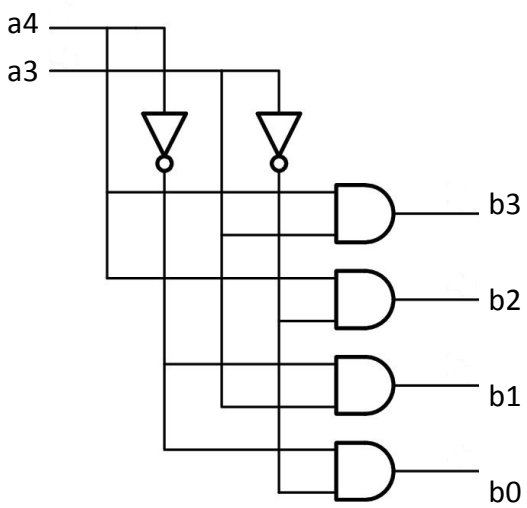
Logic Design HW5 Solution

1.

Input:

$$a \quad = a_4\ a_3\ a_2\ a_1\ a_0$$

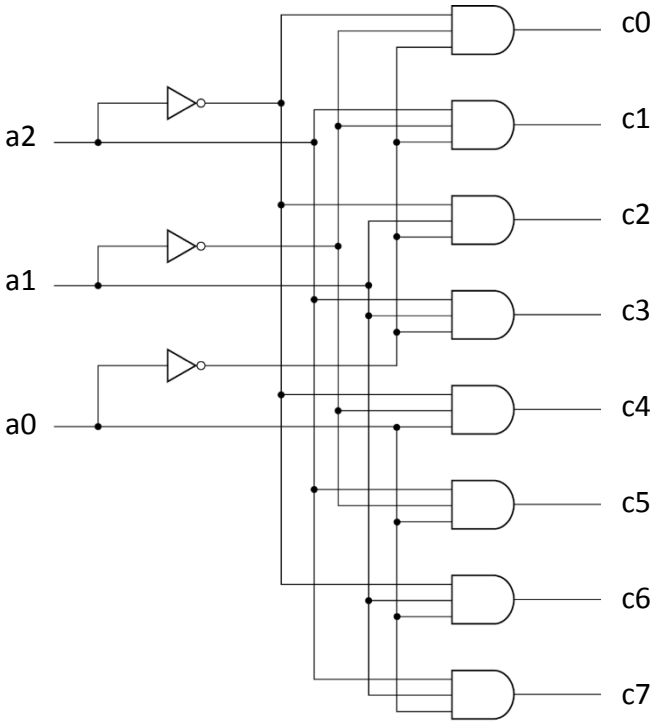


2 to 4 decoder:

$$b_3 = a_4\ a_3$$
$$b_2 = a_4\ a_3'$$
$$b_1 = a_4' a_3$$
$$b_0 = a_4' a_3'$$

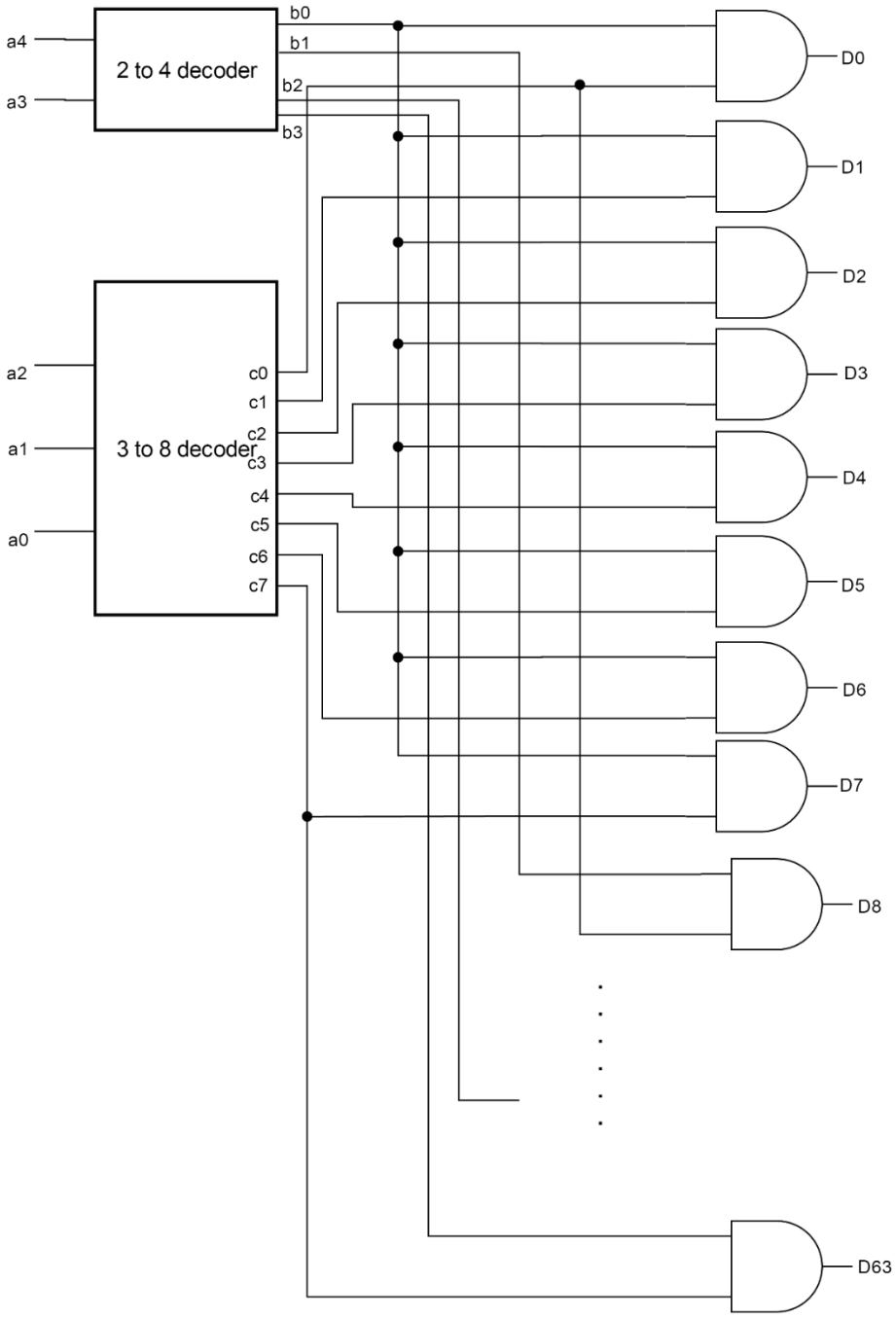
3 to 8 decoder:

$$c_7 = a_2\ a_1\ a_0$$
$$c_6 = a_2\ a_1\ a_0'$$
$$c_5 = a_2\ a_1' a_0$$
$$c_4 = a_2\ a_1' a_0'$$
$$c_3 = a_2' a_1\ a_0$$
$$c_2 = a_2' a_1\ a_0'$$
$$c_1 = a_2' a_1' a_0$$
$$c_0 = a_2' a_1' a_0'$$



Output:

$$D_0 = b_0\ c_0$$
$$D_1 = b_0\ c_1$$
$$D_2 = b_0\ c_2$$
$$D_3 = b_0\ c_3$$
$$D_4 = b_0\ c_4$$
$$D_5 = b_0\ c_5$$
$$D_6 = b_0\ c_6$$
$$D_7 = b_0\ c_7$$
$$D_8 = b_1\ c_0$$
$$D_9 = b_1\ c_1$$
$$\ldots$$
$$D_{63} = b_3\ c_7$$


a4
a3
2 to 4 decoder
b0
b1
b2
b3
a2
a1
a0
3 to 8 decoder
c0
c1
c2
c3
c4
c5
c6
c7
D0
D1
D2
D3
D4
D5
D6
D7
D8
D63

2.

以 4-bit input 為例

Magnitude Comparator:

