ca6

810101469 محمد صدرا عباسی

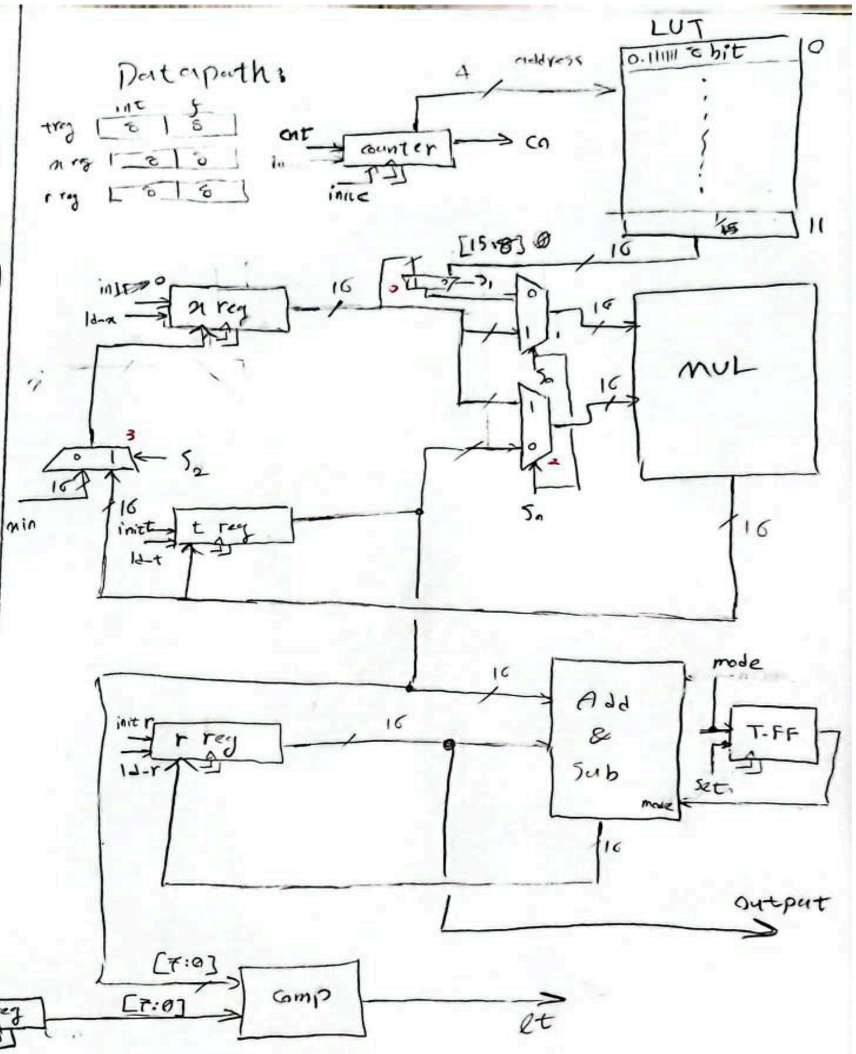
part1)

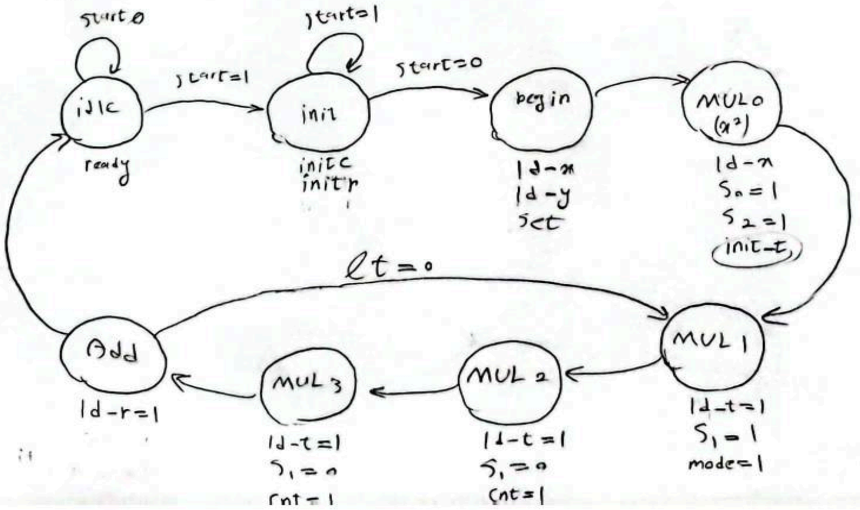
Design flow)

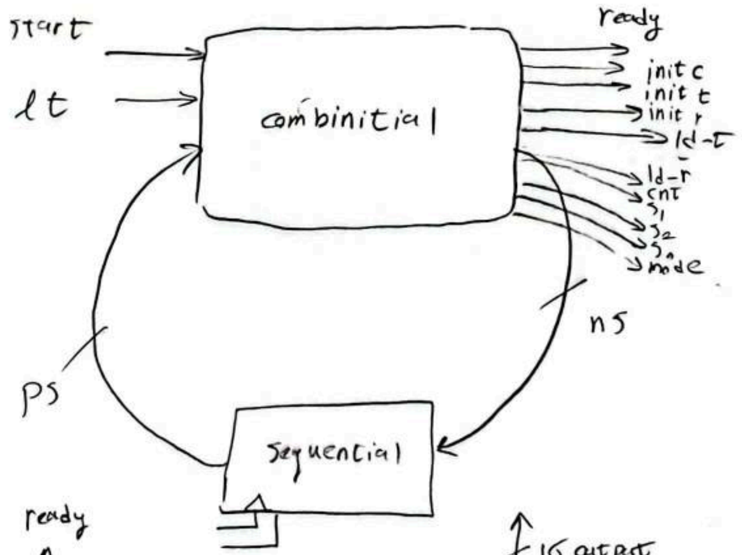
Algorithm:

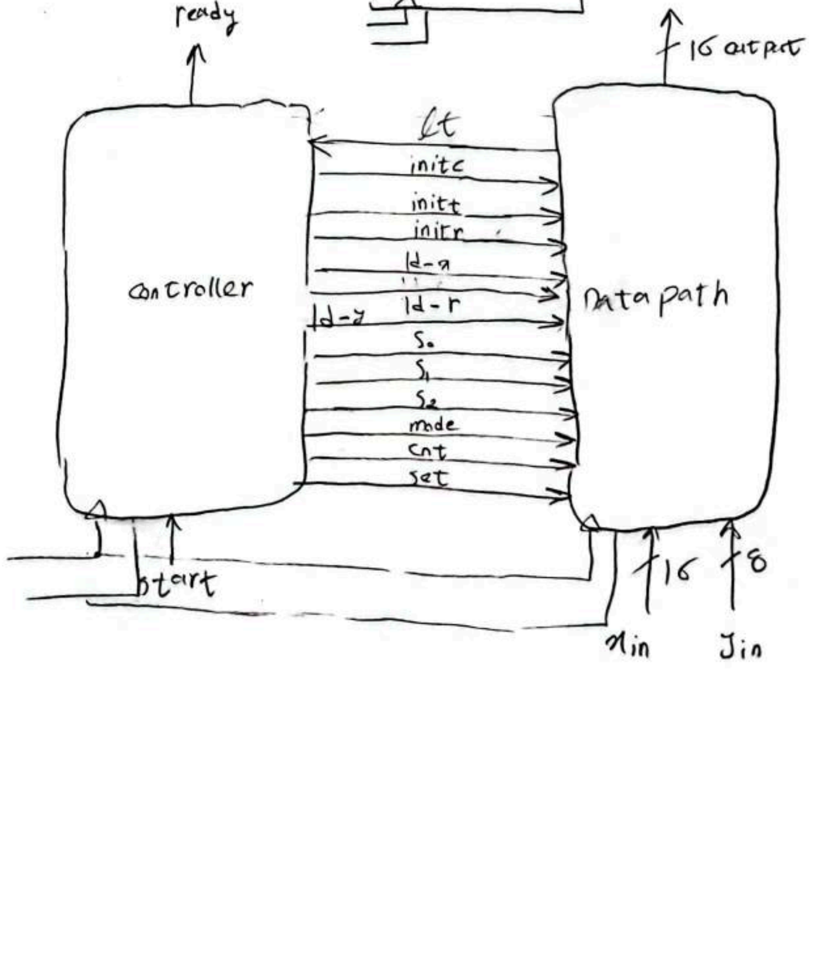
```
t=1;
r=1;
mode=1
for (k=0 ; k<n ; k+2)
{
    t = t×x^2;
    t = t×1/k;
    t = t×1/(k+1);
    if (mode)  r=r+t;
    else       r=r+t;
    mode = ~mode;
}
```

