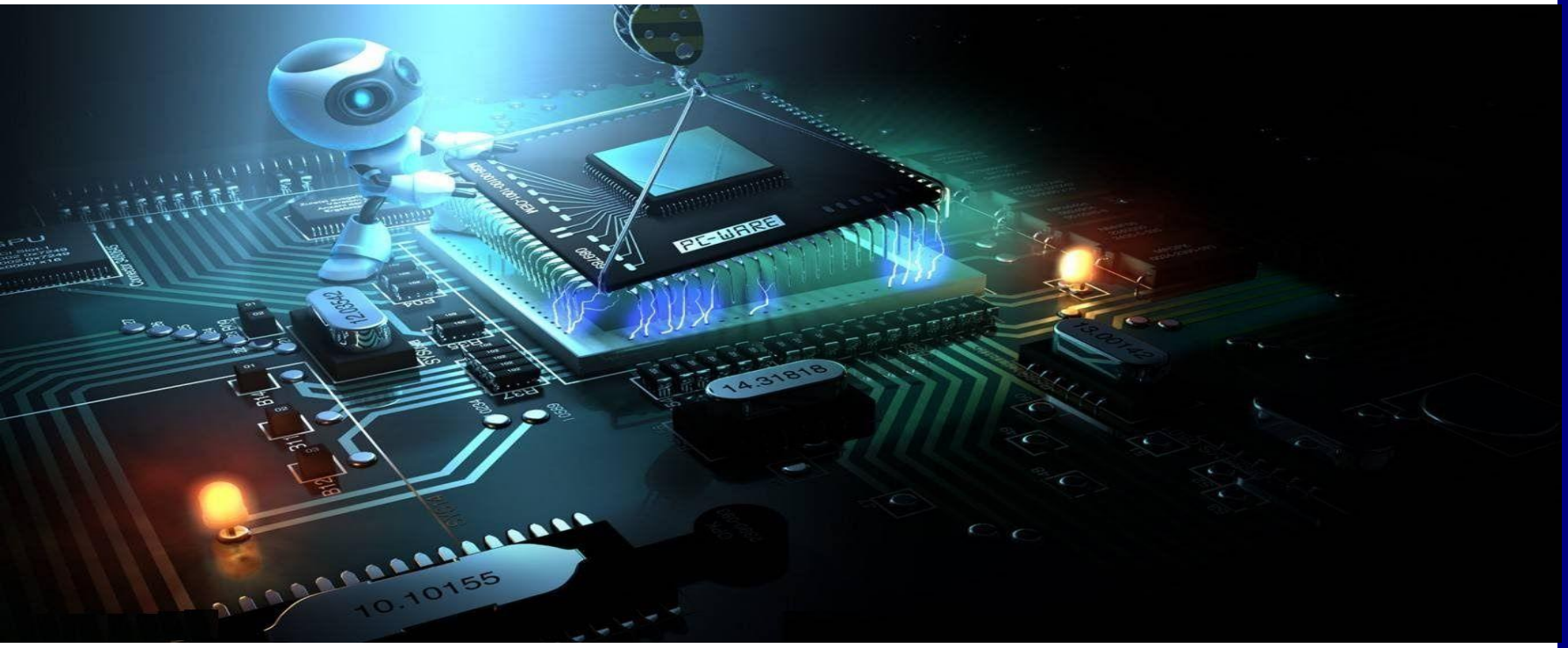
جامعة القاهرة الجديدة التكنولوجية

Fall 2023

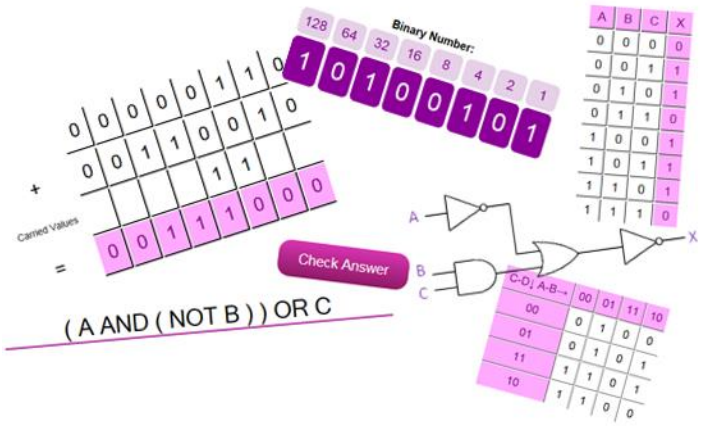
(7)