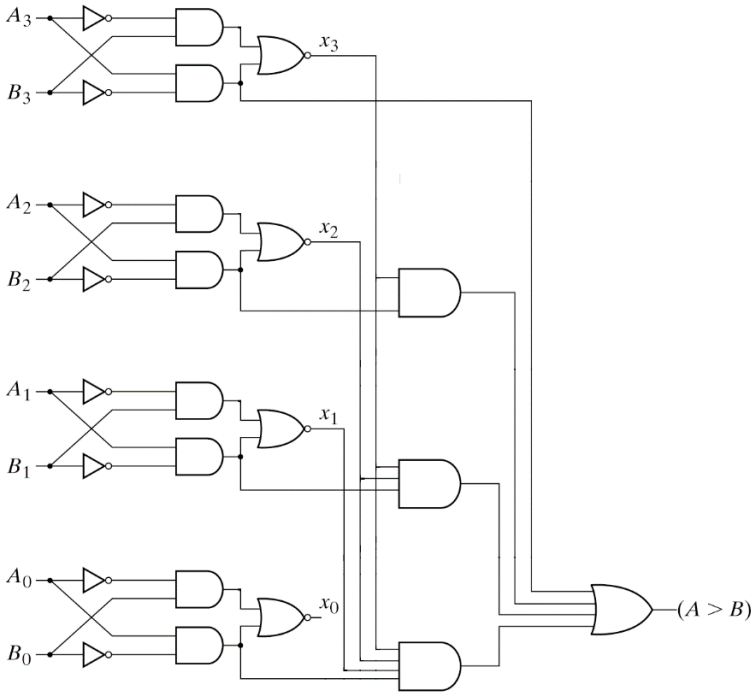
Input: A={A3,A2,A1,A0},B={B3,B2,B1,B0}

(A>B) = $A_3\ B_3' + x_3\ A_2\ B_2' + x_3\ x_2\ A_1\ B_1' + x_3\ x_2\ x_1\ A_0\ B_0'$

$x_i\ = A_i\ B_i\ + A_i' B_i'$

If A>B, output = 1

Else output =0



Comparator1:

If a>b, output=1

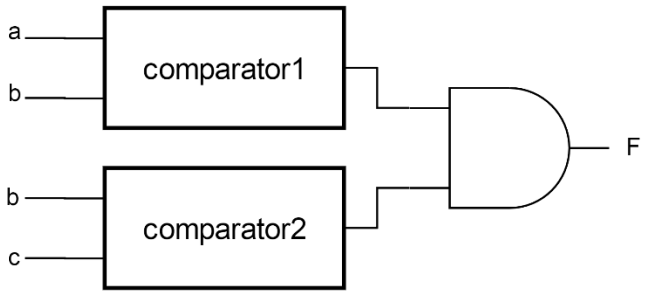
Else output=0

Comparator2:

If b>c, output=1

Else output=0

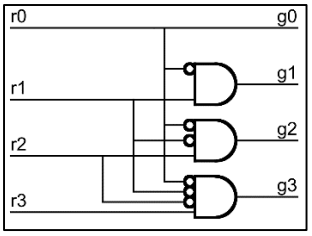
F = 1, only if a>b>c



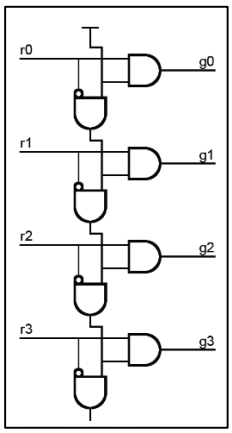
3.

4-bit arbiter:

Priority: r0>r1>r2>r3



or



Input: { r3, r2, r1, r0 }

Arbiter0:

Priority: r0>r3>r2>r1 (rightward rotation)

Arbiter1:

Priority: r1>r0>r3>r2 (rightward rotation)

