Combinational logic circuits

EIC 0844091

Digital Circuit and Logic Design

Associate Prof. Luo Jie

Huazhong University of Science & Technology

* << >> ↵

Presentation Outline

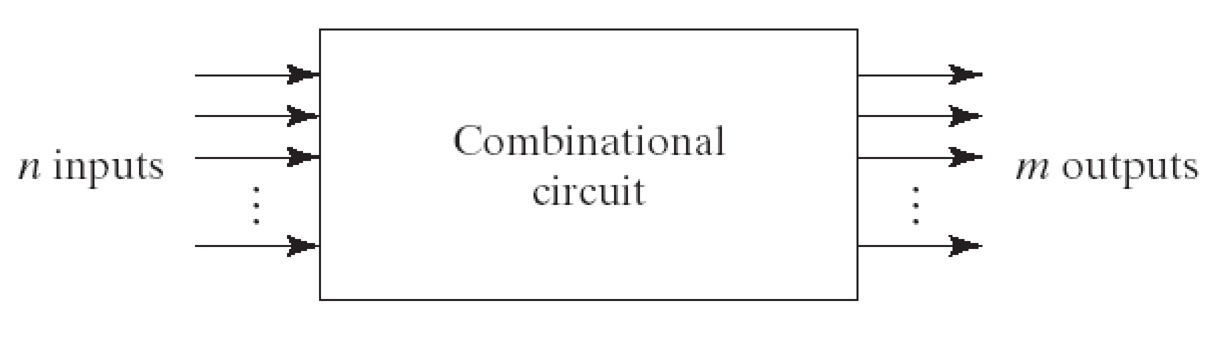
- Combinational logic circuits
- Logic circuit analysis
- Logic circuit design

Combinational logic circuits

□ Logic circuits are classified into two types, “combinational” and “sequential.”

□ A combinational logic circuit is one whose outputs depend only on its current inputs.

Structurally, there are no storage elements in combinational logic circuit.



Block Diagram of Combinational Circuit

Next topic

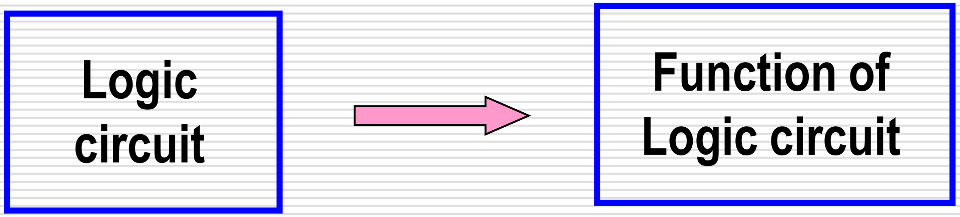
- Combinational logic circuits
- Logic circuit analysis
- Logic circuit design

* << >> ↵

Logic circuit analysis

□ Analysis

The process to obtain the logic function of a logic circuit is called analysis.



The process usually includes following steps:

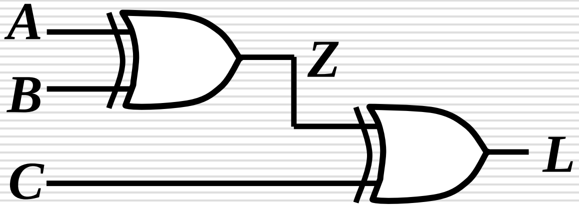
1. Derive the Boolean expressions of outputs from logic circuit.
2. Derive the truth table from Boolean expressions.
3. Determine the logic function according to truth table or Boolean expressions.

Logic circuit analysis

□ Analysis

Example 1

A logic diagram is shown in figure. Determine its logic function.



Solution:

Boolean expressions

$$Z = A\overline{B} + \overline{A}B = A \oplus B$$

$$L = Z\overline{C} + \overline{Z}C = Z \oplus C = A \oplus B \oplus C$$

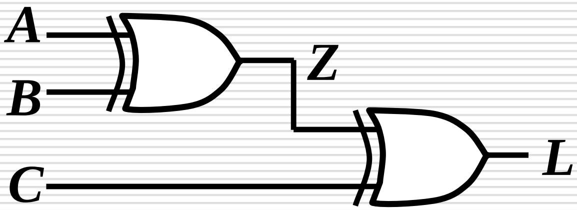
Truth table

Truth table

A B C	*Z*	*L*
0 0 0	0	0
0 0 1	0	1
0 1 0	1	1
0 1 1	1	0
1 0 0	1	1
1 0 1	1	0
1 1 0	0	0
1 1 1	0	1

Logic circuit analysis

□ Analysis



Solution:

Boolean expressions

$$Z = A\overline{B} + \overline{A}B = A \oplus B$$

$$L = Z\overline{C} + \overline{Z}C = Z \oplus C = A \oplus B \oplus C$$

Truth table

Analyzing the logic function

When the number of 1s on the inputs is odd, the output L is 1. It is called 3-input odd parity checker.

How to construct a 4-input odd parity checker?

