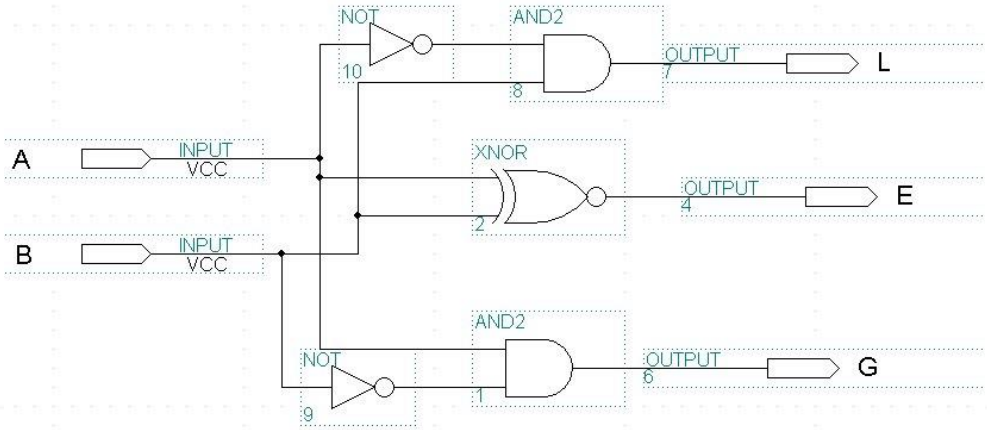


Logic Circuit Design

… we have studied…

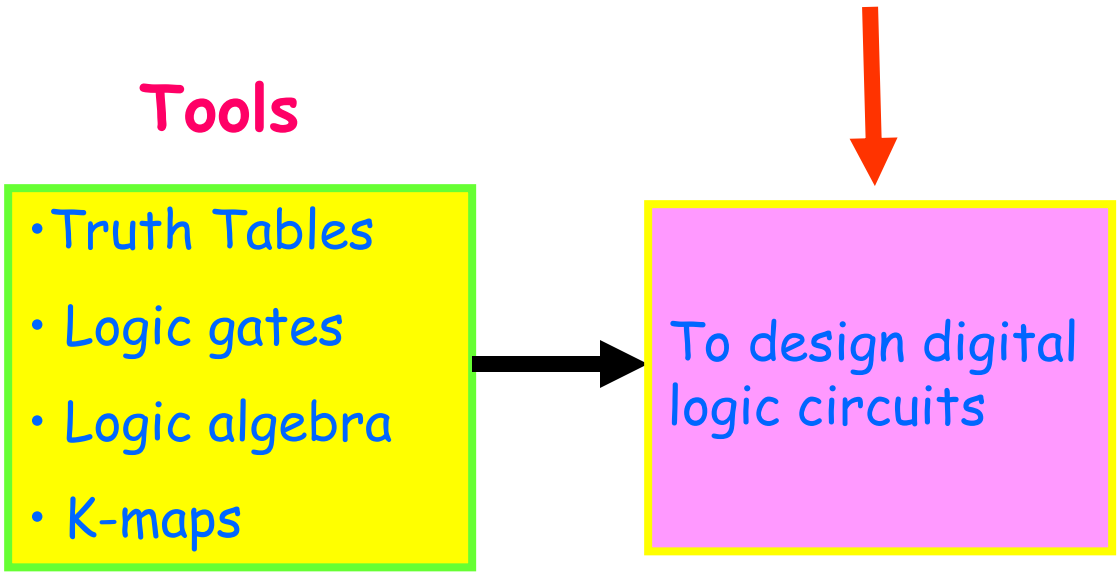
- Truth Tables
- Logic gates
- Logic algebra
- K-maps

All these are tools …

Tools

- Truth Tables
- Logic gates
- Logic algebra
- K-maps

All these are tools …



We know to simplify and implement

- Truth Tables
- Logic gates
- Logic algebra
- K-maps

Today we will learn

- Truth Tables
- Logic gates
- Logic algebra
- K-maps

- **To set-up and implement logic word problems (Logic design problem)**

4-Questions for a design problem

1. What do we need to know for a design problem?
   - Specifications (inputs, outputs, function)
2. Can we set-up a truth table ?
   - Truth tables determine the function of the problem
3. Can we simplify ?

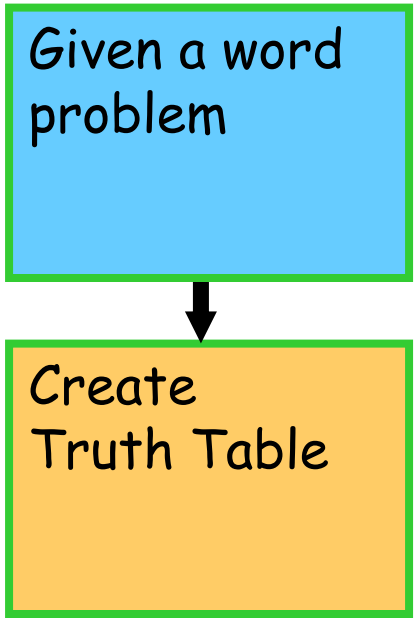
   - Use K- maps or logic algebra
4. Can we implement the result with a circuit?

   - Use gates (VHDL)

