19L303-DIGITAL ELECTRONICS PRESENTATION

BCD RIPPLE COUNTER

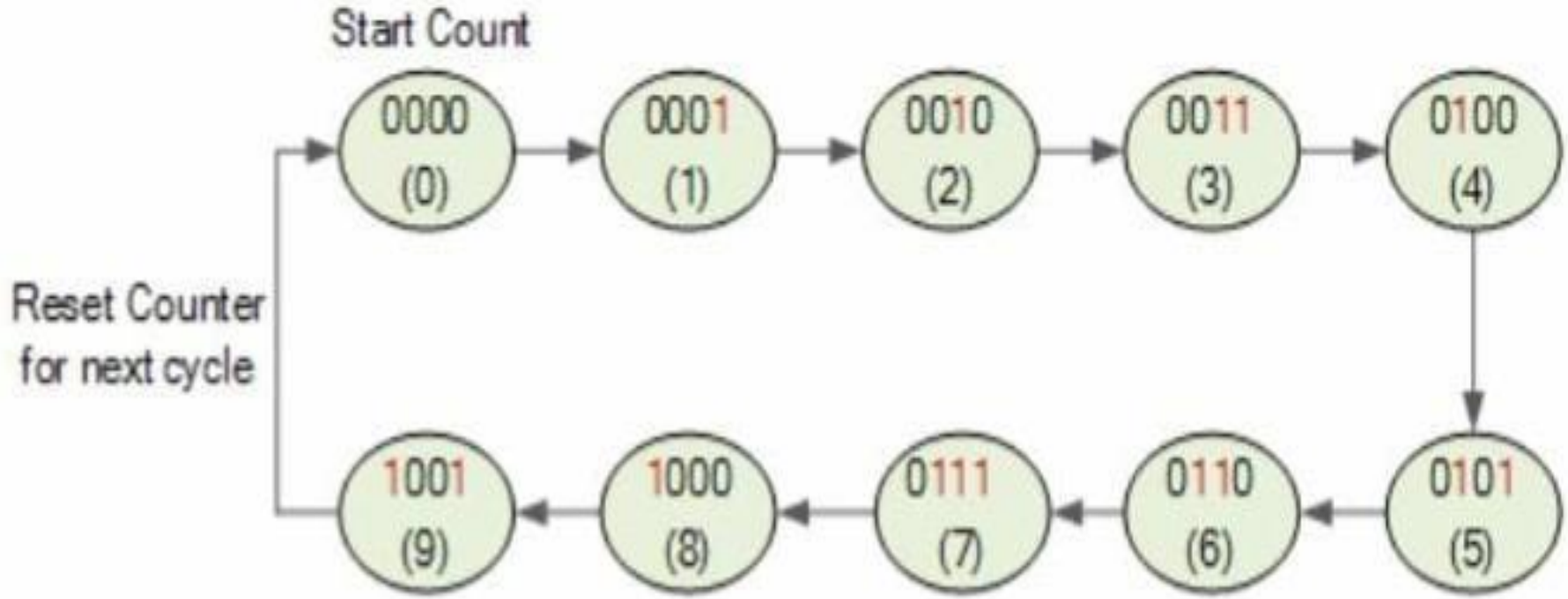
BY:
20L108-DHEVAA DHARSHINI B
20L116-JEYASHRI R
20L141-SHEENA S

