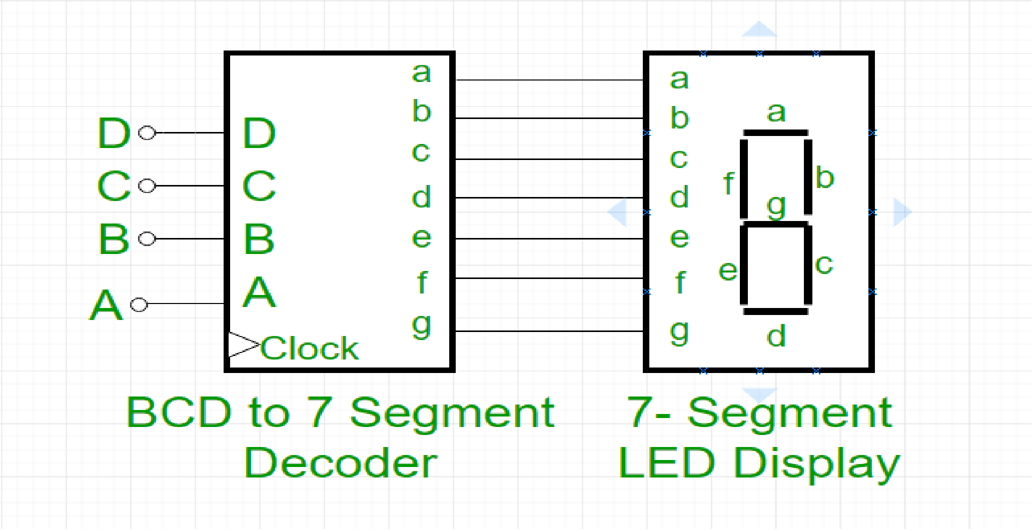
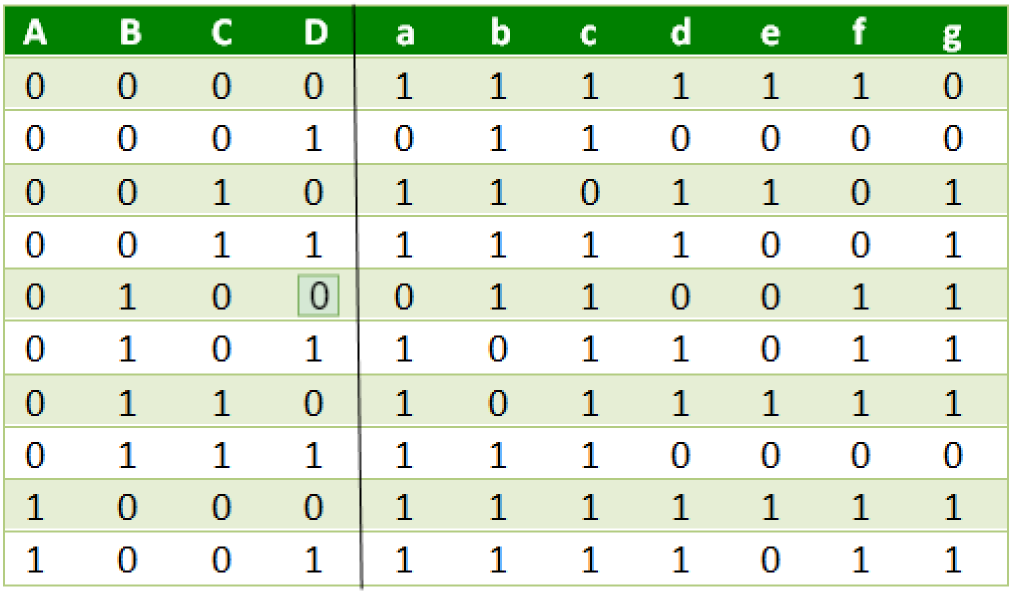
Day 25

BCD Seven Segment Decoder:

BCD to seven segment decoder has four input lines (A, B, C and D) and 7 output lines (a, b, c, d, e, f and g), this output is given to seven segment LED display which displays the decimal number depending upon inputs.



