Write a verilog code to design, simulate and a test circuit of 3 to 8 decoder.

Design module :-

```
module dec(
  input a,
  input b,
  input c,
  output [0:7] outr
);
  assign outr[0] = (~a & ~b & ~c);
  assign outr[1] = (~a & ~b & c);
  assign outr[2] = (~a & b & ~c);
  assign outr[3] = (~a & b & c);
  assign outr[4] = (a & ~b & ~c);
  assign outr[5] = (a & ~b & c);
  assign outr[6] = (a & b & ~c);
  assign outr[7] = (a & b & c);
//      assign or = (~a & ~b & ~c) | (~a & ~b & c) | (~a
& b & ~c) | (~a & b & c) | (a & ~b & ~c) | (a & ~b & c) |
(a & b & ~c) | (a & b & c);
endmodule
```

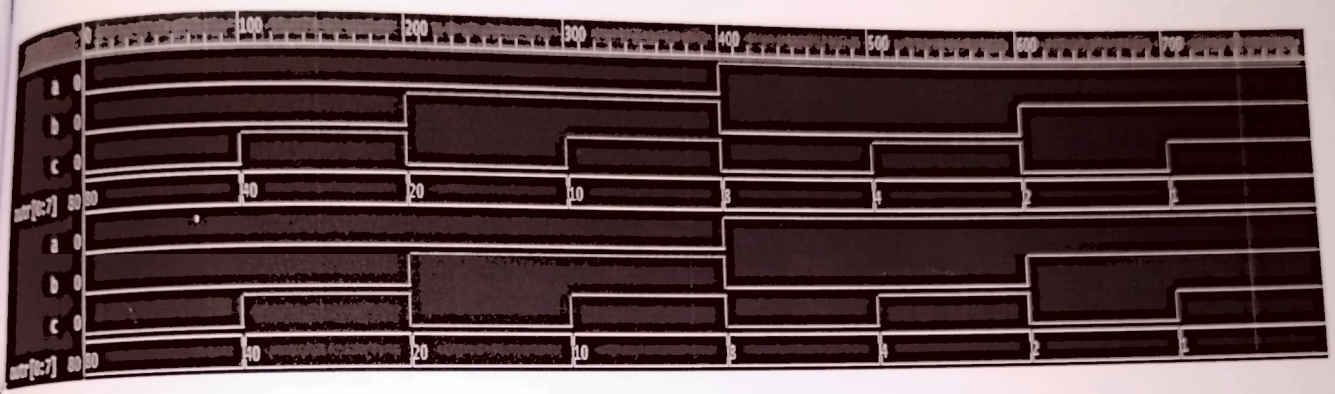
Test bench :-

```
module dec_tb;
    reg a;
    reg b;
    reg c;
    wire [0:7] outr;
    dec uut (.a(a), .b(b), .c(c), .outr(outr));
```

```
initial begin
  a = 0;
  b = 0;
  c = 0;
  # 100;
  a = 0;
  b = 0;
  c = 1;
  # 100;
  a = 0;
  b = 1;
  c = 0;
  # 100;
  a = 0;
  b = 1;
  c = 1;
  # 100;
  a = 1;
  b = 0;
  c = 0;
  # 100;
  a = 1;
  b = 0;
  c = 1;
  # 100;
  a = 1;
  b = 1;
  c = 0;
  # 100;
  a = 1;
  b = 1;
  c = 1;
  # 100;
end
initial begin
```

```
    $dumpfile ("dump.vcd");
    $dumpvars (0, dec_tb);
  end
endmodule
```

3X8 DECODER:


0
100
200
300
400
500
600
700
a 0
b 0
c 0
outr[0:7] 80
80
40
20
10
8
4
2
1
a 0
b 0
c 0
outr[0:7] 80
80
40
20
10
8
4
2
1