1. 16 bit reg ✓
2. 2 to 1 mux ✓
3. counter ✓
4. Sub-add ✓
5. Comparator ✓
6- mul

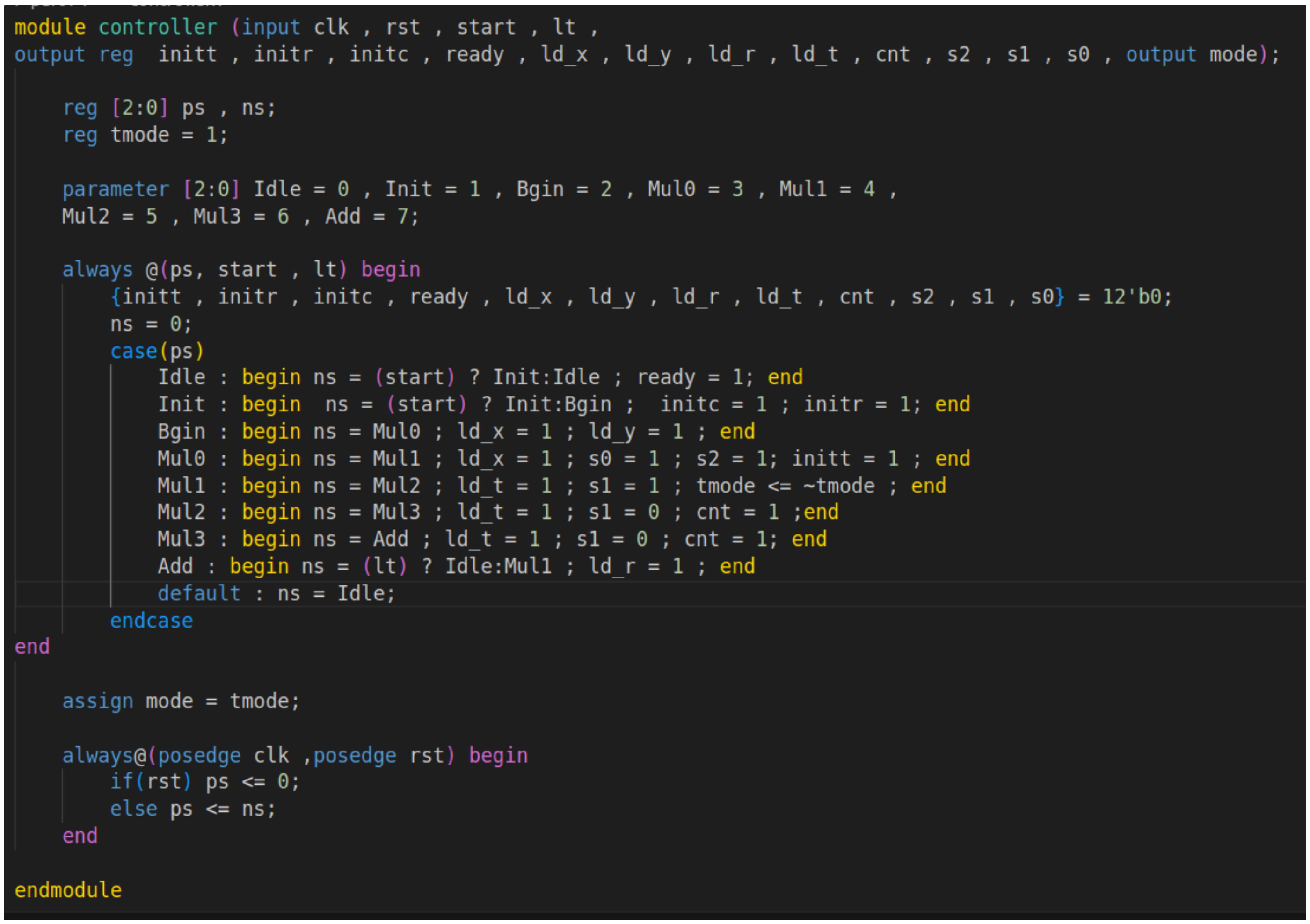




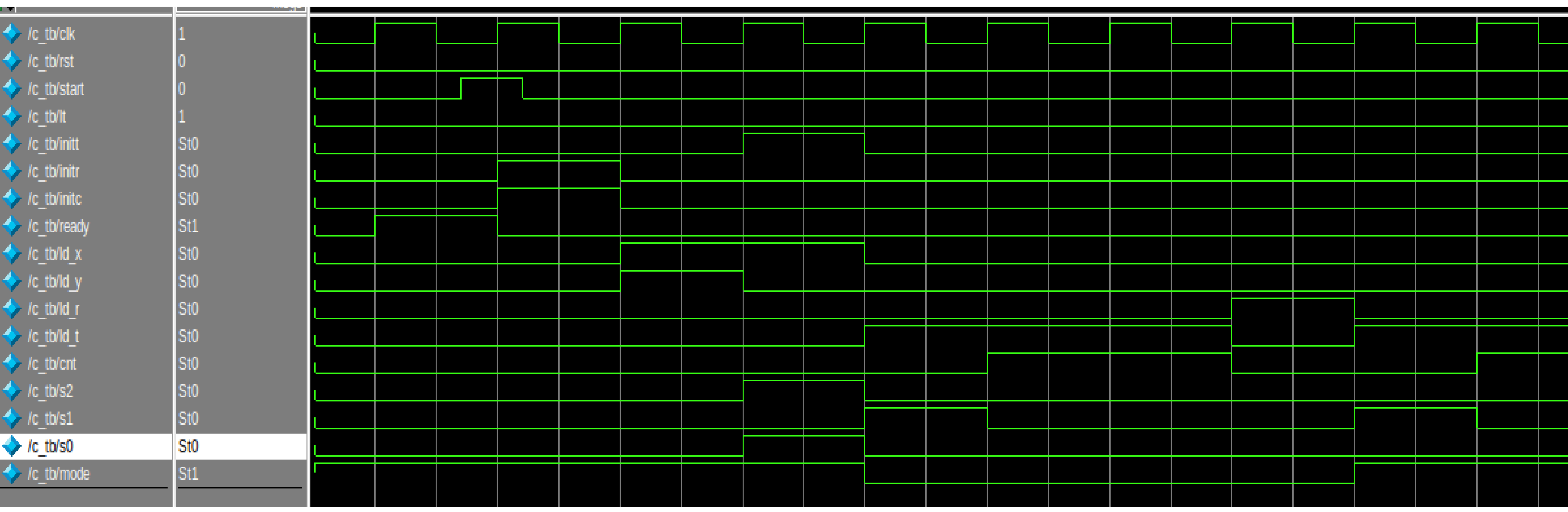

start
lt
combinitial
ready
init c
init t
init r
ld-t
ld-r
cnt
s1
s2
s0
mode
ns
ps
sequential


ready
16 output
controller
Datapath
lt
initc
initt
initr
ld-a
ld-r
ld-y
s0
s1
s2
mode
cnt
set
start
16
8
xin
Jin

descriptions)