BCD or Decade Counter Circuit

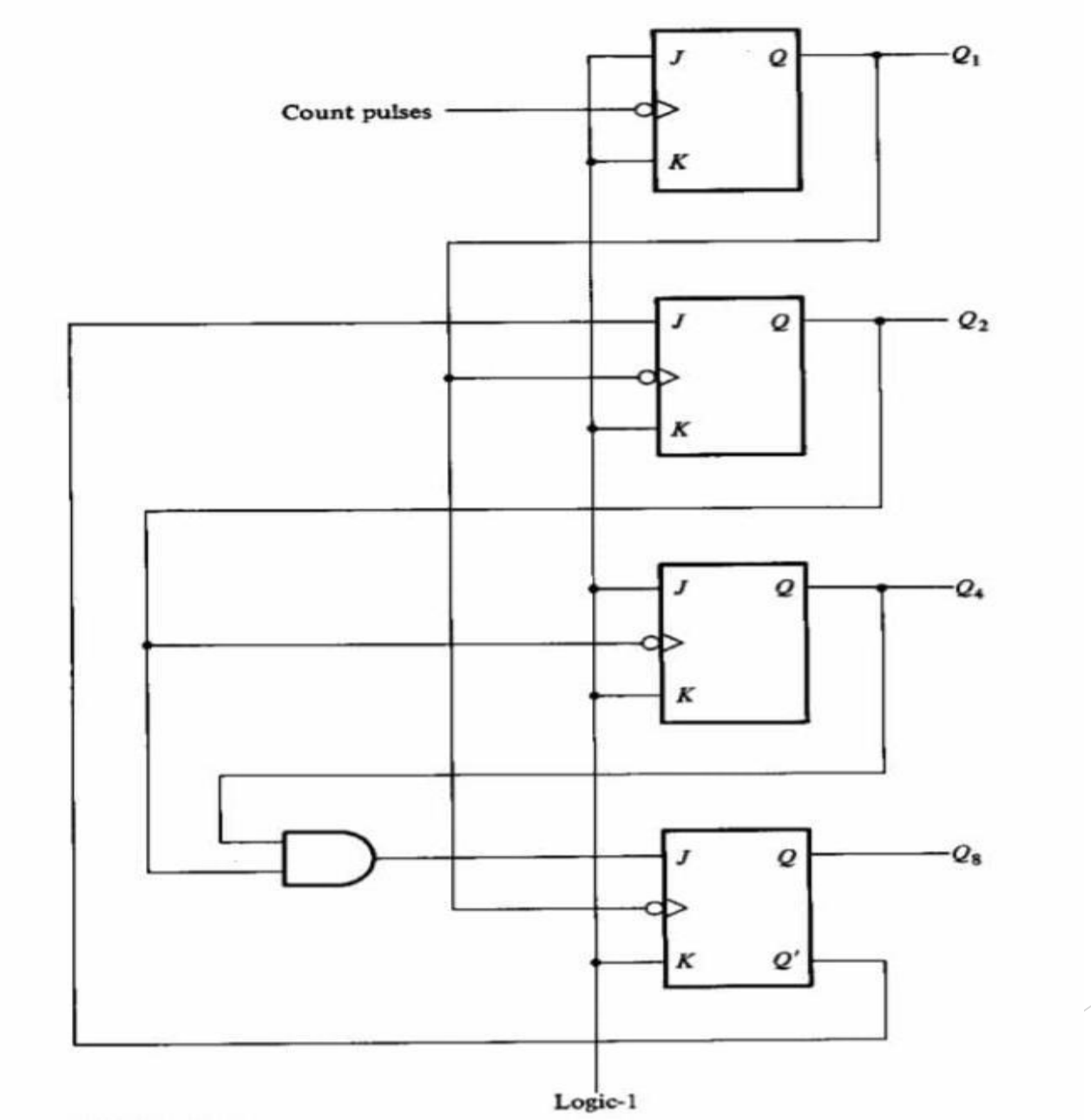
A binary coded decimal (BCD) is a serial digital counter that counts ten digits .And it resets for every new clock input. As it can go through 10 unique combinations of output, it is also called as “Decade counter”. A BCD counter can count 0000, 0001, 0010, 1000, 1001, 1010, 1011, 1110, 1111, 0000, and 0001 and so on.It is an asynchronous counter. Different flip-flops are used with a different clock pulse

It is called MOD 10 counter,because it goes through ten states.It can be constructed using a minimum of four JK flip-flops. The BCD counter can have 16 combinational states . Out of 16 states, 10 are used. When the counters are connected in series, we can count up to 100 or 1000 based on the application.