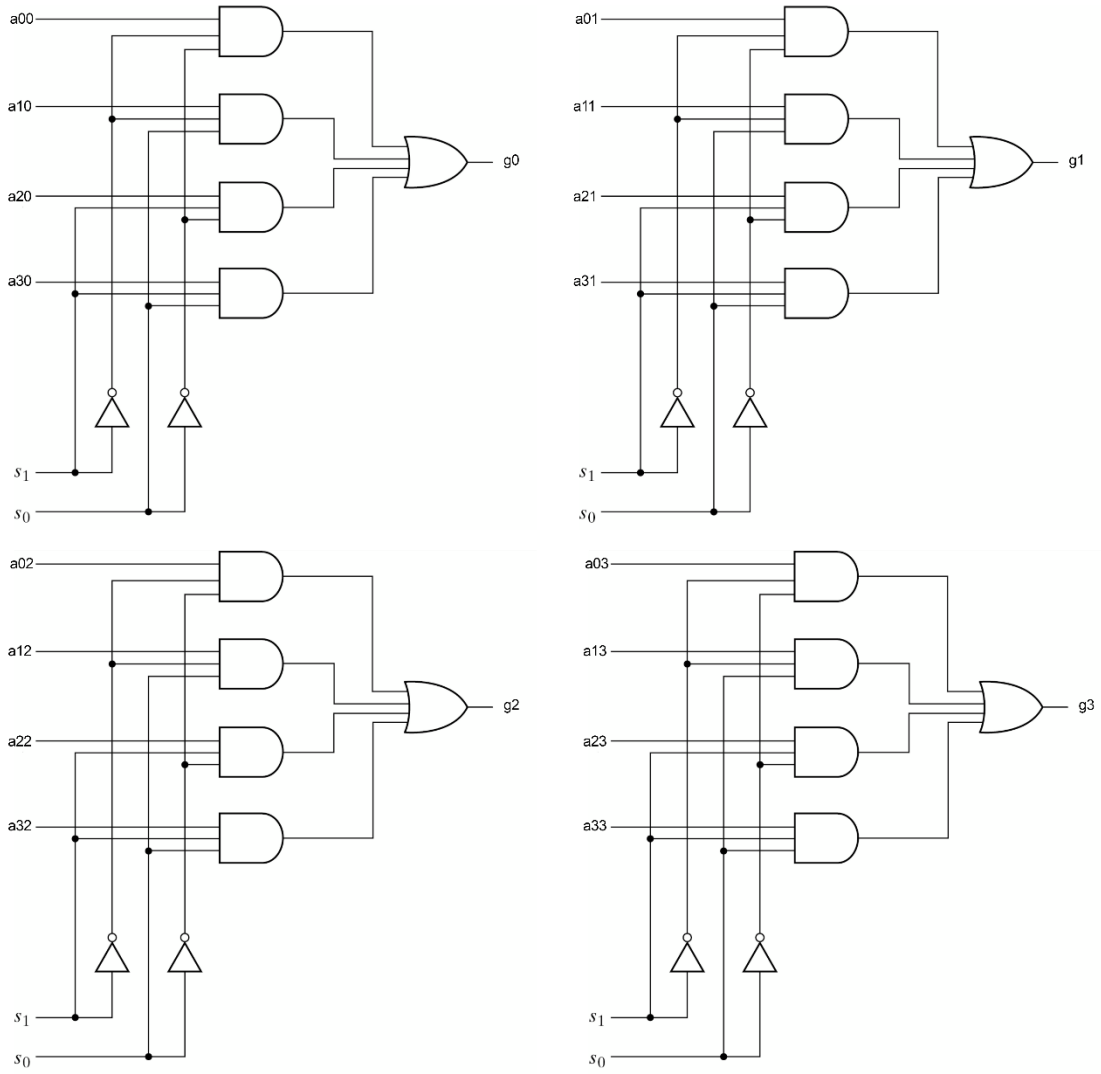
Arbiter2:

Priority: r2>r1>r0>r3 (rightward rotation)

Arbiter3:

Priority: r3> r2>r1>r0 (rightward rotation)