Digital Engineering

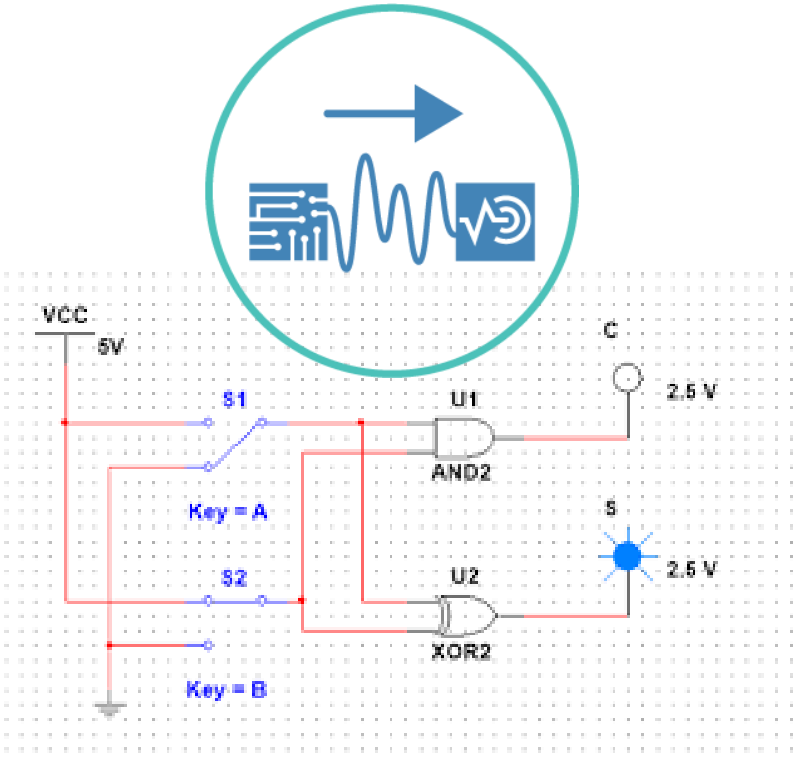
Dr. Hatem Yousry

Agenda



- **Four-variable K-Map**
- **Multisim.**

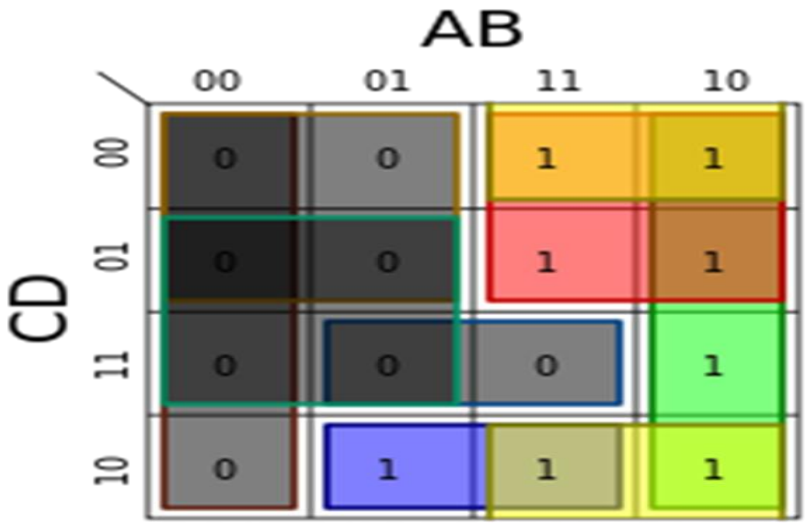
	C’	C’	C	C	
A’					B’
A’					B
A					B
A					B’
	D’	D	D	D’	



Multisim

Karnaugh Maps

- Karnaugh maps (K-maps) are ***graphical*** representations of Boolean functions.
- One ***map cell*** corresponds to a row in the truth table.
- Also, one map cell corresponds to a Minterm or a Maxterm in the Boolean expression
- **Multiple-cell areas of the map correspond to standard terms.**



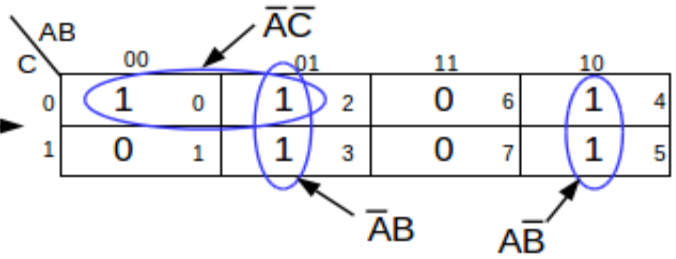
Example

Resolve the logic function

3 Variable Truth Table

	A	B	C	F
0	0	0	0	1
1	0	0	1	0
2	0	1	0	1
3	0	1	1	1
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	0

K-Map



Note : $\bar{A}B + A\bar{B} = A \oplus B$ (Exclusive OR)

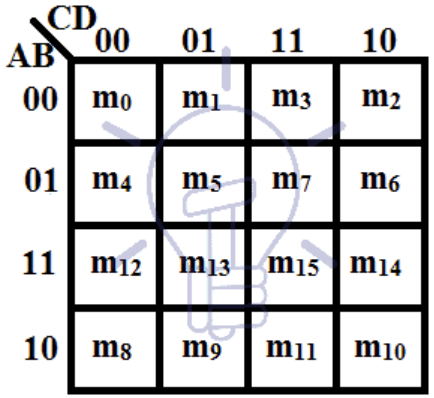
$F = \bar{A}\bar{C} + \bar{A}B + A\bar{B}$

$F = \bar{A}\bar{C} + A \oplus B$

Minterms and Maxterms for Three Binary Variables

			Minterms		Maxterms	
x	y	z	Term	Designation	Term	Designation
0	0	0	$x'y'z'$	m_0	$x + y + z$	M_0
0	0	1	$x'y'z$	m_1	$x + y + z'$	M_1
0	1	0	$x'yz'$	m_2	$x + y' + z$	M_2
0	1	1	$x'yz$	m_3	$x + y' + z'$	M_3
1	0	0	$xy'z'$	m_4	$x' + y + z$	M_4
1	0	1	$xy'z$	m_5	$x' + y + z'$	M_5
1	1	0	xyz'	m_6	$x' + y' + z$	M_6
1	1	1	xyz	m_7	$x' + y' + z'$	M_7

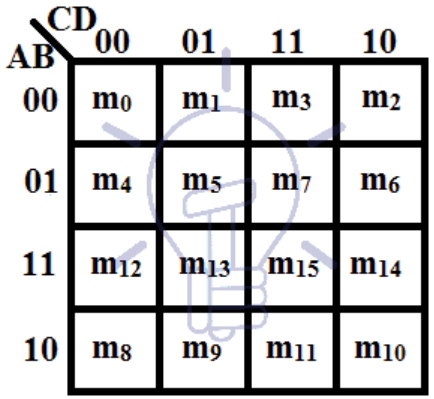
Four-Variable Map



- The map
  - **16 minterms**
  - Combinations of 2, 4, 8, and 16 adjacent squares

	C’	C’	C	C	
A’	m0	m1	m3	m2	B’
A’	m4	m5	m7	m6	B
A	m12	m13	m15	m14	B
A	m8	m9	m11	m10	B’
	D’	D	D	D’	

Four-Variable Map



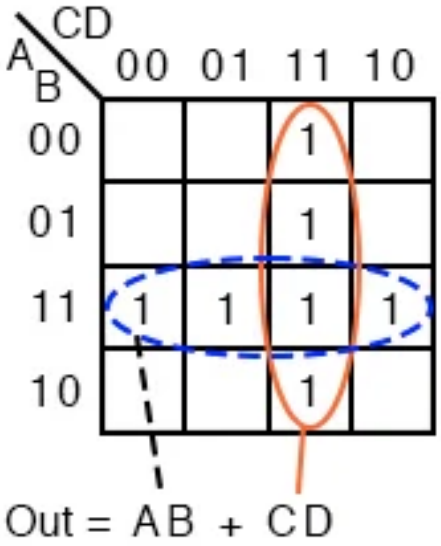
- 4 variables have $2^n=2^4=16$ minterms. So a 4-variable k-map will have 16 cells as shown in the figure given below.