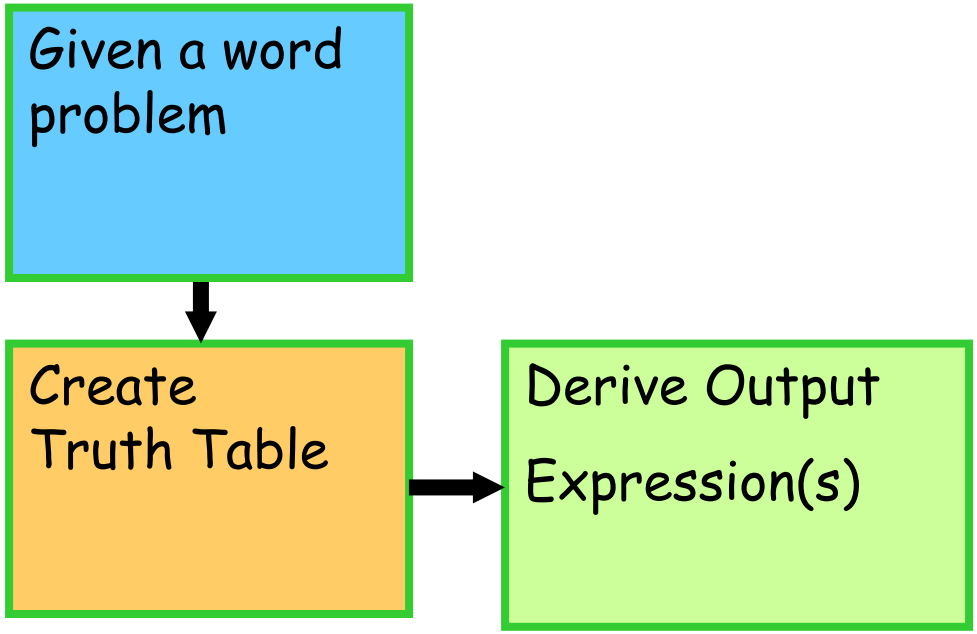
Steps to Design a Logic Circuit

Given a word problem

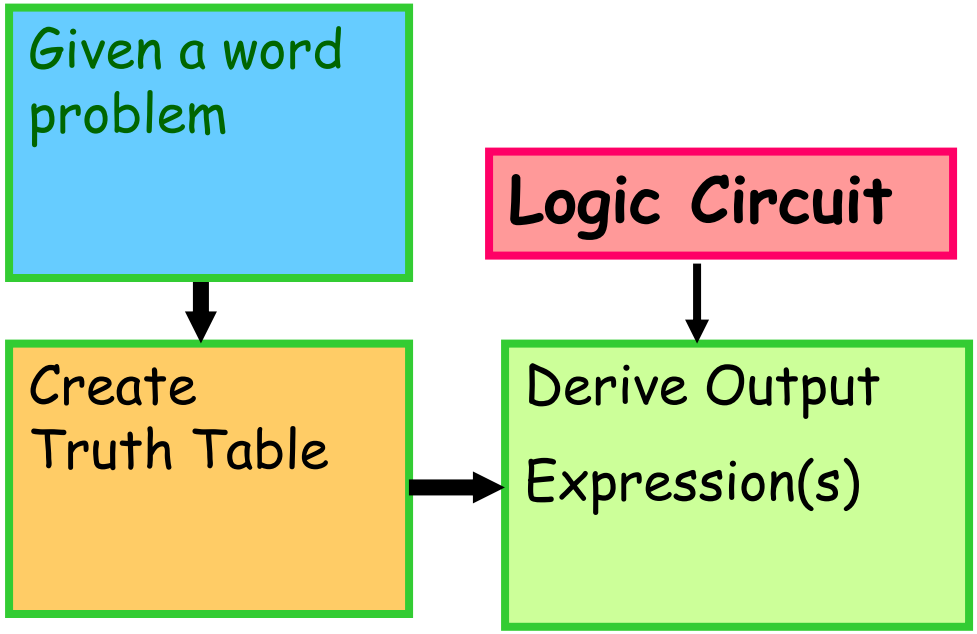
Steps to Design a Logic Circuit



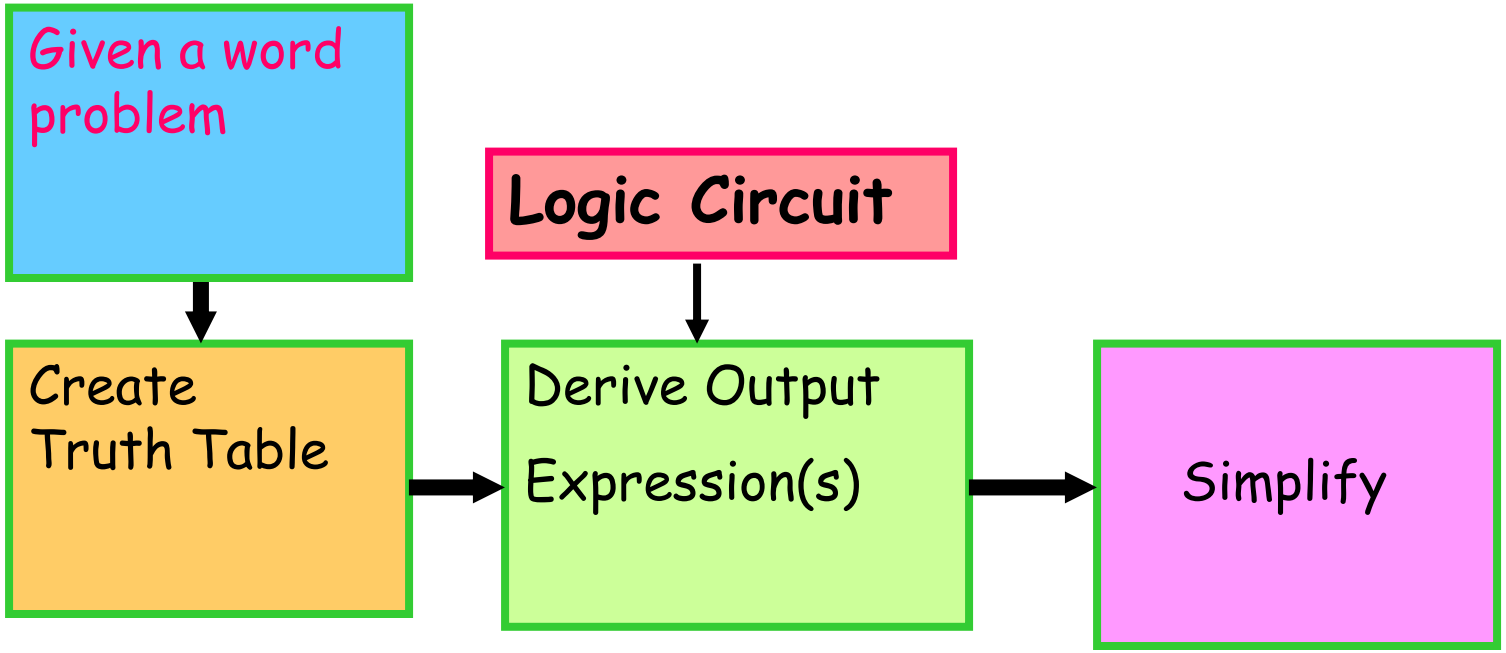
Steps to Design a Logic Circuit



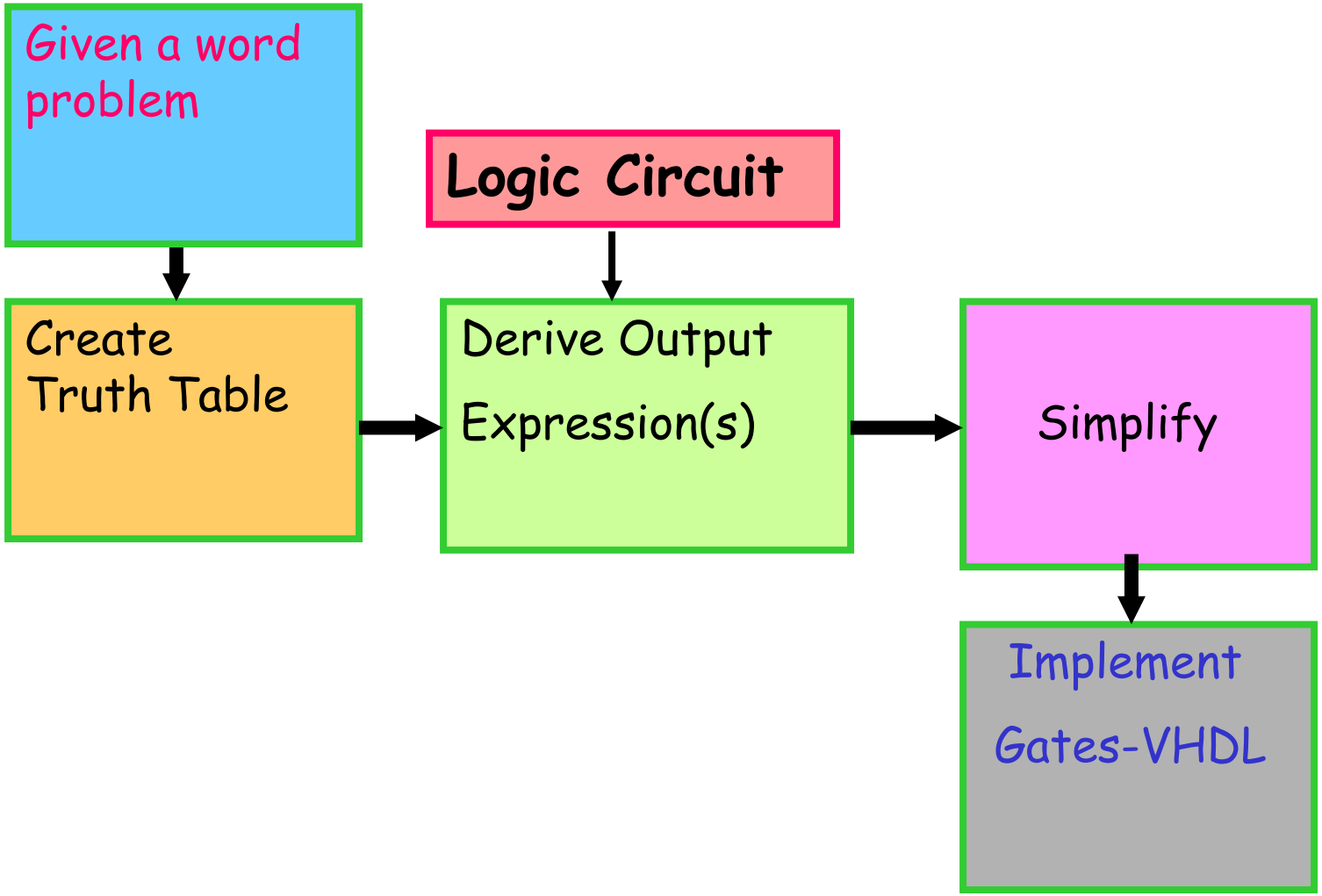
Steps to Design a Logic Circuit



Steps to Design a Logic Circuit



Steps to Design a Logic Circuit



Algorithm: Logic circuit design

1. From the specifications of the problem: Determine the required number of inputs and outputs. Assign a letter symbol to each
2. Derive the truth table
3. Obtain the Boolean expressions for each output as a function of the input variables
4. Simplify all the output equations
5. Draw the logic diagram
6. Verify the correctness of the design

Algorithm: Logic circuit design

Synthesis

1. From the specifications of the problem: Determine the required number of inputs and outputs. Assign a letter symbol to each
2. Derive the truth table
3. Obtain the Boolean expressions for each output as a function of the input variables
4. Simplify all the output equations
5. Draw the logic diagram
6. Verify the correctness of the design

Implementation

Design problem

Design a digital logic circuit with three inputs and one output. The output must be logic one$(1)_2$ when the binary value of the inputs is less than three$(11)_2$ and zero$(0)_2$ otherwise.

1. Inputs/Outputs

- Inputs = 3
- Output = 1

Inputs/Outputs/Truth table

- Inputs = 3
- Output = 1

Design a digital logic circuit with three inputs and one output. The output must be logic one$(1)_2$ when the binary value of the inputs is less than three$(11)_2$ and zero$(0)_2$ otherwise.

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

2. Function - Truth table

Design a digital logic circuit with three inputs and one output. The output must be logic one$(1)_2$ when the binary value of the inputs is less than three$(11)_2$ and zero$(0)_2$ otherwise.

