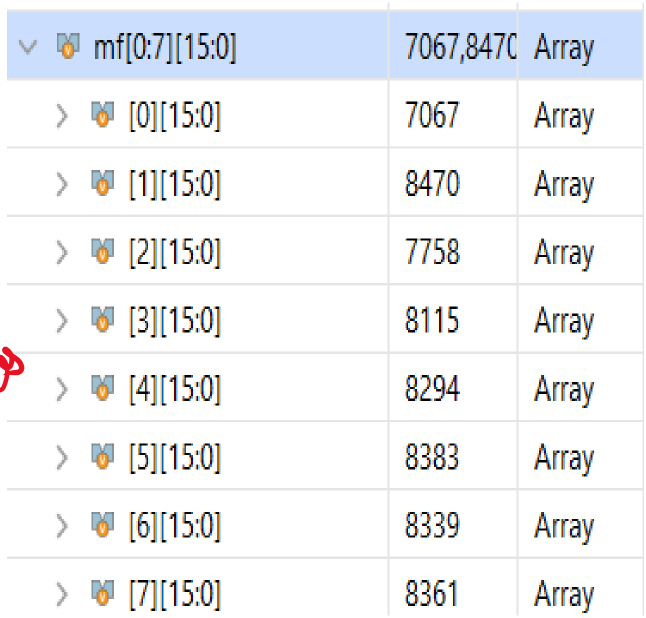
ITERATIONS



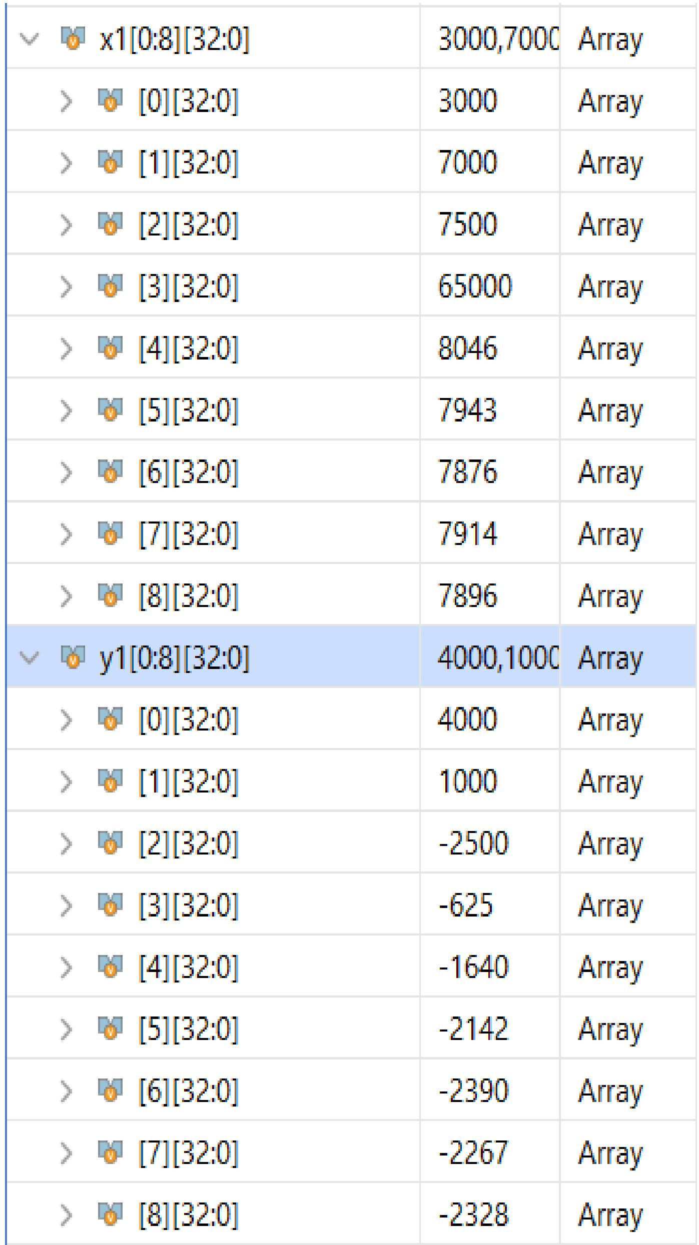
Name	Value	Data T...
m[0:8][15:0]	4500,2567	Array
[0][15:0]	4500	Array
[1][15:0]	2567	Array
[2][15:0]	1403	Array
[3][15:0]	712	Array
[4][15:0]	357	Array
[5][15:0]	179	Array
[6][15:0]	89	Array
[7][15:0]	44	Array
[8][15:0]	22	Array
angle[15:0]	20a9	Array
outputc[0:7][7:0]	1,1,0,1,1,1,	Array
[0][7:0]	1	Array
[1][7:0]	1	Array
[2][7:0]	0	Array
[3][7:0]	1	Array
[4][7:0]	1	Array
[5][7:0]	1	Array
[6][7:0]	0	Array
[7][7:0]	1	Array

