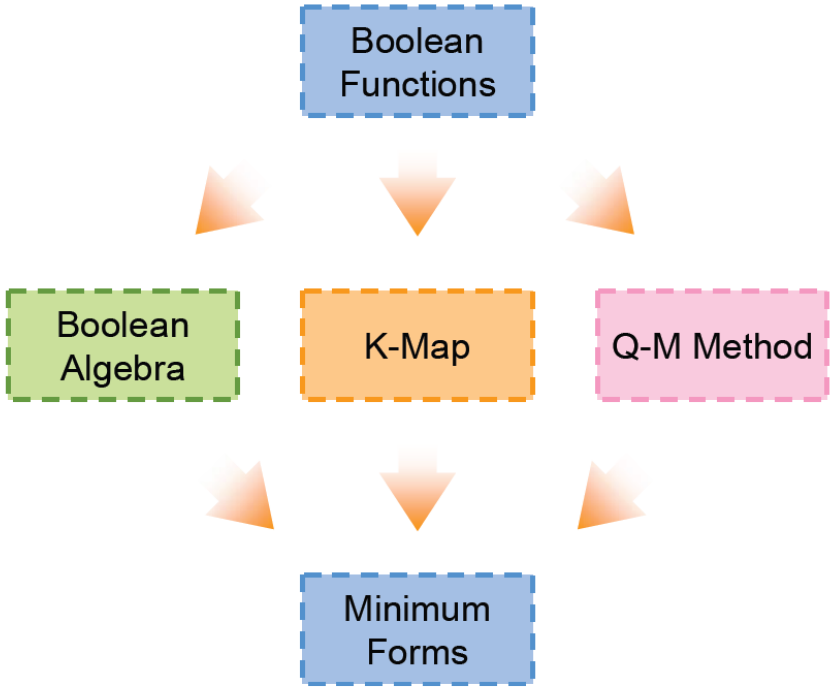
EE2000 Logic Circuit Design

Lecture 3 – Combinational System Design



Exercise

A security gate system has a binary output *Z* that is used to control the gate "open and close" where *Z* = 1 for the gate to open. The gate will close when the system detects the first two displayed numbers from a user's ID card {00, 01, 02, …, 15} with the number 06, 07, 11, 12, 13, 14, or 15. The variables *a*, *b*, *c*, and *d* determine a 4-digit binary code to represent the first two decimal numbers shown on the ID card.

Determine the truth table for the above security gate system.

State the case

Design a circuit to control the gate

- The gate is closed when ID numbers are 06, 07, 11, 12, 13, 14, 15.
- The gate is open for other ID numbers.
- Open/Close gate is a binary decision output (*Z*): 1 for Open and 0 for Close.
- 4-digit binary codes are the inputs (*a*, *b*, *c*, *d*).