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

Terms of the output expression

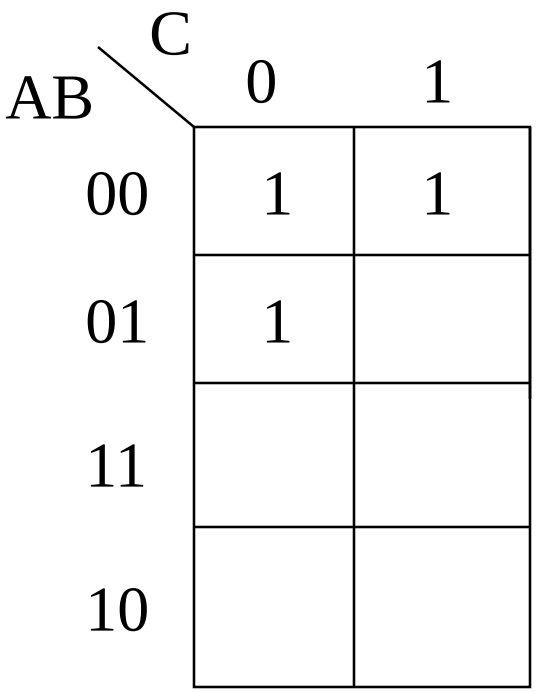
A	B	C	X		
0	0	0	1	⟶	$\bar{A}\bar{B}\bar{C}$
0	0	1	1	⟶	$\bar{A}\bar{B}C$
0	1	0	1	⟶	$\bar{A}B\bar{C}$
0	1	1	0		
1	0	0	0		
1	0	1	0		
1	1	0	0		
1	1	1	0		

3. Output Boolean expression

$$X = \bar{A}\bar{B}\bar{C} + \bar{A}\bar{B}C + \bar{A}B\bar{C}$$

A	B	C	X		
0	0	0	1	⟶	$\bar{A}\bar{B}\bar{C}$
0	0	1	1	⟶	$\bar{A}\bar{B}C$
0	1	0	1	⟶	$\bar{A}B\bar{C}$
0	1	1	0		
1	0	0	0		
1	0	1	0		
1	1	0	0		
1	1	1	0		

4. Simplification K-Map

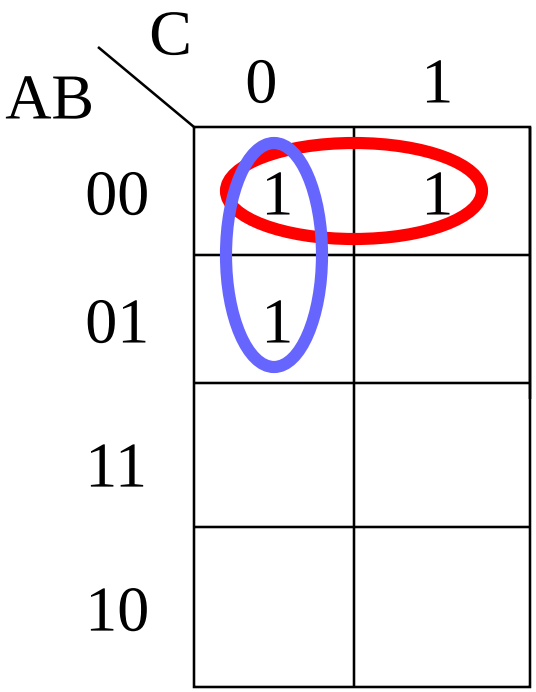


$\rightarrow \overline{A}\overline{B}\overline{C}$

$\rightarrow \overline{A}\overline{B}C$

$\rightarrow \overline{A}B\overline{C}$

4. Simplification K-Map



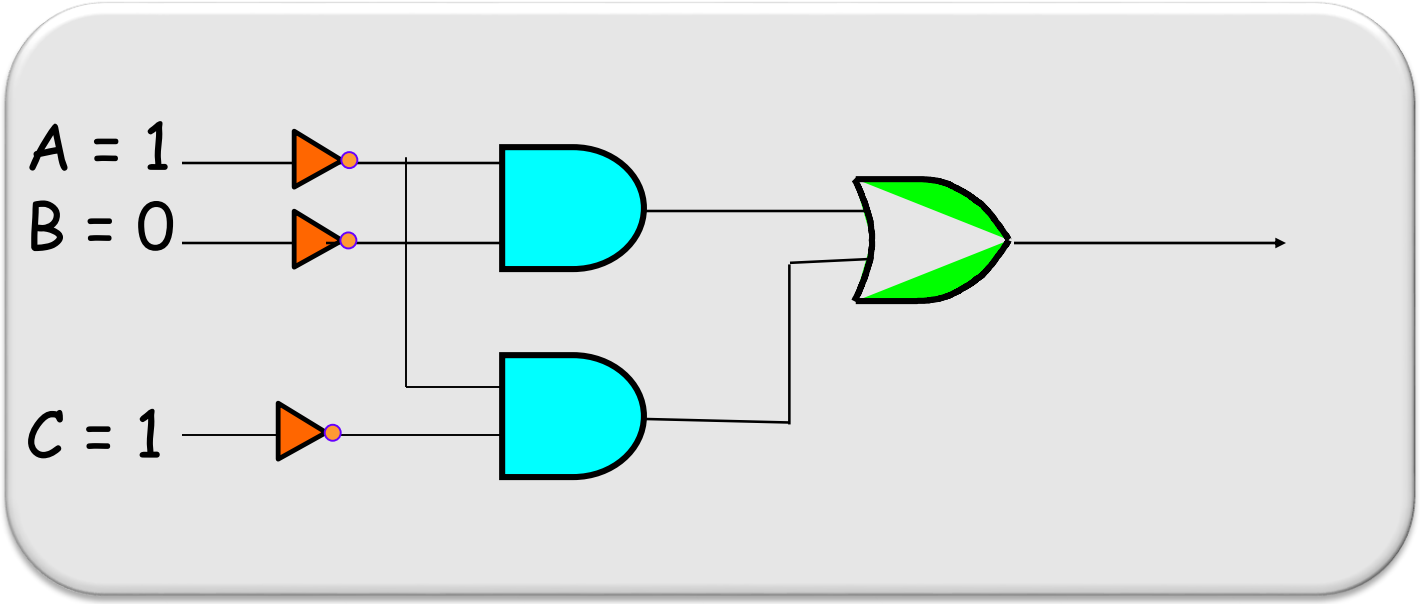
$\overline{A}\,\overline{B}\overline{C}$

$\overline{A}\,\overline{B}C$

$\overline{A}B\overline{C}$

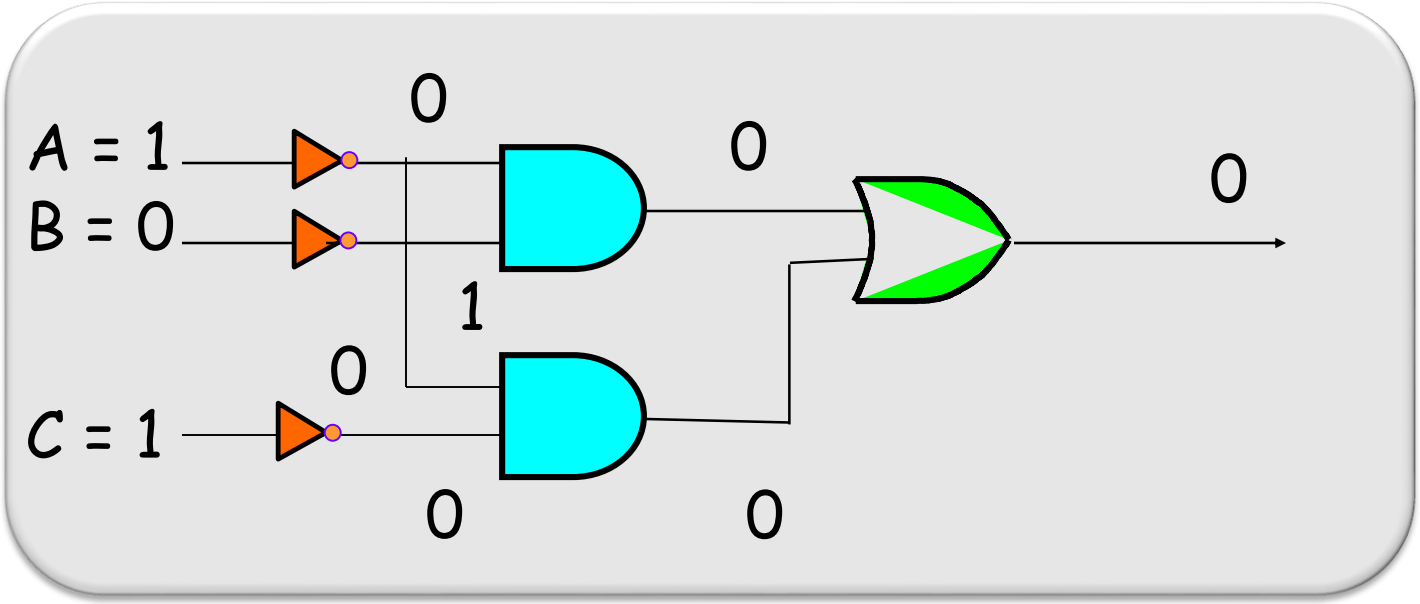
$$X = \overline{A}\,\overline{B} + \overline{A}\,\overline{C}$$