4-bit 4to1 MUX

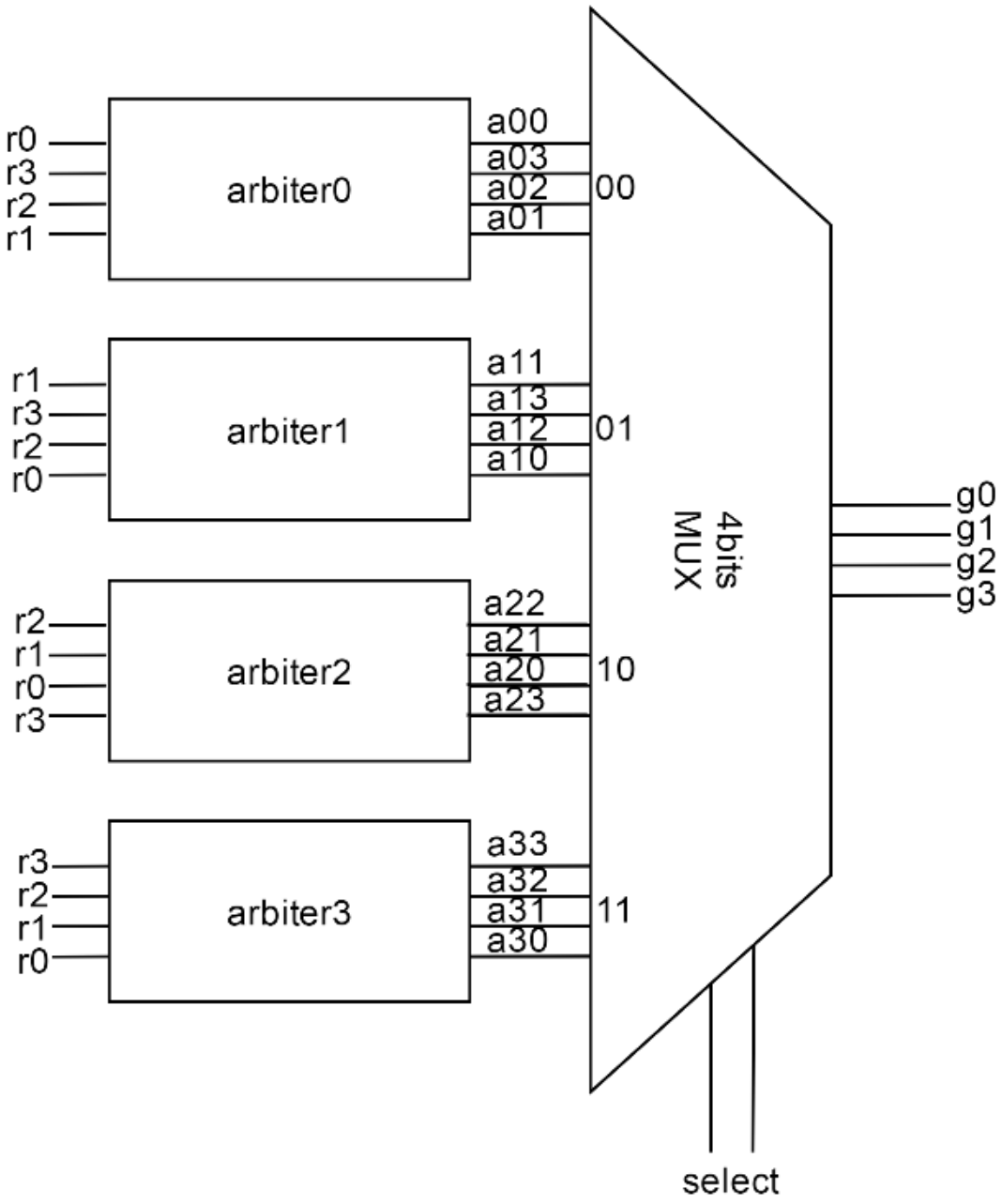


Select{s1,s0} =00: output = arbiter0's output

Select{s1,s0} =01: output = arbiter1's output

Select{s1,s0} =10: output = arbiter2's output

Select{s1,s0} =11: output = arbiter3's output


r0
r3
r2
r1
arbiter0
a00
a03
a02
a01
00
r1
r3
r2
r0
arbiter1
a11
a13
a12
a10
01
r2
r1
r0
r3
arbiter2
a22
a21
a20
a23
10
r3
r2
r1
r0
arbiter3
a33
a32
a31
a30
11
4bits
MUX
g0
g1
g2
g3
select

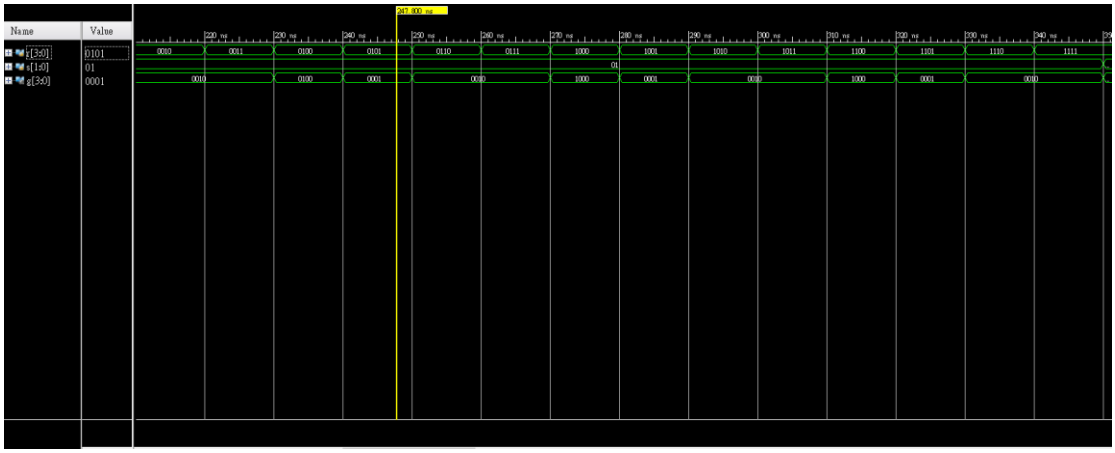
4.

For example,

Input r=0101, s=01,

The priority is r1>r0>r3>r2

Output g=0001



更多更詳盡 verilog 在 ※HW5.4, HW5.6 資料夾

5.