	C'	C'	C	C	
A'	m0	m1	m3	m2	B'
A'	m4	m5	m7	m6	B
A	m12	m13	m15	m14	B
A	m8	m9	m11	m10	B'
	D'	D	D	D'	

Example

- Simplify F(A,B,C,D) =

$$Out = \bar{A}\bar{B}CD + \bar{A}B\bar{C}D + ABCD + A\bar{B}CD + AB\bar{C}\bar{D} + AB\bar{C}D + ABC\bar{D}$$

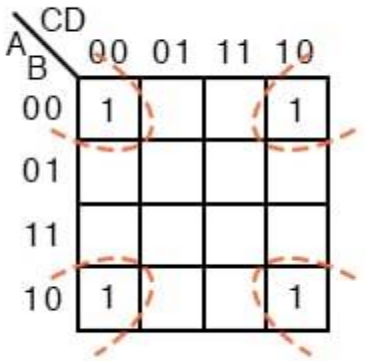


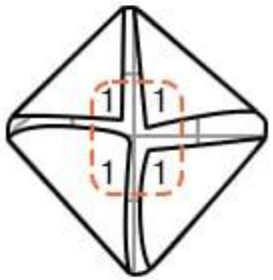
	C'	C'	C	C	
A'	m0	m1	m3	m2	B'
A'	m4	m5	m7	m6	B
A	m12	m13	m15	m14	B
A	m8	m9	m11	m10	B'
	D'	D	D	D'	

Example

- Simplify F(A,B,C,D) =

Out = $\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}\overline{B}C\overline{D} + A\overline{B}\overline{C}\overline{D} + A\overline{B}C\overline{D}$





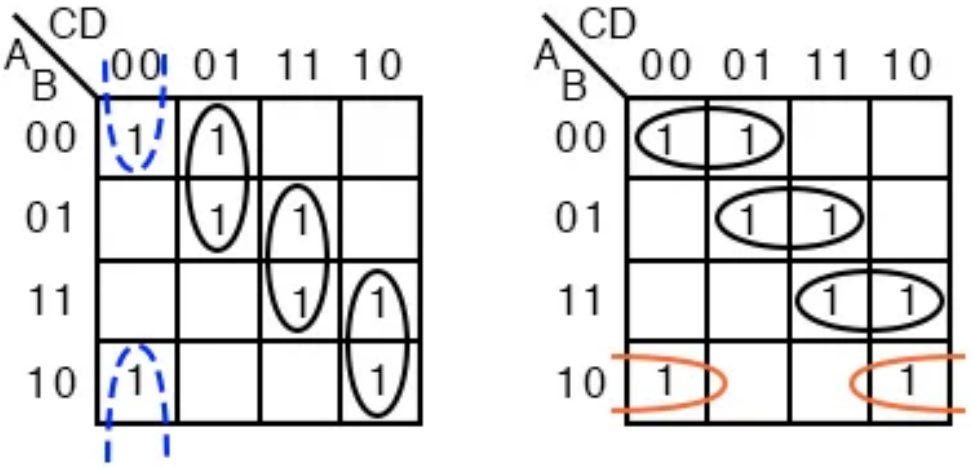
Out = $\overline{B}\overline{D}$

	C'	C'	C	C	
A'	m0	m1	m3	m2	B'
A'	m4	m5	m7	m6	B
A	m12	m13	m15	m14	B
A	m8	m9	m11	m10	B'
	D'	D	D	D'	

Example

- Simplify F(A,B,C,D) =

$$\text{Out} = \bar{A}\bar{B}\bar{C}\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}B\bar{C}D + \bar{A}BCD + ABCD + ABC\bar{D} + A\bar{B}\bar{C}\bar{D} + A\bar{B}C\bar{D}$$



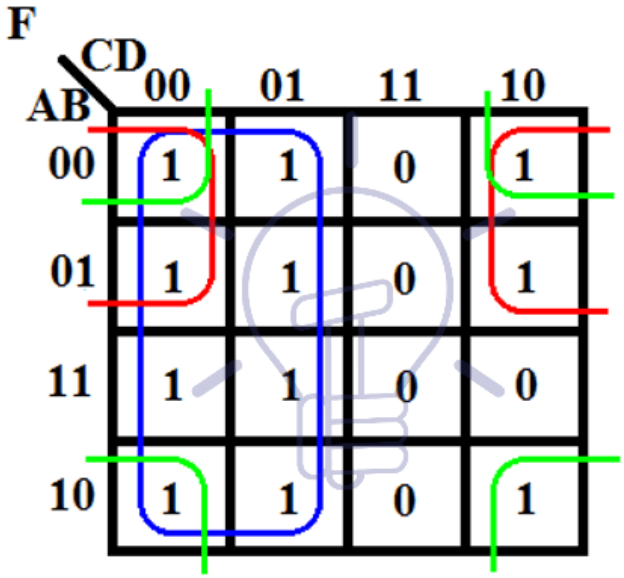
$$\text{Out} = \bar{B}\bar{C}\bar{D} + \bar{A}\bar{C}D + BCD + AC\bar{D}$$

$$\text{Out} = \bar{A}\bar{B}\bar{C} + \bar{A}BD + ABC + A\bar{B}\bar{D}$$

Example

- Simplify F(A,B,C,D) = ∑(m0, m1, m2, m4, m5, m6, m8, m9, m12, m13, m14)

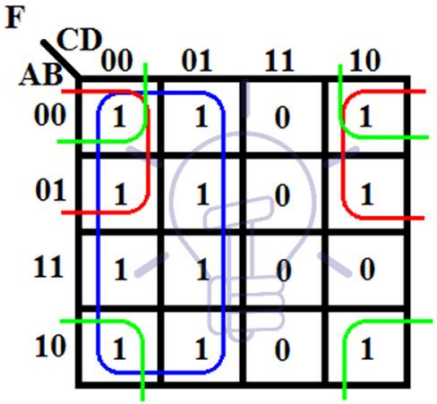
	C’	C’	C	C	
A’	m0	m1	m3	m2	B’
A’	m4	m5	m7	m6	B
A	m12	m13	m15	m14	B
A	m8	m9	m11	m10	B’
	D’	D	D	D’	