5. Implementation: Simplified logic circuit

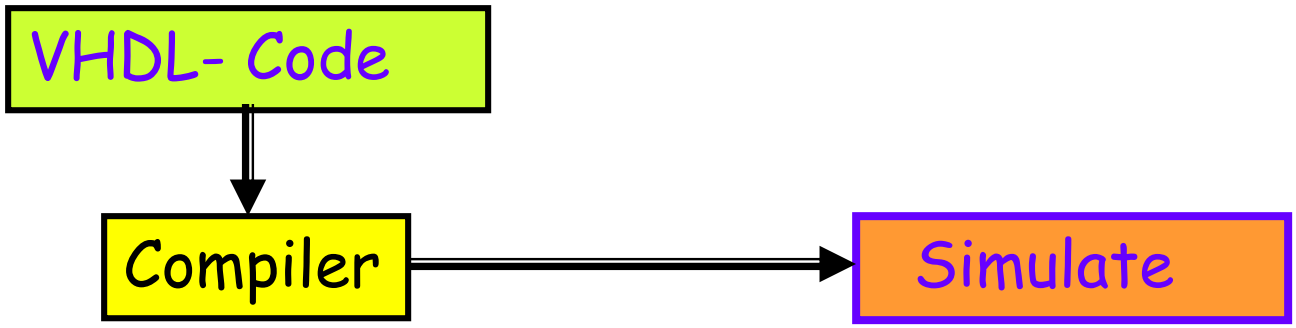


X = A' B' + A' C'

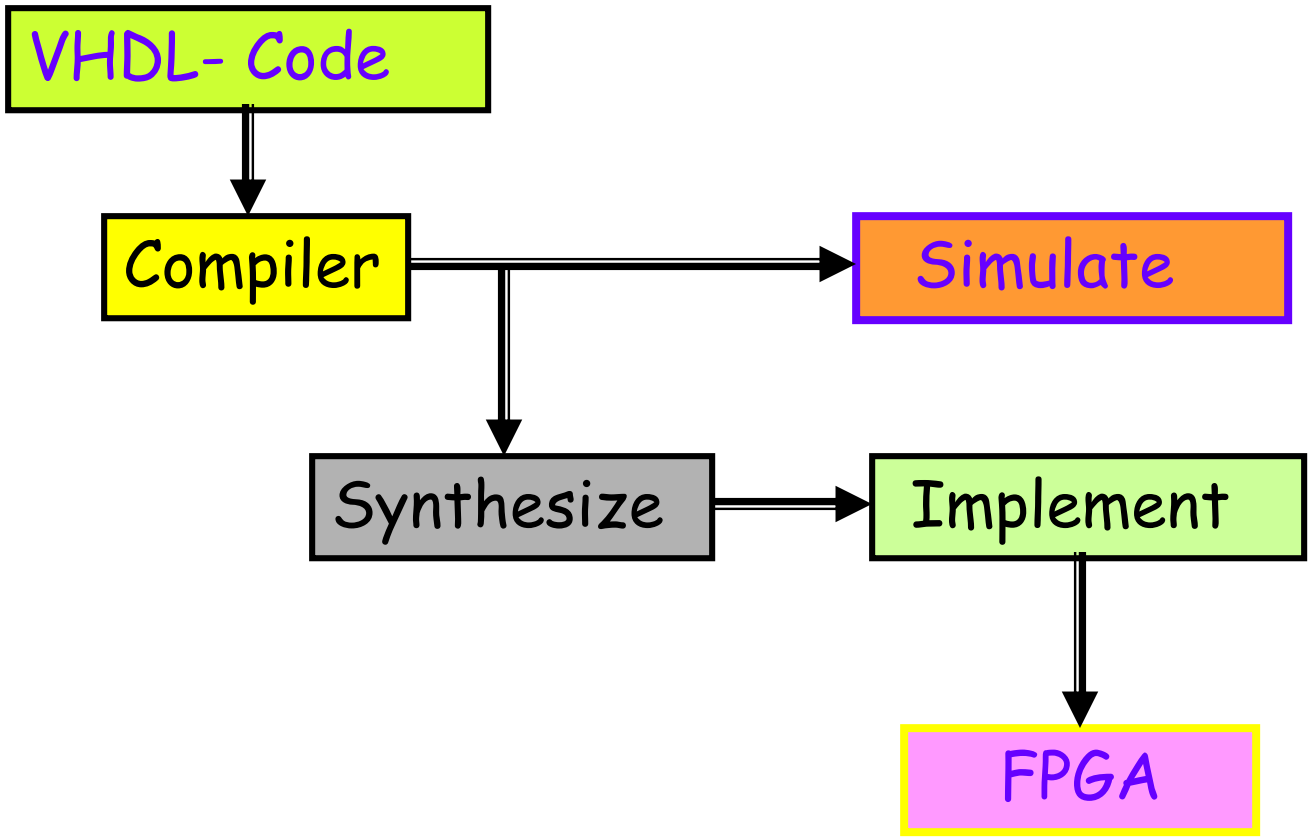
6. Verification



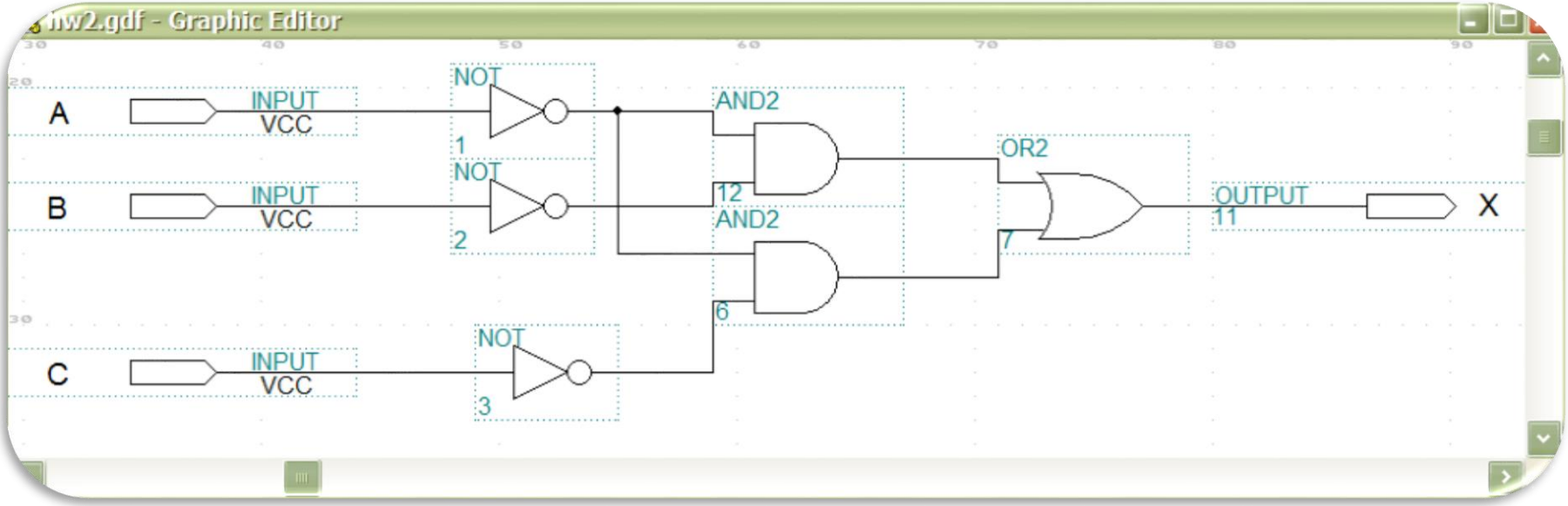
Simulation: VHDL



VHDL to Chips



Logic diagram-VHDL editor

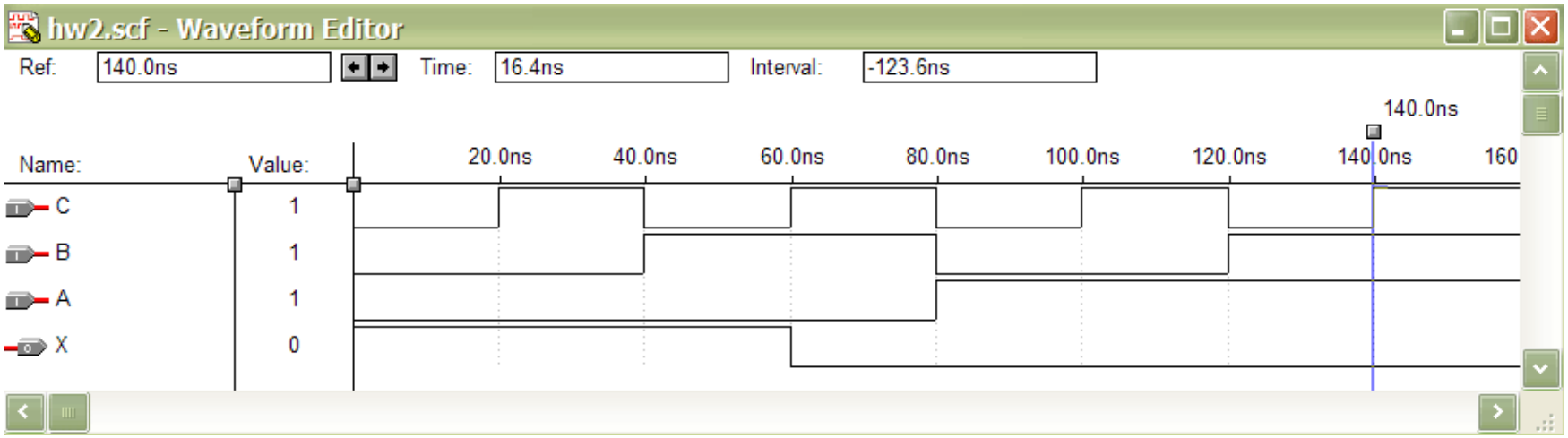


VHDL Code

```
ENTITY hw2 IS
        PORT ( A, B, C          : IN BIT;
                        X       : OUT BIT ) ;
END hw2 ;

ARCHITECTURE LogicFunc OF hw2 IS
BEGIN
X = (NOT A AND NOT B) OR (NOT A AND NOT C) ;
END LogicFunc ;
```

VHDL/Simulation

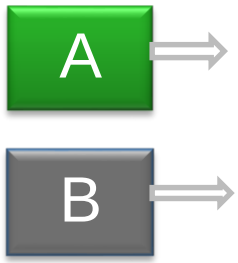


Design: 1-bit comparator

New Example

Design: 1-bit comparator

- A comparator compares the values of two bits and produces the proper result.



