1A)

```
module decoder(input a0, a1, a2, output x0, x1, x2, x3, x4, x5, x6, x7);
assign x0 = (~a0) & (~a1) & (~a2);
assign x1 = zz(~a0) & (~a1) & (a2);
assign x2 = (~a0) & (a1) & (~a2);
assign x3 = (~a0) & (a1) & (a2);
assign x4 = (a0) & (~a1) & (~a2);
assign x5 = (a0) & (~a1) & (a2);
assign x6 = (a0) & (a1) & (~a2);
assign x7 = (a0) & (a1) & (a2);
endmodule

module pg(input a, b, c, output x, y);
wire x0, x1, x2, x3, x4, x5, x6, x7;
decoder(a, b, c, x0, x1, x2, x3, x4, x5, x6, x7);
assign x = x1 | x4 | x6 | x7;
assign y = x0 | x3 | x4 | x5 | x7;
endmodule


module decoder(input a0, a1, a2, output x0, x1, x2, x3, x4, x5, x6, x7);
and(x0, ~a2, ~a1, ~a0);
and(x1, ~a2, ~a1, a0);
and(x2, ~a2, a1, ~a0);
and(x3, ~a2, a1, a0);
and(x4, a2, ~a1, ~a0);
and(x5, a2, ~a1, a0);
and(x6, a2, a1, ~a0);
and(x7, a2, a1, a0);
endmodule

module pg(input a, b, c, output x, y);
wire x0, x1, x2, x3, x4, x5, x6, x7;
decoder(a, b, c, x0, x1, x2, x3, x4, x5, x6, x7);
or(x, x1, x4, x6, x7);
or(y, x0, x3, x4, x5, x7);
endmodule


module decoder(input a2, a1, a0, output reg x0, x1, x2, x3, x4, x5, x6, x7);
always @ * begin
if(a2 == 0 & a1 == 0 & a0 == 0) begin
x0 = 1;
x1 = 0; x2 = 0; x3 = 0; x4 = 0; x5 = 0; x6 = 0; x7 = 0;
end
else if(a2 == 0 & a1 == 0 & a0 == 1) begin
x1 = 1;
x0 = 0; x2 = 0; x3 = 0; x4 = 0; x5 = 0; x6 = 0; x7 = 0;
end
else if(a2 == 0 & a1 == 1 & a0 == 0) begin
x2 = 1;
x1 = 0; x0 = 0; x3 = 0; x4 = 0; x5 = 0; x6 = 0; x7 = 0;
end
```

```
else if(a2 == 0 & a1 == 1 & a0 == 1) begin
x3 = 1;
x1 = 0; x2 = 0; x0 = 0; x4 = 0; x5 = 0; x6 = 0; x7 = 0;
end
else if(a2 == 1 & a1 == 0 & a0 == 0) begin
x4 = 1;
x1 = 0; x2 = 0; x3 = 0; x0 = 0; x5 = 0; x6 = 0; x7 = 0;
end
else if(a2 == 1 & a1 == 0 & a0 == 1) begin
x5 = 1;
x1 = 0; x2 = 0; x3 = 0; x4 = 0; x0 = 0; x6 = 0; x7 = 0;
end
else if(a2 == 1 & a1 == 1 & a0 == 0) begin
x6 = 1;
x1 = 0; x2 = 0; x3 = 0; x4 = 0; x5 = 0; x0 = 0; x7 = 0;
end
else if(a2 == 1 & a1 == 1 & a0 == 1) begin
x7 = 1;
x1 = 0; x2 = 0; x3 = 0; x4 = 0; x5 = 0; x6 = 0; x0 = 0;
end
end
endmodule

module pg(input a, b, c, output x, y);
wire x0, x1, x2, x3, x4, x5, x6, x7;
decoder INST0(a, b, c, x0, x1, x2, x3, x4, x5, x6, x7);
or(x, x1, x4, x6, x7);
or(y, x0, x3, x4, x5, x7);
endmodule


module pg_tb;
wire x, y;
reg a, b, c;
pg inst0(a, b, c, x, y);
initial begin
repeat(10) begin
a = $random;
b = $random;
c = $random;
end
end
endmodule
```

1B)

```
module pg(input w, x, y, z, output f1, f2, f3);
assign f1 = x&(~z) | w&x&y | (~w)&(~y)&(z) | (~w)&(~x)&y;
assign f2 = y&z | (~w)&(~x) | (~x)&z | (~w)&y;
assign f3 = (~y)&z | w&z | (~x)&z | (~w)&x&(~y) | (w)&(~x)&y;
endmodule

module pg(input w, x, y, z, output reg f1, f2, f3);
```

```
always @ * begin
if(w == 0 & x == 0 & y == 0 & z == 0) begin
f1 = 0;f2 = 0;f3 = 0;
end
else if(w == 0 & x == 0 & y == 0 & z == 1) begin