$F = \overline{C} + \overline{B}\overline{D} + \overline{A}\overline{D}$

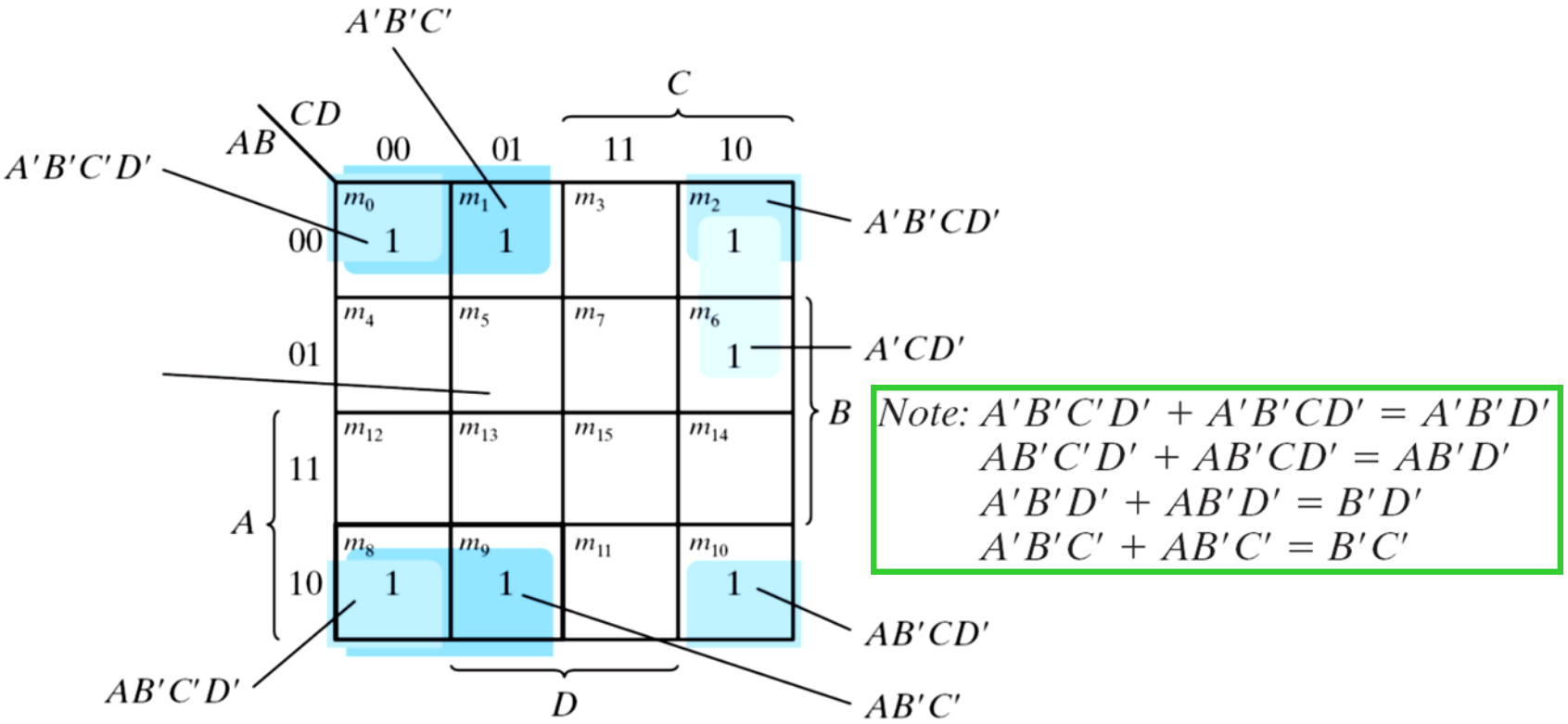
Example

- First of all, try to make the **biggest possible groups** as shown in this example. Corner 1s can also be made into a group of 4. The remaining last 1 should be combined with the pre-made group to make a big overlapping group.
- Group of 8 will give a term of 1 literal that remains unchanged i.e. $\overline{C}$
- Corner group of 4 will give term with 2 literals that remain unchanged i.e. $\overline{B}\overline{D}$
- The last group of 4 will give $\overline{A}\overline{D}$ because they remained unchanged in the group.
- So the expression will be
- $F = \overline{C} + \overline{B}\overline{D} + \overline{A}\overline{D}$



Example

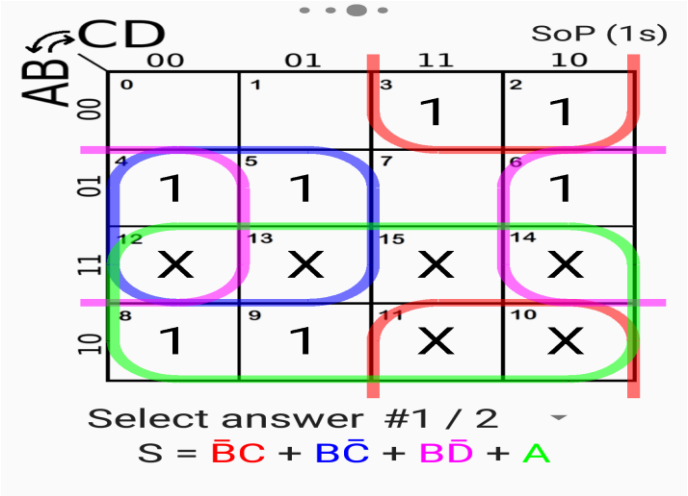
- Simplify $F = A'B'C' + B'CD' + A'B'C'D' + AB'C'$



$$B'D' + B'C' + A'CD'$$

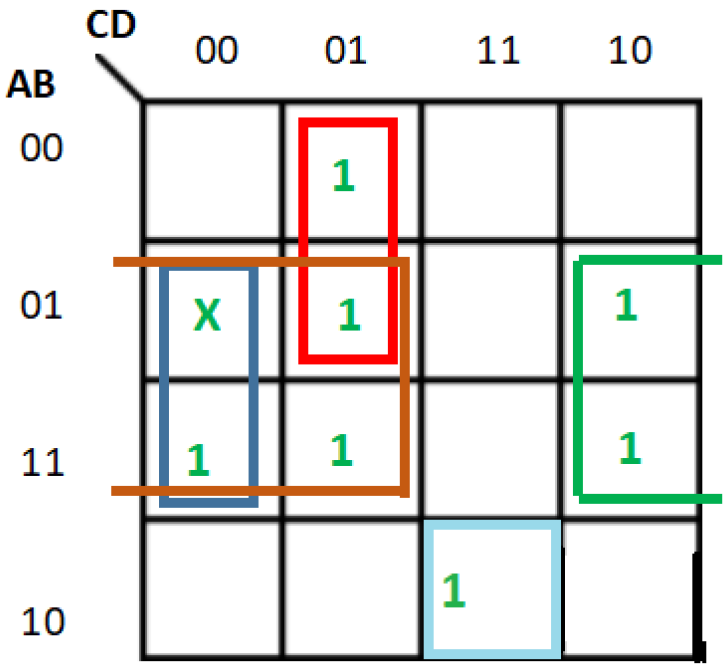
Don't-Care Conditions

- One of the very significant and useful concepts in simplifying the output expression using K-Map is the concept of “Don’t Care”.
- The “Don’t Care” conditions allow us to **replace the empty cell of a K-Map to form a grouping of the variables which is larger than that of forming groups without don’t care.**
- While forming groups of cells, we can consider a “Don’t Care” cell as 1 or 0 or we can also ignore that cell. Therefore, the “Don’t Care” condition can **help us to form a larger group of cells.**