BCD Counter State Diagram



Circuit diagram:



Truth Table of Decade Counter

Input Pulses	D	C	B	A
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0
0	0	0	0	0 (resets)

SIMULATION:

VERILOG CODE:

TESTBENCH:

testbench.sv

SV/Verilog Testbe

```
1  // Code your testbench here
2  // or browse Examples
3  module bcdtb();
4    reg j,k,clk;
5    wire q1,q2,q4,q8,_q1,_q2,_q4,_q8;
6    bcd b1(j,k,clk,q1,q2,q4,q8,_q1,_q2,_q4,_q8);
7    initial begin
8      clk=0;
9      repeat(50)
10     #5clk=~clk;
11   end
12   initial begin
13     j=1;k=1;#10;
14   end
15   initial begin
16     $dumpfile("dump*.vcd");$dumpvars();end
17 endmodule
```

DESIGN:

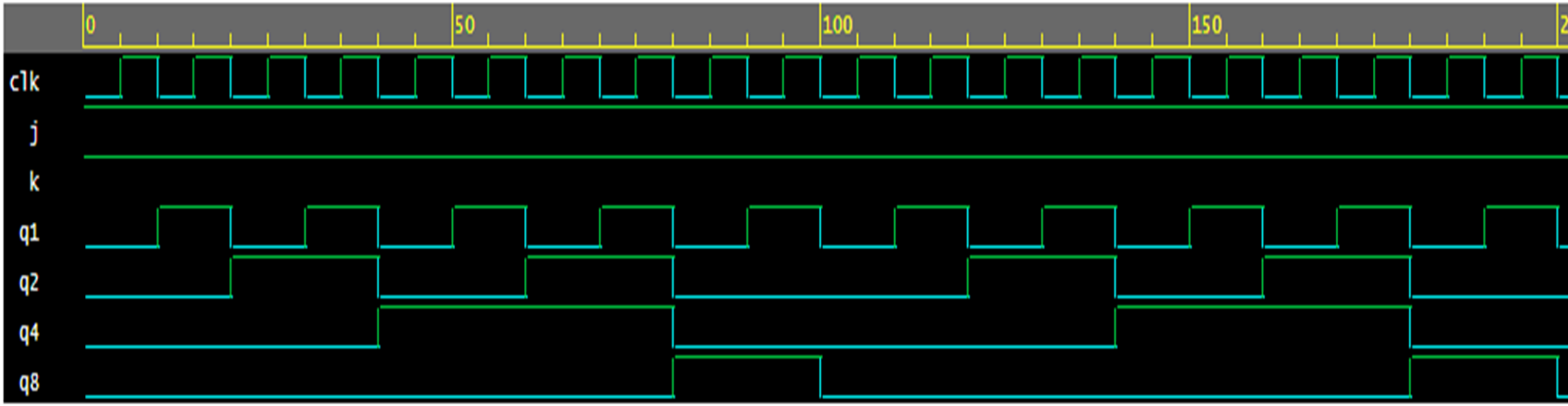
design.sv

SV/Verilog Design

```
// Code your design here
module bcd(j,k,clk,q1,q2,q4,q8,_q1,_q2,_q4,_q8);
  input j,k,clk;
  output  q1,q2,q4,q8;
  output  _q1,_q2,_q4,_q8;
  wire w1;
    jkff j1(j,k,clk,q1,_q1);
    jkff j2(_q8,k,q1,q2,_q2);
    jkff j3(j,k,q2,q4,_q4);
    and(w1,q4,q2);
    jkff j4(w1,k,q1,q8,_q8);
endmodule

    module jkff(j,k,cl,q,_q);
      input j,k,cl;
      output wire _q;
      output reg q;
      initial begin
        q=1;end
      assign _q=~q;
      always @ (negedge(cl))
      begin
        if(j==1 && k==0)
          begin q=1; end
        if(j==0 && k==1)
          begin q=0;end
        if(j==1 && k==1)
          begin q=~q ;end
        if(j==0 && k==0)
          begin q=q; end
      end
    endmodule
```

OUTPUT WAVEFORM:



Note: To revert to EPWave opening in a new browser window, set that option on your user page.

Applications of BCD Counter

- Clock generation
- Clock division
- Integrated oscillator
- Low power cmos
- TTL compatible inputs
- In frequency counting circuits