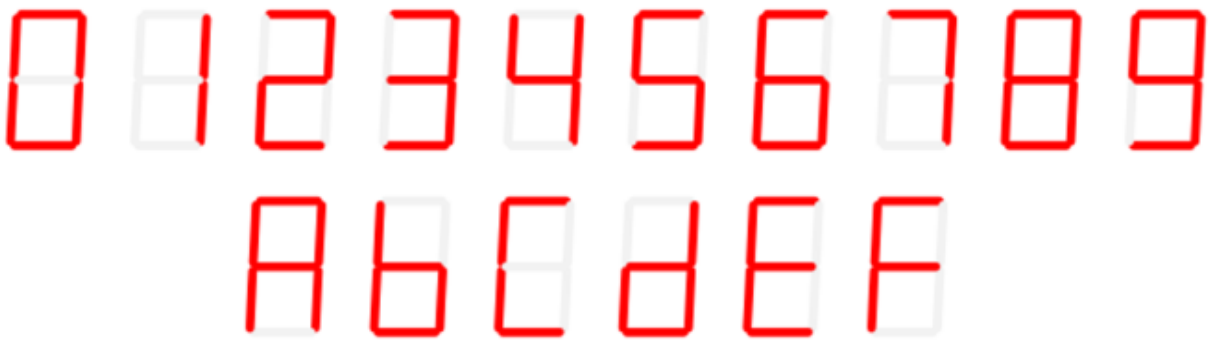
CSARCH1

Design Problem: Euler's Totient Function Counter

This is an individual exercise. Discussion between peers is allowed but showing of codes and circuit verse design will be considered cheating.

Euler's totient function (phi function), denoted as φ(n), is a function that counts the number of positive integers less than or equal to n that are relatively prime to n. Euler's totient function is used in various algorithms in particular, the RSA algorithm used in cryptography.

You need to design a circuit that calculates Euler's totient function for the first 16 numbers and displays the results on a 7-segment display. The BCD display will change to the next number at the rising edge of the clock. The first 16 numbers of the Euler's totient function are [1,1,2,2,4,2,6,4,6,4,10,4,12,6,8,8]. Display 10 and 12 as A and C. so the display should be [1,1,2,2,4,2,6,4,6,4, A,4, C,6,8,8]. It will then repeat, showing the pattern again. An input R=1, resets the sequence to the first element. Provide an output for your 7-segment as A, B,C, D, E, F, G, similar to the Design Exercise 1.



Grading System:

- Running Verilog Design [80%]
- Documentation [20%]
- Bonus 20 points: Instead of going back immediately from the first display, it will display them first, reverse, and then repeat the pattern again. Example: [1,1,2,2,4,2,6,4,6,4,A,4,C,6,8,8,8,8,6,C,4,A,4,6,4,6,2,4,2,2,1,1]

Testbench:

```
`timescale 1ns / 1ps

module TestBench();
  reg R, clk_0;
  wire A, B, C, D, E, F, G;

  EulerTotient DUT0(R, clk_0, A, B, C, D, E, F, G);

  always begin
    clk_0 = 0;
    forever #1000 clk_0 = ~clk_0;
  end

  always @(posedge clk_0) begin
                R=0;
                #100
                R=1;
                #2000
                $display("R=%b clk_0=%b ABCDEFG = %b%b%b%b%b%b%b", R, clk_0, A, B, C, D, E ,F, G); // 1
                #2000
                $display("R=%b clk_0=%b ABCDEFG = %b%b%b%b%b%b%b", R, clk_0, A, B, C, D, E ,F, G); // 1
                #2000
                $display("R=%b clk_0=%b ABCDEFG = %b%b%b%b%b%b%b", R, clk_0, A, B, C, D, E ,F, G); // 1
                R=0;
                #2000
                $display("R=%b clk_0=%b ABCDEFG = %b%b%b%b%b%b%b", R, clk_0, A, B, C, D, E ,F, G); // 1
                #2000
                $display("R=%b clk_0=%b ABCDEFG = %b%b%b%b%b%b%b", R, clk_0, A, B, C, D, E ,F, G); // 2
                #2000
                $display("R=%b clk_0=%b ABCDEFG = %b%b%b%b%b%b%b", R, clk_0, A, B, C, D, E ,F, G); // 2
                #2000
                $display("R=%b clk_0=%b ABCDEFG = %b%b%b%b%b%b%b", R, clk_0, A, B, C, D, E ,F, G); // 4
                #2000
                $display("R=%b clk_0=%b ABCDEFG = %b%b%b%b%b%b%b", R, clk_0, A, B, C, D, E ,F, G); // 2
                #2000
                $display("R=%b clk_0=%b ABCDEFG = %b%b%b%b%b%b%b", R, clk_0, A, B, C, D, E ,F, G); // 6
                #2000
                $display("R=%b clk_0=%b ABCDEFG = %b%b%b%b%b%b%b", R, clk_0, A, B, C, D, E ,F, G); // 4

                /*add more tests as needed*/
     $finish;
  end

endmodule
```