Decoders (A)

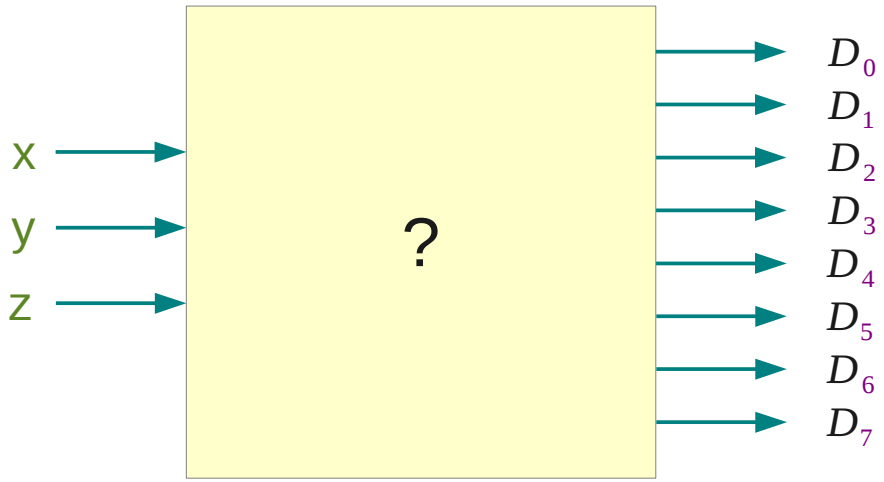
Please send corrections (or suggestions) to youngwlim@hotmail.com.

This document was produced by using OpenOffice and Octave.

Decoder Truth Table

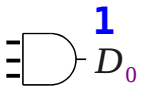
x	y	z	D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7
0	0	0	1	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0
1	0	1	0	0	0	0	0	1	0	0
1	1	0	0	0	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	1

inputs output

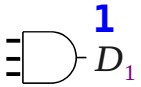


Truth Table and minterms

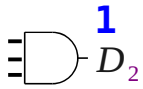
x	y	z	D_0
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



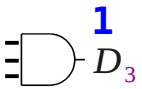
x	y	z	D_1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



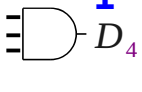
x	y	z	D_2
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



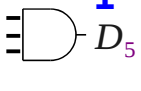
x	y	z	D_3
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	0



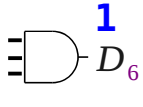
x	y	z	D_4
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	0



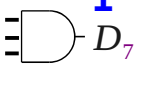
x	y	z	D_5
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0