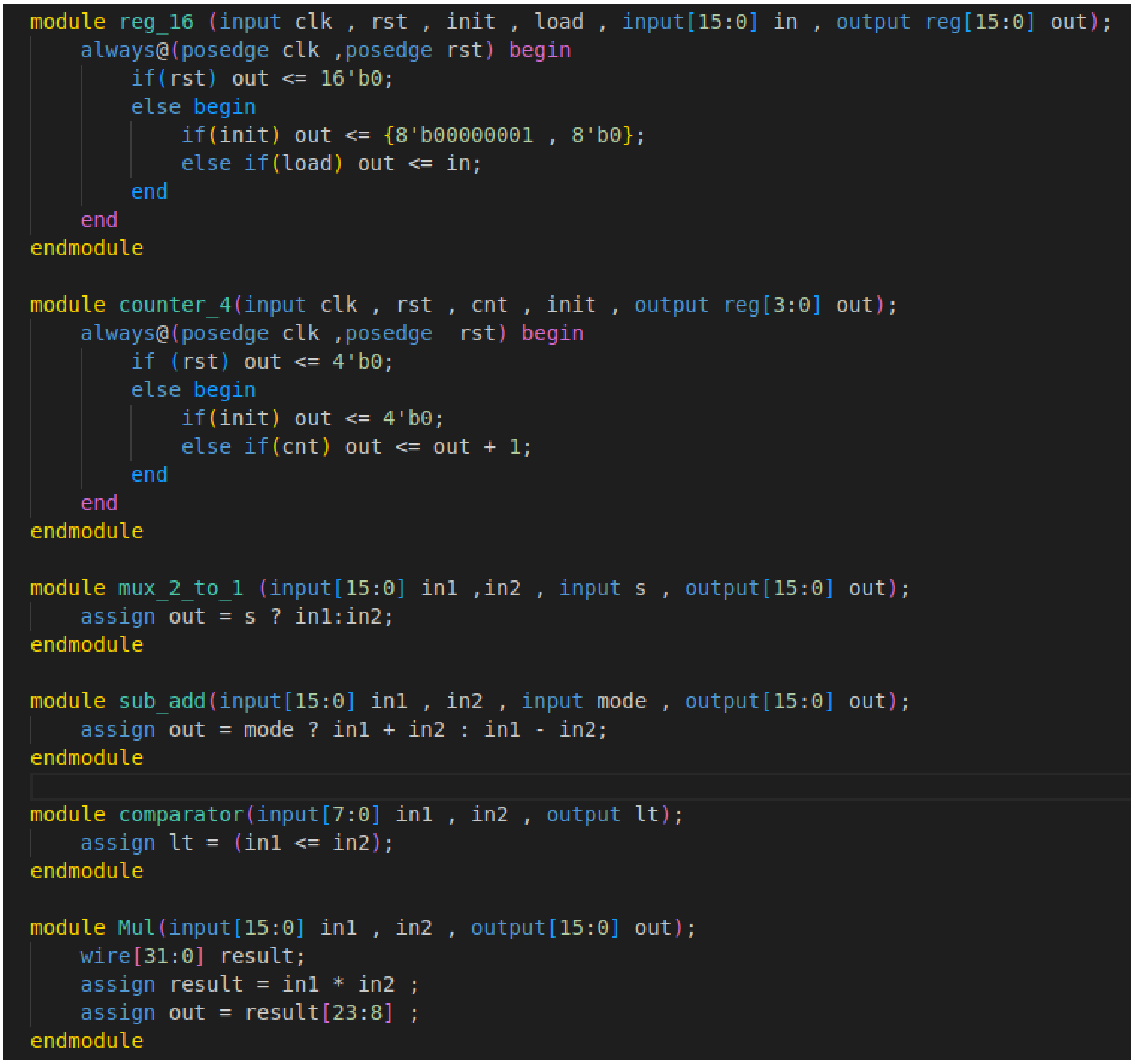
```
module controller (input clk , rst , start , lt ,
output reg  initt , initr , initc , ready , ld_x , ld_y , ld_r , ld_t , cnt , s2 , s1 , s0 , output mode);

    reg [2:0] ps , ns;
    reg tmode = 1;

    parameter [2:0] Idle = 0 , Init = 1 , Bgin = 2 , Mul0 = 3 , Mul1 = 4 ,
    Mul2 = 5 , Mul3 = 6 , Add = 7;

    always @(ps, start , lt) begin
        {initt , initr , initc , ready , ld_x , ld_y , ld_r , ld_t , cnt , s2 , s1 , s0} = 12'b0;
        ns = 0;
        case(ps)
            Idle : begin ns = (start) ? Init:Idle ; ready = 1; end
            Init : begin  ns = (start) ? Init:Bgin ;  initc = 1 ; initr = 1; end
            Bgin : begin ns = Mul0 ; ld_x = 1 ; ld_y = 1 ; end
            Mul0 : begin ns = Mul1 ; ld_x = 1 ; s0 = 1 ; s2 = 1; initt = 1 ; end
            Mul1 : begin ns = Mul2 ; ld_t = 1 ; s1 = 1 ; tmode <= ~tmode ; end
            Mul2 : begin ns = Mul3 ; ld_t = 1 ; s1 = 0 ; cnt = 1 ;end
            Mul3 : begin ns = Add ; ld_t = 1 ; s1 = 0 ; cnt = 1; end
            Add : begin ns = (lt) ? Idle:Mul1 ; ld_r = 1 ; end
            default : ns = Idle;
        endcase
end

    assign mode = tmode;

    always@(posedge clk ,posedge rst) begin
        if(rst) ps <= 0;
        else ps <= ns;
    end

endmodule
```



part2)

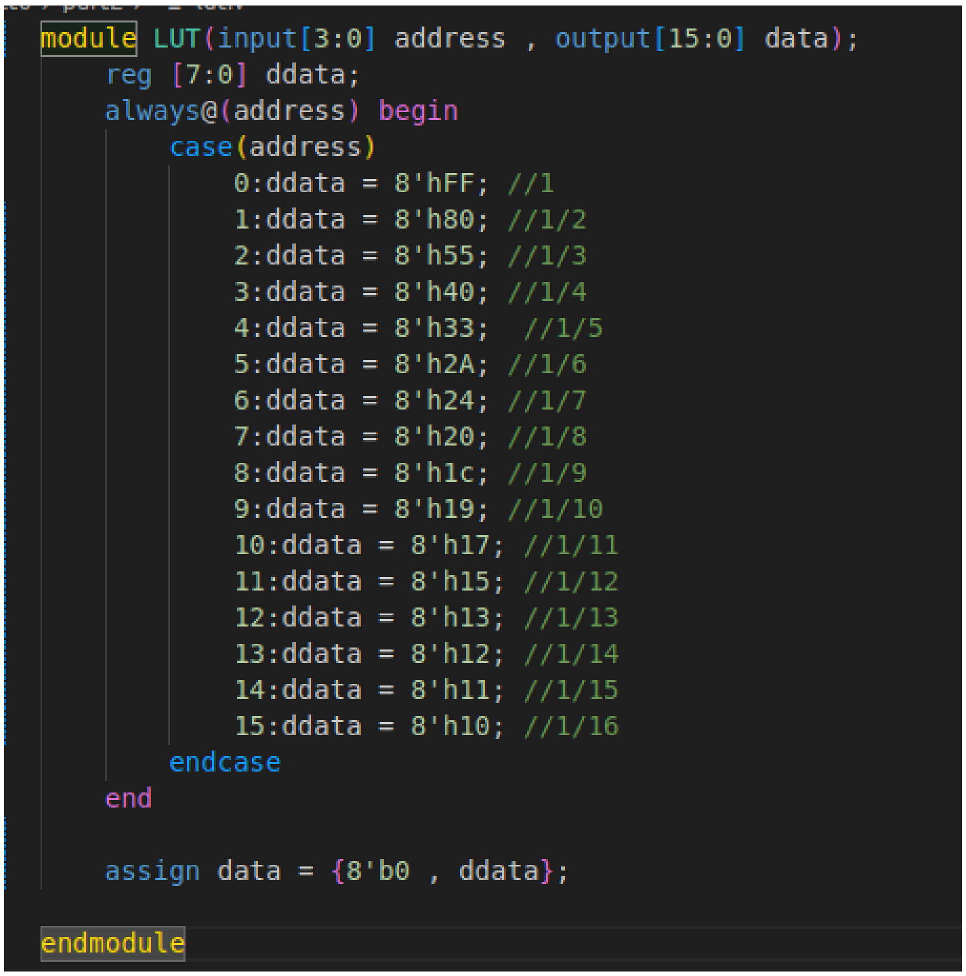
descriptions)

Components:



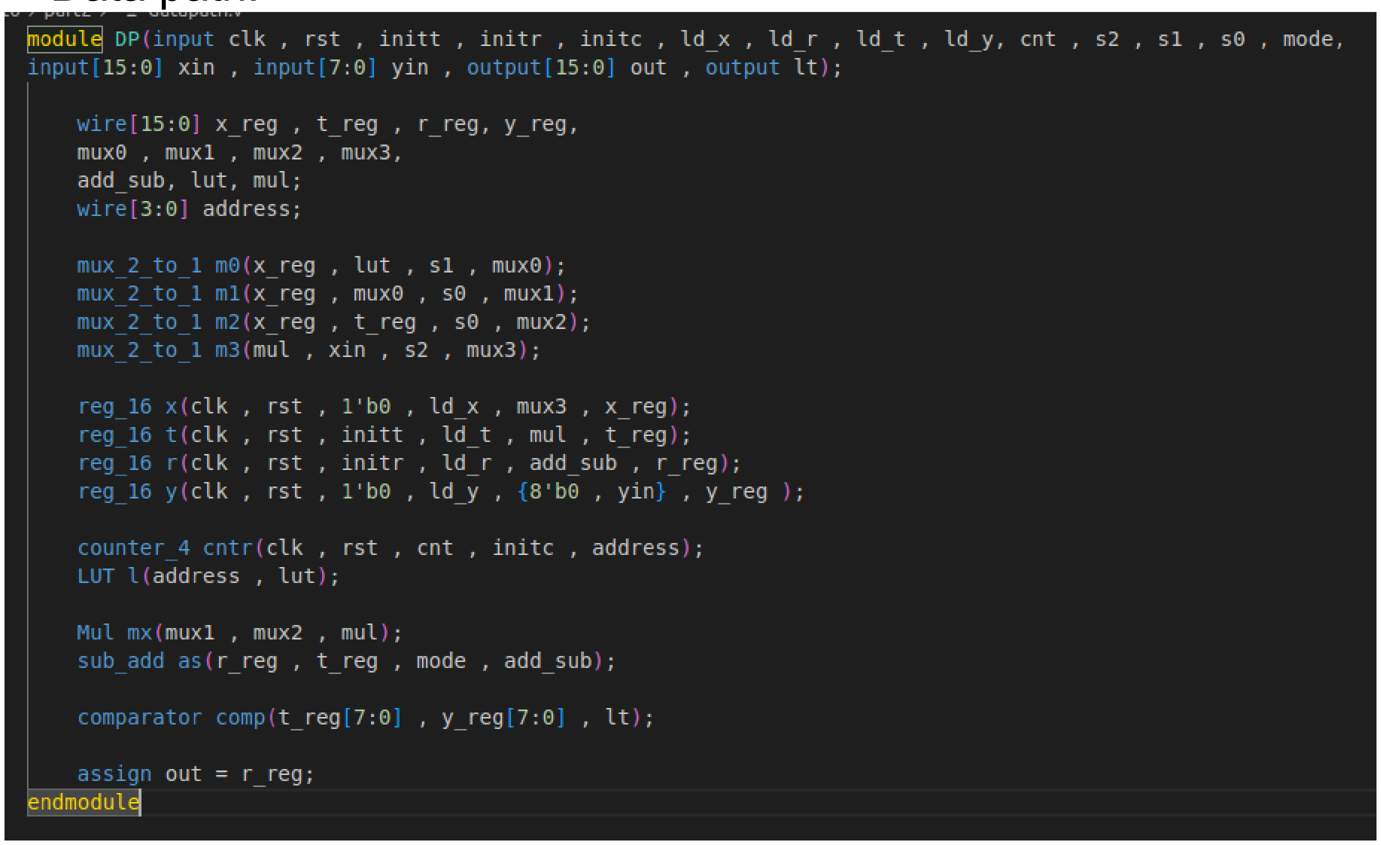
```
module reg_16 (input clk , rst , init , load , input[15:0] in , output reg[15:0] out);
    always@(posedge clk ,posedge rst) begin
        if(rst) out <= 16'b0;
        else begin
            if(init) out <= {8'b00000001 , 8'b0};
            else if(load) out <= in;
        end
    end
endmodule

module counter_4(input clk , rst , cnt , init , output reg[3:0] out);
    always@(posedge clk ,posedge  rst) begin
        if (rst) out <= 4'b0;
        else begin
            if(init) out <= 4'b0;
            else if(cnt) out <= out + 1;
        end
    end
endmodule

module mux_2_to_1 (input[15:0] in1 ,in2 , input s , output[15:0] out);
    assign out = s ? in1:in2;
endmodule

module sub_add(input[15:0] in1 , in2 , input mode , output[15:0] out);
    assign out = mode ? in1 + in2 : in1 - in2;
endmodule

module comparator(input[7:0] in1 , in2 , output lt);
    assign lt = (in1 <= in2);
endmodule

module Mul(input[15:0] in1 , in2 , output[15:0] out);
    wire[31:0] result;
    assign result = in1 * in2 ;
    assign out = result[23:8] ;
endmodule
```

LUT:



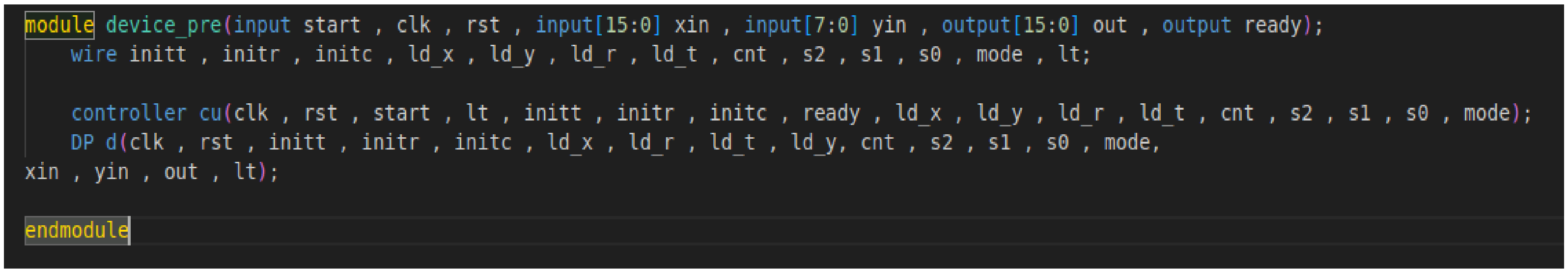
```
module LUT(input[3:0] address , output[15:0] data);
    reg [7:0] ddata;
    always@(address) begin
        case(address)
            0:ddata = 8'hFF; //1
            1:ddata = 8'h80; //1/2
            2:ddata = 8'h55; //1/3
            3:ddata = 8'h40; //1/4
            4:ddata = 8'h33;  //1/5
            5:ddata = 8'h2A; //1/6
            6:ddata = 8'h24; //1/7
            7:ddata = 8'h20; //1/8
            8:ddata = 8'h1c; //1/9
            9:ddata = 8'h19; //1/10
            10:ddata = 8'h17; //1/11
            11:ddata = 8'h15; //1/12
            12:ddata = 8'h13; //1/13
            13:ddata = 8'h12; //1/14
            14:ddata = 8'h11; //1/15
            15:ddata = 8'h10; //1/16
        endcase
    end

    assign data = {8'b0 , ddata};

endmodule
```

