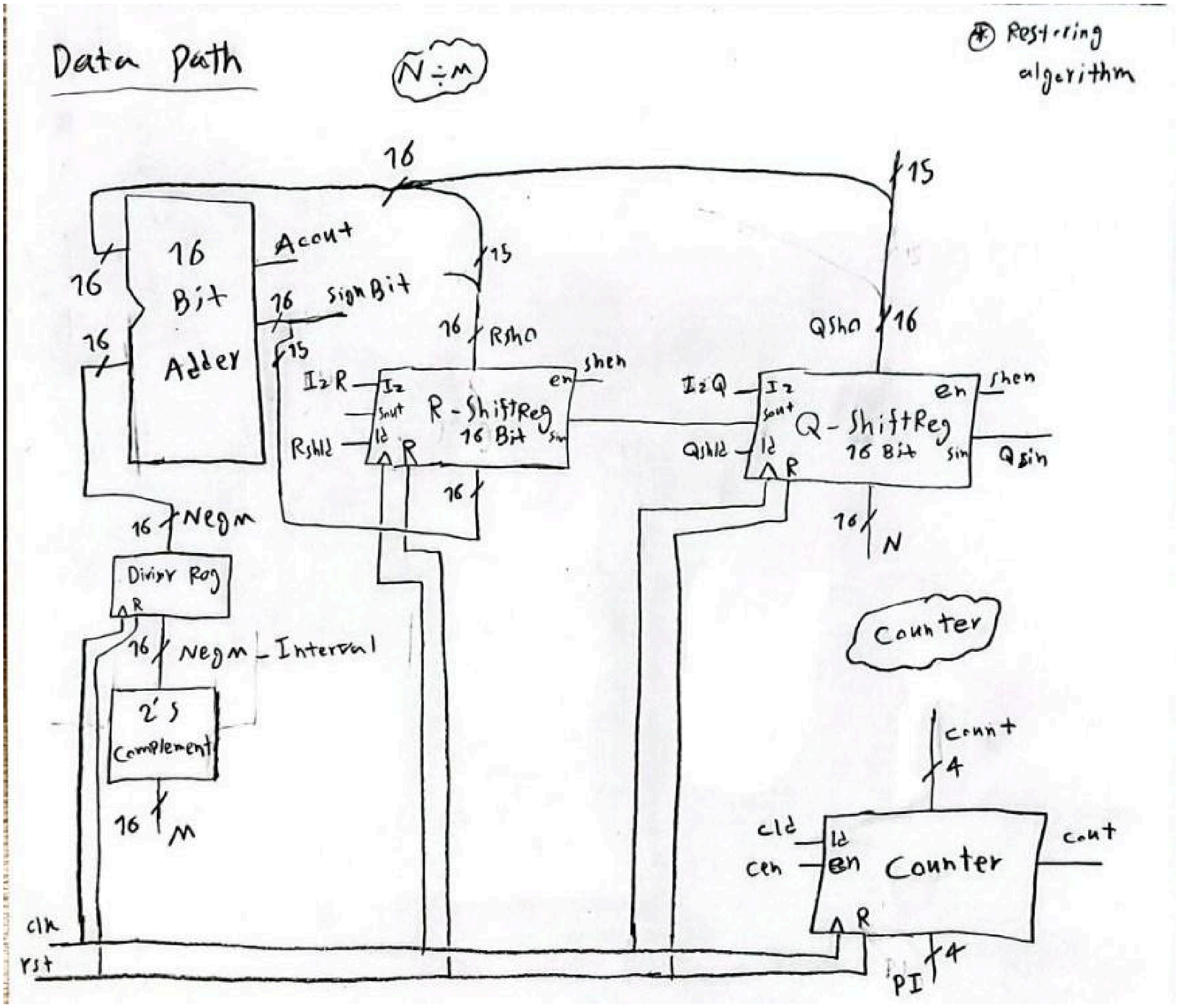
Erfan Falahati SID: 810102491

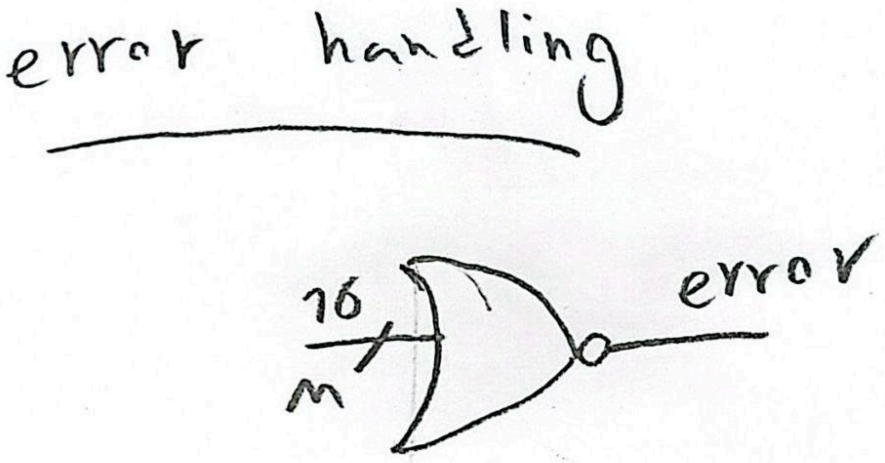
CA#5

Divider RTL

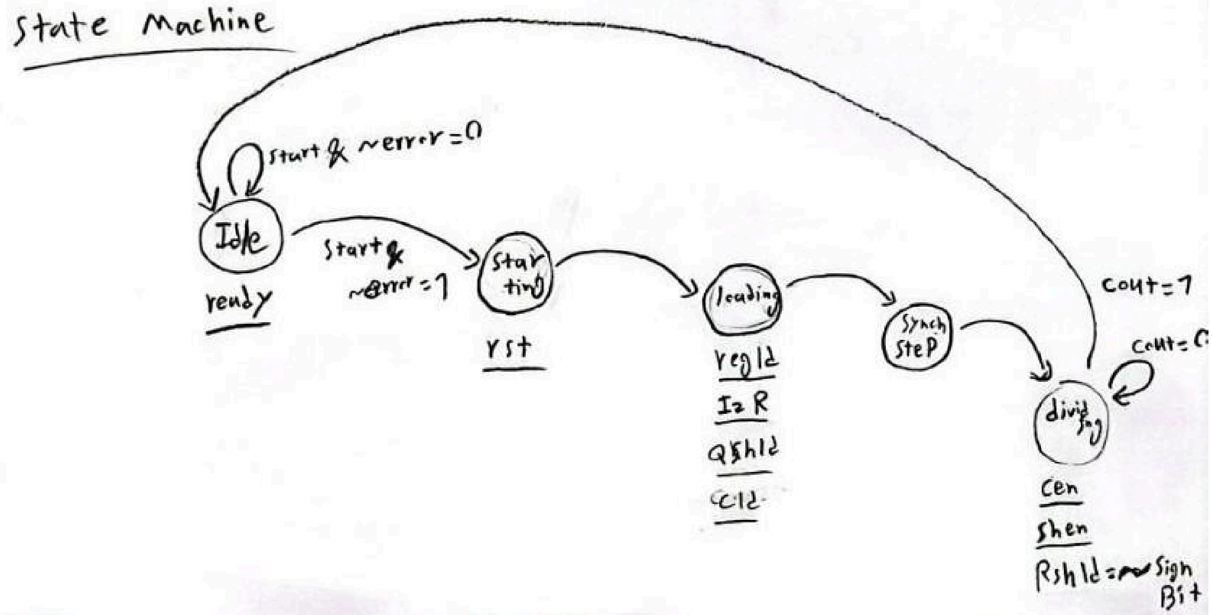
## Pre Synthesis

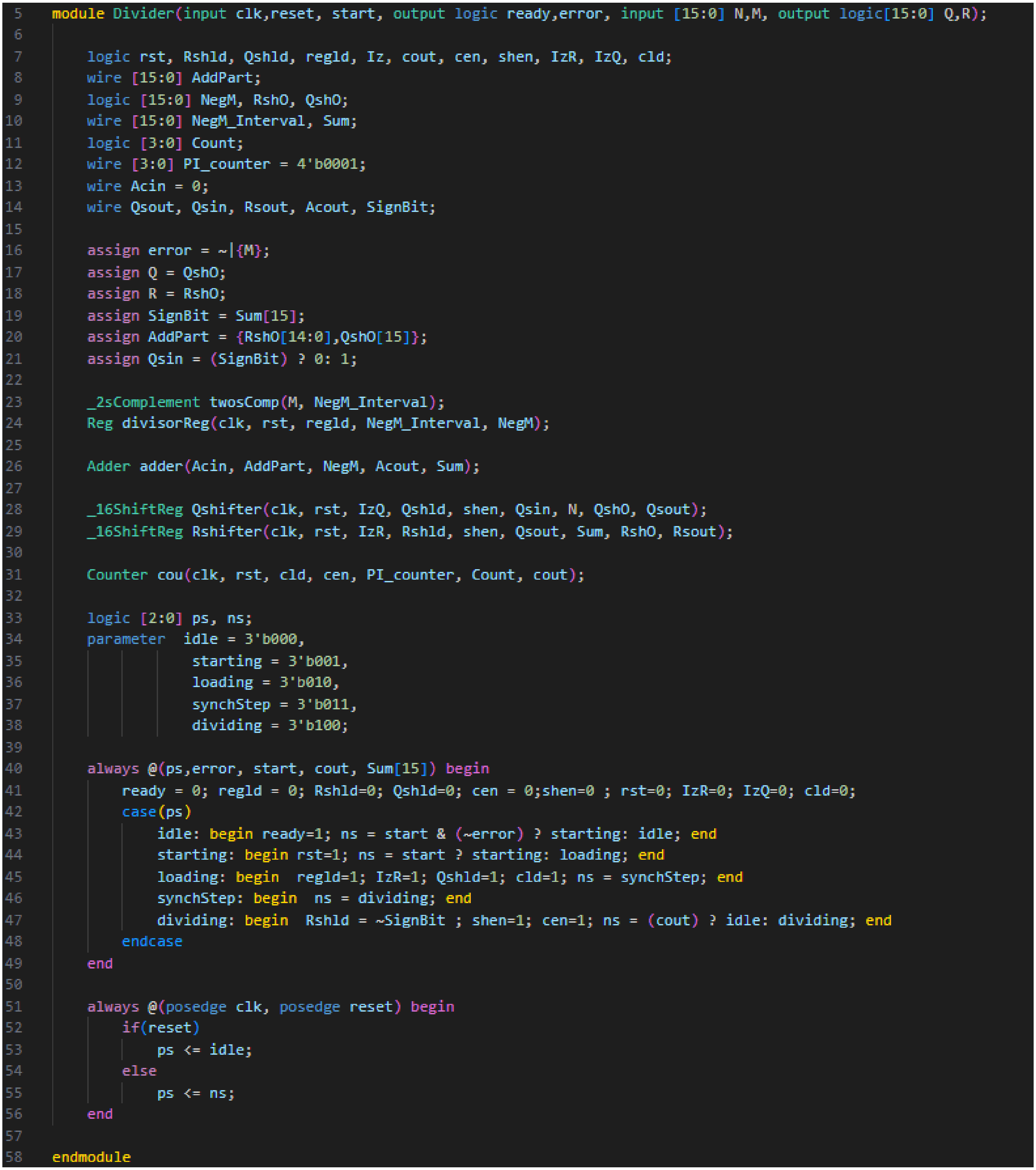
### Data Path



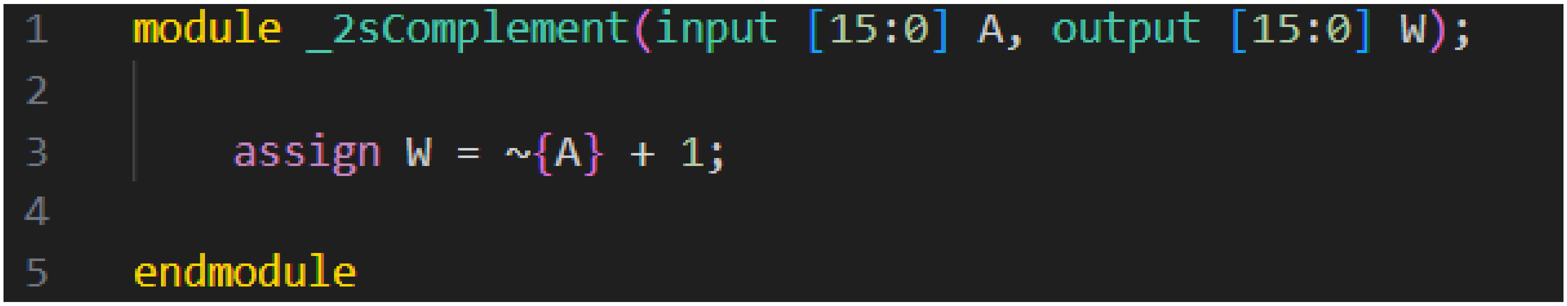

error handling
16
M
error

# Controller

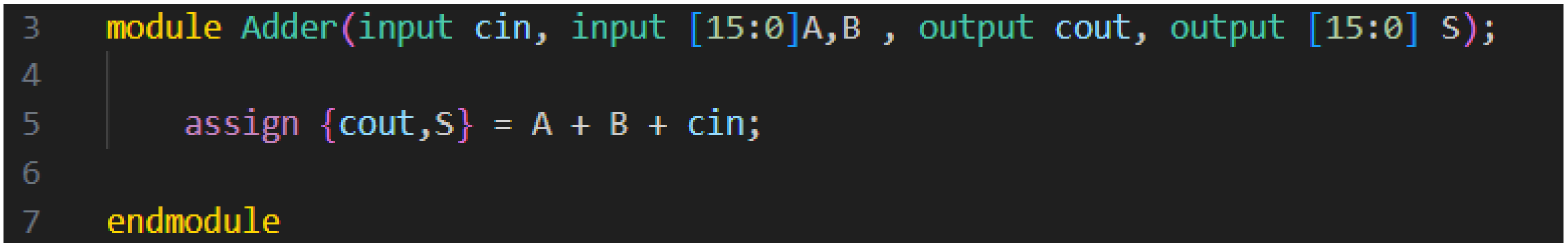

State Machine
start & ~error=0
Idle
ready
start & ~error=1
Starting
rst
loading
regld
I=R
Qshld
Cld
Synch Step
cout=1
cout=0
dividing
cen
shen
Rshld=~Sign Bit



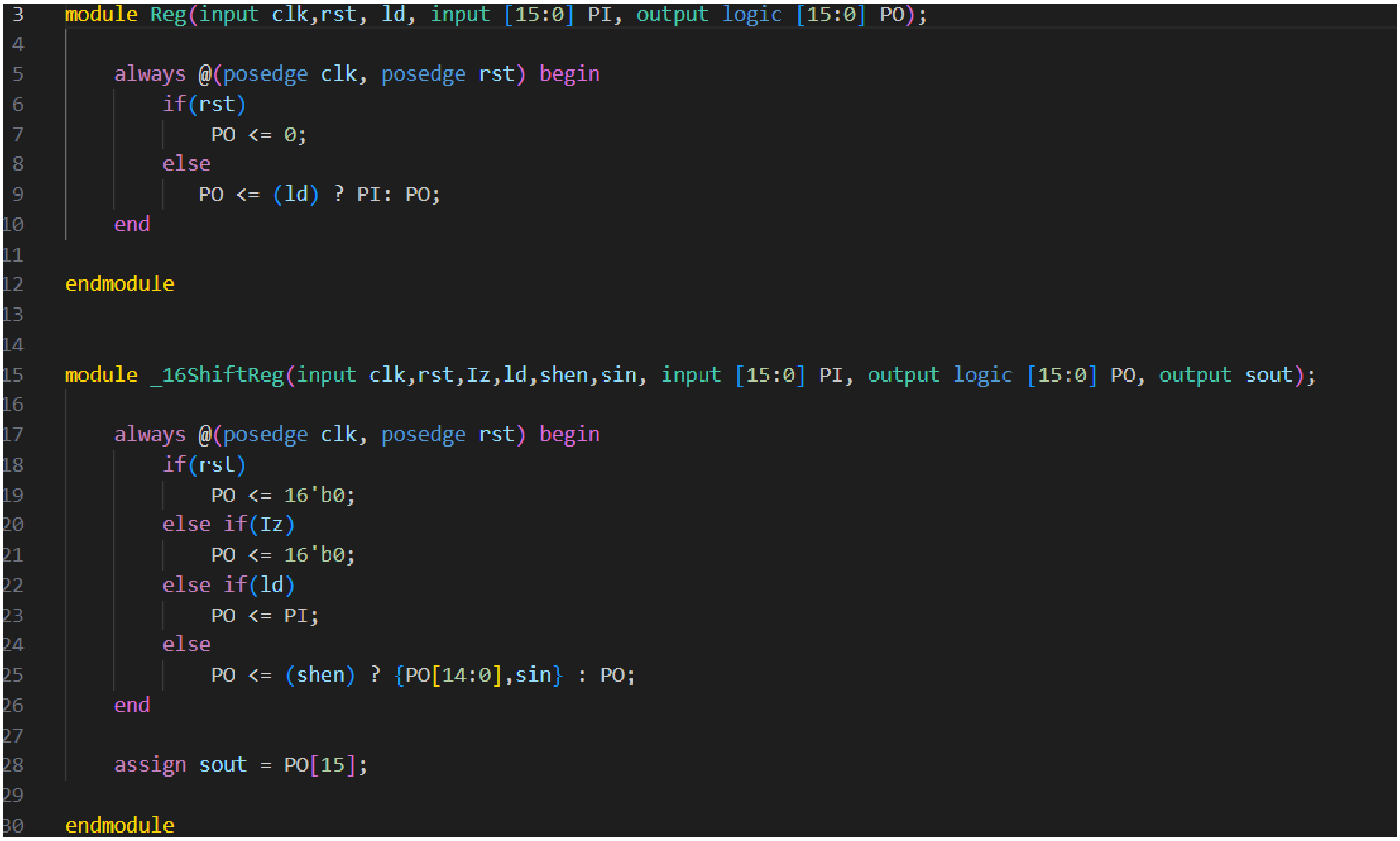
```
module Divider(input clk,reset, start, output logic ready,error, input [15:0] N,M, output logic[15:0] Q,R);

    logic rst, Rshld, Qshld, regld, Iz, cout, cen, shen, IzR, IzQ, cld;
    wire [15:0] AddPart;