A, B = 0 or 1

Design: 1-bit comparator

- A comparator compares the values of two bits and produces the proper result.

- What is the proper result?

1-bit comparator

- There are three cases:

1. The two bits are equal
2. One bit has a greater value that the other
3. One bit has a smaller value than the other

Inputs/Outputs/Function

- Inputs: ?
- Outputs: ?
- Function: Comparator

Inputs/Outputs/Function

- Inputs: 2
- Outputs: 3
- Function: Comparator

Label the Input/Output

Truth table?

		E	G	L
A	B	A = B	A > B	A < B
0	0			
0	1			
1	0			
1	1			

Truth table

		E	G	L
A	B	A = B	A > B	A < B
0	0	1		
0	1	0		
1	0	0		
1	1	1		

Truth table

		E	G	L
A	B	A = B	A > B	A < B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

We have 3 output equations

		E	G	L
A	B	A = B	A > B	A < B
0	0	1	0	0
0	1	0	0	1
1	0	0	1	0
1	1	1	0	0

$E = A'B' + AB$

$G = AB'$

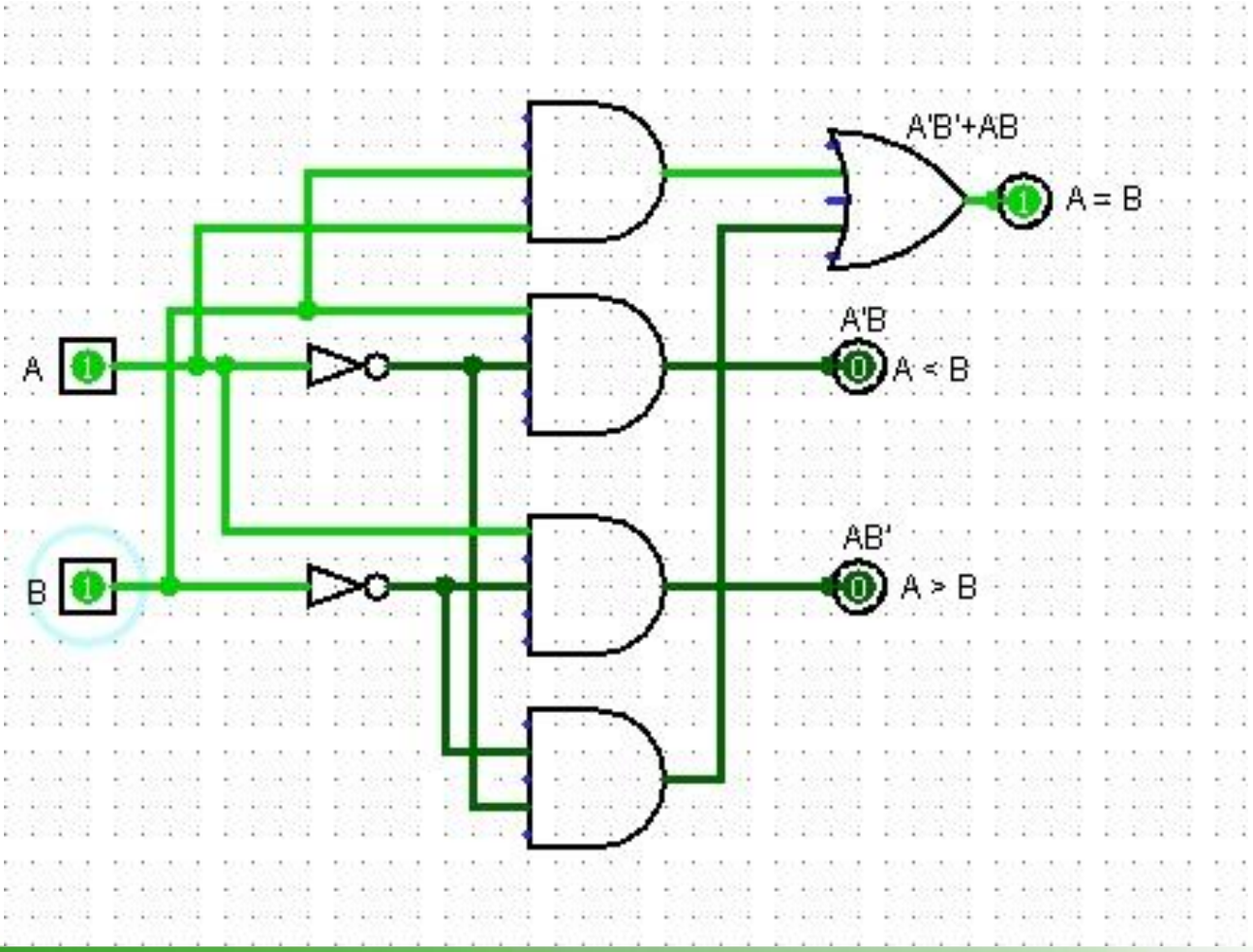
$L = A'B$

1-bit Comparator (Logic Circuit)

E = A'B' + AB

G = AB'

L = A'B



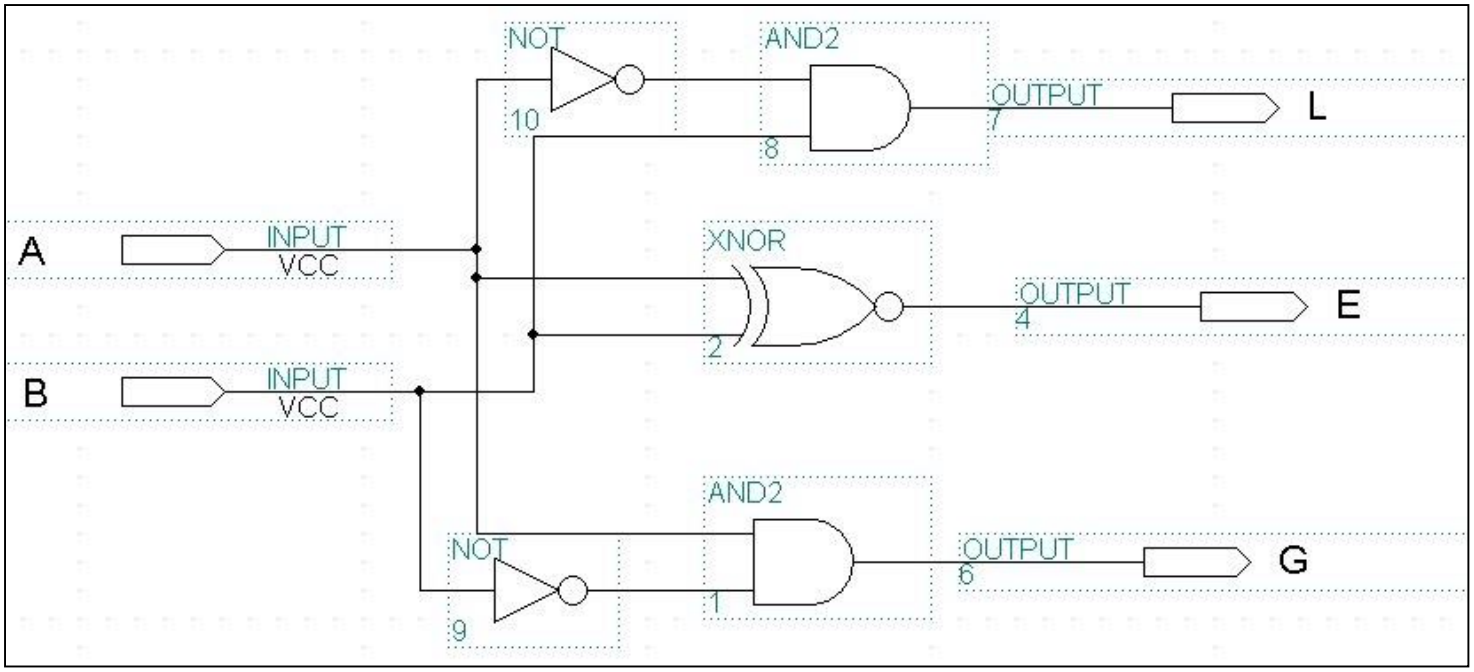
VHDL: Logic circuit

E = A'B' + AB = A XNOR B

G = AB'