Example

- Simplify F(A,B,C,D) = ∑ (1, 5, 6, 11, 12, 13, 14) + d(4)

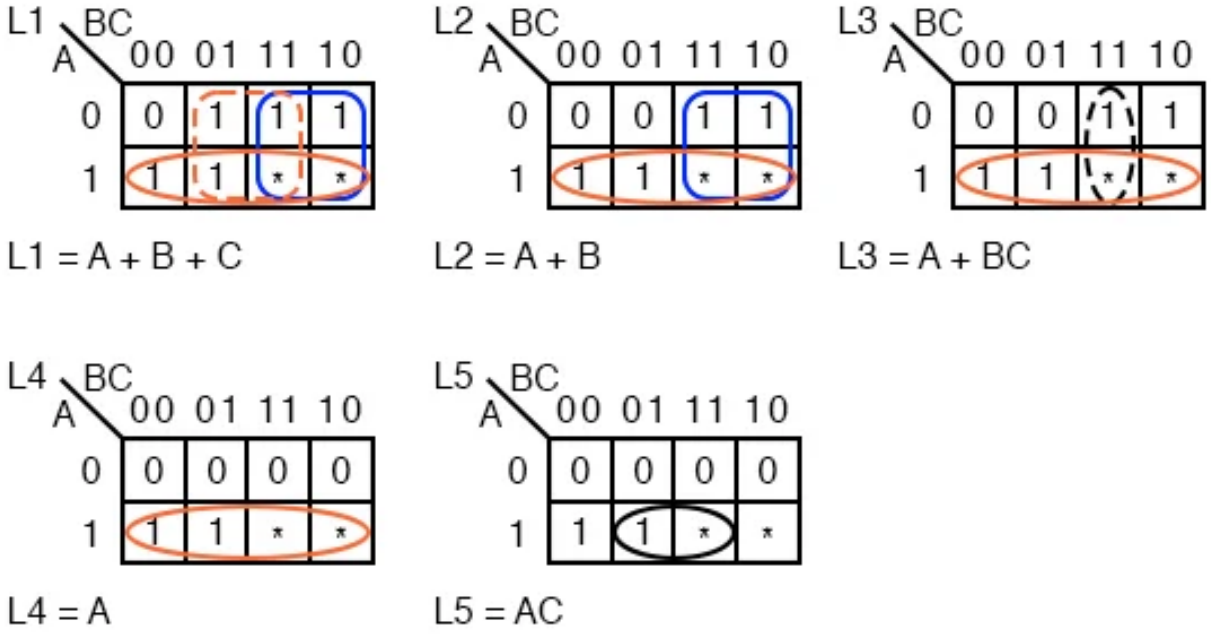


	C’	C’	C	C	
A’	m0	m1	m3	m2	B’
A’	m4	m5	m7	m6	B
A	m12	m13	m15	m14	B
A	m8	m9	m11	m10	B’
	D’	D	D	D’	

f = BC' + BCD' + A'C'D + AB'CD

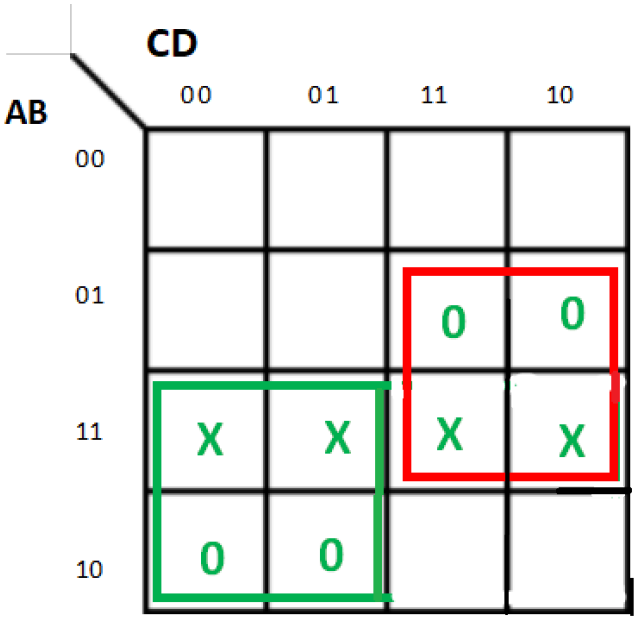
Don't-Care Conditions

- **Can be implemented as 0 or 1**



Example

- Simplify F(A,B,C,D) = ∏ (0, 1, 2, 3, 4, 5) + d(10, 11, 12, 13, 14, 15)



	C’	C’	C	C	
A’	m0	m1	m3	m2	B’
A’	m4	m5	m7	m6	B
A	m12	m13	m15	m14	B
A	m8	m9	m11	m10	B’
	D’	D	D	D’	

F = (A'+ C)(B' + C')

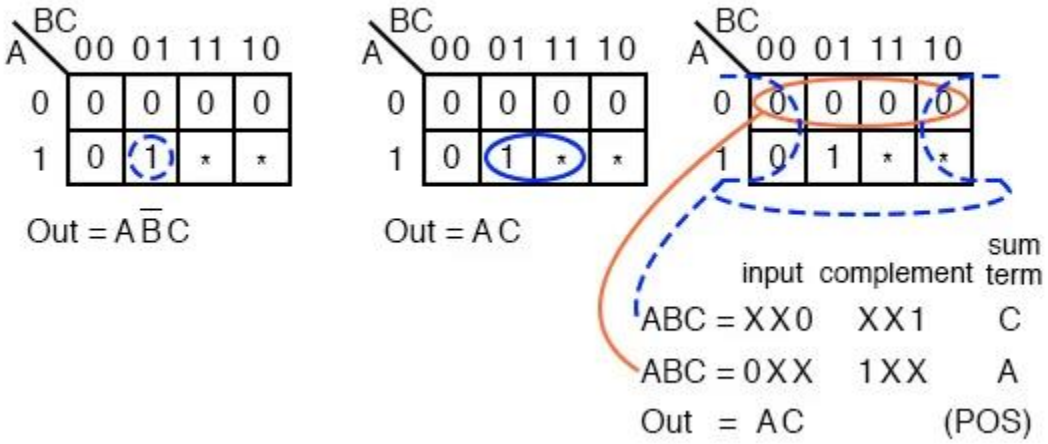
Significance of "Don't Care" Conditions:

- Don't Care conditions has the following significance in designing of the digital circuits:

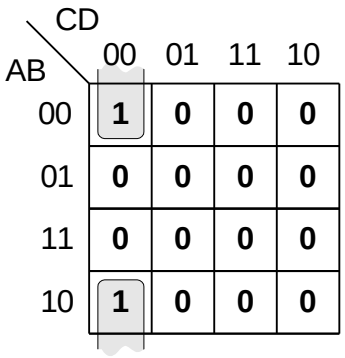
1. **Simplification of the output.**
2. **Reduction in number of gates required.**
3. **Reduced Power Consumption.**
4. **Prevention of Hazards in Digital Circuits.**

Don't-Care Conditions

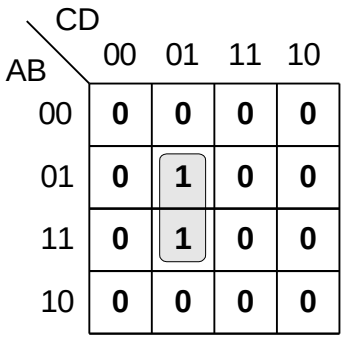
- The value of a function is not specified for certain combinations of variables
  - BCD; 1010-1111: don't care
- The don't-care conditions can be utilized in logic minimization
  - **Can be implemented as 0 or 1**



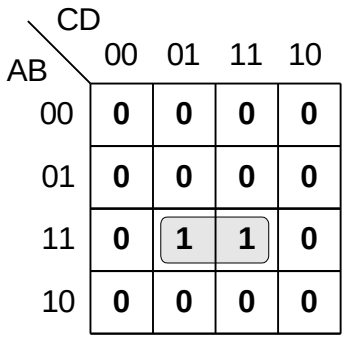
Four-Variable K-Maps



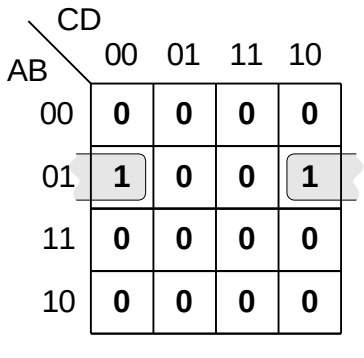
$f = \sum(0,8) = \overline{B} \bullet \overline{C} \bullet \overline{D}$



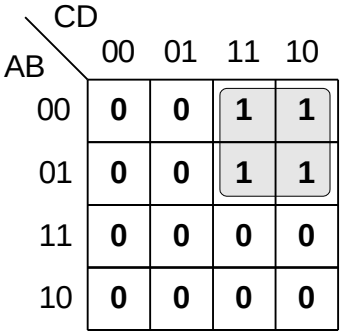
$f = \sum(5,13) = B \bullet \overline{C} \bullet D$



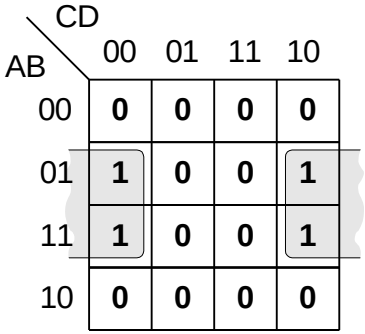
$f = \sum(13,15) = A \bullet B \bullet D$



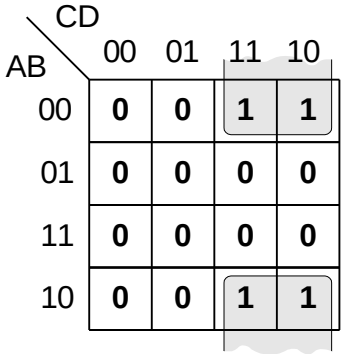
$f = \sum(4,6) = \overline{A} \bullet B \bullet \overline{D}$



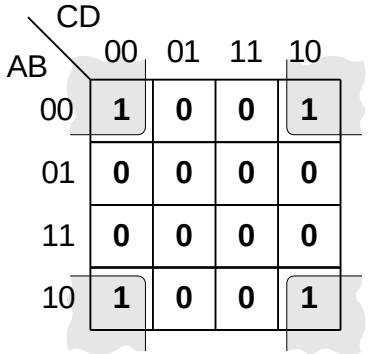
$f = \sum(2,3,6,7) = \overline{A} \bullet C$



$f = \sum(4,6,12,14) = B \bullet \overline{D}$



$f = \sum(2,3,10,11) = \overline{B} \bullet C$



$f = \sum(0,2,8,10) = \overline{B} \bullet \overline{D}$

Four-Variable K-Maps

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

$f = \sum(4,5,6,7) = \overline{A} \bullet B$

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

$f = \sum(3,7,11,15) = C \bullet D$

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

$f = \sum(0,3,5,6,9,10,12,15)$

$f = A \otimes B \otimes C \otimes D$

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

$f = \sum(1,2,4,7,8,11,13,14)$

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

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

$f = \sum(1,3,5,7,9,11,13,15)$

$f = D$

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

$f = \sum(0,2,4,6,8,10,12,14)$

$f = \overline{D}$

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

$f = \sum(4,5,6,7,12,13,14,15)$

$f = B$

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

$f = \sum(0,1,2,3,8,9,10,11)$

$f = \overline{B}$

Design of combinational digital circuits

- Steps to design a combinational digital circuit:
  - From the problem statement derive the truth table
  - From the truth table derive the unsimplified logic expression
  - Simplify the logic expression
  - From the simplified expression draw the logic circuit