    logic [15:0] NegM, RshO, QshO;
    wire [15:0] NegM_Interval, Sum;
    logic [3:0] Count;
    wire [3:0] PI_counter = 4'b0001;
    wire Acin = 0;
    wire Qsout, Qsin, Rsout, Acout, SignBit;

    assign error = ~|{M};
    assign Q = QshO;
    assign R = RshO;
    assign SignBit = Sum[15];
    assign AddPart = {RshO[14:0],QshO[15]};
    assign Qsin = (SignBit) ? 0: 1;

    _2sComplement twosComp(M, NegM_Interval);
    Reg divisorReg(clk, rst, regld, NegM_Interval, NegM);

    Adder adder(Acin, AddPart, NegM, Acout, Sum);

    _16ShiftReg Qshifter(clk, rst, IzQ, Qshld, shen, Qsin, N, QshO, Qsout);
    _16ShiftReg Rshifter(clk, rst, IzR, Rshld, shen, Qsout, Sum, RshO, Rsout);

    Counter cou(clk, rst, cld, cen, PI_counter, Count, cout);

    logic [2:0] ps, ns;
    parameter  idle = 3'b000,
                starting = 3'b001,
                loading = 3'b010,
                synchStep = 3'b011,
                dividing = 3'b100;

    always @(ps,error, start, cout, Sum[15]) begin
        ready = 0; regld = 0; Rshld=0; Qshld=0; cen = 0;shen=0 ; rst=0; IzR=0; IzQ=0; cld=0;
        case(ps)
            idle: begin ready=1; ns = start & (~error) ? starting: idle; end
            starting: begin rst=1; ns = start ? starting: loading; end
            loading: begin  regld=1; IzR=1; Qshld=1; cld=1; ns = synchStep; end
            synchStep: begin  ns = dividing; end
            dividing: begin  Rshld = ~SignBit ; shen=1; cen=1; ns = (cout) ? idle: dividing; end
        endcase
    end

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

endmodule
```