x	y	z	D_6
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	0



x	y	z	D_7
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1



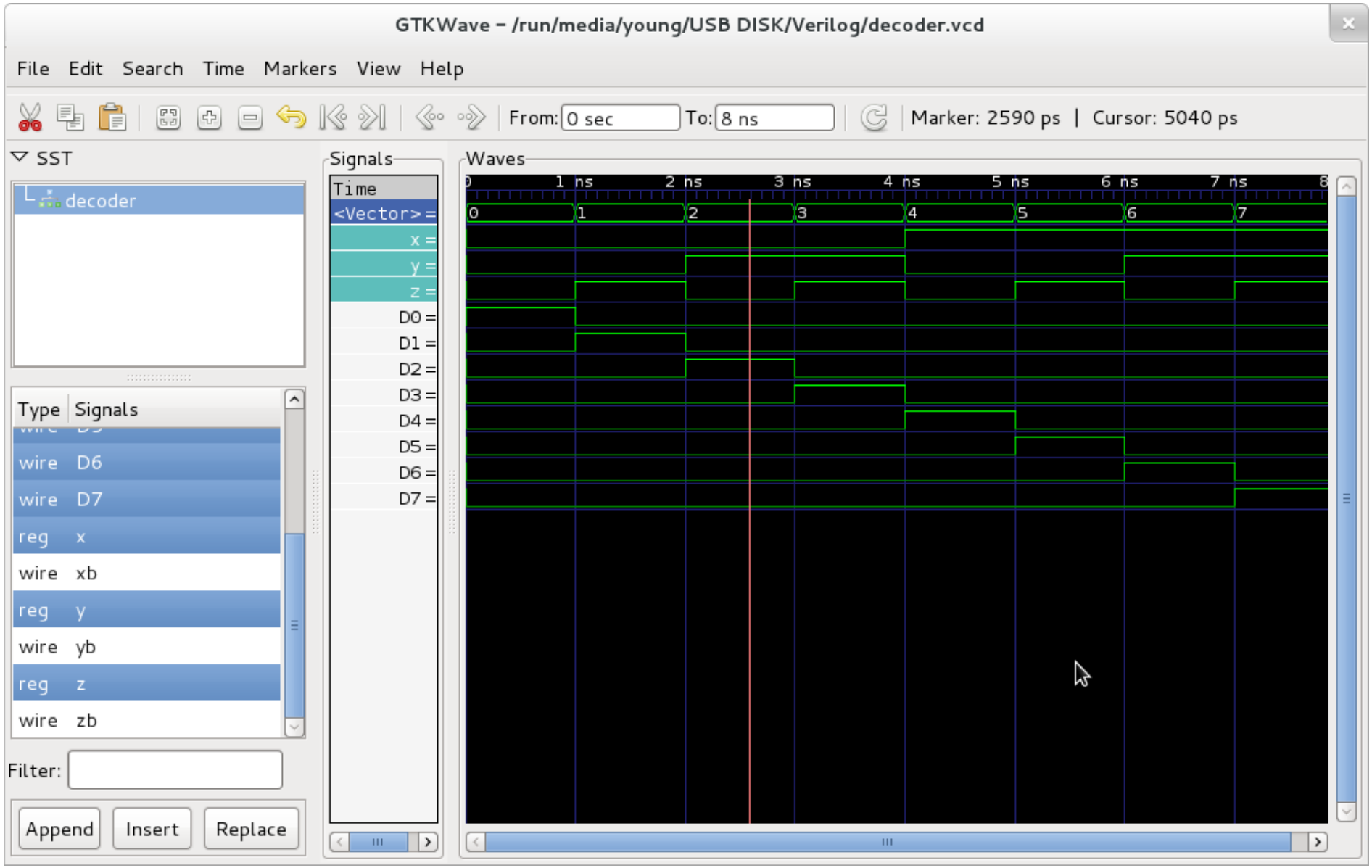
$$\bar{x}\,\bar{y}\,\bar{z} = D_0$$
$$\bar{x}\,\bar{y}\,z = D_1$$
$$\bar{x}\,y\,\bar{z} = D_2$$
$$\bar{x}\,y\,z = D_3$$
$$x\,\bar{y}\,\bar{z} = D_4$$
$$x\,\bar{y}\,z = D_5$$
$$x\,y\,\bar{z} = D_6$$
$$x\,y\,z = D_7$$

decoder.v

```
`timescale 1ns/100ps


module decoder();
  wire  D0, D1, D2, D3, D4, D5, D6, D7;
  reg x, y, z;


  not (xb, x);
  not (yb, y);
  not (zb, z);

  and (D0, xb, yb, zb);  // D0 = minterm m0
  and (D1, xb, yb, z );  // D1 = minterm m1
  and (D2, xb, y , zb);  // D2 = minterm m2
  and (D3, xb, y , z );  // D3 = minterm m3
  and (D4, x , yb, zb);  // D4 = minterm m4
  and (D5, x , yb, z );  // D5 = minterm m5
  and (D6, x , y , zb);  // D6 = minterm m6
  and (D7, x , y , z );  // D7 = minterm m7
```

```
// Testbench Code goes here
initial begin
  $dumpfile("decoder.vcd");
  $dumpvars(0, decoder);

  $monitor ("[x y z] = %b%b%b [D0, D1, D2, D3]
                = %b%b%b%b%b%b%b%b",
               x, y, z, D0, D1, D2, D3, D4, D5, D6, D7);

  x = 0;
  y = 0;
  z = 0;


  #8 $finish;
end



always #4 x = ~x;
always #2 y = ~y;
always #1 z = ~z;


endmodule
```