Truth table

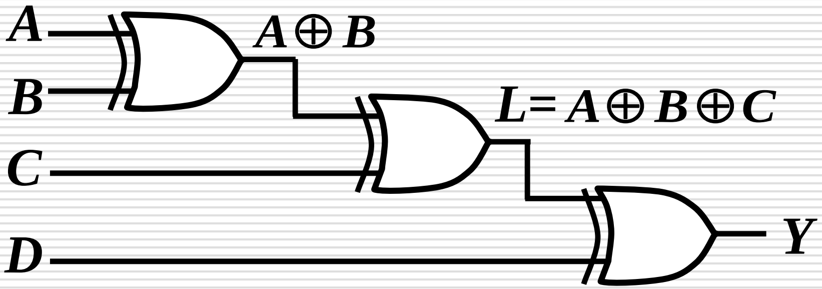
A B C	Z	L
0 0 0	0	0
0 0 1	0	1
0 1 0	1	1
0 1 1	1	0
1 0 0	1	1
1 0 1	1	0
1 1 0	0	0
1 1 1	0	1

Logic circuit analysis

even parity output

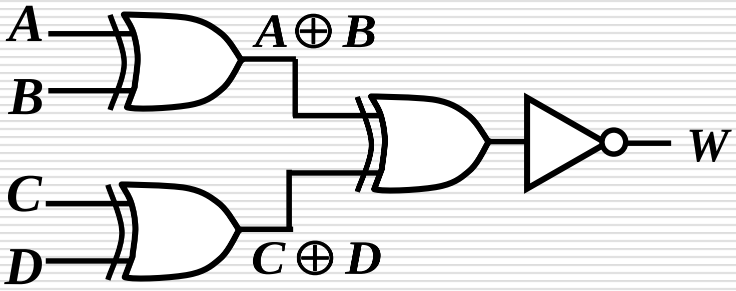
How to construct a 4-input odd parity checker?

$$L = A\oplus B\oplus C$$



$$Y = A\oplus B\oplus C\oplus D$$

$$=(A\oplus B)\oplus(C\oplus D)=A\oplus(B\oplus C\oplus D)$$



How to construct a 4-input even parity checker?

Truth table

$ABCD$	L	Y	W
0 0 0 0	0	0	1
0 0 0 1	0	1	0
0 0 1 0	1	1	0
0 0 1 1	1	0	1
0 1 0 0	1	1	0
0 1 0 1	1	0	1
0 1 1 0	0	0	1
0 1 1 1	0	1	0
1 0 0 0	1	1	0
1 0 0 1	1	0	1
1 0 1 0	0	0	1
1 0 1 1	0	1	0
1 1 0 0	0	0	1
1 1 0 1	0	1	0
1 1 1 0	1	1	0
1 1 1 1	1	0	1

Logic circuit analysis

Note:
Ignoring the transition time and propagation delay usually when analyzing only logic function unless otherwise stated.

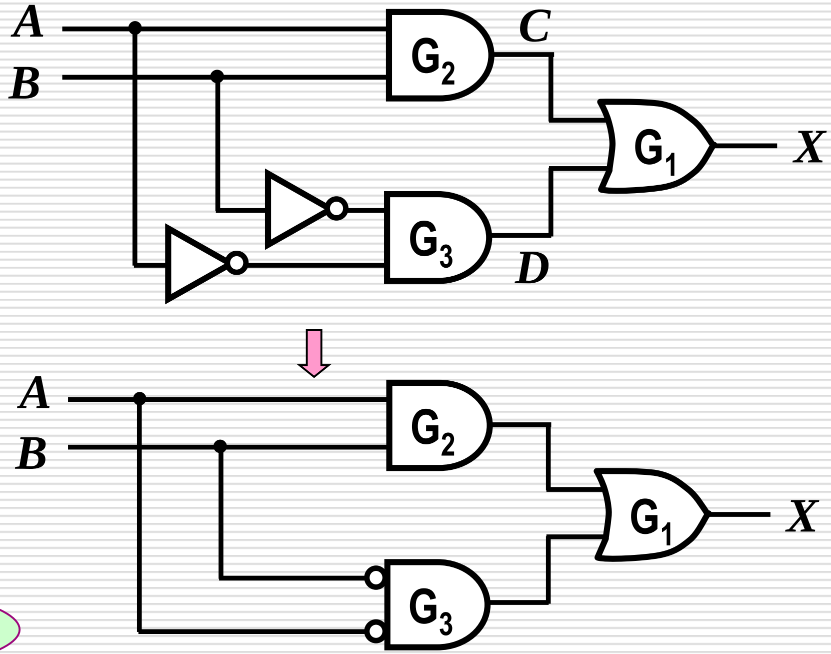
□ Analysis

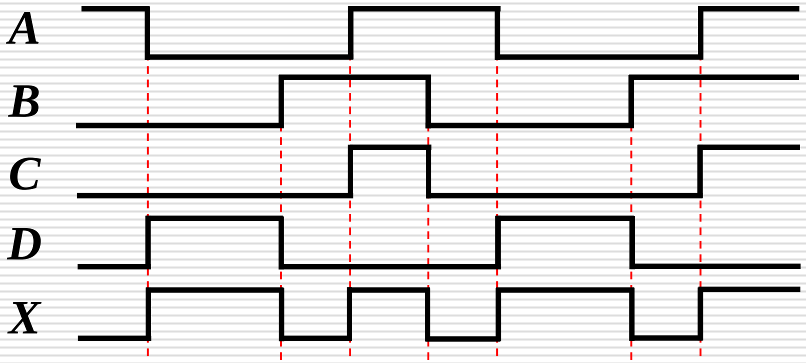
Example 2

Determine the output waveform X for the logic circuit in figure, with input waveforms A and B as shown.