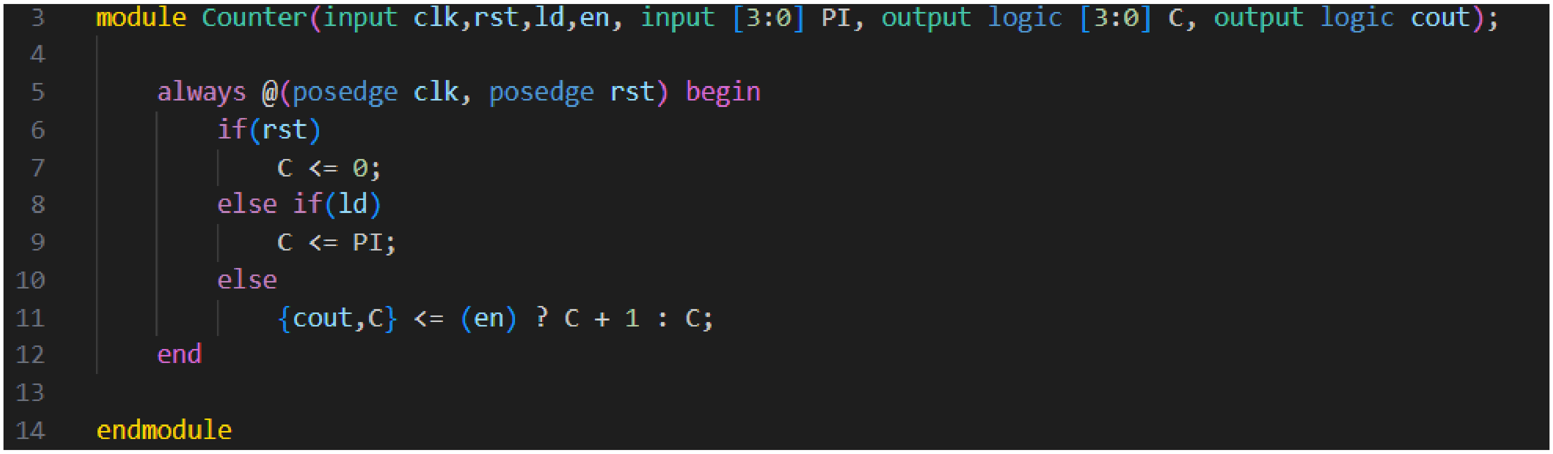
```verilog
module _2sComplement(input [15:0] A, output [15:0] W);

    assign W = ~{A} + 1;

endmodule
```



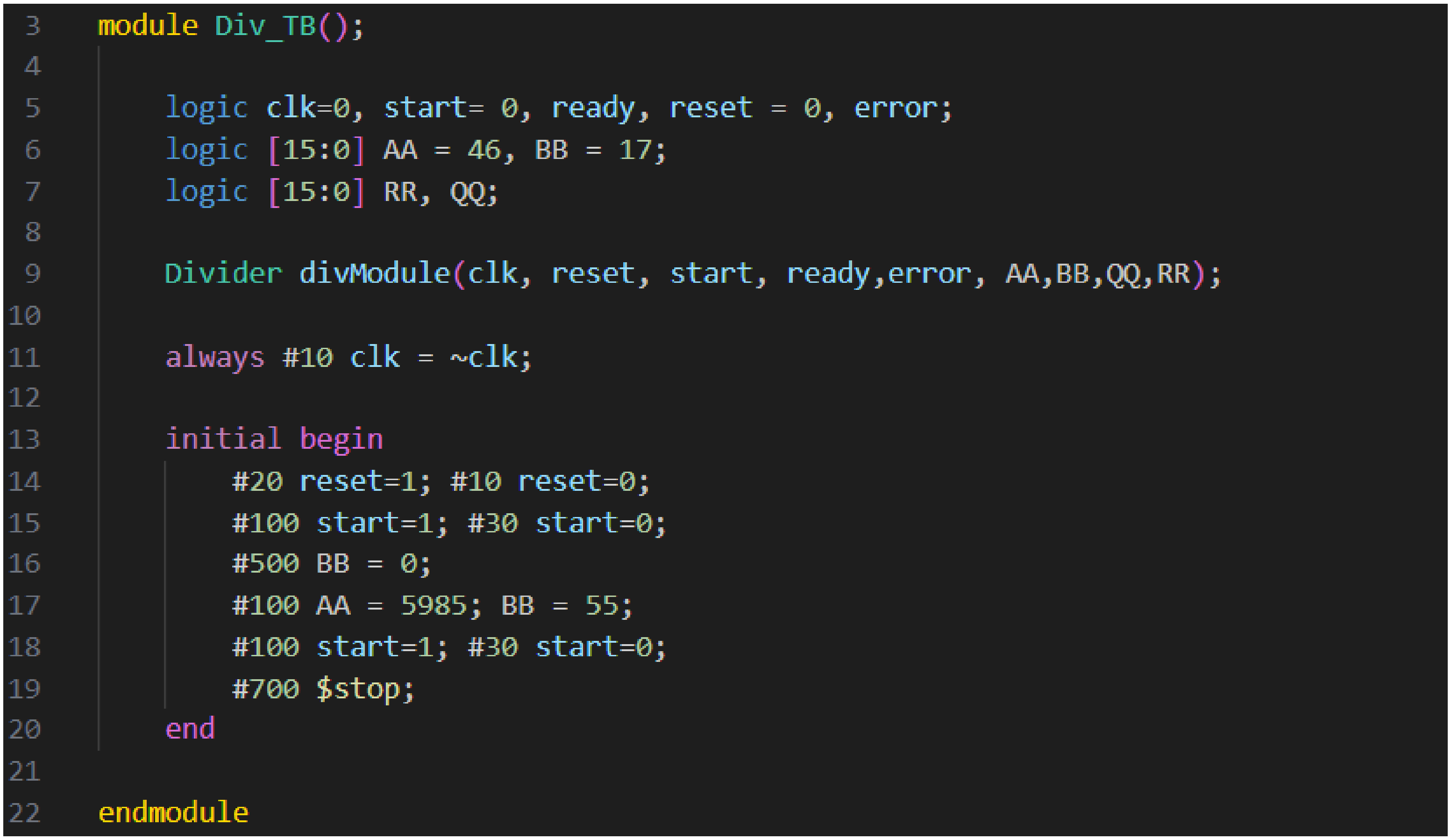
```verilog
module Adder(input cin, input [15:0]A,B , output cout, output [15:0] S);

    assign {cout,S} = A + B + cin;

endmodule
```