f1 = 1;f2 = 1;f3 = 1;
end
else if(w == 0 & x == 0 & y == 1 & z == 0) begin
f1 = 1;f2 = 1;f3 = 0;
end
else if(w == 0 & x == 0 & y == 1 & z == 1) begin
f1 = 1;f2 = 1;f3 = 1;
end
else if(w == 0 & x == 1 & y == 0 & z == 0) begin
f1 = 1;f2 = 1;f3 = 1;
end
else if(w == 0 & x == 1 & y == 0 & z == 1) begin
f1 = 1;f2 = 1;f3 = 1;
end
else if(w == 0 & x == 1 & y == 1 & z == 0) begin
f1 = 1;f2 = 1;f3 = 0;
end
else if(w == 0 & x == 1 & y == 1 & z == 1) begin
f1 = 0;f2 = 1;f3 = 0;
end
else if(w == 1 & x == 0 & y == 0 & z == 0) begin
f1 = 0;f2 = 0;f3 = 0;
end
else if(w == 1 & x == 0 & y == 0 & z == 1) begin
f1 = 0;f2 = 1;f3 = 1;
end
else if(w == 1 & x == 0 & y == 1 & z == 0) begin
f1 = 0;f2 = 0;f3 = 1;
end
else if(w == 1 & x == 0 & y == 1 & z == 1) begin
f1 = 0;f2 = 1;f3 = 1;
end
else if(w == 1 & x == 1 & y == 0 & z == 0) begin
f1 = 1;f2 = 0;f3 = 0;
end
else if(w == 1 & x == 1 & y == 0 & z == 1) begin
f1 = 0;f2 = 0;f3 = 1;
end
else if(w == 1 & x == 1 & y == 1 & z == 0) begin
f1 = 1;f2 = 0;f3 = 0;
end
else if(w == 1 & x == 1 & y == 1 & z == 1) begin
f1 = 1;f2 = 1;f3 = 1;
end
end
endmodule


module q1b_gate(input w,x,y,z,output f1,f2,f3);
wire notw, notx, noty, notz;
```

```
wire p,q,r,s,a,b,c,d,e,f,g,h;
not(notw,w);
not(notx,x);
not(noty,y);
not(notz,z);
//for f1
and(p,x,notz);
and(q,w,x,y);
and(r,notw,noty,z);
and(s,notw,notx,y);
or(f1,p,q,r,s);
//for f2
and(a,y,z);
and(b,notw,notx);
and(c,notx,z);
and(d,notw,y);
or(f2,a,b,c,d);
// for f3
and(e,noty,z);
and(f,w,z);
and(g,notw,x,noty);
and(h,w,y,notx);
or(f3,e,f,c,g,h);
endmodule


module q1b_tb;
wire x, y, z;
reg a, b, c, d;
pg inst0(a, b, c, d, x, y, z);
initial begin
repeat(10) begin
a = $random;
b = $random;
c = $random;
d = $random;
#10;
end
end
endmodule
```

1C)

```
module Q1c(input a,b,c,d,output f1,f2);
wire [4:0]w1,w2;
and(w1[0],b,!d);
and(w1[1],a,b,c);
and(w1[2],!a,b,!c);
and(w1[3],!c,!d,!a);
and(w1[4],!a,!b,c,d);
and(w2[0],!c,d);
and(w2[1],!a,b);
and(w2[2],!c,b);
```

```
and(w2[3],b,d);
and(w2[4],c,d,!a);
or(f1,w1[0],w1[1],w1[2],w1[3],w1[4]);
or(f2,w2[0],w2[1],w2[2],w2[3],w2[4]);
endmodule

module Q1c(input a,b,c,d,output f1,f2);
wire [4:0]w1,w2;
assign w1[0]=b&!d;
assign w1[1]=a&b&c;
assign w1[2]=a&b&!c;
assign w1[3]=!c&!d&!a;
assign w1[4]=!a&!b&c&d;
assign w2[0]=!c&d;
assign w2[1]=!a&b;
assign w2[2]=!c&b;
assign w2[3]=b&d;
assign w2[4]=c&d&!a;
assign f1=|w1;
assign f2=|w2;
endmodule

module Q1c(input a,b,c,d,output reg f1,f2);
reg [4:0]w1,w2;
always@*
begin
w1[0]=b&!d;
w1[1]=a&b&c;
w1[2]=a&b&!c;
w1[3]=!c&!d&!a;
w1[4]=!a&!b&c&d;
w2[0]=!c&d;
w2[1]=!a&b;
w2[2]=!c&b;
w2[3]=b&d;
w2[4]=c&d&!a;
f1=|w1;
f2=|w2;
end
endmodule


module Q1c_tb;
reg a,b,c,d;
wire f1,f2;
Q1c c1(a,b,c,d,f1,f2);
initial
begin
repeat(10)
begin
{a,b,c,d}=$random();
#10;
end
end
endmodule
```