3 to 8 decoder:

{s2, s1, s0} is sel =(000, 001, 010, 011, 100, 101, 110, 111)

D0~D7 Only one is “1”, others are “0”

8x2 AND OR

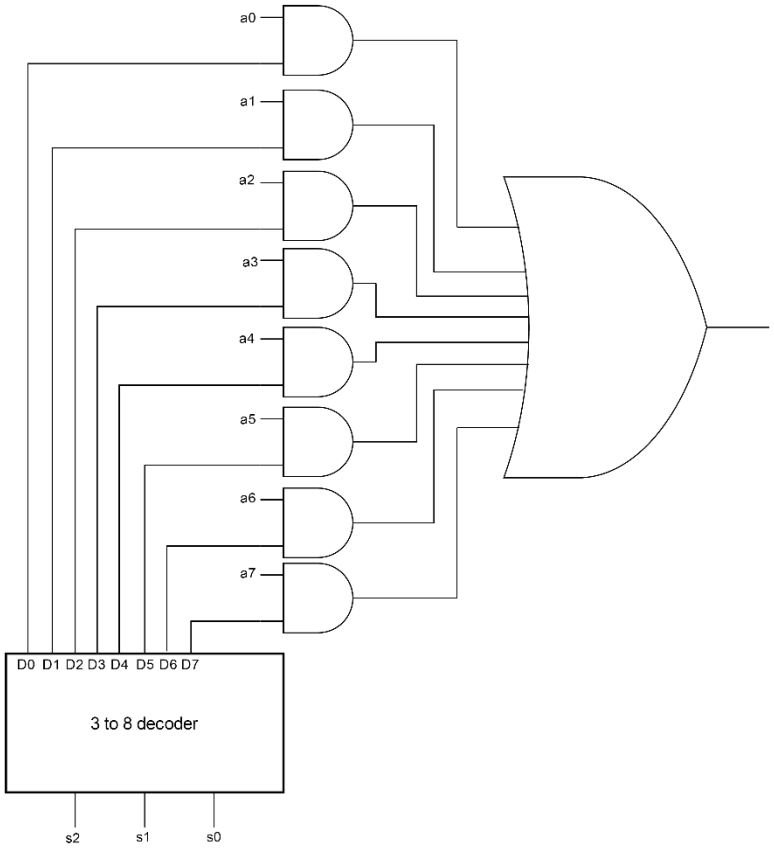
For example, {s0,s1,s2} =000,

D0=1, D1~D7=0,

a0D0=a0 (because D0=1)

a1D1~a7D7=0 (because D1~D7=0)

output is a0 (It’s a 8to1 MUX!)



s2	s1	s0	a0D0	a1D1	a2D2	a3D3	a4D4	a5D5	a6D6	a7D7	F
0	0	0	a0	0	0	0	0	0	0	0	a0
0	0	1	0	a1	0	0	0	0	0	0	a1
0	1	0	0	0	a2	0	0	0	0	0	a2
0	1	1	0	0	0	a3	0	0	0	0	a3
1	0	0	0	0	0	0	a4	0	0	0	a4
1	0	1	0	0	0	0	0	a5	0	0	a5
1	1	0	0	0	0	0	0	0	a6	0	a6
1	1	1	0	0	0	0	0	0	0	a7	a7

6.

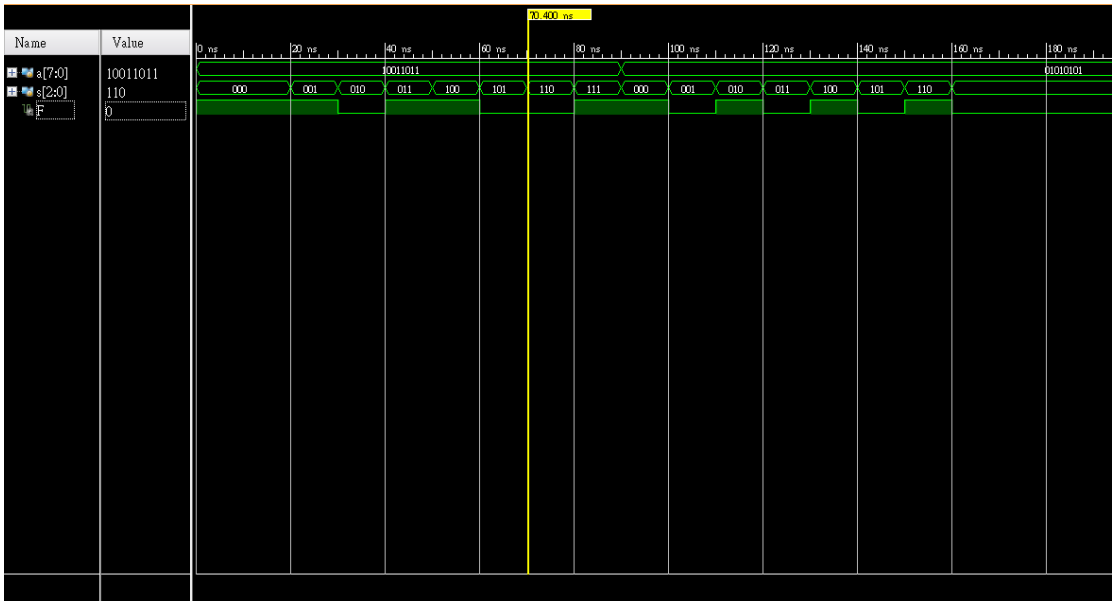
For example

A=10011011

s2	s1	s0	a0D0	a1D1	a2D2	a3D3	a4D4	a5D5	a6D6	a7D7	F
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	0	1	0	0	0	0	0	0	1
0	1	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0	0	0	0	1
1	0	0	0	0	0	0	1	0	0	0	1
1	0	1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	1	1