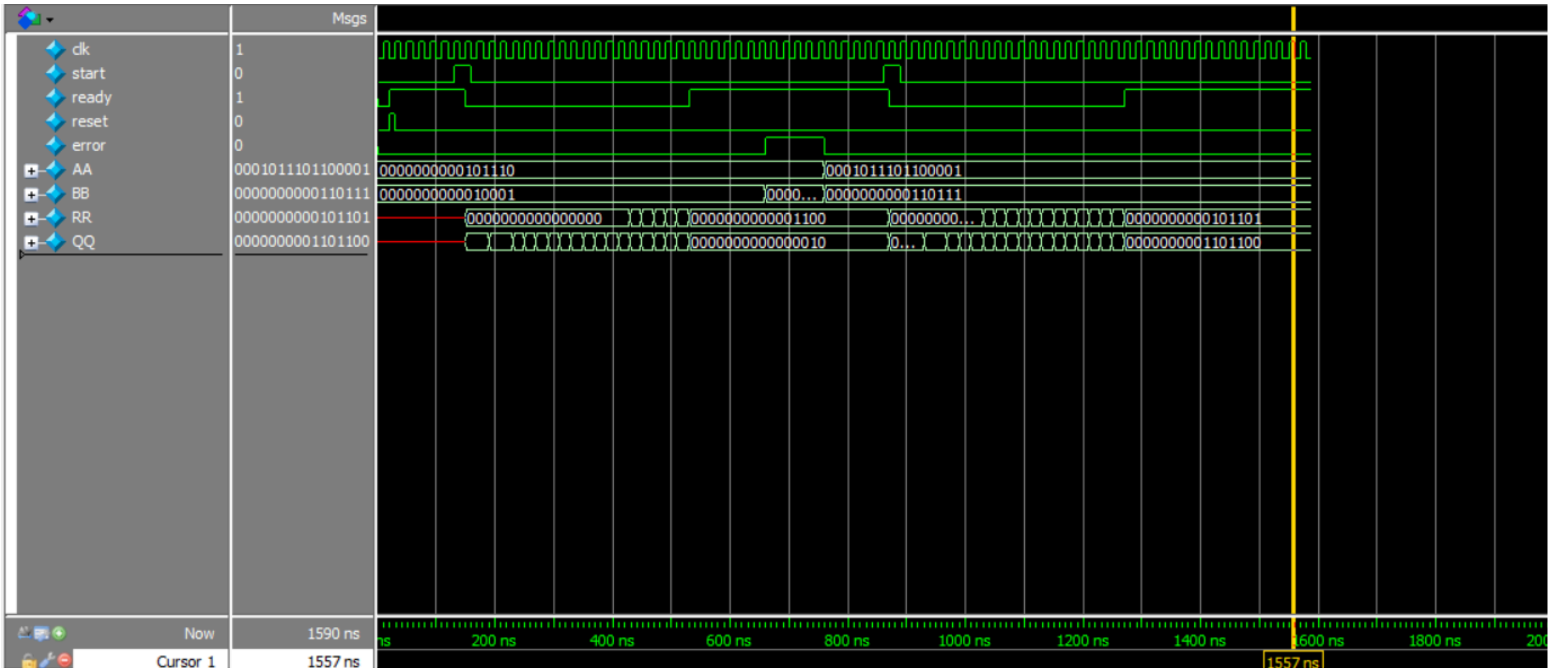
```verilog
module Reg(input clk,rst, ld, input [15:0] PI, output logic [15:0] PO);

    always @(posedge clk, posedge rst) begin
        if(rst)
            PO <= 0;
        else
           PO <= (ld) ? PI: PO;
    end

endmodule


module _16ShiftReg(input clk,rst,Iz,ld,shen,sin, input [15:0] PI, output logic [15:0] PO, output sout);

    always @(posedge clk, posedge rst) begin
        if(rst)
            PO <= 16'b0;
        else if(Iz)
            PO <= 16'b0;
        else if(ld)
            PO <= PI;
        else
            PO <= (shen) ? {PO[14:0],sin} : PO;
    end

    assign sout = PO[15];

endmodule
```



```
module Counter(input clk,rst,ld,en, input [3:0] PI, output logic [3:0] C, output logic cout);

    always @(posedge clk, posedge rst) begin
        if(rst)
            C <= 0;
        else if(ld)
            C <= PI;
        else
            {cout,C} <= (en) ? C + 1 : C;
    end

endmodule
```

