Lab#6
Prime number decremental counter with 4-Bits, FFT

Digital design principles.

Teacher:
Gloria Castro Muñoz

Work made by:
Christian Aaron Ortega Blanco

Hermosillo, Son. 11 de mayo de 2022.

The prime numbers are those who only were divisible by 1 or by themselves, if we want to represent a decremental count of prime number in a 4-Bit format, firstly it's necessary to find out which is the biggest number that its possible to do with 4-bits (thirteen, 1101). The rest numbers are as follow the next table.

13	.1101.
11	.1011.
7	.0111.
5	.0101.
3	.0011.
2	.0010.

Table 1 prime numbers in 4-bit format

The next step is the next state table, it must contain the present and the next state that leads the next prime number.

fa	fb	fc	fd	qa	qb	qc	qd
0	0	0	0	x	x	x	x
0	0	0	1				
0	0	1	0	1	1	0	1
0	0	1	1	0	0	1	0
0	1	0	0				
0	1	0	1	0	0	1	1
0	1	1	0				
0	1	1	1	0	1	0	1
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1	0	1	1	1
1	1	0	0				
1	1	0	1	1	0	1	1
1	1	1	0				
1	1	1	1				

Table 2 descendent count state table 4-bits

From the table 2, fa, fb, fc and fd represent the present state on the circuit and qa, ,qb, qc, qd are the next state, notice that only the green rows has an future state, that means that the rest of the possible combinations were omitted.

To implement the circuit with T flip-flops, its necessary to add another column to compare "f" and "q" with the state table of the T flip-flops.

T	Q	Q'
0	0	0
1	0	1
0	1	0
1	1	0

Table 3 state table of the T-flipflop

Following the next table, we obtain the next resulting column.

fa	fb	fc	fd	qa	qb	qc	qd	ta	tb	tc	td
0	0	0	0	x	x	x	x	x	x	x	x
0	0	0	1								
0	0	1	0	1	1	0	1	1	1	1	1
0	0	1	1	0	0	1	0	0	0	0	1
0	1	0	0								
0	1	0	1	0	0	1	1	0	1	1	0
0	1	1	0								
0	1	1	1	0	1	0	1	0	0	1	0
1	0	0	0								
1	0	0	1								
1	0	1	0								
1	0	1	1	0	1	1	1	1	1	0	0
1	1	0	0								
1	1	0	1	1	0	1	1	0	1	1	0
1	1	1	0								
1	1	1	1								

Table 4 next state table

Now we can obtain the Boolean expression using every column ta, tb, tc, td, those represent the input for each flipflop.

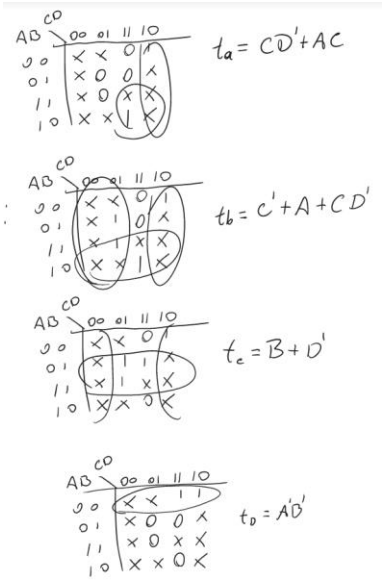


Figure 1 Karnaugh simplification

Its recommendable to test the behavior of the circuit using any simulation software.