Data path:



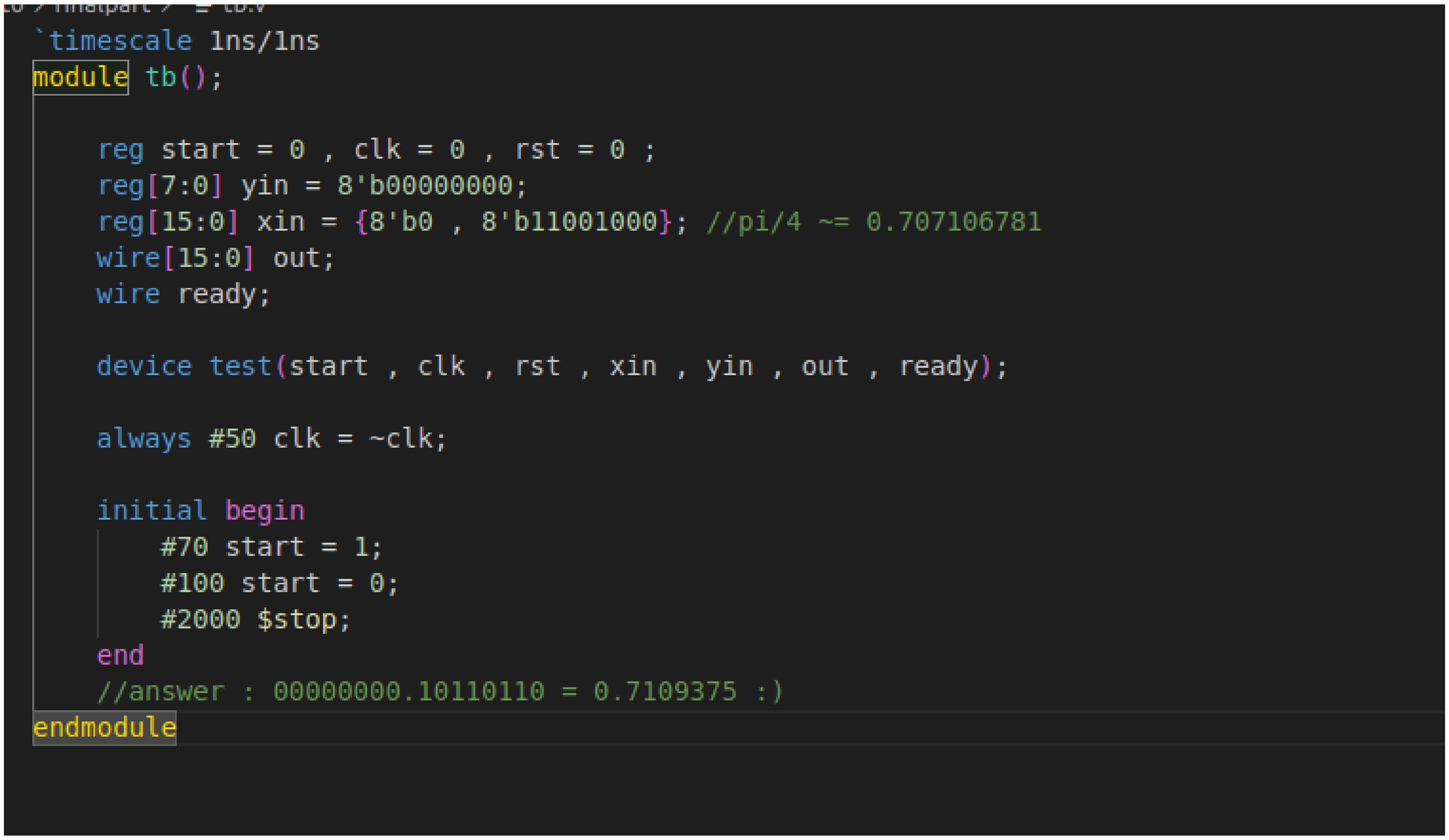
```
module DP(input clk , rst , initt , initr , initc , ld_x , ld_r , ld_t , ld_y, cnt , s2 , s1 , s0 , mode,
input[15:0] xin , input[7:0] yin , output[15:0] out , output lt);

    wire[15:0] x_reg , t_reg , r_reg, y_reg,
    mux0 , mux1 , mux2 , mux3,
    add_sub, lut, mul;
    wire[3:0] address;

    mux_2_to_1 m0(x_reg , lut , s1 , mux0);
    mux_2_to_1 m1(x_reg , mux0 , s0 , mux1);
    mux_2_to_1 m2(x_reg , t_reg , s0 , mux2);
    mux_2_to_1 m3(mul , xin , s2 , mux3);

    reg_16 x(clk , rst , 1'b0 , ld_x , mux3 , x_reg);
    reg_16 t(clk , rst , initt , ld_t , mul , t_reg);
    reg_16 r(clk , rst , initr , ld_r , add_sub , r_reg);
    reg_16 y(clk , rst , 1'b0 , ld_y , {8'b0 , yin} , y_reg );

    counter_4 cntr(clk , rst , cnt , initc , address);
    LUT l(address , lut);

    Mul mx(mux1 , mux2 , mul);
    sub_add as(r_reg , t_reg , mode , add_sub);

    comparator comp(t_reg[7:0] , y_reg[7:0] , lt);

    assign out = r_reg;
endmodule
```