Solution:

Marking intermediate variables





$$X = AB + \overline{A}\,\overline{B}$$

XNOR operation

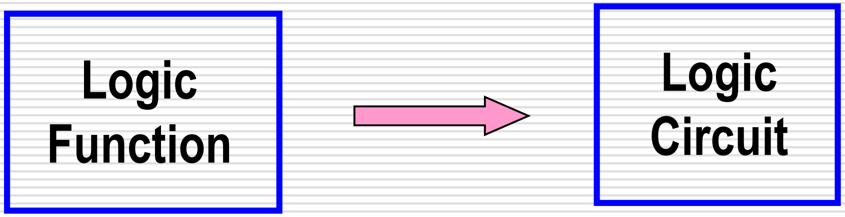
Next topic

- Combinational logic circuits
- Logic circuit analysis
- Logic circuit design

Logic circuit design

□ Design

The process to implement the logic function using a certain logic circuit is called design.



Logic Function

Logic Circuit

The process usually includes following steps:

1. From the specification of the logic function, determine the required number of inputs and outputs and assign a symbol to each.

2. Derive the truth table that defines the required relationship between inputs and outputs.

3. Obtain the simplified Boolean expressions for each output.

4. Draw the logic diagram.

Logic circuit design

□ Design

Example 1

Design a logic circuit for converting BCD code to Excess-3 code.

Solution:

Inputs: A_3, A_2, A_1, A_0

Outputs: Y_3, Y_2, Y_1, Y_0

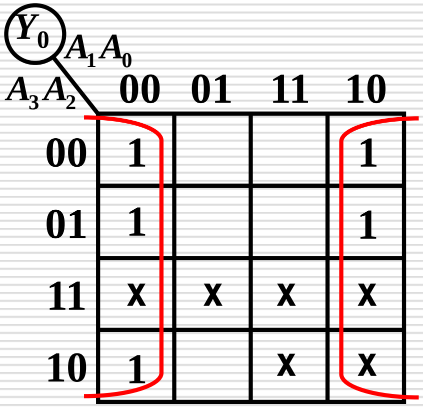
Truth table

Using K-Map to simplify the function for each output

Dec	$A_3A_2A_1A_0$	$Y_3Y_2Y_1Y_0$
0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 1 0 1
3	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 0 0
6	0 1 1 0	1 0 0 1
7	0 1 1 1	1 0 1 0
8	1 0 0 0	1 0 1 1
9	1 0 0 1	1 1 0 0
10	1 0 1 0	x x x x
11	1 0 1 1	x x x x
12	1 1 0 0	x x x x
13	1 1 0 1	x x x x
14	1 1 1 0	x x x x
15	1 1 1 1	x x x x

Design examples

Using K-Map to simplify the function for each output

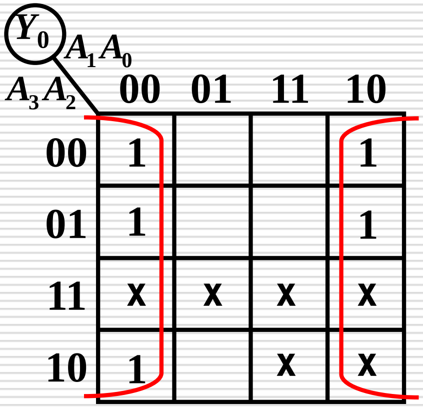


Y_0 (A_3A_2 \ A_1A_0)	00	01	11	10
00	1			1
01	1			1
11	x	x	x	x
10	1		x	x

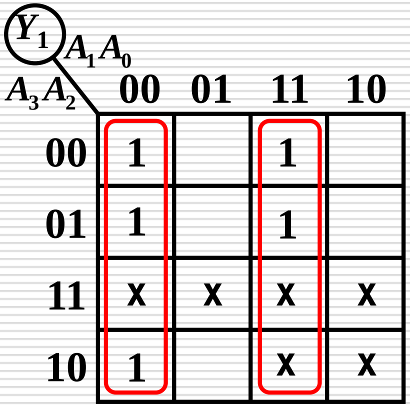
$A_3A_2A_1A_0$	$Y_3Y_2Y_1Y_0$
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0
1 0 0 0	1 0 1 1
1 0 0 1	1 1 0 0
1 0 1 0	x x x x
1 0 1 1	x x x x
1 1 0 0	x x x x
1 1 0 1	x x x x
1 1 1 0	x x x x
1 1 1 1	x x x x

