

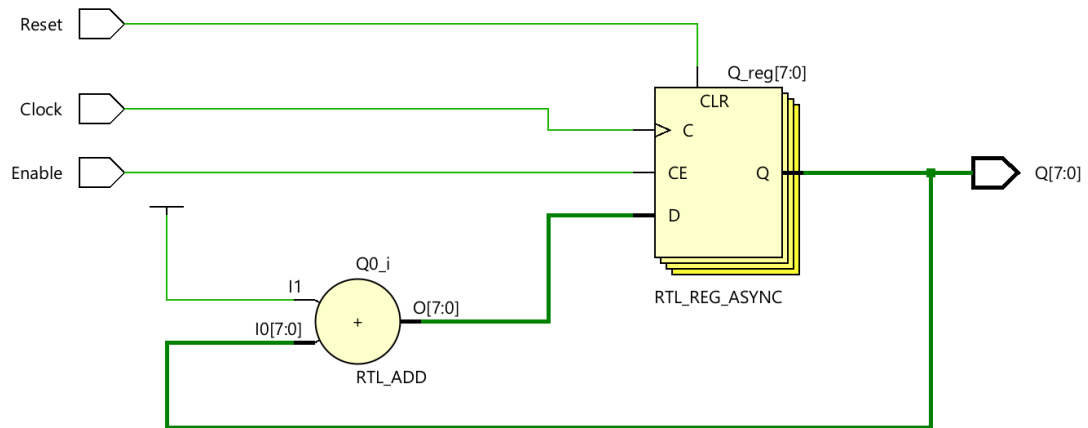
Design Code:-

```
module top (input logic Clock, Reset, Enable, output logic [7:0] Q);
    timeunit 1ns;
    timeprecision 1ns;
    always_ff @(posedge Clock or posedge Reset) begin
        if (Reset) begin
            Q <= 0;
        end
        else if (Enable) begin
            Q <= Q + 1;
        end
    end
    assert property (@(posedge Clock) Reset |-> Q == 0);
    assert property (@(posedge Clock) disable iff (Reset) !Enable |=> Q ==
$past(Q));
    assert property (@(posedge Clock) disable iff (Reset) Enable |=> Q ==
$past(Q)+8'b1);
endmodule
```

Testbench Code:-

```
module tb;
    timeunit 1ns;
    timeprecision 1ns;
    logic Clock, Reset, Enable;
    logic [7:0] Q;
    always begin
        #5 Clock = 1;
        #5 Clock = 0;
    end
    initial begin
        $dumpfile("File.vcd");
        $dumpvars(0);
    end
    initial begin
        Enable = 0;
        Reset = 1;
        #10;
        Reset = 0;
        #10;
        Enable = 1;
        #2560;
        $stop;
    end
    top dut (
        .Clock(Clock),
        .Reset(Reset),
        .Enable(Enable),
        .Q(Q)
    );
endmodule
```

Schematic:-



EP WAVEFORM:-