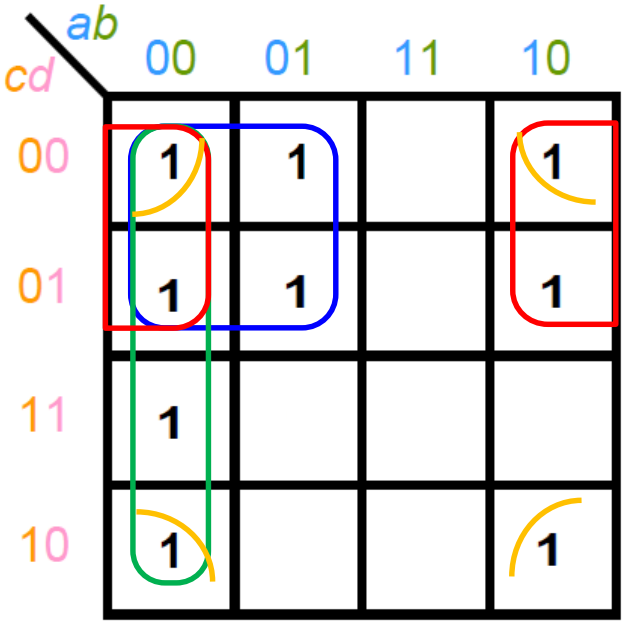
Exercise

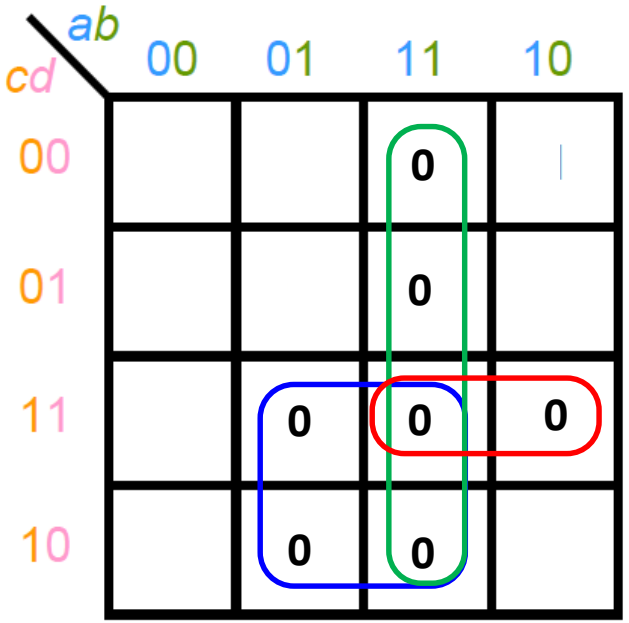
$f(a, b, c, d) = Z = 0$ when ID numbers 06, 07, 11, 12, 13, 14 or 15

Else $Z = 1$

Decimal numbers	Inputs				Output
	a	b	c	d	Z
00	0	0	0	0	1
01	0	0	0	1	1
02	0	0	1	0	1
03	0	0	1	1	1
04	0	1	0	0	1
05	0	1	0	1	1
06	0	1	1	0	0
07	0	1	1	1	0
08	1	0	0	0	1
09	1	0	0	1	1
10	1	0	1	0	1
11	1	0	1	1	0
12	1	1	0	0	0
13	1	1	0	1	0
14	1	1	1	0	0
15	1	1	1	1	0

Exercise





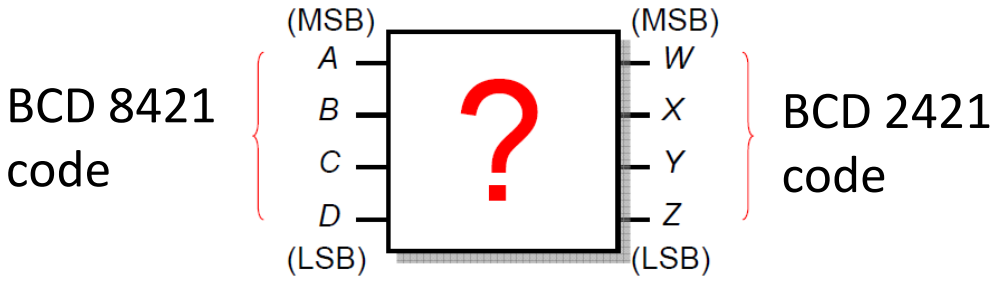
$f(a,b,c,d)$
$= a'b' + a'c' + b'c' + b'd'$

$f'(a,b,c,d) = ab + bc + acd$

$f(a,b,c,d)$
$= (a' + b')(b' + c')(a' + c' + d')$

Exercise

- Design a logic circuit that perform code conversion



- Input is BCD 8421 code
- Output is BCD 2421 code

State the case

Design a circuit to convert the BCD 8421 to the BCD 2421 code

- *A, B, C, D* are the input.
- *W, X, Y, Z* are the output.
- The output functions are:

  W (*A, B, C, D*)
  X (*A, B, C, D*)
  Y (*A, B, C, D*)
  Z (*A, B, C, D*)

Formulation

Decimal numbers	Inputs (8421)				Outputs (2421)			
	A	B	C	D	W	X	Y	Z
0	0	0	0	0	0	0	0	0
1	0	0	0	1	0	0	0	1
2	0	0	1	0	0	0	1	0
3	0	0	1	1	0	0	1	1
4	0	1	0	0	0	1	0	0
5	0	1	0	1	1	0	1	1
6	0	1	1	0	1	1	0	0
7	0	1	1	1	1	1	0	1
8	1	0	0	0	1	1	1	0
9	1	0	0	1	1	1	1	1
Unused	x	x	x	x	x	x	x	x
Unused	x	x	x	x	x	x	x	x
Unused	x	x	x	x	x	x	x	x
Unused	x	x	x	x	x	x	x	x
Unused	x	x	x	x	x	x	x	x
Unused	x	x	x	x	x	x	x	x