Design examples

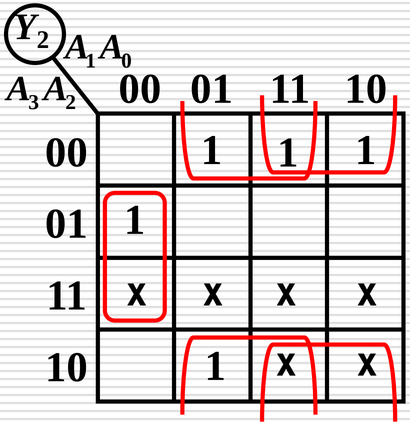
Using K-Map to simplify the function for each output



Y_0: A_3A_2 \ A_1A_0	00	01	11	10
00	1			1
01	1			1
11	x	x	x	x
10	1		x	x



Y_1: A_3A_2 \ A_1A_0	00	01	11	10
00	1		1	
01	1		1	
11	x	x	x	x
10	1		x	x



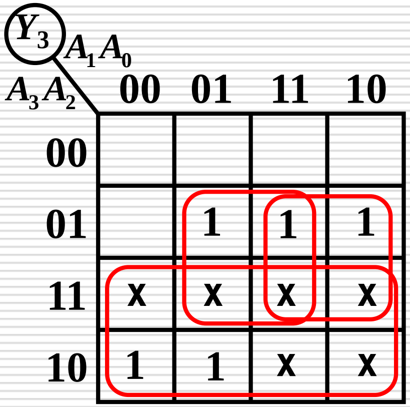
Y_2: A_3A_2 \ A_1A_0	00	01	11	10
00		1	1	1
01	1			
11	x	x	x	x
10		1	x	x

$$Y_0 = \overline{A_0}$$

$$Y_1 = \overline{A_1}\,\overline{A_0} + A_1A_0$$

$$Y_2 = \overline{A_2}A_0 + \overline{A_2}A_1 + A_2\overline{A_1}\,\overline{A_0}$$

$$Y_3 = A_3 + A_2A_0 + A_2A_1$$



Y_3: A_3A_2 \ A_1A_0	00	01	11	10
00				
01		1	1	1
11	x	x	x	x
10	1	1	x	x

$A_3A_2A_1A_0$	$Y_3Y_2Y_1Y_0$
0 0 0 0	0 0 1 1
0 0 0 1	0 1 0 0
0 0 1 0	0 1 0 1
0 0 1 1	0 1 1 0
0 1 0 0	0 1 1 1
0 1 0 1	1 0 0 0
0 1 1 0	1 0 0 1
0 1 1 1	1 0 1 0
1 0 0 0	1 0 1 1
1 0 0 1	1 1 0 0
1 0 1 0	x x x x
1 0 1 1	x x x x
1 1 0 0	x x x x
1 1 0 1	x x x x
1 1 1 0	x x x x
1 1 1 1	x x x x

Design examples

Note:

when the circuit have more than one output, the common parts in output expressions are more, the final circuit is more simpler.

In this case, we hope the expressions contain most possible common parts.

The logic diagram

$$Y_0 = \overline{A_0}$$

$$Y_1 = \overline{A_1}\,\overline{A_0} + A_1A_0$$

$$Y_2 = \overline{A_2}A_0 + \overline{A_2}A_1 + A_2\overline{A_1}\,\overline{A_0}$$

$$Y_3 = A_3 + A_2A_0 + A_2A_1$$

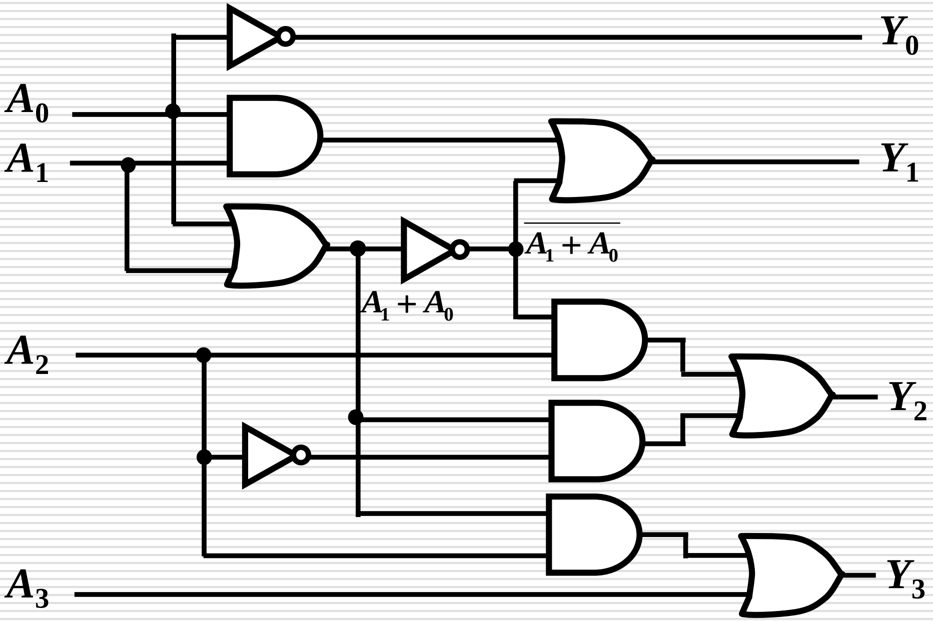
Transformation

$$Y_0 = \overline{A_0}$$

$$\begin{aligned} Y_1 &= \overline{A_1}\,\overline{A_0} + A_1A_0 \\ &= \overline{A_1 + A_0} + A_1A_0 \end{aligned}$$