A=01010101

s2	s1	s0	a0D0	a1D1	a2D2	a3D3	a4D4	a5D5	a6D6	a7D7	F
0	0	0	1	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	0	0	0
0	1	0	0	0	1	0	0	0	0	0	1
0	1	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	1	0	0	0	1
1	0	1	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	1	0	1
1	1	1	0	0	0	0	0	0	0	0	0



更多更詳盡 verilog 在 ※HW5.4, HW5.6 資料夾

7.

By DeMorgan's law

D0= x’y’z’= (x+y+z)’

D1= x’y’z= (x+y+z’)’

D2= x’yz’= (x+y’+z)’

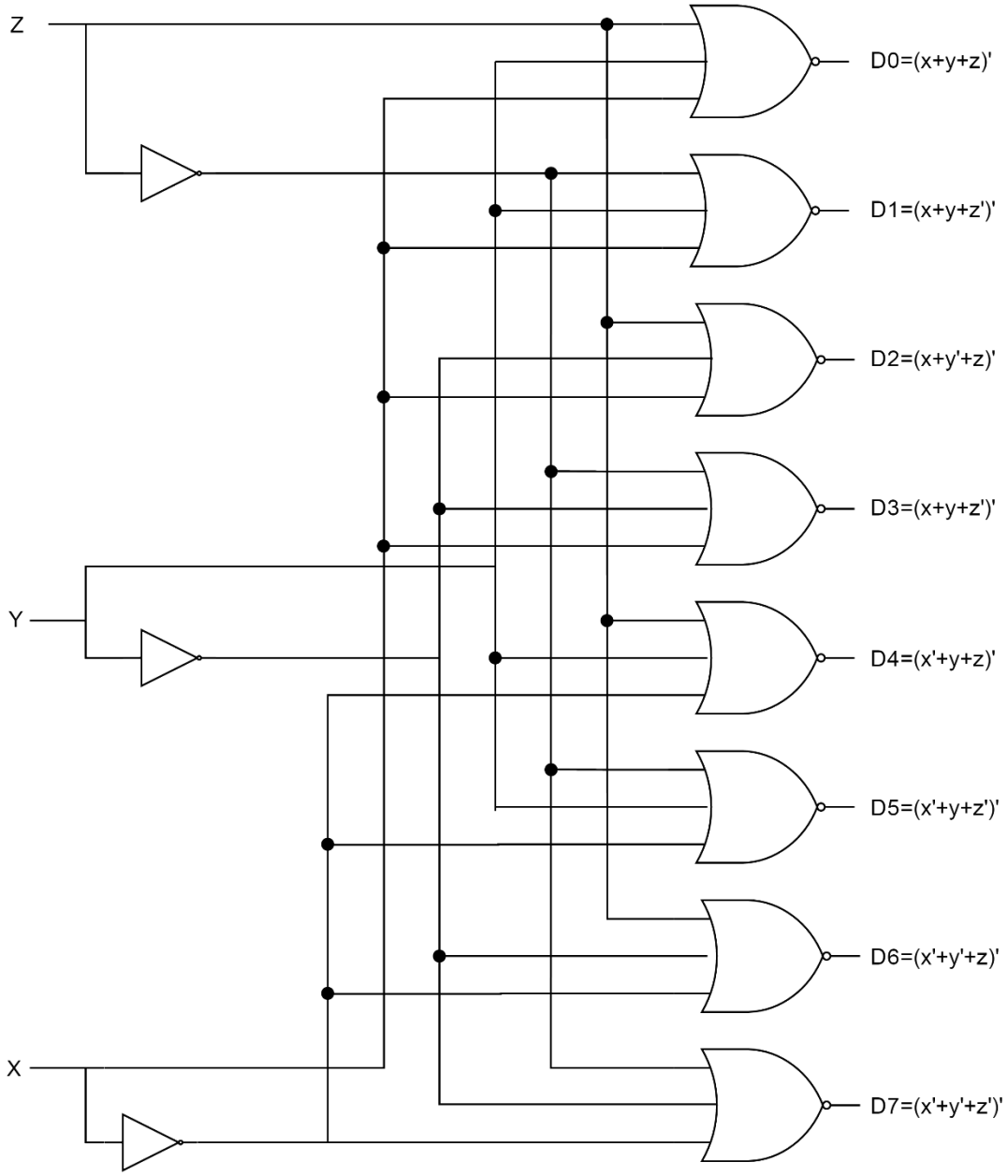
D3= x’yz= (x+y’+z’)’