$$W(A,B,C,D) = \sum m(5,6,7,8,9) + \sum d(10,11,12,13,14,15)$$

$$X(A,B,C,D) = \sum m(4,6,7,8,9) + \sum d(10,11,12,13,14,15)$$

$$Y(A,B,C,D) = \sum m(2,3,5,8,9) + \sum d(10,11,12,13,14,15)$$

$$Z(A,B,C,D) = \sum m(1,3,5,7,9) + \sum d(10,11,12,13,14,15)$$

K-maps

cd \ ab	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}

K-map for *W*:

CD \ AB	00	01	11	10
00			x	1
01		1	x	1
11		1	x	x
10		1	x	x

K-map for *X*:

CD \ AB	00	01	11	10
00		1	x	1
01			x	1
11		1	x	x
10		1	x	x

K-map for *Y*:

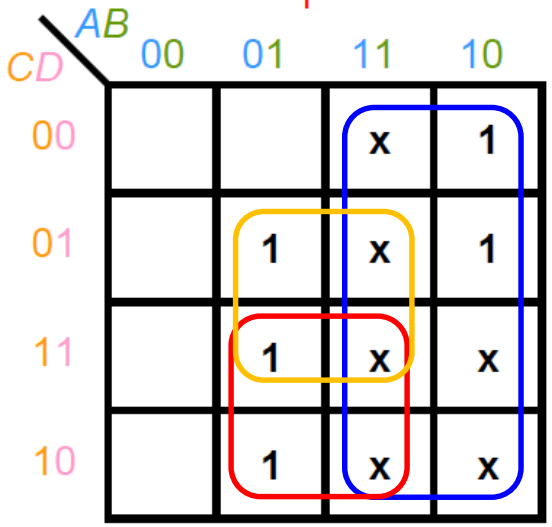
CD \ AB	00	01	11	10
00			x	1
01		1	x	1
11	1		x	x
10	1		x	x

K-map for *Z*:

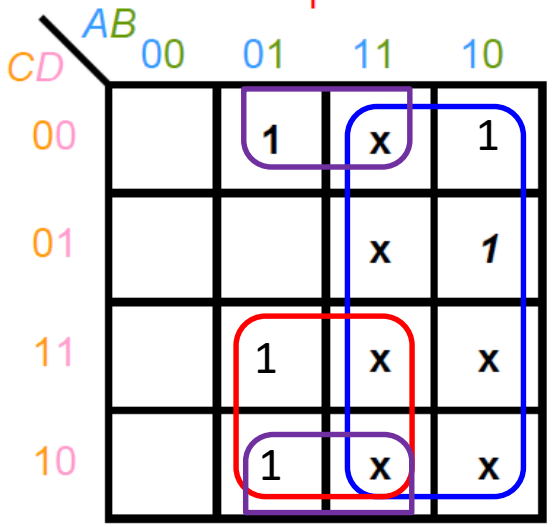
CD \ AB	00	01	11	10
00			x	
01	1	1	x	1
11	1	1	x	x
10			x	x

Simplification

K-map for *W*:



K-map for *X*:



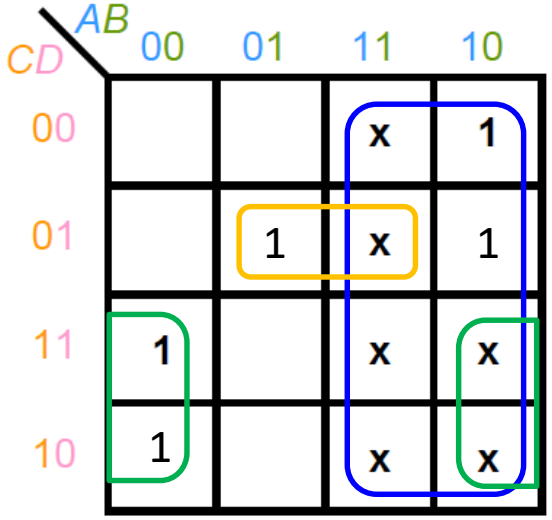
$$W = A + BC + BD$$

$$X = A + BC + BD'$$

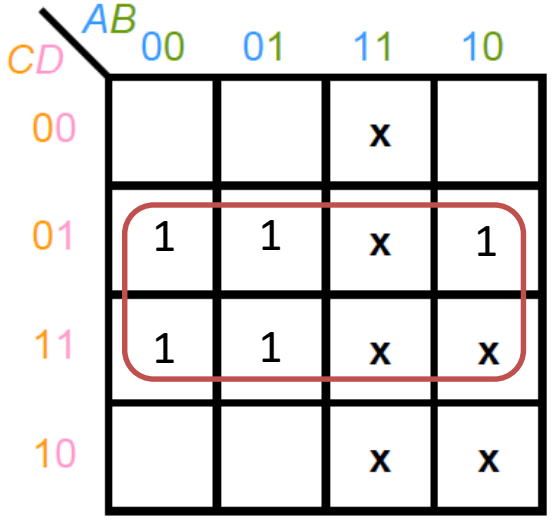
$$Y = A + B'C + BC'D$$

$$Z = D$$

K-map for *Y*:



K-map for *Z*:



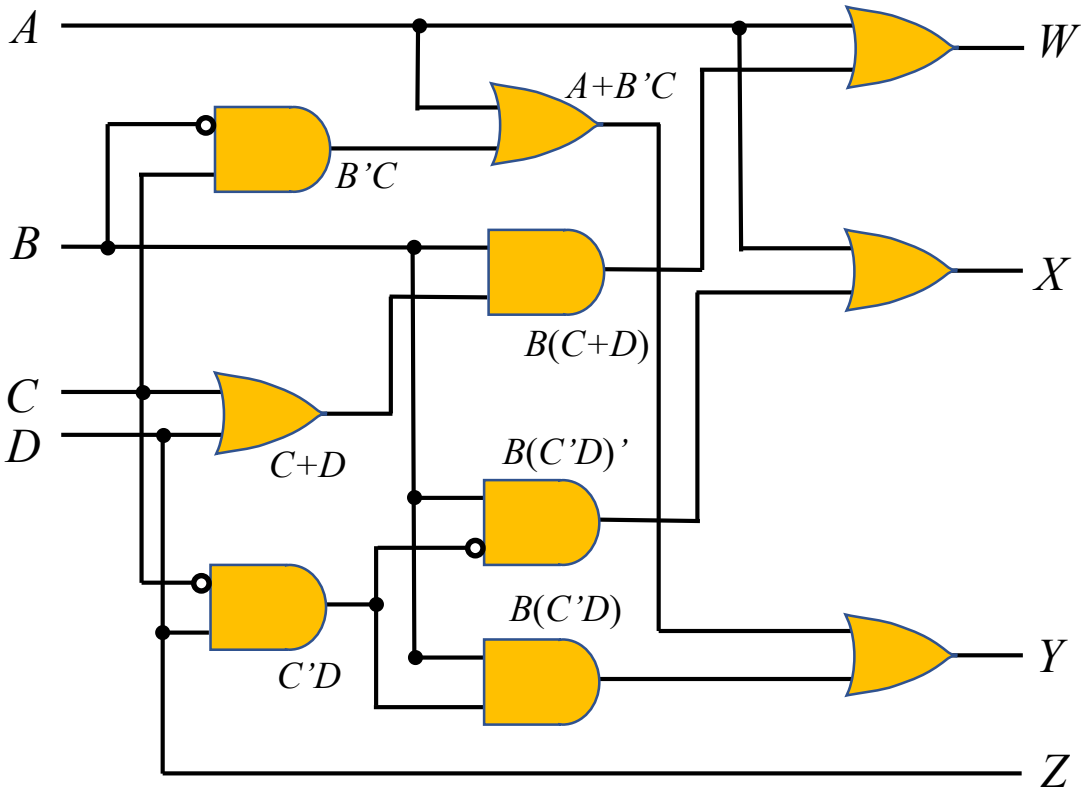
Logic Circuit

$$W = A + BC + BD = A + B(C + D)$$

$$X = A + BC + BD' = A + B(C + D') = A + B(C'D)'$$

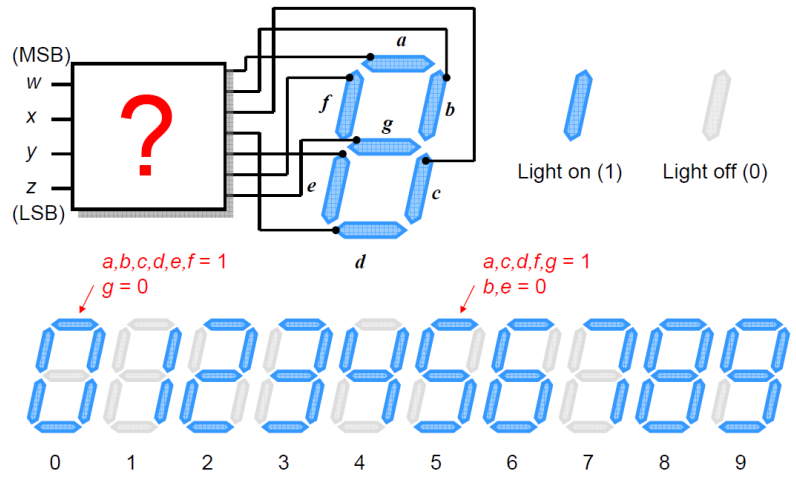
$$Y = A + B'C + B(C'D)$$

$$Z = D$$



*Using only Two-input Gates and NOT Gates.

Formulation



State the case

Design a circuit to convert the BCD 8421 to the 7-segment display

- *w, x, y, z* are the input.
- *a, b, c, d, e, f, g* are the output.

numbers	Inputs				7-segment display						
	w	x	y	z	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1
10	1	0	1	0	x	x	x	x	x	x	x
11	1	0	1	1	x	x	x	x	x	x	x
12	1	1	0	0	x	x	x	x	x	x	x
13	1	1	0	1	x	x	x	x	x	x	x
14	1	1	1	0	x	x	x	x	x	x	x
15	1	1	1	1	x	x	x	x	x	x	x

$$a\,(w,x,y,z) =$$

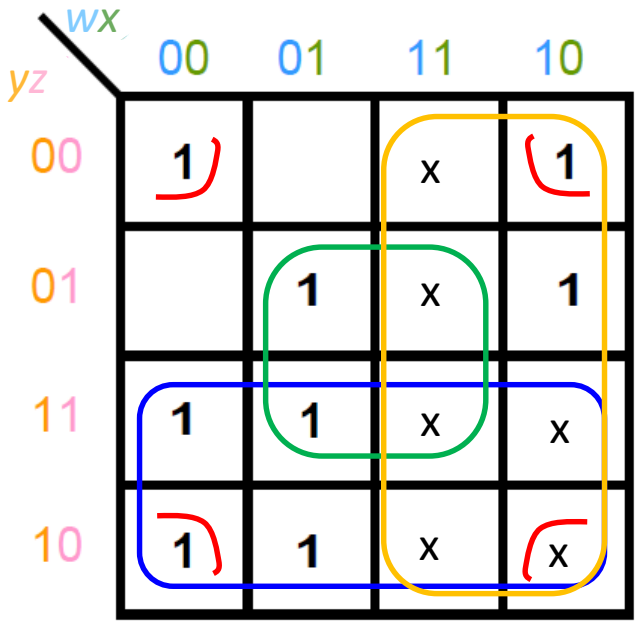
$$\Sigma\, m\,(0,2,3,5,6,7,8,9) + \Sigma d(10,11,12,13,14,15)$$

K-map for segment ‘a’

cd \ ab	00	01	11	10
00	m_0	m_4	m_{12}	m_8
01	m_1	m_5	m_{13}	m_9
11	m_3	m_7	m_{15}	m_{11}
10	m_2	m_6	m_{14}	m_{10}