Complete circuit:

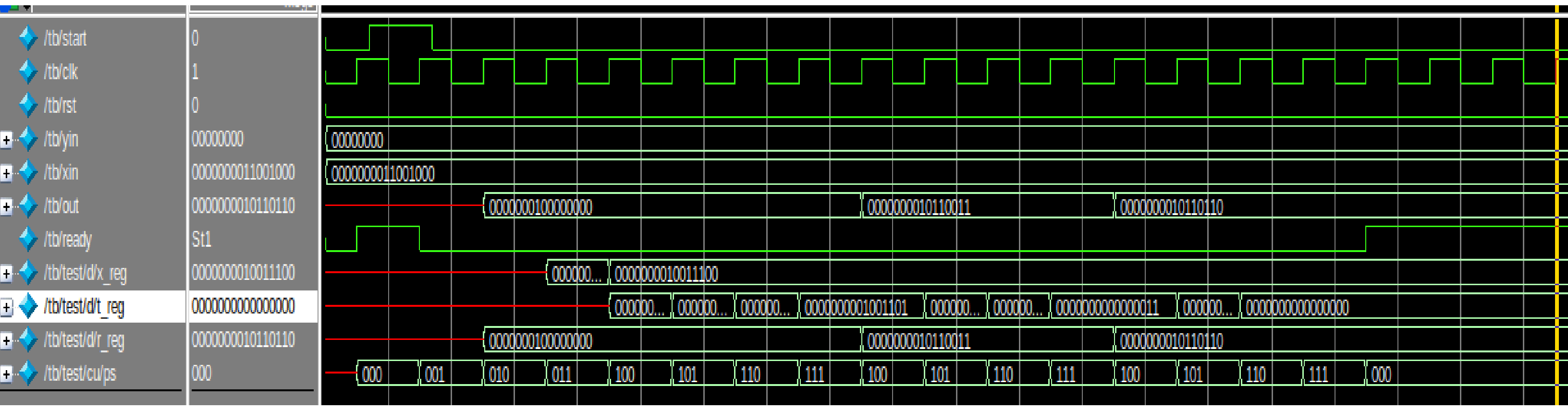


```
module device_pre(input start , clk , rst , input[15:0] xin , input[7:0] yin , output[15:0] out , output ready);
    wire initt , initr , initc , ld_x , ld_y , ld_r , ld_t , cnt , s2 , s1 , s0 , mode , lt;

    controller cu(clk , rst , start , lt , initt , initr , initc , ready , ld_x , ld_y , ld_r , ld_t , cnt , s2 , s1 , s0 , mode);
    DP d(clk , rst , initt , initr , initc , ld_x , ld_r , ld_t , ld_y, cnt , s2 , s1 , s0 , mode,
xin , yin , out , lt);

endmodule
```

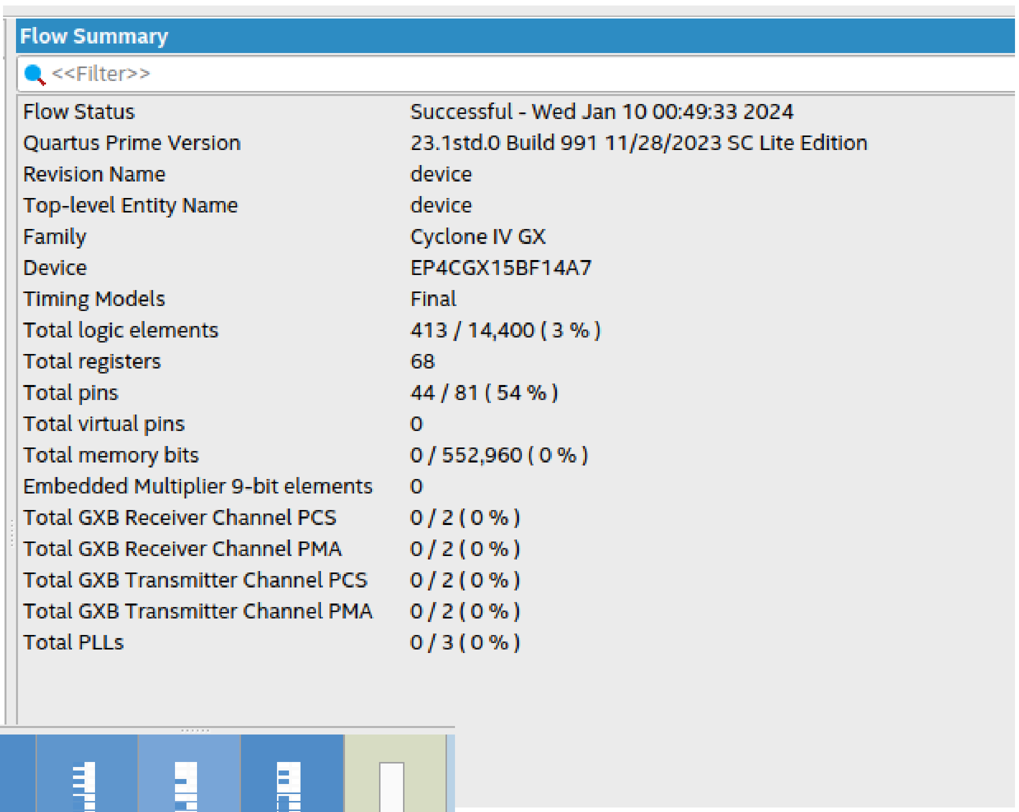
Test bench)



```
`timescale 1ns/1ns
module tb();

    reg start = 0 , clk = 0 , rst = 0 ;
    reg[7:0] yin = 8'b00000000;
    reg[15:0] xin = {8'b0 , 8'b11001000}; //pi/4 ~= 0.707106781
    wire[15:0] out;
    wire ready;

    device test(start , clk , rst , xin , yin , out , ready);

    always #50 clk = ~clk;

    initial begin
        #70 start = 1;
        #100 start = 0;
        #2000 $stop;
    end
    //answer : 00000000.10110110 = 0.7109375 :)
endmodule
```