$$\begin{aligned} Y_2 &= \overline{A_2}A_0 + \overline{A_2}A_1 + A_2\overline{A_1}\,\overline{A_0} \\ &= \overline{A_2}(A_1 + A_0) + A_2(\overline{A_1 + A_0}) \end{aligned}$$

$$\begin{aligned} Y_3 &= A_3 + A_2A_0 + A_2A_1 \\ &= A_3 + A_2(A_1 + A_0) \end{aligned}$$



Logic circuit design

□ Design

Example 2

Design a logic circuit for converting Gray code to binary code.

Solution:

Inputs: G_3, G_2, G_1, G_0

Outputs: B_3, B_2, B_1, B_0

Truth table

Binary	Gray
$B_3B_2B_1B_0$	$G_3G_2G_1G_0$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 0	0 0 1 1
0 0 1 1	0 0 1 0
0 1 0 0	0 1 1 0
0 1 0 1	0 1 1 1
0 1 1 0	0 1 0 1
0 1 1 1	0 1 0 0
1 0 0 0	1 1 0 0
1 0 0 1	1 1 0 1
1 0 1 0	1 1 1 1
1 0 1 1	1 1 1 0
1 1 0 0	1 0 1 0
1 1 0 1	1 0 1 1
1 1 1 0	1 0 0 1
1 1 1 1	1 0 0 0

Logic circuit design

□ Design

Example 2

Design a logic circuit for converting Gray code to binary code.

Solution:

Inputs: G_3, G_2, G_1, G_0

Outputs: B_3, B_2, B_1, B_0

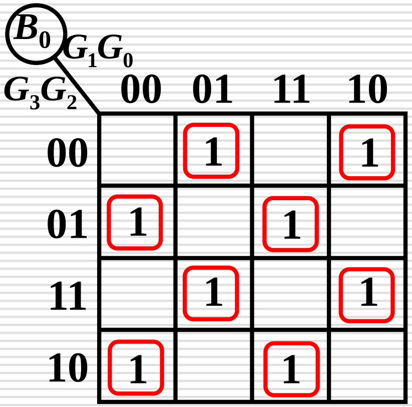
Truth table

Using K-Map to simplify the function for each output

Input	Output
Gray	Binary
$G_3G_2G_1G_0$	$B_3B_2B_1B_0$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 1	0 0 1 0
0 0 1 0	0 0 1 1
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 1	1 0 1 0
1 1 1 0	1 0 1 1
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 0 0 1	1 1 1 0
1 0 0 0	1 1 1 1

Design examples

Using K-Map to simplify the function for each output



B_0 G_3G_2 \ G_1G_0	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

Input	Output
Gray	Binary
$G_3G_2G_1G_0$	$B_3B_2B_1B_0$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 1	0 0 1 0
0 0 1 0	0 0 1 1
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 1	1 0 1 0
1 1 1 0	1 0 1 1
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 0 0 1	1 1 1 0
1 0 0 0	1 1 1 1

Design examples

Using K-Map to simplify the function for each output

B_0: G_3G_2 \ G_1G_0	00	01	11	10
00		1		1
01	1		1	
11		1		1
10	1		1	

B_1: G_3G_2 \ G_1G_0	00	01	11	10
00			1	1
01	1	1		
11			1	1
10	1	1		

B_2: G_3G_2 \ G_1G_0	00	01	11	10
00				
01	1	1	1	1
11				
10	1	1	1	1

B_3: G_3G_2 \ G_1G_0	00	01	11	10
00				
01				
11	1	1	1	1
10	1	1	1	1

$$B_3 = G_3$$

$$B_2 = \overline{G_3}G_2 + G_3\overline{G_2}$$

$$B_1 = G_3\overline{G_2}\,\overline{G_1} + G_3G_2G_1 + \overline{G_3}G_2\overline{G_1} + \overline{G_3}\,\overline{G_2}G_1$$

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

Input	Output
Gray	Binary
$G_3G_2G_1G_0$	$B_3B_2B_1B_0$
0 0 0 0	0 0 0 0
0 0 0 1	0 0 0 1
0 0 1 1	0 0 1 0
0 0 1 0	0 0 1 1
0 1 1 0	0 1 0 0
0 1 1 1	0 1 0 1
0 1 0 1	0 1 1 0
0 1 0 0	0 1 1 1
1 1 0 0	1 0 0 0
1 1 0 1	1 0 0 1
1 1 1 1	1 0 1 0
1 1 1 0	1 0 1 1
1 0 1 0	1 1 0 0
1 0 1 1	1 1 0 1
1 0 0 1	1 1 1 0
1 0 0 0	1 1 1 1