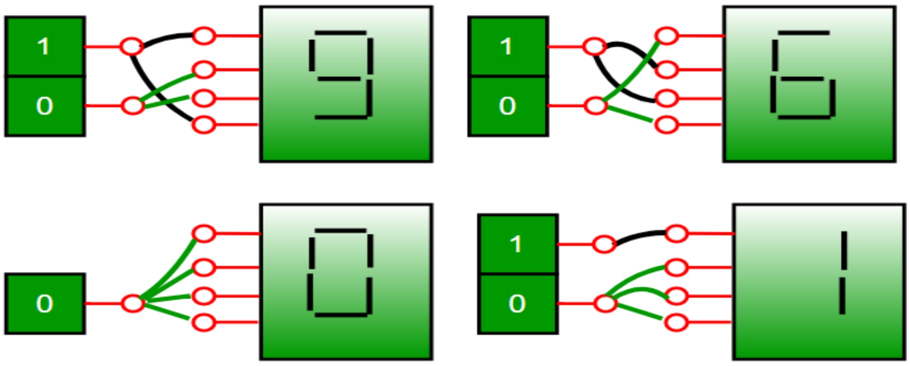
Truth Table – For common cathode type BCD to seven segment decoder:



A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1

- For Common Anode type seven segment LED display, we only have to interchange all ‘0s’ and ‘1s’ in the output side i.e., (for a, b, c, d, e, f, and g replace all ‘1’ by ‘0’ and vice versa) and solve using K-map.

- Output for first combination of inputs (A, B, C and D) in Truth Table corresponds to ‘0’ and last combination corresponds to ‘9’. Similarly rest corresponds from 2 to 8 from top to bottom.
- BCD numbers only range from 0 to 9,thus rest inputs from 10-F are invalid inputs.

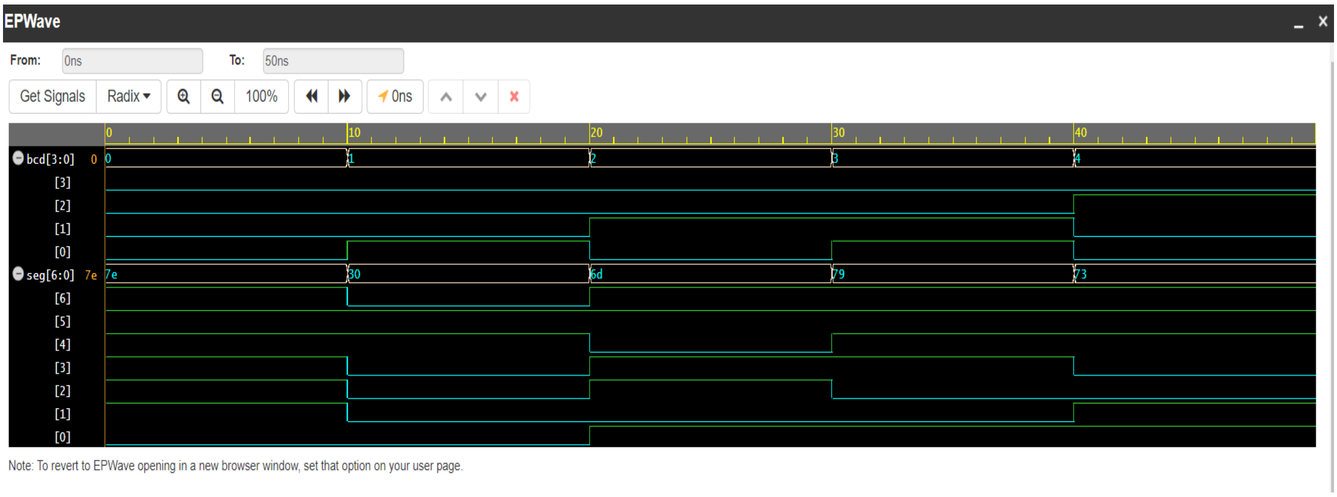


RTL CODE:

```
module BCDtoSevenSegment(input [3:0] bcd,output reg [6:0]
seg);
  always @ (bcd) begin
    case (bcd)
      4'b0000: seg = 7'b1111110;
      4'b0001: seg = 7'b0110000;
      4'b0010: seg = 7'b1101101;
      4'b0011: seg = 7'b1111001;
      4'b0100: seg = 7'b1110011;
      4'b0101: seg = 7'b1011011;
      4'b0110: seg = 7'b1011111;
      4'b0111: seg = 7'b1110100;
      4'b1000: seg = 7'b1111111;
      4'b1001: seg = 7'b1110011;
      default: seg = 7'b0000000;
    endcase
  end
endmodule
```

TESTBENCH:

```
module testbench;
  reg [3:0]bcd;
  wire [6:0]seg;
  BCDtoSevenSegment a1( bcd, seg);
  initial
    begin
      $dumpfile(".vcd");
      $dumpvars(1);
    end
  initial
    begin
      bcd = 0;
      #10 bcd =1;
      #10 bcd =2;
      #10 bcd =3;
      #10 bcd =4;
      #10 bcd =5;
      #10 bcd =6;
      #10 bcd =7;
      #10 bcd =8;
      #10 bcd =9;
    end
  initial
    begin
      #50 $finish();
    end
endmodule
```



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