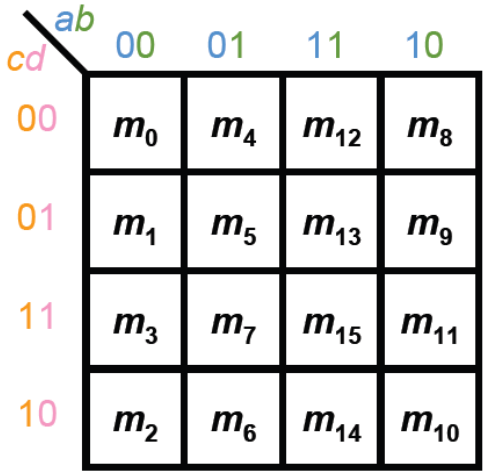
$$a\,(w,x,y,z) = \Sigma m(0,2,3,5,6,7,8,9) + \Sigma d(10,11,12,13,14,15)$$

yz \ wx	00	01	11	10
00	1		x	1
01		1	x	1
11	1	1	x	x
10	1	1	x	x

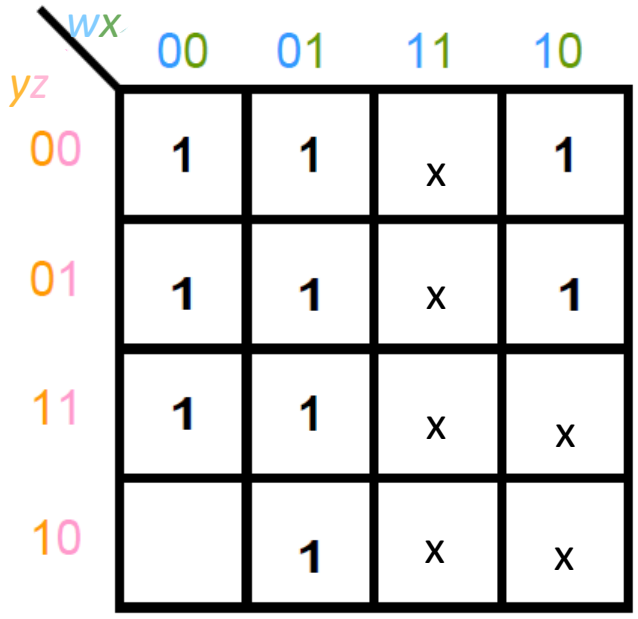


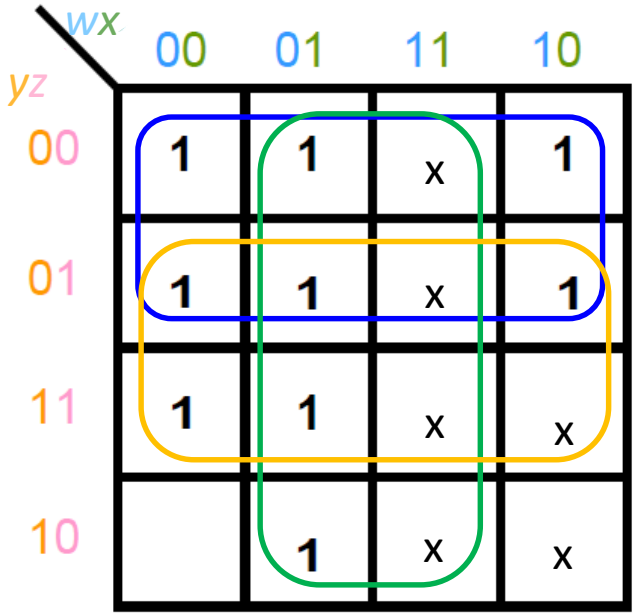
$$a(w,x,y,z) = w + y + xz + x'z'$$

K-map for segment ‘c’



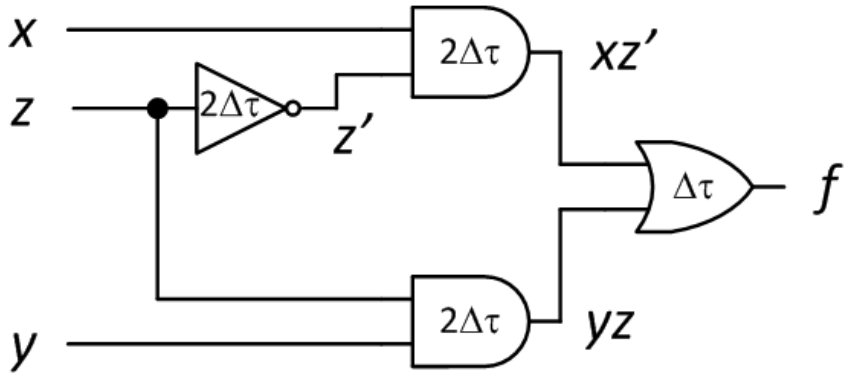
$$c\,(w,x,y,z) = \Sigma m(0,1,3,4,5,6,7,8,9) + \Sigma d(10,11,12,13,14,15)$$