Design examples

The logic diagram

$$B_3 = G_3 \qquad B_2 = \overline{G_3}G_2 + G_3\overline{G_2}$$

$$B_1 = G_3\overline{G_2}\,\overline{G_1} + G_3G_2G_1 + \overline{G_3}G_2\overline{G_1} + \overline{G_3}\,\overline{G_2}G_1$$

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0$$

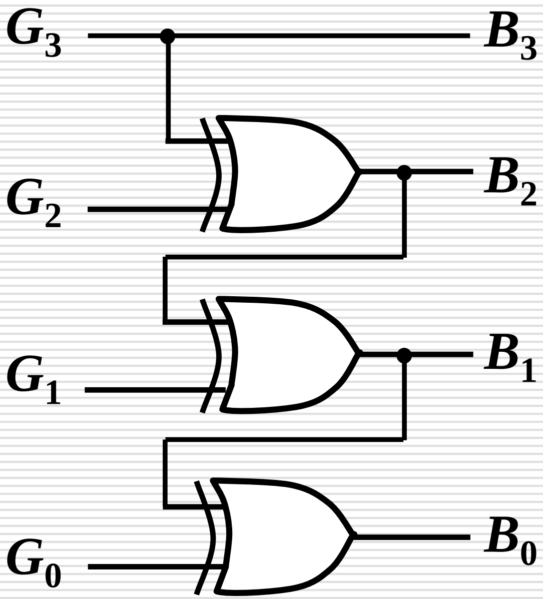
Transformation

$$B_3 = G_3$$

$$B_2 = \overline{G_3}G_2 + G_3\overline{G_2} = G_3 \oplus G_2$$

$$\begin{aligned} B_1 &= G_3\overline{G_2}\,\overline{G_1} + G_3G_2G_1 + \overline{G_3}G_2\overline{G_1} + \overline{G_3}\,\overline{G_2}G_1 \\ &= (G_3\overline{G_2} + \overline{G_3}G_2)\overline{G_1} + (G_3G_2 + \overline{G_3}\,\overline{G_2})G_1 \\ &= (G_3 \oplus G_2)\overline{G_1} + \overline{(G_3 \oplus G_2)}G_1 \\ &= G_3 \oplus G_2 \oplus G_1 = B_2 \oplus G_1 \end{aligned}$$

$$B_0 = G_3 \oplus G_2 \oplus G_1 \oplus G_0 = B_1 \oplus G_0$$



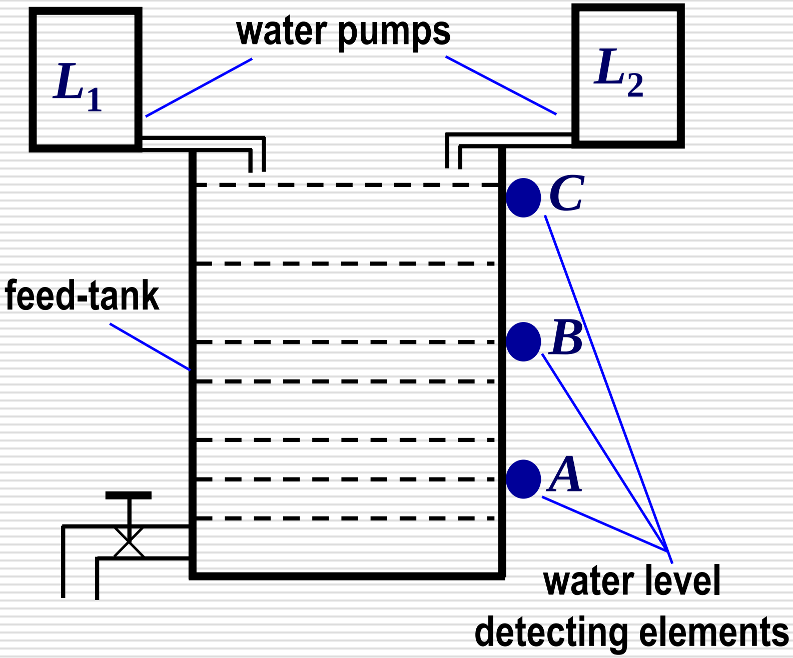
Design examples

□ Example 3

A water tank supply is shown in figure. The feed-tank are supplied with water by two water pumps L_1 and L_2. A, B and C represent three water level detecting elements respectively. When the water level is above their position on the feed-tank, the outputs of them are LOW. When the water level is below their position, the outputs of them are HIGH.

