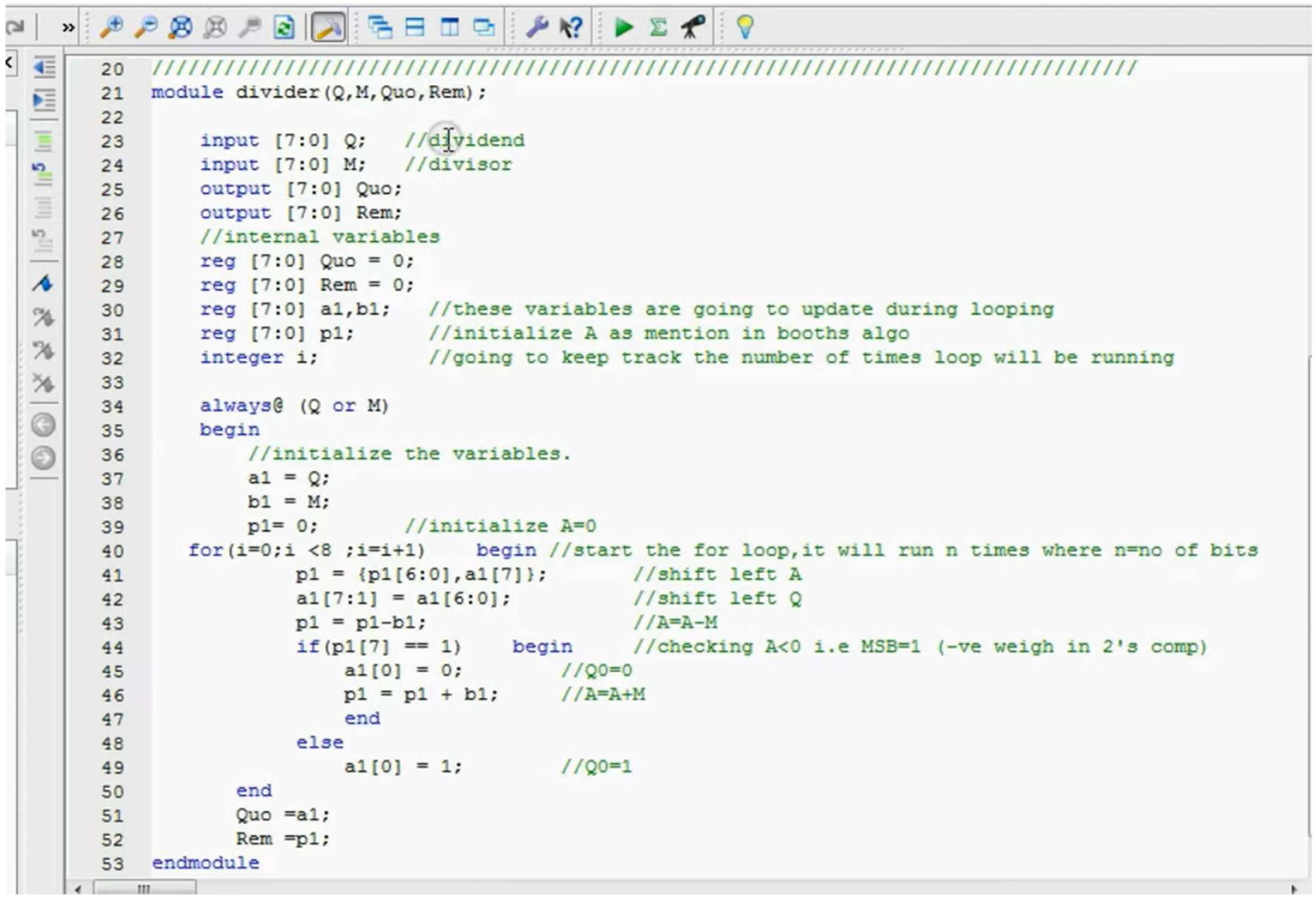
DAY #22

# 30 DAYS OF VERILOG

## AIM – TO IMPLEMENT 32 bit UNSIGNED DIVIDER

Division is a fundamental arithmetic operation we take for granted. FPGAs include dedicated hardware to perform addition, subtraction, and multiplication and will infer the necessary logic. Division is different: we need to do it ourselves. This post looks at a straightforward division algorithm for positive integers before extending it to cover fixed-point numbers and signed numbers.

