```verilog
`timescale 1ns / 1ps
//////////////////////////////////////////////////////////////////////////////////
// Company:
// Engineer:
//
// Create Date: 12/03/2020 03:38:25 PM
// Design Name:
// Module Name: Divisor
// Project Name:
// Target Devices:
// Tool Versions:
// Description:
//
// Dependencies:
//
// Revision:
// Revision 0.01 - File Created
// Additional Comments:
//
//////////////////////////////////////////////////////////////////////////////////


module Divider(
    input clk, Start, Clear,
    input [15:0] Input,
    output [15:0] Q_out,
    output [15:0] Div_out,
    output NPrime, Prime
    );

//    wire start;
//    Edge_Detector Starter(.clk(clk), .Btn(Start), .out(start));

    wire eq, lt, ceq, clt, ld, shift, shiftr, q, subb, sub0, add, ldr, next, c0;

//    Divider_StateMachine FSM_Divide(.clk(clk), .EQ(eq), .LT(lt), .Start(Start),
.LD(ld), .Shift(shift), .Q(q), .Sub(sub), .Add(add),
//    .NPrime(NPrime), .Prime(Prime));
    Divider_StateMachine FSM_Divide(.clk(clk), .EQ(eq), .LT(lt), .CEQ(ceq),
.CLT(clt), .Start(Start), .LD(ld), .Shift(shift), .ShiftR(shiftr),
    .Q(q), .SubB(subb), .Sub0(sub0), .Ccase(c0), .Add(add), .LDR(ldr), .Next(next));

    // Shift Registers
    wire [15:0] A;
    wire [15:0] B;
    wire [15:0] R;
    wire [15:0] Q;
    wire [15:0] subout;
    //Shift needs to happen earlier
    Shift_Register Dividend(.clk(clk), .SHin(1'b0), .SHL(shift), .LD(ld),
.Din(Input), .out(A));
    Shift_Register Remainder(.clk(clk), .SHin(A[15]), .SHL(shiftr), .LD(ldr),
.Din(subout), .R(ld), .out(R));
    Shift_Register Quotient(.clk(clk), .SHin(q), .SHL(shift), .R(next), .out(Q));


    wire [15:0] C;
    wire Count;
    wire deq, dlt;
    wire req;

    // Comparators
    Comparator BvA(.A(B), .B(Input), .EQ(deq), .LT(dlt));
    Comparator RvB(.A(R), .B(B), .EQ(eq), .LT(lt));
    Comparator Cv16(.A(C), .B(16'd16), .EQ(ceq), .LT(clt));
    Comparator Cv0(.A(C), .B(16'd0), .EQ(c0), .LT());
    Comparator Rv0(.A(R), .B(16'd0), .EQ(req), .LT());

    // Subtractor
    wire [15:0] subin;
    MUX2_1x16 selector(.in0(B), .in1(16'b0), .sel(lt), .out(subin));
    Subtractor16bit Remainder_sub(.A(R), .B(B), .out(subout));

    // Counter
    counterUD16L Counter(.clk(clk), .Up(add), .Dw(1'b0), .LD(1'b0), .din(16'b0),
.R(ld), .Q(C));

    // Divisor adder
    wire stop;
    counterUD16L Divisor_Adder(.clk(clk),
.Up(next&Start&~NPrime/*(~Prime&~NPrime)*/), .Dw(1'b0), .LD(Clear), .R(1'b0),
.din(16'd2), .Q(B));
//    FDRE #(.INIT(1'b0)) Toggle (.C(clk), .R(Clear), .CE(Prime|NPrime), .D(1'b1),
.Q(stop));

    //outputs
    assign Div_out = B;
    assign Q_out = Q;
    assign Prime = (~deq&~dlt)|((|R[15:0])&~dlt);
    assign NPrime = next&~|R;

endmodule
```