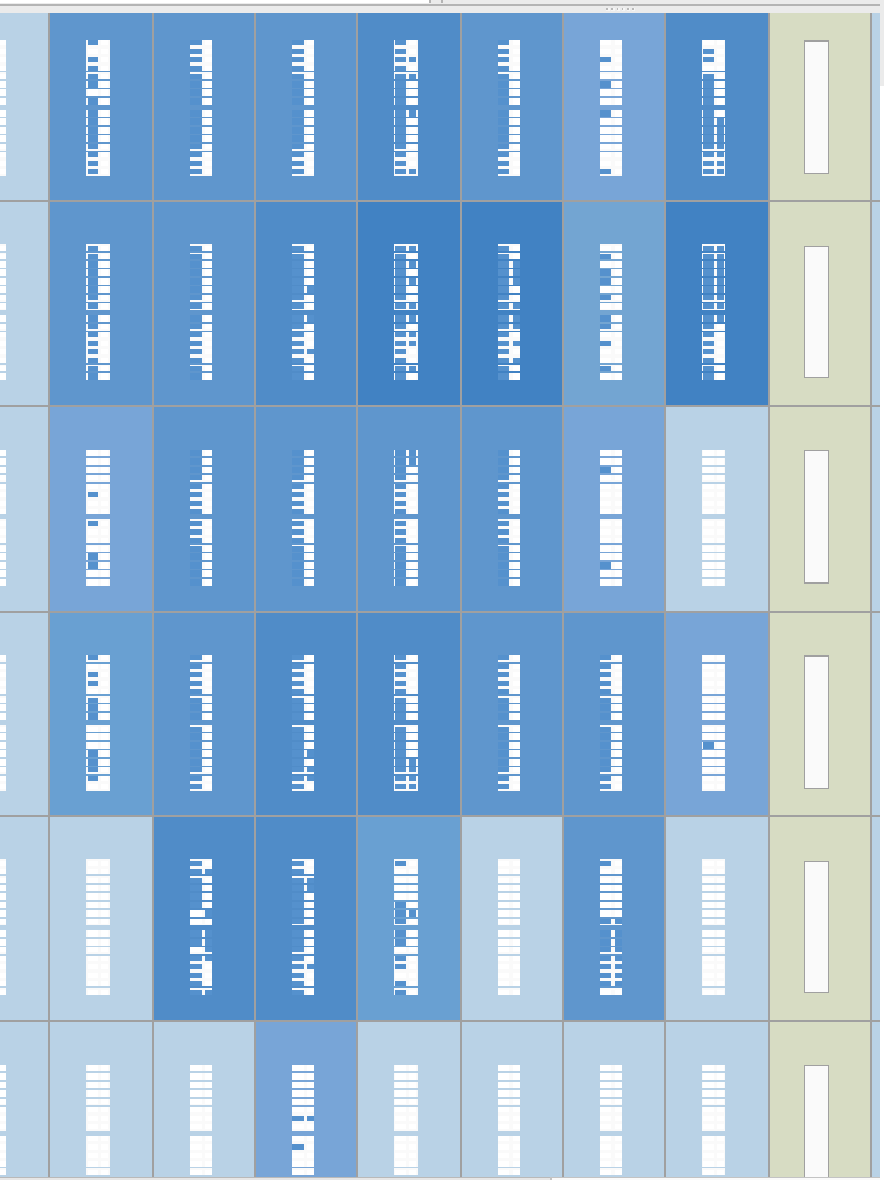
synthesize)

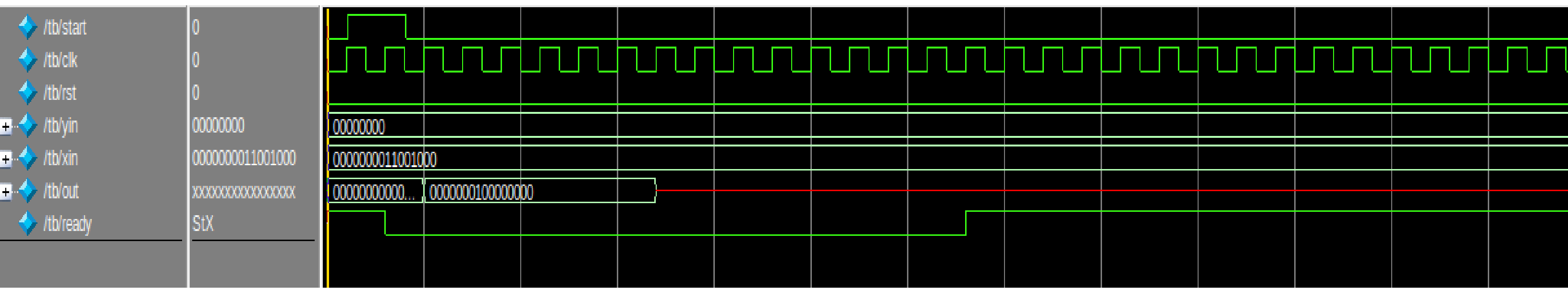


Flow Summary

<<Filter>>

Flow Status	Successful - Wed Jan 10 00:49:33 2024
Quartus Prime Version	23.1std.0 Build 991 11/28/2023 SC Lite Edition
Revision Name	device
Top-level Entity Name	device
Family	Cyclone IV GX
Device	EP4CGX15BF14A7
Timing Models	Final
Total logic elements	413 / 14,400 (3 %)
Total registers	68
Total pins	44 / 81 (54 %)
Total virtual pins	0
Total memory bits	0 / 552,960 (0 %)
Embedded Multiplier 9-bit elements	0
Total GXB Receiver Channel PCS	0 / 2 (0 %)
Total GXB Receiver Channel PMA	0 / 2 (0 %)
Total GXB Transmitter Channel PCS	0 / 2 (0 %)
Total GXB Transmitter Channel PMA	0 / 2 (0 %)
Total PLLs	0 / 3 (0 %)





```
/tb/start   0
/tb/clk     0
/tb/rst     0
/tb/yin     00000000
/tb/xin     0000000011001000
/tb/out     xxxxxxxxxxxxxxxx
/tb/ready   StX
```