CODE –

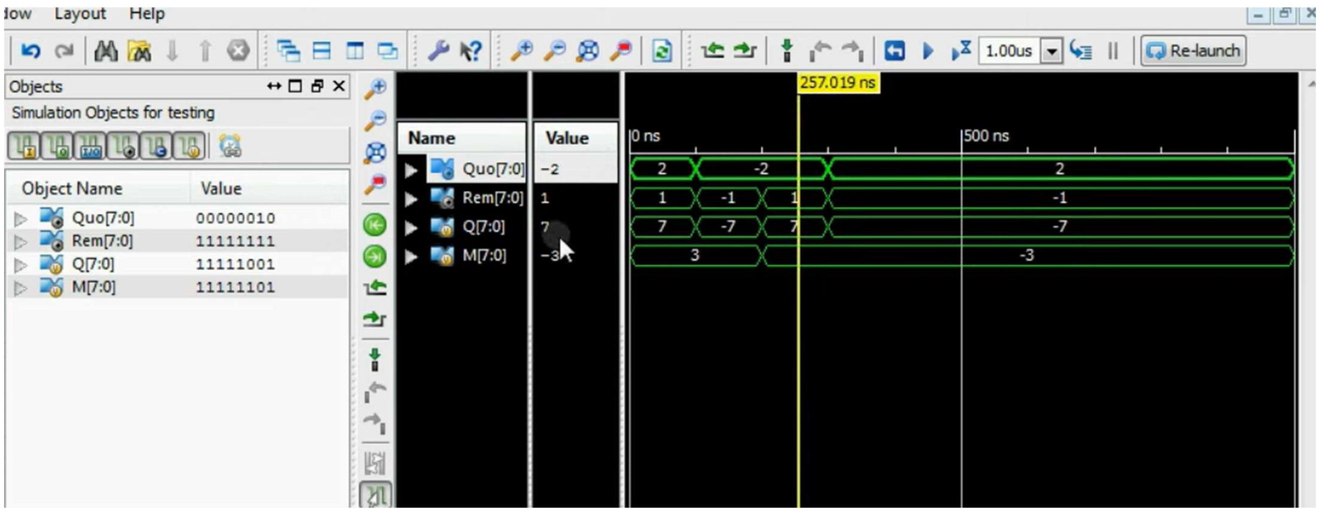


```verilog
/////////////////////////////////////////////////////////////////////////////////
module divider(Q,M,Quo,Rem);

    input [7:0] Q;    //dividend
    input [7:0] M;    //divisor
    output [7:0] Quo;
    output [7:0] Rem;
    //internal variables
    reg [7:0] Quo = 0;
    reg [7:0] Rem = 0;
    reg [7:0] a1,b1;   //these variables are going to update during looping
    reg [7:0] p1;      //initialize A as mention in booths algo
    integer i;         //going to keep track the number of times loop will be running

    always@ (Q or M)
    begin
        //initialize the variables.
        a1 = Q;
        b1 = M;
        p1= 0;        //initialize A=0
   for(i=0;i <8 ;i=i+1)    begin //start the for loop,it will run n times where n=no of bits
            p1 = {p1[6:0],a1[7]};       //shift left A
            a1[7:1] = a1[6:0];          //shift left Q
            p1 = p1-b1;                 //A=A-M
            if(p1[7] == 1)    begin     //checking A<0 i.e MSB=1 (-ve weigh in 2's comp)
                a1[0] = 0;         //Q0=0
                p1 = p1 + b1;      //A=A+M
                end
            else
                a1[0] = 1;         //Q0=1
        end
        Quo =a1;
        Rem =p1;
endmodule
```

WAVEFORM –



| Object Name | Value |
|---|---|
| Quo[7:0] | 00000010 |
| Rem[7:0] | 11111111 |
| Q[7:0] | 11111001 |
| M[7:0] | 11111101 |

| Name | Value |
|---|---|
| Quo[7:0] | -2 |
| Rem[7:0] | 1 |
| Q[7:0] | 7 |
| M[7:0] | -3 |