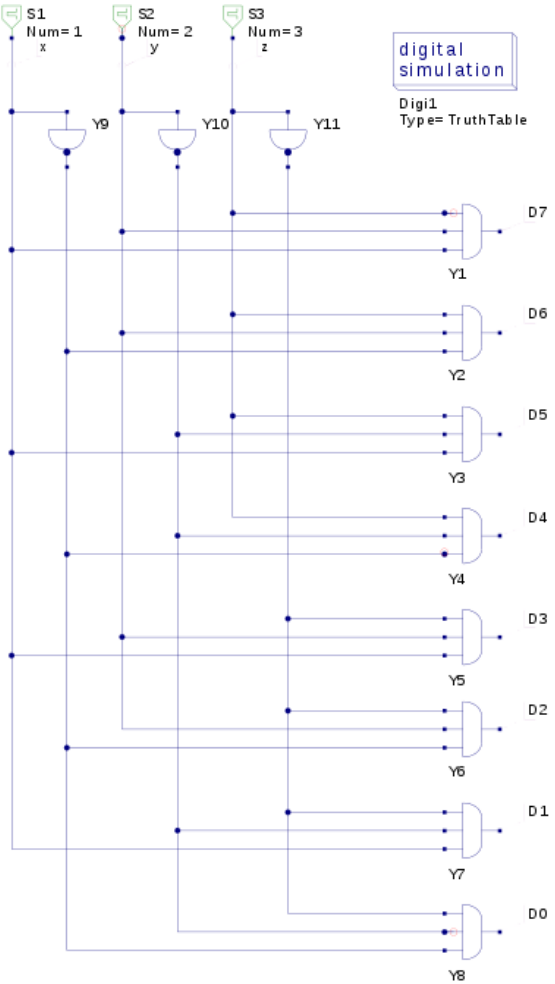
decoder.v simulation



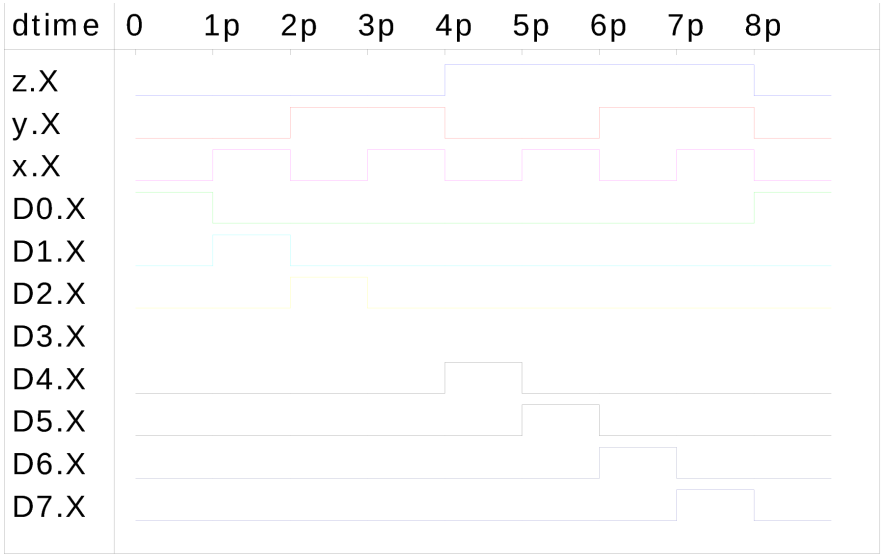
Behavioral Decoder

```
`timescale 1ns/100ps


module decoder(A, D);
  input [2:0] A;
  output [7:0] D;
  reg      [7:0] D;

   always @(A)
     begin
         case (A)
             0: D = 8'b00000001;
             1: D = 8'b00000010;
             2: D = 8'b00000100;
             3: D = 8'b00001000;
             4: D = 8'b00010000;
             5: D = 8'b00100000;
             6: D = 8'b01000000;
             7: D = 8'b10000000;
             default: D = 8b'X;
        endcase
     end

endmodule
```

Decoder Schematic



Decoder Schematic



	z.X	y.X	x.X	D0.X	D1.X	D2.X	D3.X	D4.X	D5.X	D6.X	D7.X
0000	0	0	0	1	0	0	0	0	0	0	0
0001	0	0	1	0	1	0	0	0	0	0	0
0010	0	1	0	0	0	1	0	0	0	0	0
0011	0	1	1	0	0	0	1	0	0	0	0
0100	1	0	0	0	0	0	0	1	0	0	0
0101	1	0	1	0	0	0	0	0	1	0	0
0110	1	1	0	0	0	0	0	0	0	1	0
0111	1	1	1	0	0	0	0	0	0	0	1
1000	0	0	0	1	0	0	0	0	0	0	0

References

[1] http://en.wikipedia.org/