D4= xy’z’= (x’+y+z)’

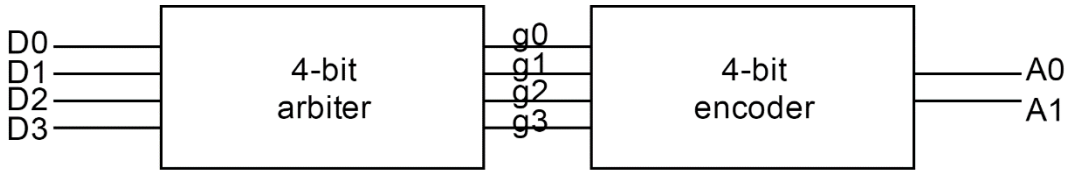
D5= xy’z= (x’+y+z’)’

D6= xyz’= (x’+y’+z)’

D7= xyz = (x’+y’+z’)’

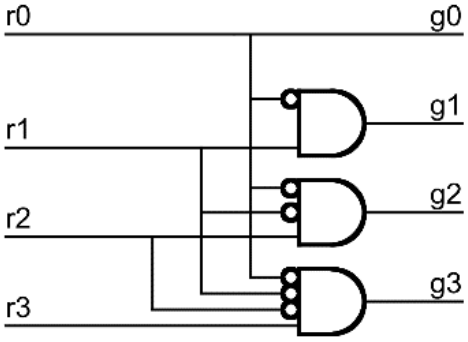


8.



4-bit arbiter:

Priority: r0>r1>r2>r3 (D0>D1>D2>D3)



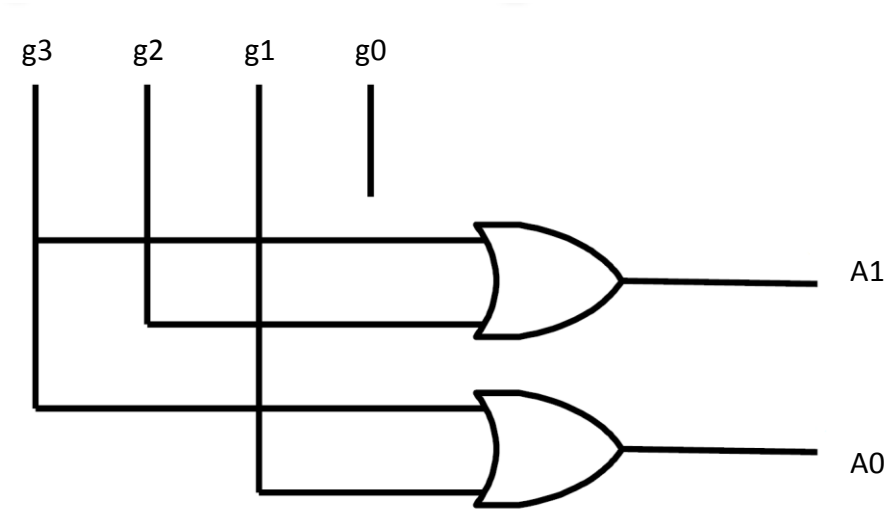
4-bit encoder :

Input: {g3, g2, g1, g0}

Output: {A1, A0}

A1 = g3 + g2

A0 = g3 + g1



D3	D2	D1	D0	g3	g2	g1	g0	A1	A0
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1	0	0
0	0	1	0	0	0	1	0	0	1
0	0	1	1	0	0	0	1	0	0
0	1	0	0	0	1	0	0	1	0
0	1	0	1	0	0	0	1	0	0
0	1	1	0	0	0	1	0	0	1
0	1	1	1	0	0	0	1	0	0
1	0	0	0	1	0	0	0	1	1
1	0	0	1	0	0	0	1	0	0
1	0	1	0	0	0	1	0	0	1
1	0	1	1	0	0	0	1	0	0
1	1	0	0	0	1	0	0	1	0
1	1	0	1	0	0	0	1	0	0
1	1	1	0	0	0	1	0	0	1
1	1	1	1	0	0	0	1	0	0