Design of combinational digital circuits

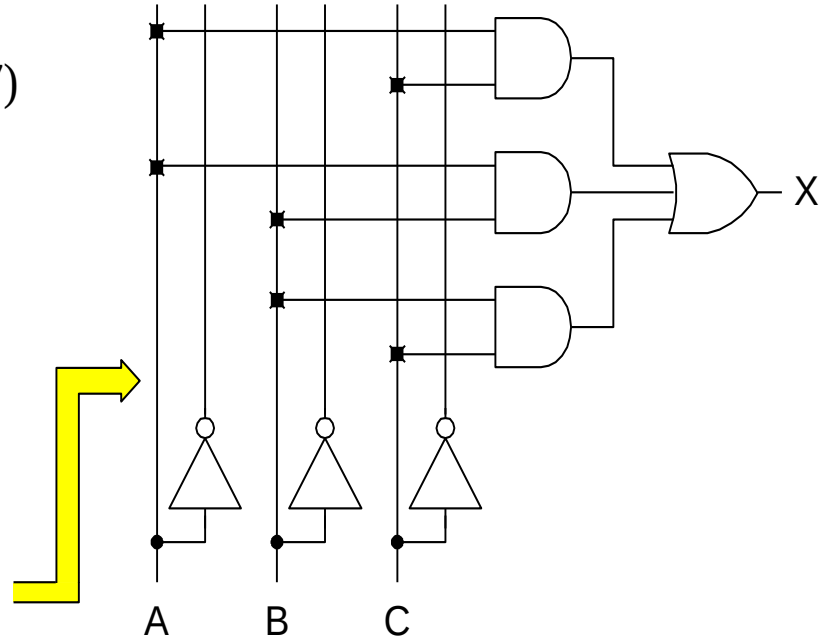
- Example: Design a 3-input (A,B,C) digital circuit that will give at its output (X) a logic 1 only if the binary number formed at the input has more ones than zeros.

	Inputs A	B	C	Output X
0	0	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	1	1	1
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

$$X = \sum(3,5,6,7)$$

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

$$X = AC + AB + BC$$



Design of combinational digital circuits

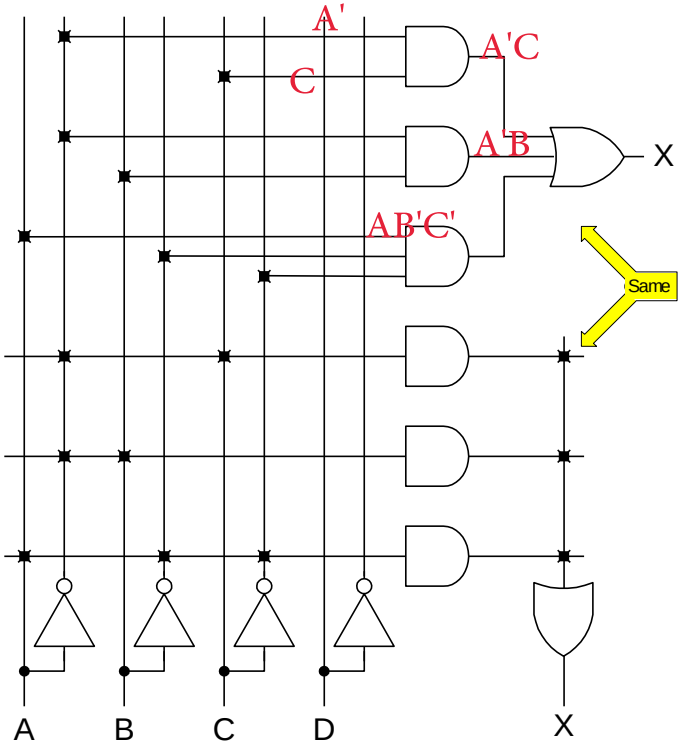
- Example: Design a 4-input (A,B,C,D) digital circuit that will give at its output (X) a logic 1 only if the binary number formed at the input is between 2 and 9 (including).

	Inputs				Output
	A	B	C	D	X
0	0	0	0	0	0
1	0	0	0	1	0
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	1
5	0	1	0	1	1
6	0	1	1	0	1
7	0	1	1	1	1
8	1	0	0	0	1
9	1	0	0	1	1
10	1	0	1	0	0
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

$X = \sum(2,3,4,5,6,7,8,9)$

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

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

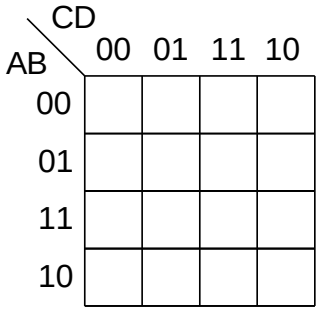


Design of combinational digital circuits (Example)

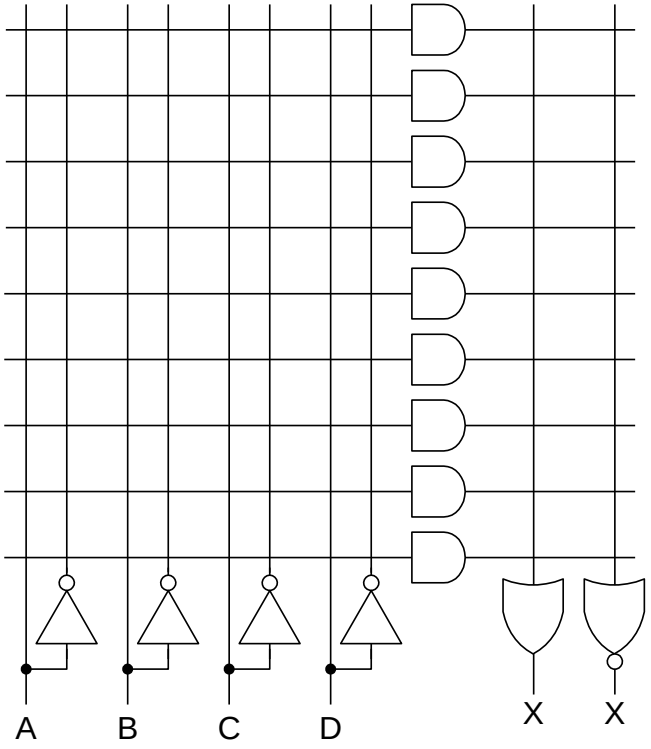
- Example: Design a 4-input (A,B,C,D) digital circuit that will give at its output (X) a logic 1 only if there more ones than zeros in the binary number formed at the input.