L = A'B

XNOR gates will be studied in the next lecture

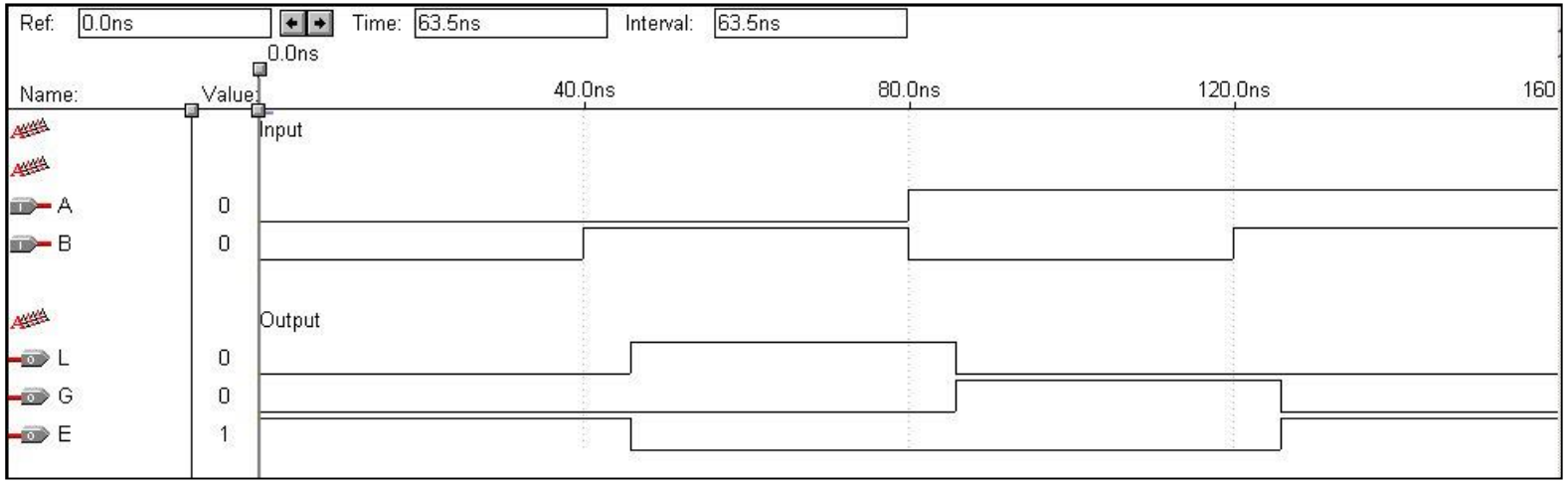


VHDL

```
--Homework asignment 4 part A
--1-bit Comparator

ENTITY hwfourA IS
    PORT (  A, B        :   IN BIT ;
            G, E, L     :   OUT BIT ) ;
END hwfourA ;

ARCHITECTURE LogicFunc OF hwfourA IS
BEGIN
    G <= (A and not B) ;
    E <= (A XNOR B) ;
    L <= (B and not A) ;
    END LogicFunc ;
```



Another design example

- Design a BCD-to-7 Segment Display converter

New Example

Input, outputs(s), function ?

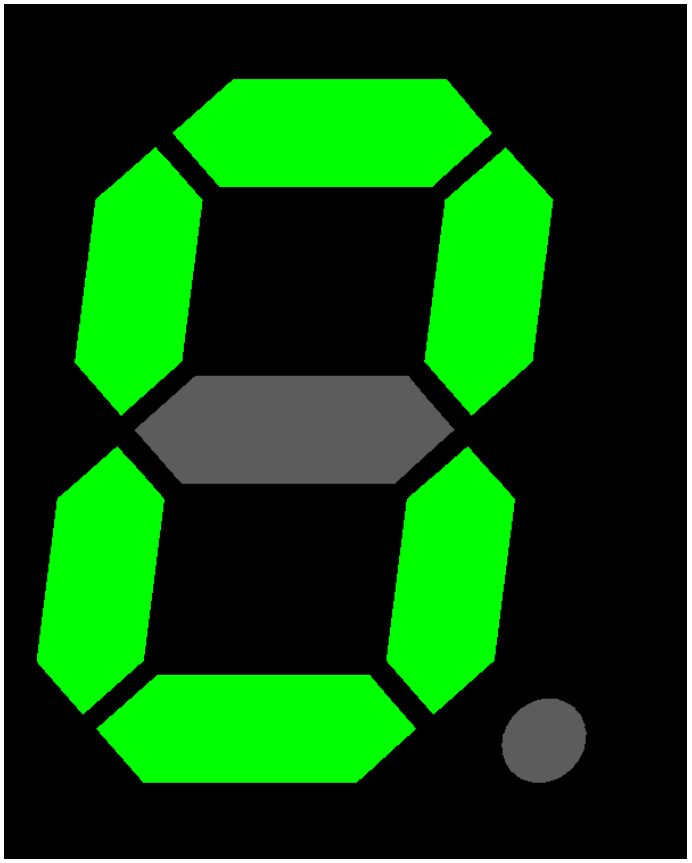
- BCD
- 7SD (7-Segment-Display)