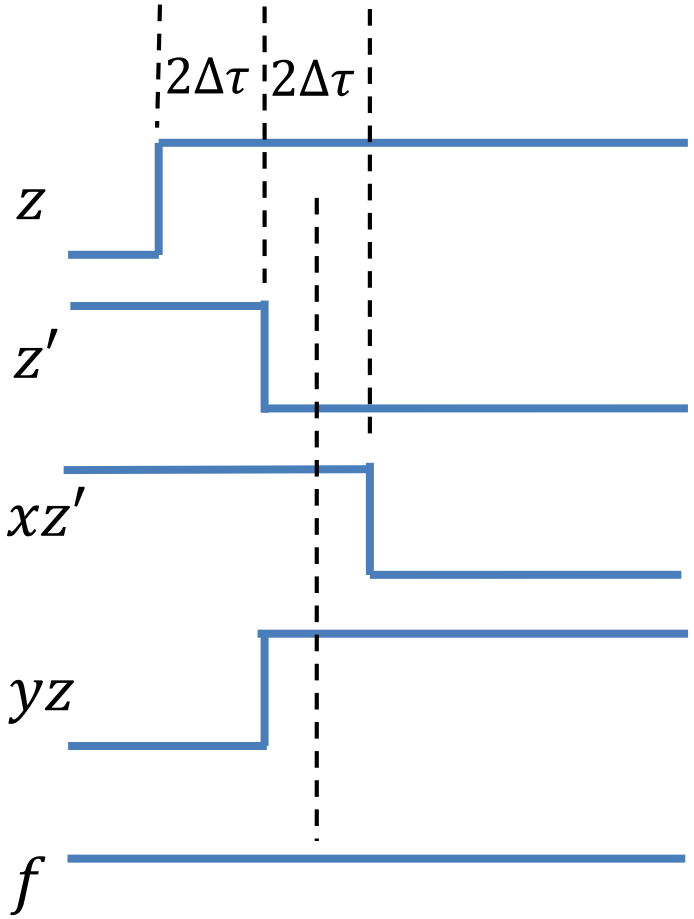




$$c(w,x,y,z) = x + y' + z$$

Exercise



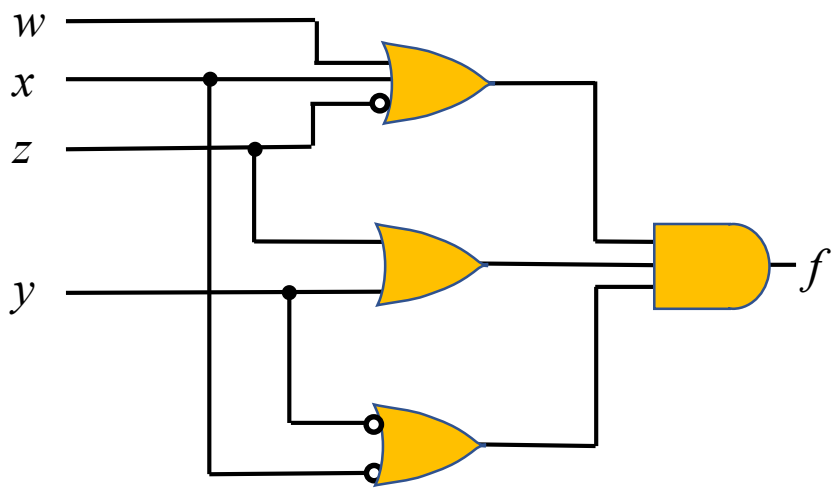


Assume that the propagation delay of each gate are as shown above.

Work out the timing diagram to identify the presence of any timing hazard when the input condition changes from $(x, y, z) = (1,1,0)$ to $(1,1,1)$.