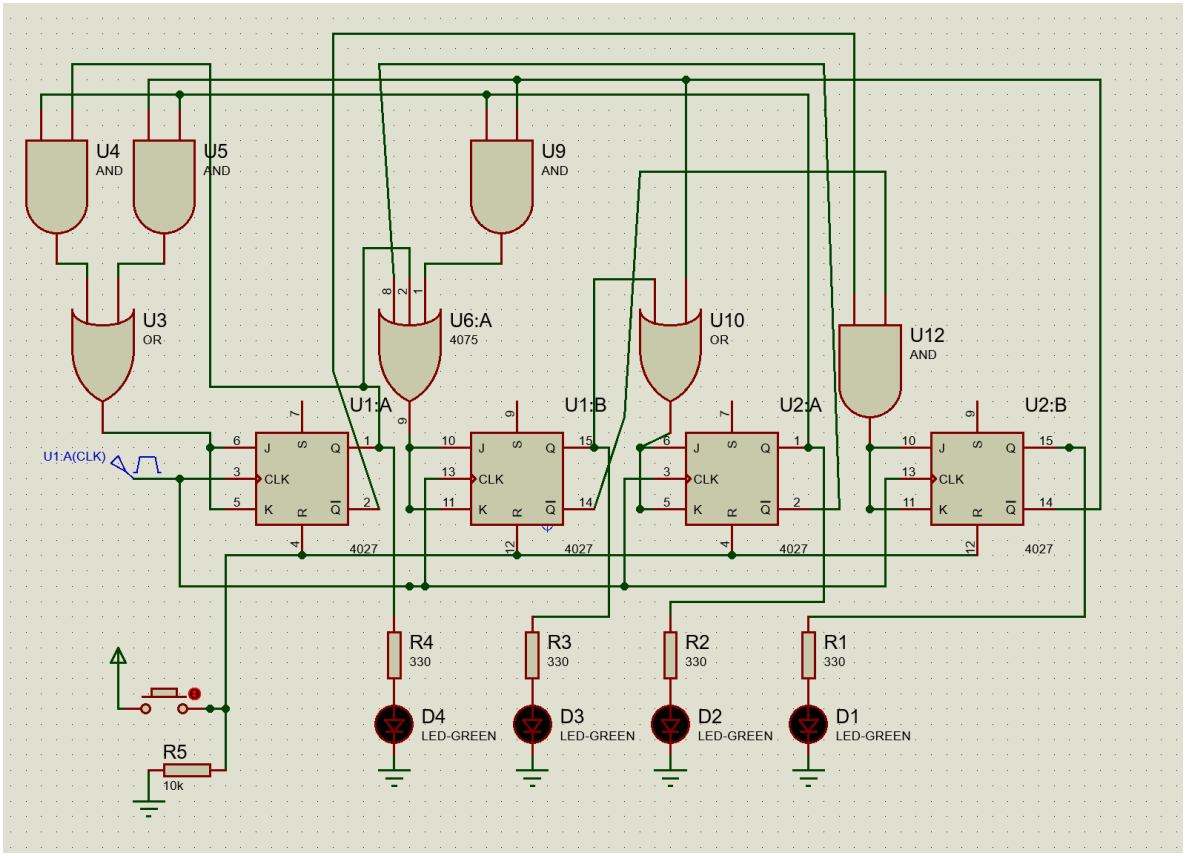


Figure 2 descendent prime number counter

Then if every number is displayed correctly, the circuit is converted into a Verilog scrip as bellow;

Module definition

```
module upcount(clk,rst,Count);
  input rst,clk;
  output [3:0] Count;
  wire TA,TB,TC,TD,QA,QB,QC,QD,QAneg,QBneg,QCneg,QDneg,n1,n2,n3;


  Tflipflop FFA(clk,rst,TA,QA,QAneg);
  Tflipflop FFB(clk,rst,TB,QB,QBneg);
  Tflipflop FFC(clk,rst,TC,QC,QCneg);
  Tflipflop FFD(clk,rst,TD,QD,QDneg);

  And_gate_2 U1(QC,QDneg,n1);
  And_gate_2 U2(QA,QC,n2);
  And_gate_2 U3(QC,QDneg,n3);
   And_gate_2 U4(QAneg,QBneg,TD);

  Or_gate_2 U5(n1,n2,TA);
  Or_gate_2 U6(QB,QDneg,TC);

  Or_gate U7(QCneg,QA,n3,TB);


  assign Count = {QA,QB,QC,QD};
endmodule
```

Bench test

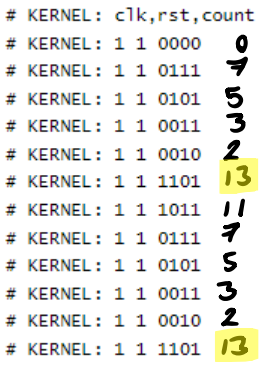
```
`timescale 1ns/1ps
`include "Or_gate_2.sv"
`include "And_gate_2.sv"
`include "Or_gate.sv"
`include "And_gate.sv"
`include "FFT.sv"

module upcount_TB;
  reg clk,rst;
  wire [3:0] count;

  upcount UUT(clk,rst,count);

  initial begin
     $dumpfile("upcount.vcd");
     $dumpvars(1,upcount_TB);
     $display("clk,rst,count");


     rst=1'b0;
     clk=1'b0;
     #1;
     rst=1'b1;
     #24

     $finish;
  end
  always forever begin
    if (clk)$display ("%0b %0b %b",clk,rst,count);;
    #1 clk=~clk;
  end
```

T- flipflop module

```
module Tflipflop ( input clk, input rstn, input t, output reg q,output reg qneg);

  always @(posedge clk,rstn) begin
    if (~rstn) begin
      q <= 0;
      qneg <= 1;
    end

    else

     begin

        if (t) begin
            q <= ~q;
          qneg <= q;
        end
        else  begin
            q <= q;
          qneg <= ~q;
        end

     end

  end
endmodule
```

Results

As we start the program on 0000 and there's no next state on the state table 4 for this combination, the next state is going to be deterministic, but it is going to be a valid state, then is going to fallow the sequence of the prime numbers.



13	.1101.
11	.1011.
7	.0111.
5	.0101.
3	.0011.
2	.0010.

Figure 3 scrip results