BCD

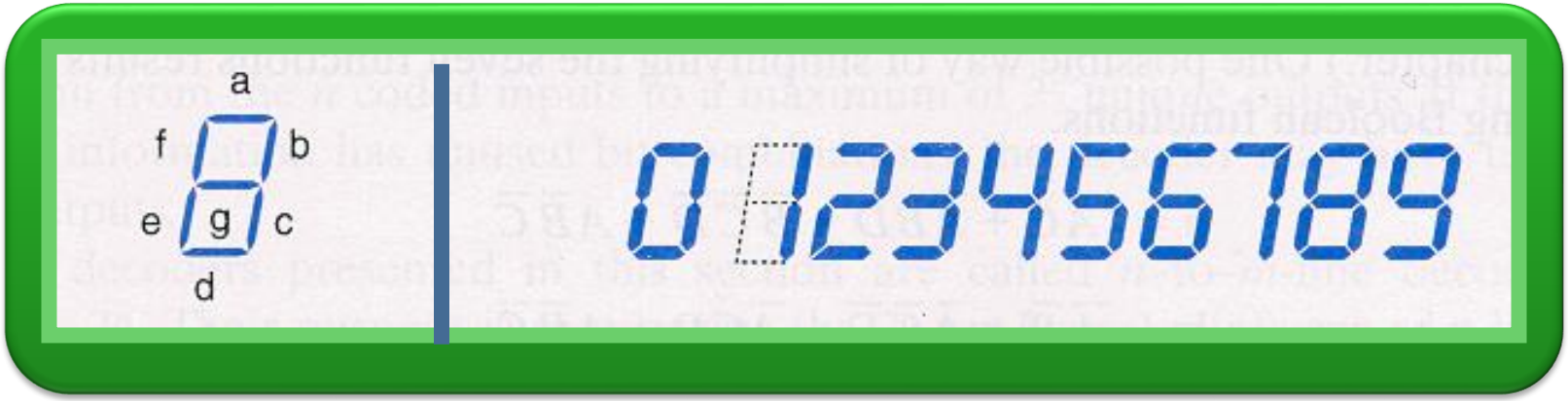
- BCD = Binary Coded Decimal

	A	B	C	D
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1

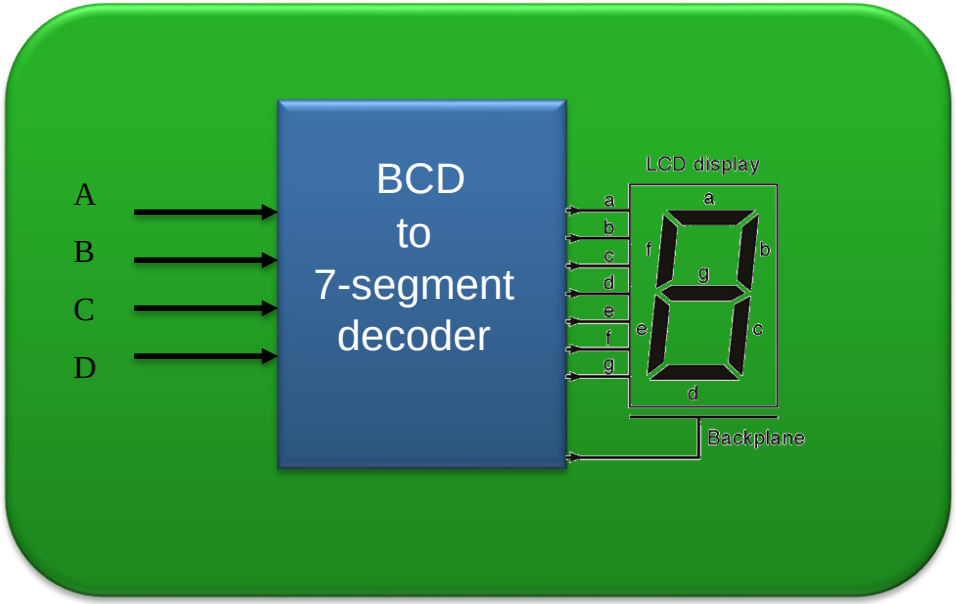
7SD

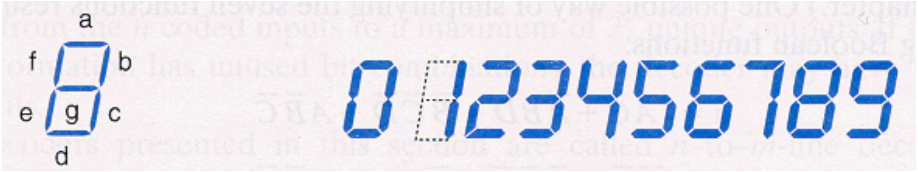


7SD



BCD to 7SD converter





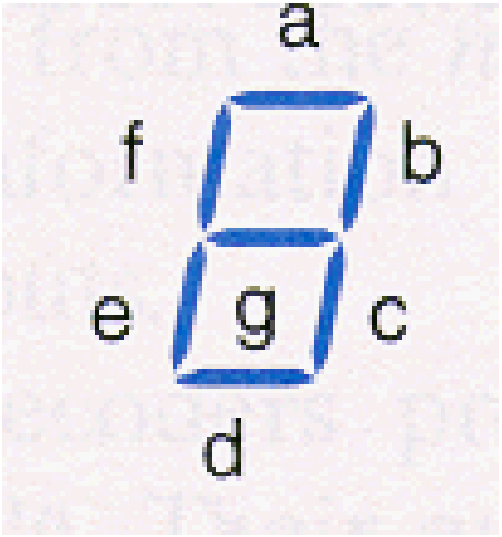
Truth Table for BCD–to–Seven-Segment Decoder

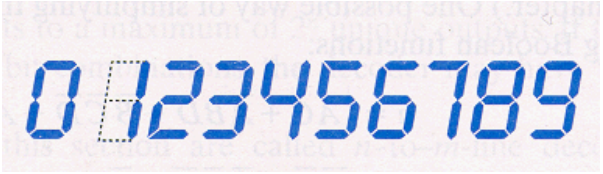
BCD Input				Seven-Segment Decoder						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
All other inputs				0	0	0	0	0	0	0

Simplify

Truth Table for BCD–to–Seven-Segment Decoder

BCD Input				Seven-Segment Decoder						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
All other inputs				0	0	0	0	0	0	0





K-maps

Truth Table for BCD–to–Seven-Segment Decoder

BCD Input				Seven-Segment Decoder						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
All other inputs				0	0	0	0	0	0	0

a)

AB \ CD	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	0	0	0	0
10	1	1	0	0

$a = \bar{A}C + \bar{A}B\bar{D} + A\bar{B}\bar{C} + \bar{B}\bar{C}\bar{D}$

b)

AB \ CD	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	0	0	0	0
10	1	0	0	1

$b = \bar{A}\bar{B} + \bar{A}\bar{C}\bar{D} + \bar{A}CD + B\bar{C}D + \bar{B}\bar{C}\bar{D}$

c)

AB \ CD	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	0	0	0	0
10	1	1	0	0

$c = \bar{A}\bar{C} + \bar{A}\bar{B} + \bar{A}CD + A\bar{B}\bar{C}$

d)

AB \ CD	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	0	0	0	0
10	1	1	0	0

$d = \bar{B}\bar{C}\bar{D} + A\bar{B}\bar{C} + \bar{A}B$

e)

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	0	0	0	0
10	1	0	0	0

$e = \bar{B}\bar{C}\bar{D} + \bar{A}C\bar{D}$

f)

AB \ BC	00	01	11	10
00	1	0	0	0
01	1	1	0	0
11	0	0	0	0
10	1	1	0	0

$f = \bar{A}C\bar{D} + \bar{A}B\bar{D} + \bar{A}B\bar{C} + AB\bar{C}$

g)

AB \ CD	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	0	0	0	0
10	1	1	0	0

$g = \bar{A}B\bar{C} + \bar{A}C\bar{D} + \bar{A}B\bar{C} + AB\bar{C}$

Simplified equations

Truth Table for BCD–to–Seven-Segment Decoder

BCD Input				Seven-Segment Decoder						
A	B	C	D	a	b	c	d	e	f	g
0	0	0	0	1	1	1	1	1	1	0
0	0	0	1	0	1	1	0	0	0	0
0	0	1	0	1	1	0	1	1	0	1
0	0	1	1	1	1	1	1	0	0	1
0	1	0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0	1	1
0	1	1	0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	0	0	0	0
1	0	0	0	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	0	1	1
All other inputs				0	0	0	0	0	0	0

$$a = \overline{A}C + \overline{A}BD + \overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}$$

$$b = \overline{A}\,\overline{B} + \overline{A}\,\overline{C}\,\overline{D} + \overline{A}CD + A\overline{B}\,\overline{C}$$

$$c = \overline{A}B + \overline{A}D + \overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C}$$

$$d = \overline{A}C\overline{D} + \overline{A}\,\overline{B}C + \overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C} + \overline{A}B\overline{C}D$$

$$e = \overline{A}C\overline{D} + \overline{B}\,\overline{C}\,\overline{D}$$

$$f = \overline{A}B\overline{C} + \overline{A}\,\overline{C}\,\overline{D} + \overline{A}B\overline{D} + A\overline{B}\,\overline{C}$$

$$g = \overline{A}C\overline{D} + \overline{A}\,\overline{B}C + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}$$

a)

AB \ CD	00	01	11	10
00	1	0	1	1
01	0	1	1	1
11	0	0	0	0
10	1	1	0	0

$a = \overline{A}C + \overline{A}B\overline{D} + A\overline{B}\,\overline{C} + \overline{B}\,\overline{C}\,\overline{D}$

b)

AB \ CD	00	01	11	10
00	1	1	1	1
01	1	0	1	0
11	0	0	0	0
10	1	0	0	1

$b = \overline{A}\,\overline{B} + \overline{A}\,\overline{C}\,\overline{D} + \overline{A}CD + \overline{B}\,\overline{C}\,\overline{D} + \overline{B}\,\overline{C}\,\overline{D}$

c)

AB \ CD	00	01	11	10
00	1	1	1	0
01	1	1	1	1
11	0	0	0	0
10	1	1	0	0

$c = \overline{A}\,\overline{C} + \overline{A}B + \overline{A}CD + A\overline{B}\,\overline{C}$

d)

AB \ CD	00	01	11	10
00	1	0	1	1
01	0	1	0	1
11	0	0	0	0
10	1	1	0	0

$d = \overline{B}\,\overline{C}\,\overline{D} + A\overline{B}\,\overline{C} + \overline{A}B$

e)

AB \ CD	00	01	11	10
00	1	0	0	1
01	0	0	0	1
11	0	0	0	0
10	1	0	0	0

$e = \overline{B}\,\overline{C}\,\overline{D} + \overline{A}C\overline{D}$

f)

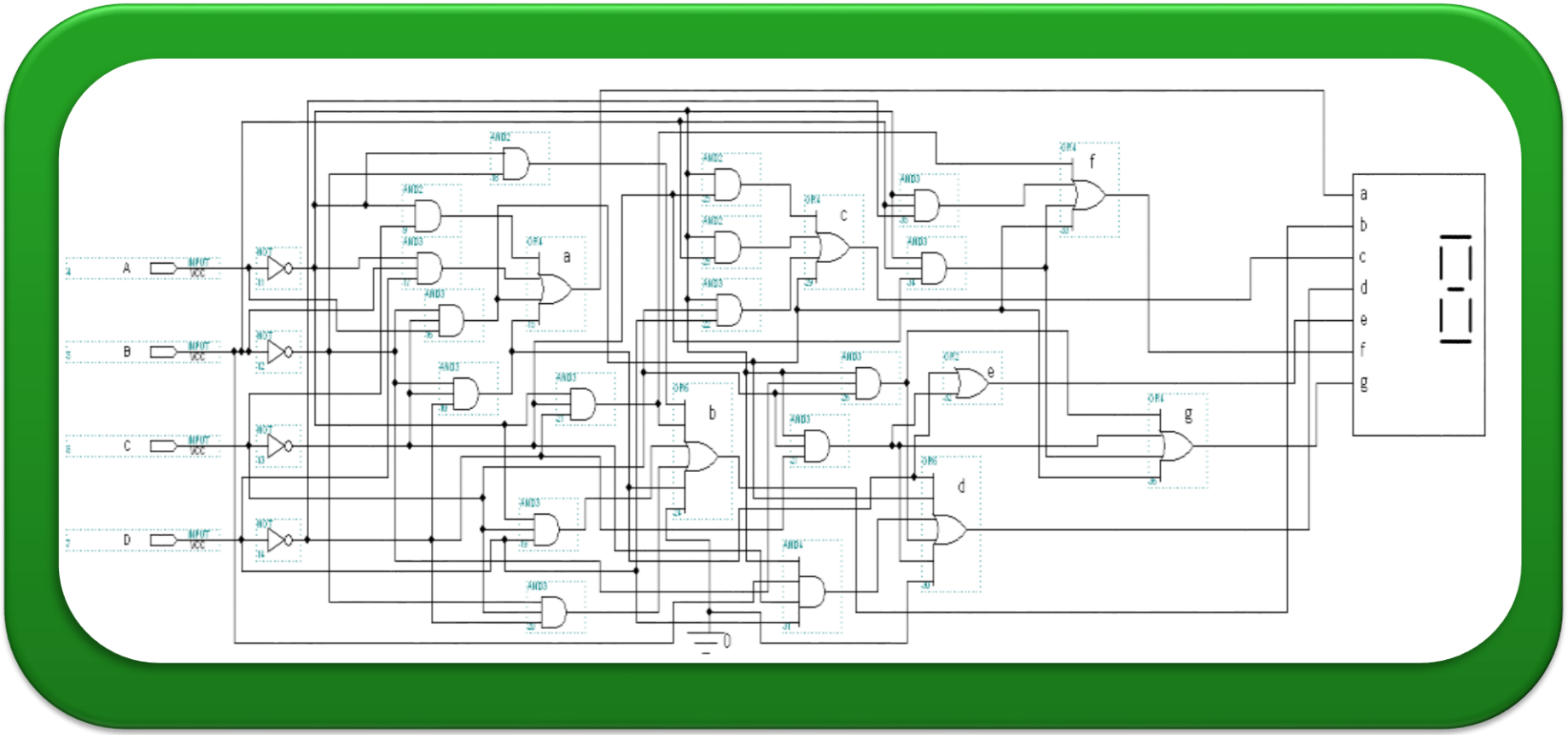
AB \ BC	00	01	11	10
00	1	0	0	0
01	1	1	0	0
11	0	0	0	0
10	1	1	0	0