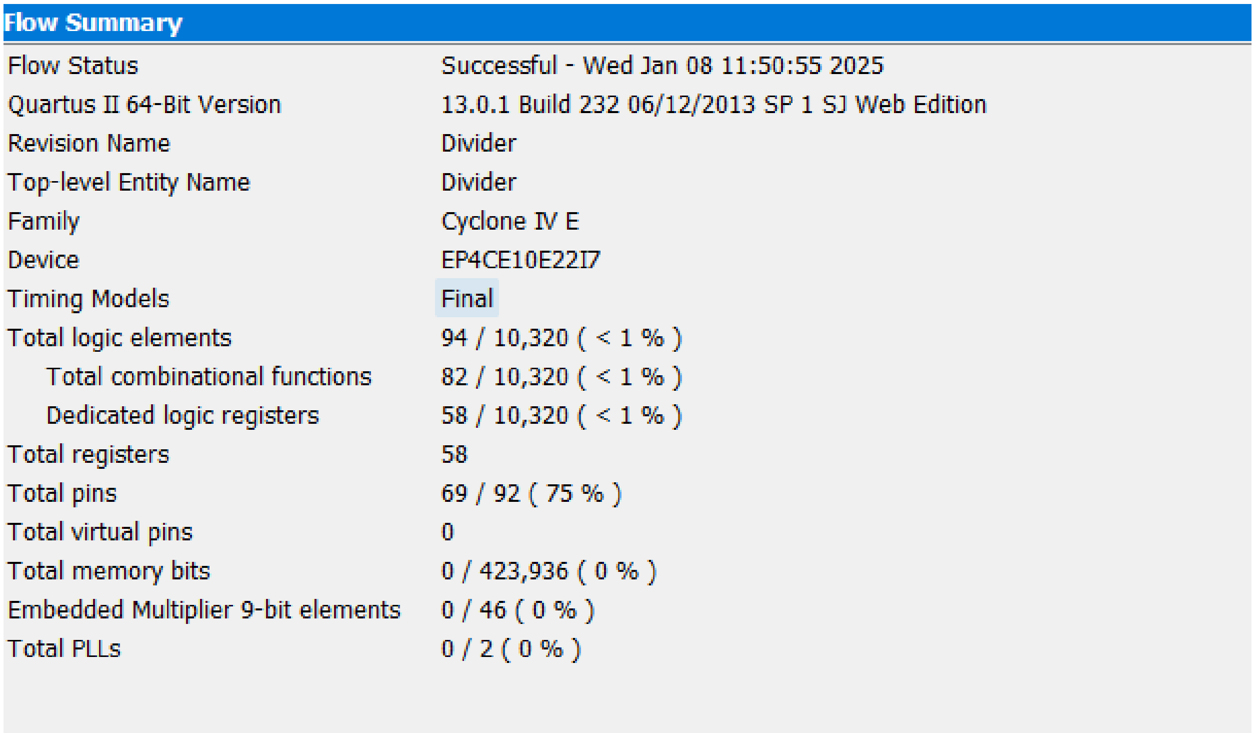


```
module Div_TB();

    logic clk=0, start= 0, ready, reset = 0, error;
    logic [15:0] AA = 46, BB = 17;
    logic [15:0] RR, QQ;

    Divider divModule(clk, reset, start, ready,error, AA,BB,QQ,RR);

    always #10 clk = ~clk;

    initial begin
        #20 reset=1; #10 reset=0;
        #100 start=1; #30 start=0;
        #500 BB = 0;
        #100 AA = 5985; BB = 55;
        #100 start=1; #30 start=0;
        #700 $stop;
    end

endmodule
```

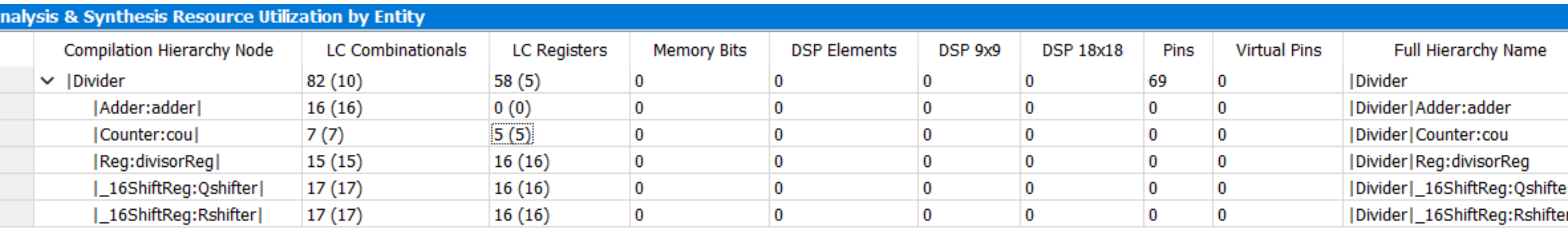


| | Msgs |
|---|---|
| clk | 1 |
| start | 0 |
| ready | 1 |
| reset | 0 |
| error | 0 |
| AA | 0001011101100001 |
| BB | 0000000000110111 |
| RR | 0000000000101101 |
| QQ | 0000000001101100 |
| Now | 1590 ns |
| Cursor 1 | 1557 ns |