↗ ← iterations

→ cordic values



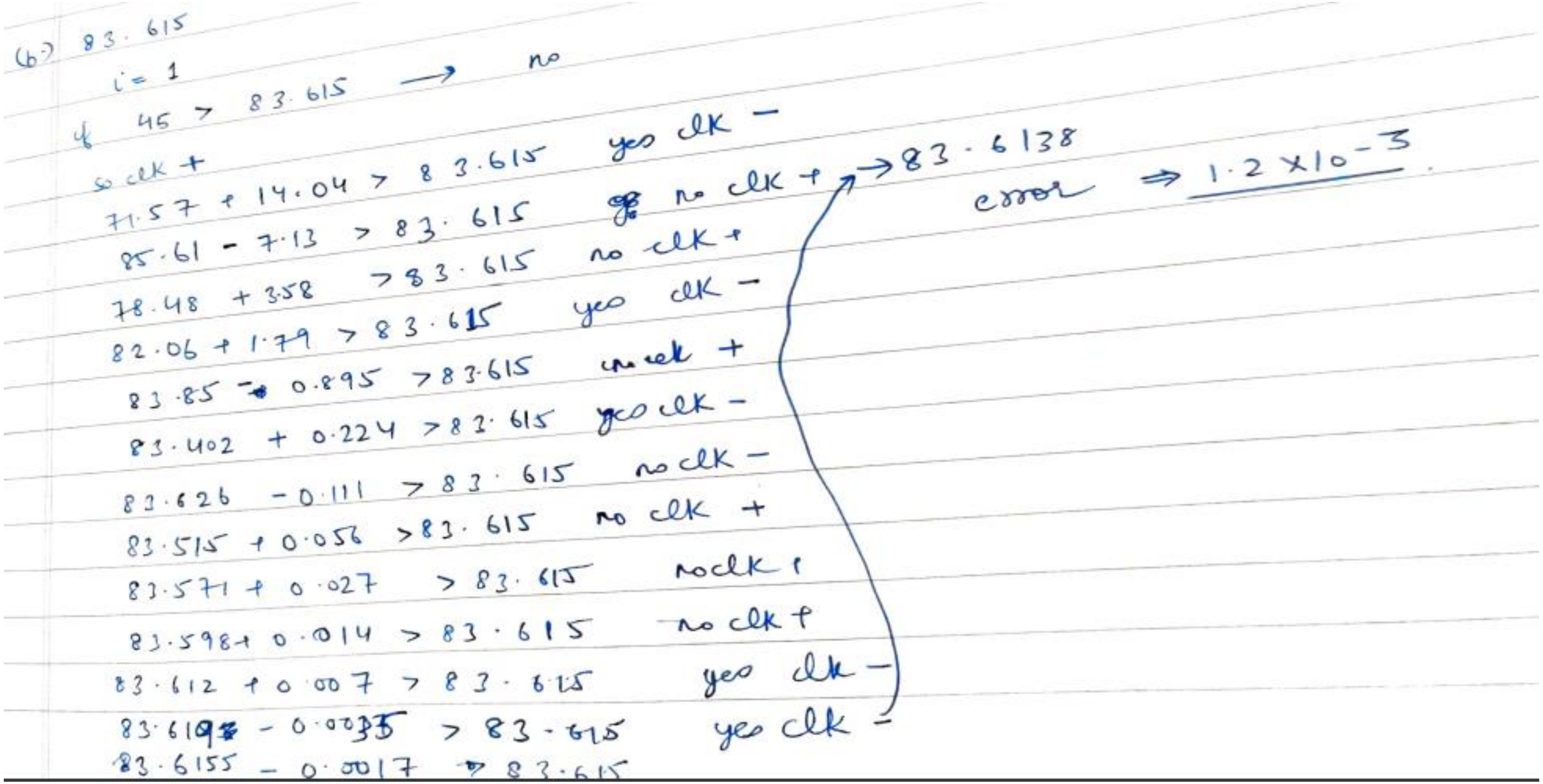
mf[0:7][15:0]	7067,8470	Array
[0][15:0]	7067	Array
[1][15:0]	8470	Array
[2][15:0]	7758	Array
[3][15:0]	8115	Array
[4][15:0]	8294	Array
[5][15:0]	8383	Array
[6][15:0]	8339	Array
[7][15:0]	8361	Array



x1[0:8][32:0]	3000,7000	Array
[0][32:0]	3000	Array
[1][32:0]	7000	Array
[2][32:0]	7500	Array
[3][32:0]	65000	Array
[4][32:0]	8046	Array
[5][32:0]	7943	Array
[6][32:0]	7876	Array
[7][32:0]	7914	Array
[8][32:0]	7896	Array
y1[0:8][32:0]	4000,1000	Array
[0][32:0]	4000	Array
[1][32:0]	1000	Array
[2][32:0]	-2500	Array
[3][32:0]	-625	Array
[4][32:0]	-1640	Array
[5][32:0]	-2142	Array
[6][32:0]	-2390	Array
[7][32:0]	-2267	Array
[8][32:0]	-2328	Array

x & y iteratio

VALIDATIONS



(b) 83.615

i = 1

if 45 > 83.615 → no

so clk +

71.57 + 14.04 > 83.615 yes clk −

85.61 − 7.13 > 83.615 no clk + → 83.6138

78.48 + 3.58 > 83.615 no clk +

82.06 + 1.79 > 83.615 yes clk −

83.85 − 0.895 > 83.615 no clk +

83.402 + 0.224 > 83.615 yes clk −

83.626 − 0.111 > 83.615 no clk −

83.515 + 0.056 > 83.615 no clk +

83.571 + 0.027 > 83.615 no clk +

83.598 + 0.014 > 83.615 no clk +

83.612 + 0.007 > 83.615 yes clk −

83.619 − 0.0035 > 83.615 yes clk −

83.6155 − 0.0017 > 83.615

error ⇒ 1.2×10^{-3}

The code can be found here:
https://github.com/Amreen-Kaur/FPGA-LAB_IS21MTECH14002

THANKYOU