$f = \overline{A}C\overline{D} + \overline{A}B\overline{D} + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}$

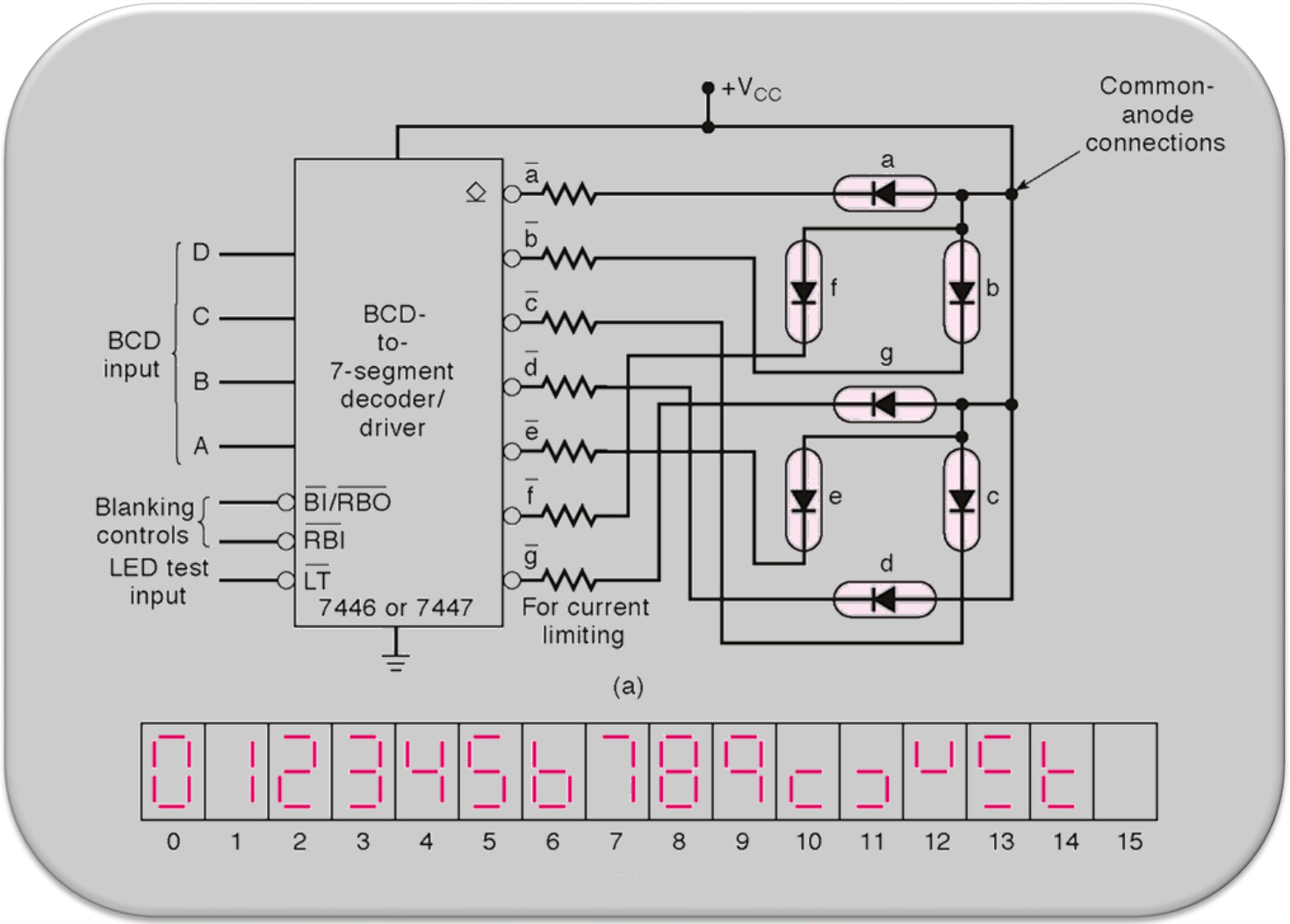
AB \ CD	00	01	11	10
00	0	0	1	1
01	1	1	0	1
11	0	0	0	0
10	1	1	0	0

$g = \overline{A}\,\overline{B}C + \overline{A}C\overline{D} + \overline{A}B\overline{C} + A\overline{B}\,\overline{C}$

Implement VHDL

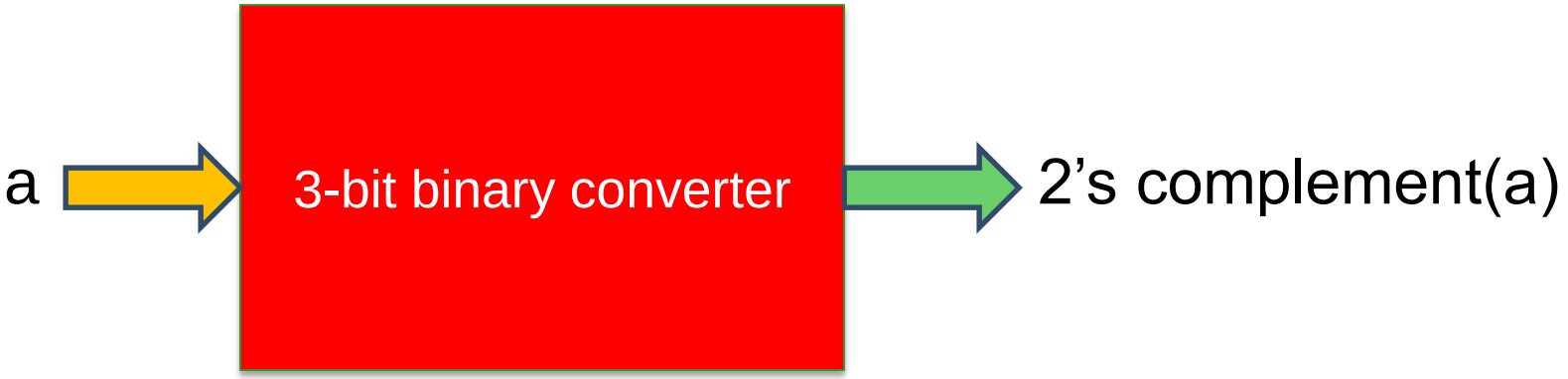


Implement … using a chip (7446)



Design a binary converter

- Design a binary converter, to convert a 3-bit binary number to it's 2's complement.



New Example

Design a binary converter

- Design a binary converter, to convert a 3-bit binary number to it's 2's complement.
  - How many inputs and outputs?
  - Inputs = 3
  - Outputs = ?
  - Need to set up the truth table

Design a binary converter

- Design a binary converter, to convert a 3-bit binary number to it's 2's complement.

Binary	1's complement	2's complement
abc	xyz	xyzw
000	111	1000
001	110	0111
010	101	0110
011	100	0101
100	011	0100
101	010	0011
110	001	0010
111	000	0001

Design a binary converter

- Design a binary converter, to convert a 3-bit binary number to it's 2's complement.

Binary	1's complement	2's complement
abc	xyz	xyzw
000	111	1000
001	110	0111
010	101	0110
011	100	0101
100	011	0100
101	010	0011
110	001	0010
111	000	0001

- Write the output equations
- Simplify
- Implement

Homework

- Design two binary converters, to convert a 3-bit binary number to:
  - 2's Complement
  - 1's Complement

Use LogiSim