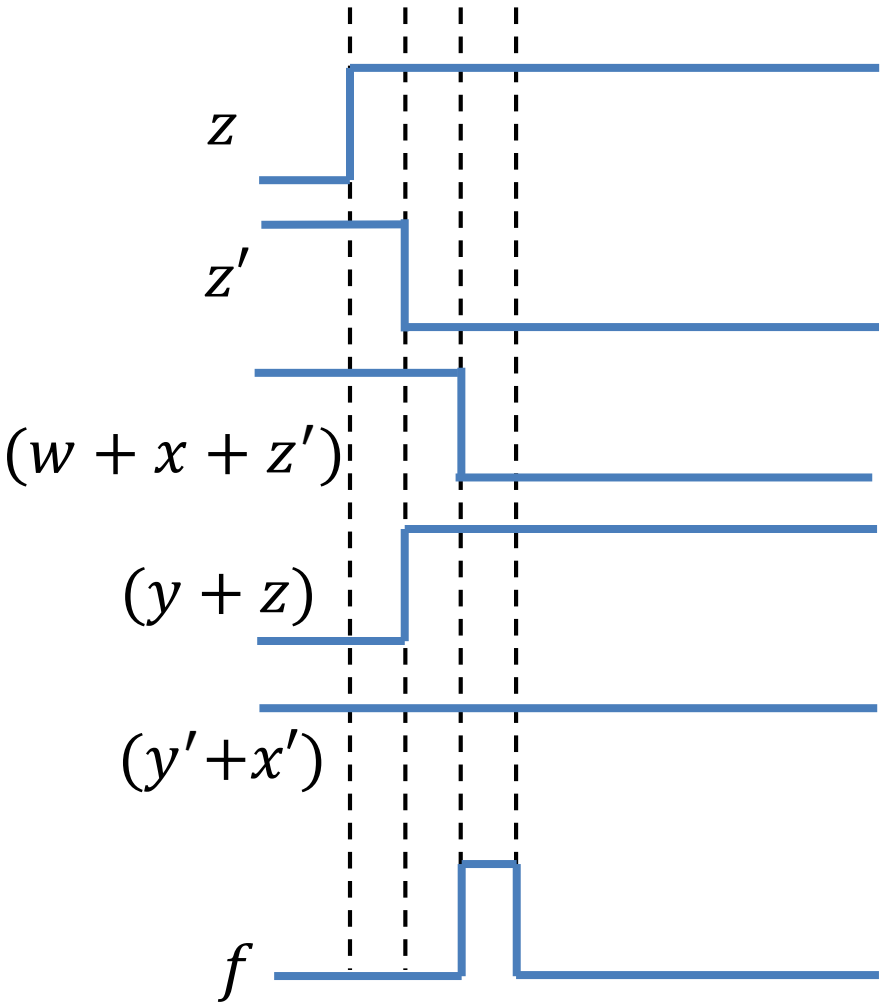
No hazard!!!

Exercise



Assume that the propagation delay of all gates is $\Delta\tau$.

Work out the timing diagram to identify the presence of any timing hazard when the input condition changes from $(w, x, y, z) = (0,0,0,0)$ to $(0,0,0,1)$.



Static-0 hazard!!!

Exercise

Determine the Hamming code using both odd and even parity bit for a data core of 11100

For a 5-bit data $d_5 d_4 d_3 d_2 d_1$, $n = 5$

$$2^k \geq 6 + k$$

Therefore, minimum value of k is 4. We need **4 parity bits**!

Exercise

Step 2: Place Parity Bits in the positions of powers of 2.

Hamming Code	H_9	H_8	H_7	H_6	H_5	H_4	H_3	H_2	H_1
	d_5	p_4	d_4	d_3	d_2	p_3	d_1	p_2	p_1
Bit	9	**8**	7	6	5	**4**	3	**2**	**1**

Exercise

Step 2: Place Parity Bits in the positions of powers of 2.

Hamming Code	H_9	H_8	H_7	H_6	H_5	H_4	H_3	H_2	H_1
	d_5	p_4	d_4	d_3	d_2	p_3	d_1	p_2	p_1
Bit	9	8	7	6	5	4	3	2	1
Binary Code	1001	1000	0111	0110	0101	0100	0011	0010	0001
p_1	1		1		0		0		
p_2			1	1			0		
p_3			1	1	0				
p_4	1								
Even Parity	1	1	1	1	0	0	0	0	0
Odd Parity	1	0	1	1	0	1	0	1	1

Step 3: Calculate the number of '1' in each parity bits

Step 4: Place '1' if odd number of '1' for even parity; else '0'; Place '0' if odd number of '1' for odd parity; else '1'