Design a logic circuit to control water pumps L_1 and L_2. It is required:

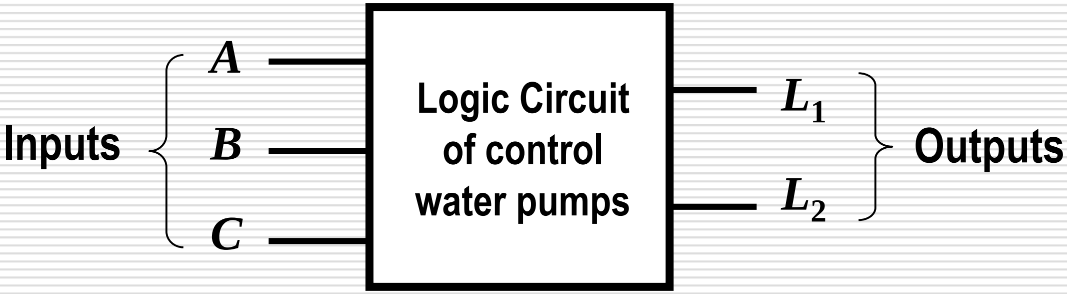
1. When the level is above the position of C, both L_1 and L_2 are stopping.
2. When the level is above the position of B and below the position of C, only L_1 is running.
3. When the level is above A and below B, only L_2 is running.
4. When the level is below A, both L_1 and L_2 are running.



Design examples

Solution:

Definition



Inputs: A, B, C

A, B and C indicate output of detecting elements respectively

Each of them is 0 if water level is above it, and it is 1 if water level is below it.

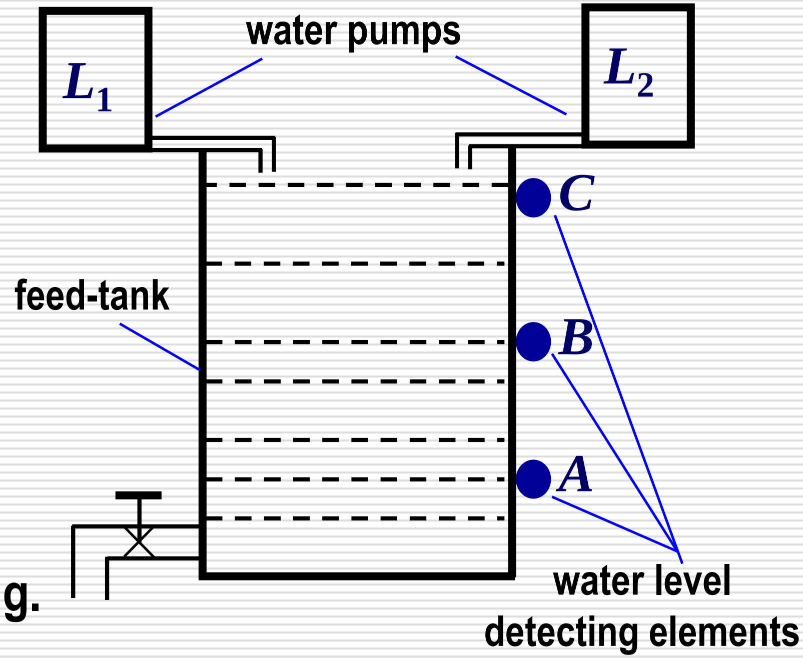
Outputs: L_2, L_1

When each of them is 1,

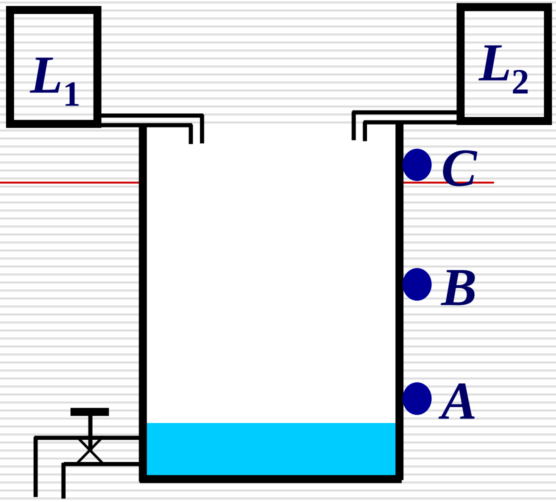
the corresponding pump is running.

When each of them is 0,

the corresponding the pump is stopping.



Design examples



Solution: Inputs: A, B, C

It is 0 if water level is above it.

It is 1 if water level is below it.

Outputs: L_2, L_1

It is 1, the corresponding pump is running.

It is 0, the corresponding the pump is stopping.

Truth table

It is required:

1. When the level is above the position of C, both L_1 and L_2 are stopping.
2. When the level is above the position of B and below the position of C, only L_1 is running.
3. When the level is above A and below B, only L_2 is running.
4. When the level is below A, both L_1 and L_2 are running.

Truth table

A	B	C	L_2	L_1
0	0	0	0	0
0	0	1	0	1
0	1	0	x	x
0	1	1	1	0
1	0	0	x	x
1	0	1	x	x
1	1	0	x	x
1	1	1	1	1

Four states never occur

Design examples

Solution:

The expressions and simplification

L_1 AB \ C	0	1
00		1
01	x	
11	x	1
10	x	x

$$L_1 = A + \overline{B}C$$

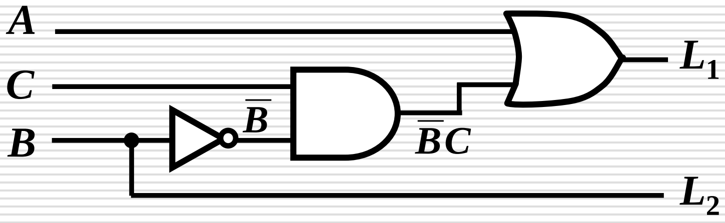
L_2 AB \ C	0	1
00		
01	x	1
11	x	1
10	x	x

$$L_2 = B$$

Truth table

$A\ B\ C$	$L_2\ L_1$
0 0 0	0 0
0 0 1	0 1
0 1 0	x x
0 1 1	1 0
1 0 0	x x
1 0 1	x x
1 1 0	x x
1 1 1	1 1