# Post Synthesis

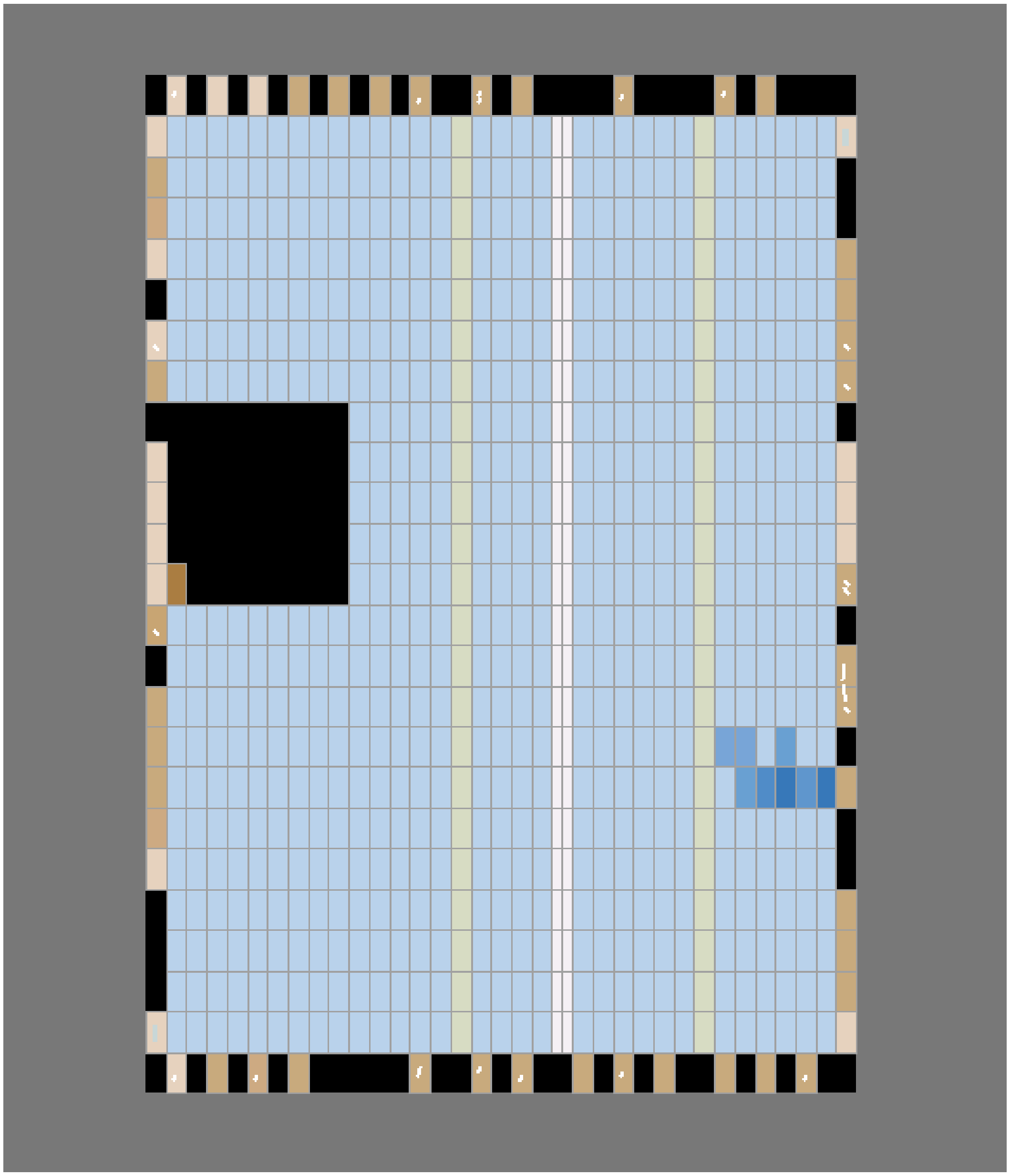


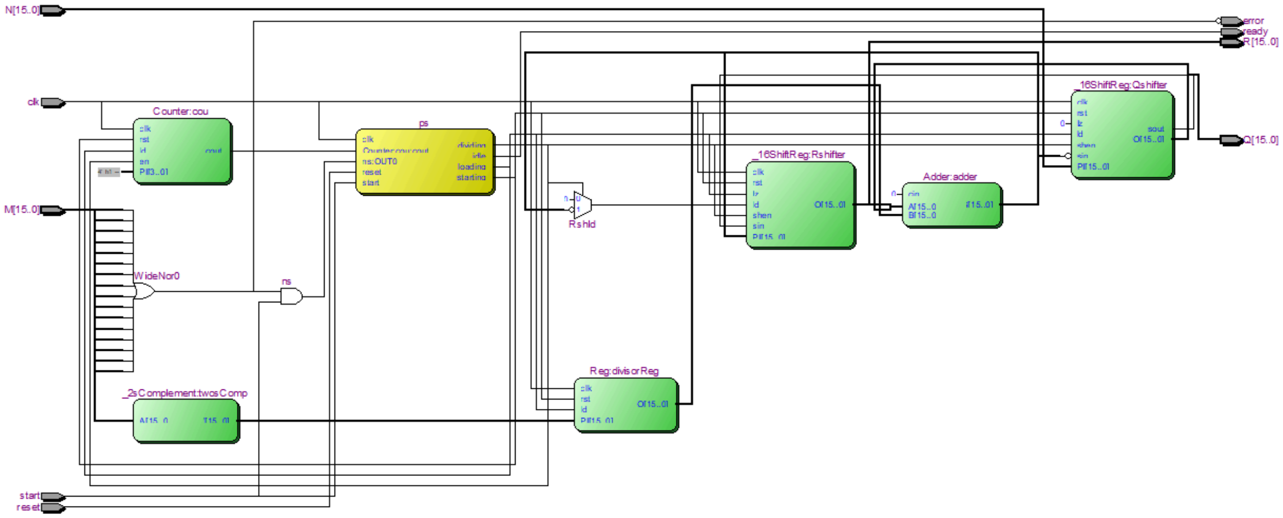
| Flow Summary | |
|---|---|
| Flow Status | Successful - Wed Jan 08 11:50:55 2025 |
| Quartus II 64-Bit Version | 13.0.1 Build 232 06/12/2013 SP 1 SJ Web Edition |
| Revision Name | Divider |
| Top-level Entity Name | Divider |
| Family | Cyclone IV E |
| Device | EP4CE10E22I7 |
| Timing Models | Final |
| Total logic elements | 94 / 10,320 ( < 1 % ) |
| Total combinational functions | 82 / 10,320 ( < 1 % ) |
| Dedicated logic registers | 58 / 10,320 ( < 1 % ) |
| Total registers | 58 |
| Total pins | 69 / 92 ( 75 % ) |
| Total virtual pins | 0 |
| Total memory bits | 0 / 423,936 ( 0 % ) |
| Embedded Multiplier 9-bit elements | 0 / 46 ( 0 % ) |
| Total PLLs | 0 / 2 ( 0 % ) |

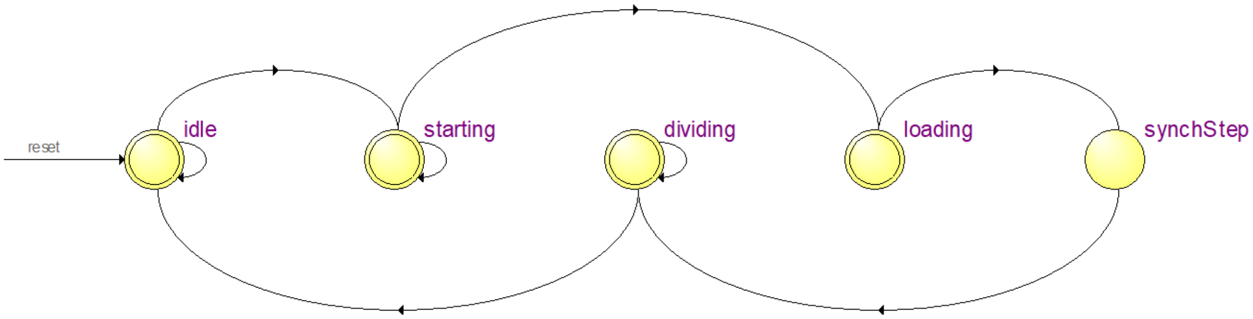


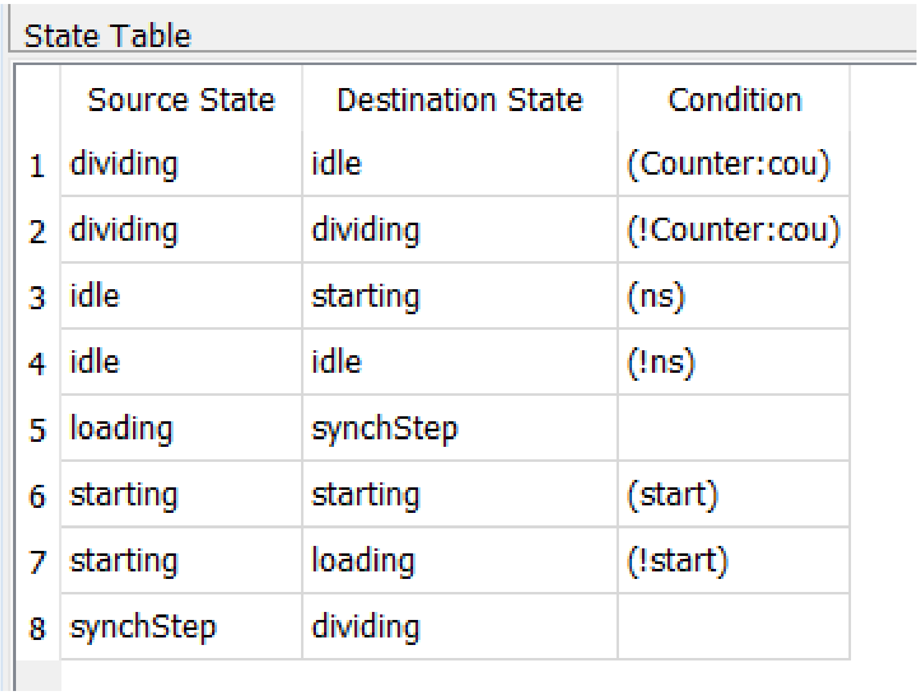
nalysis & Synthesis Resource Utilization by Entity