The logic diagram



Design examples

□ Example 4

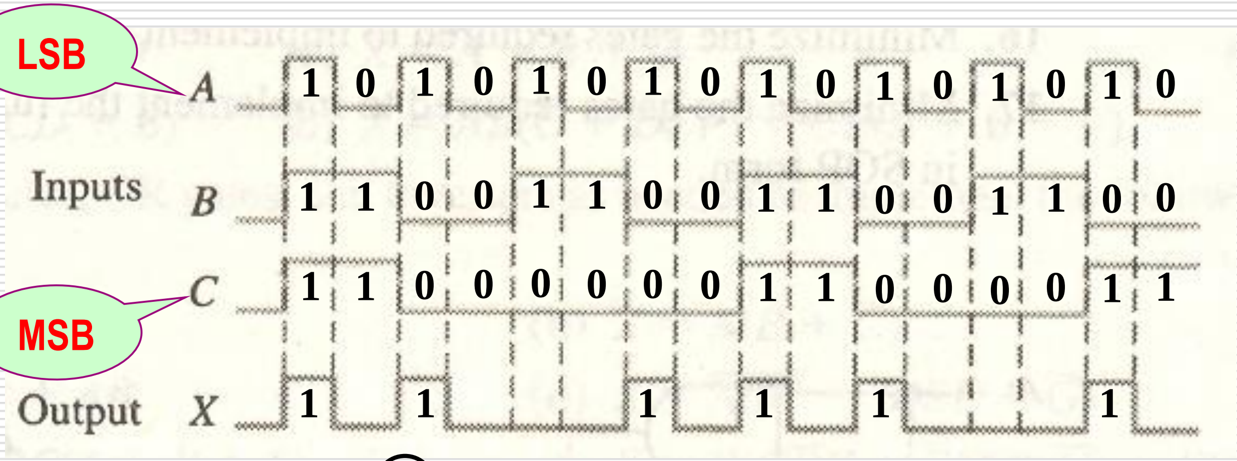
Given the input and output waveforms in figure. Design a logic circuit to implement the logic relationship between inputs and output.

Solution:

Construct the truth table from waveforms

LSB

MSB



Truth table

C	B	A	X
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

The expressions and simplification

$$X = CA + \overline{B}C$$

X CB \ A	0	1
00		1
01		
11		1
10		1

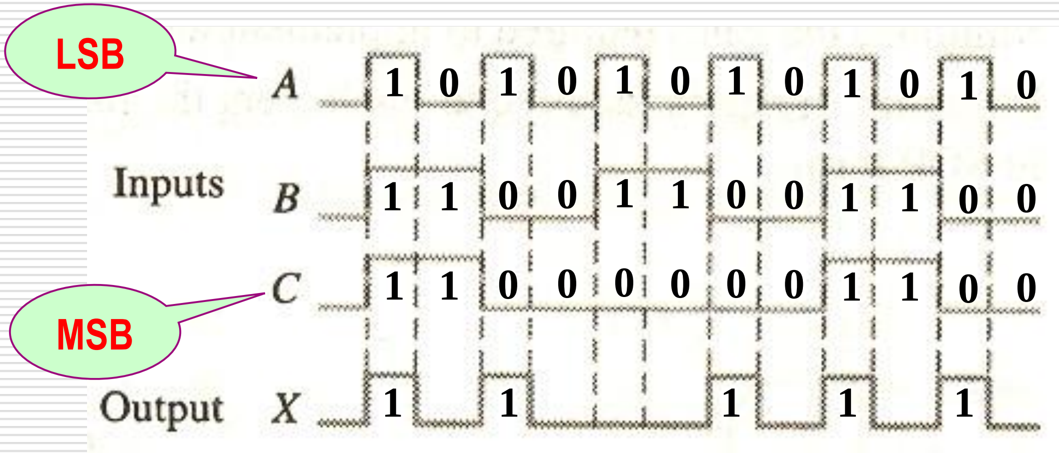
The logic diagram

(skipped)

Design examples

□ Example 4

What is its truth table if the waveforms are shown in figure ?


LSB
MSB
Inputs
A 1 0 1 0 1 0 1 0 1 0 1 0
B 1 1 0 0 1 1 0 0 1 1 0 0
C 1 1 0 0 0 0 0 0 1 1 0 0
Output
X 1 1 1 1 1

Truth table

C B A	*X*
0 0 0	0
0 0 1	1
0 1 0	0
0 1 1	0
1 0 0	x
1 0 1	x
1 1 0	0
1 1 1	1

For the combinations of inputs those don’t appear in waveforms corresponding outputs are filled with ‘x’.