| Compilation Hierarchy Node | LC Combinationals | LC Registers | Memory Bits | DSP Elements | DSP 9x9 | DSP 18x18 | Pins | Virtual Pins | Full Hierarchy Name |
|---|---|---|---|---|---|---|---|---|---|
| \|Divider | 82 (10) | 58 (5) | 0 | 0 | 0 | 0 | 69 | 0 | \|Divider |
| \|Adder:adder\| | 16 (16) | 0 (0) | 0 | 0 | 0 | 0 | 0 | 0 | \|Divider\|Adder:adder |
| \|Counter:cou\| | 7 (7) | 5 (5) | 0 | 0 | 0 | 0 | 0 | 0 | \|Divider\|Counter:cou |
| \|Reg:divisorReg\| | 15 (15) | 16 (16) | 0 | 0 | 0 | 0 | 0 | 0 | \|Divider\|Reg:divisorReg |
| \|_16ShiftReg:Qshifter\| | 17 (17) | 16 (16) | 0 | 0 | 0 | 0 | 0 | 0 | \|Divider\|_16ShiftReg:Qshifte |
| \|_16ShiftReg:Rshifter\| | 17 (17) | 16 (16) | 0 | 0 | 0 | 0 | 0 | 0 | \|Divider\|_16ShiftReg:Rshifter |

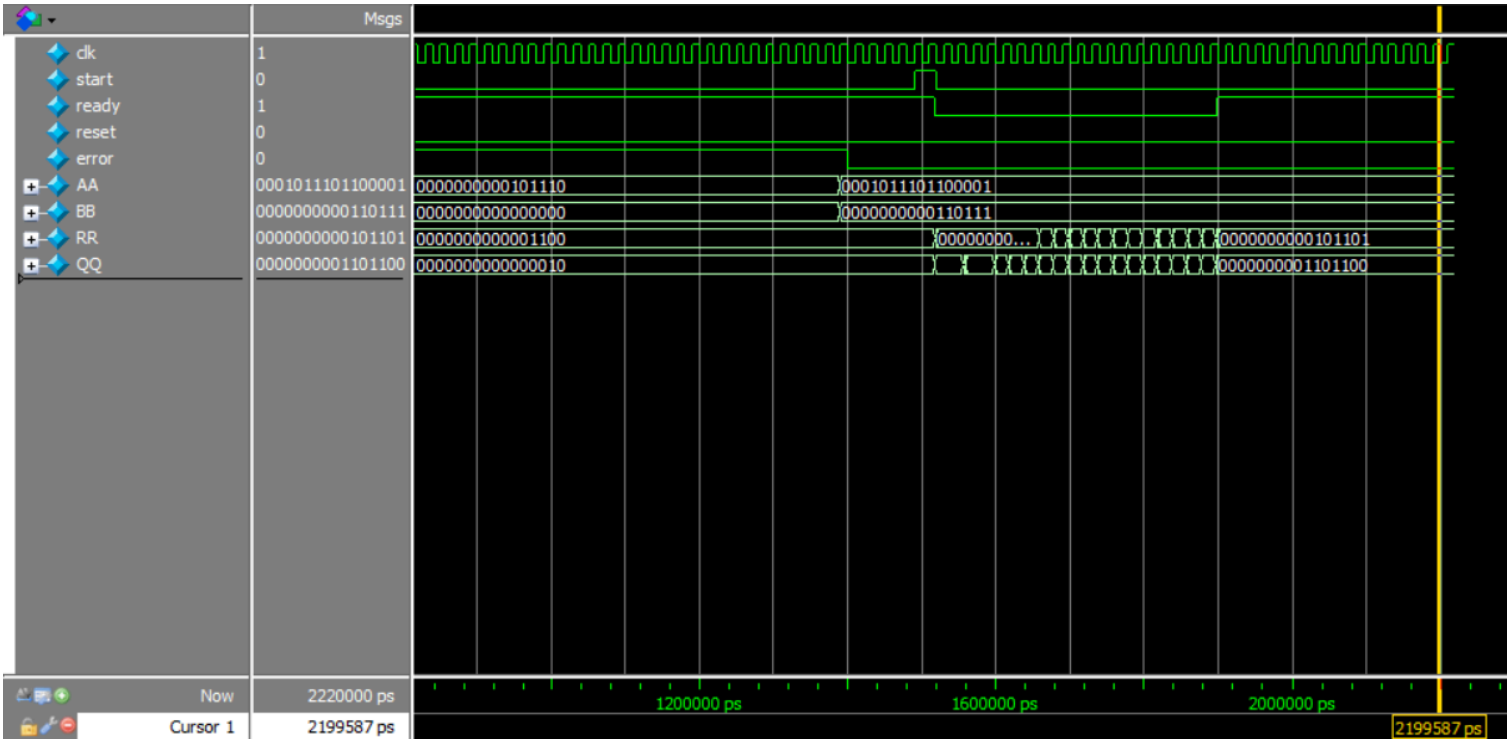







State Table

| | Source State | Destination State | Condition |
|---|---|---|---|
| 1 | dividing | idle | (Counter:cou) |
| 2 | dividing | dividing | (!Counter:cou) |
| 3 | idle | starting | (ns) |
| 4 | idle | idle | (!ns) |
| 5 | loading | synchStep | |
| 6 | starting | starting | (start) |
| 7 | starting | loading | (!start) |
| 8 | synchStep | dividing | |



```
                     Msgs
dk                   1
start                0
ready                1
reset                0
error                0
AA                   0001011101100001
BB                   0000000000110111
RR                   0000000000101101
QQ                   0000000001101100

Now                  2220000 ps
Cursor 1             2199587 ps
```