	Inputs				Output		
	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			
10	1	0	1	0			
11	1	0	1	1			
12	1	1	0	0			
13	1	1	0	1			
14	1	1	1	0			
15	1	1	1	1			

X =



X =



Name:

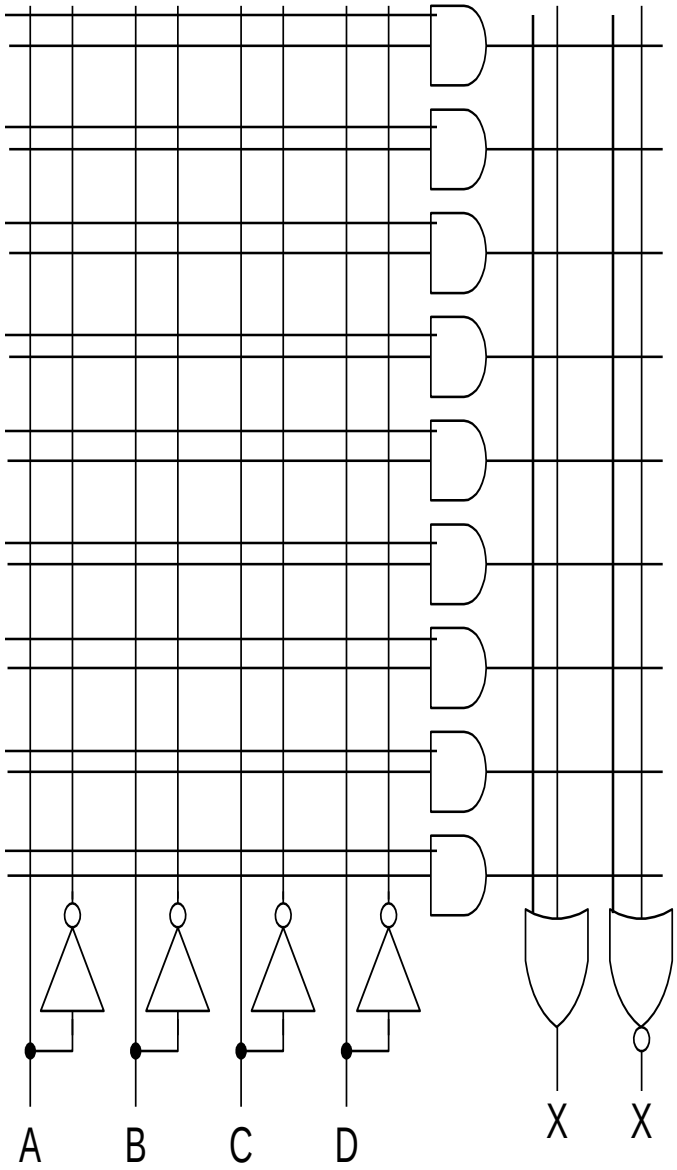
	Inputs				Output		
0	0	0	0	0			
1	0	0	0	1			
2	0	0	1	0			
3	0	0	1	1			
4	0	1	0	0			
5	0	1	0	1			
6	0	1	1	0			
7	0	1	1	1			
8	1	0	0	0			
9	1	0	0	1			
10	1	0	1	0			
11	1	0	1	1			
12	1	1	0	0			
13	1	1	0	1			
14	1	1	1	0			
15	1	1	1	1			

X =

K- Map

	y'	*y'*	*y*	*y*
x'				
x				
	z'	*z*	*z*	*z'*

X =



Name:

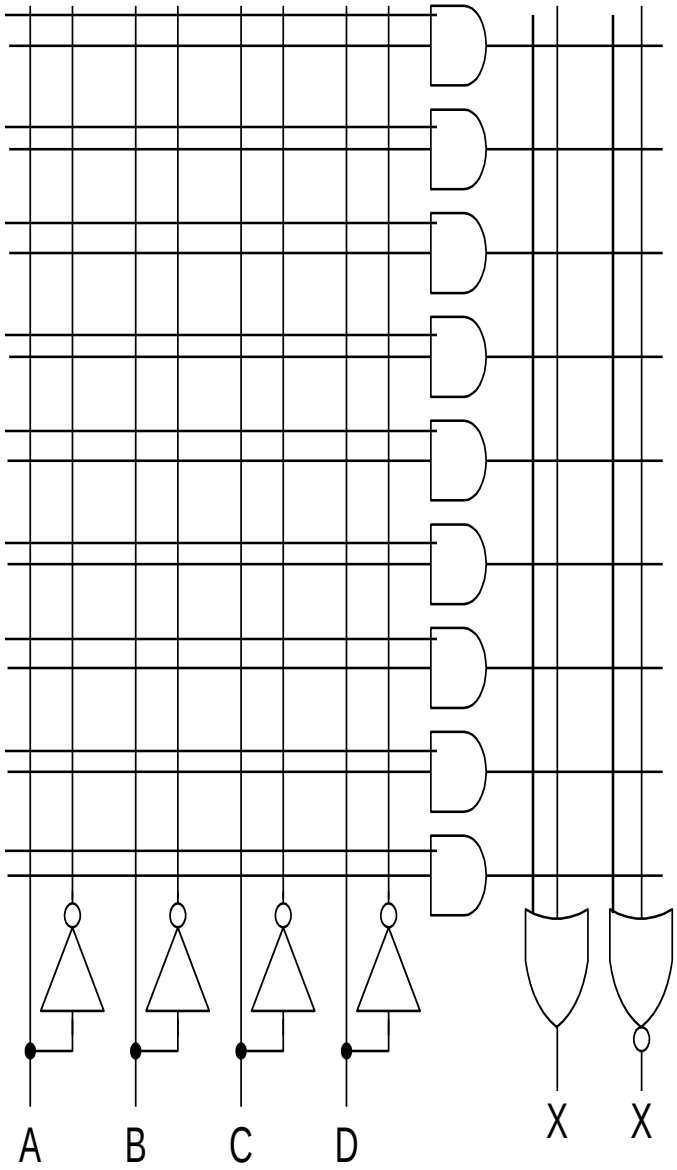
	Inputs				Output		
0	0	0	0	0			
1	0	0	0	1			
2	0	0	1	0			
3	0	0	1	1			
4	0	1	0	0			
5	0	1	0	1			
6	0	1	1	0			
7	0	1	1	1			
8	1	0	0	0			
9	1	0	0	1			
10	1	0	1	0			
11	1	0	1	1			
12	1	1	0	0			
13	1	1	0	1			
14	1	1	1	0			
15	1	1	1	1			

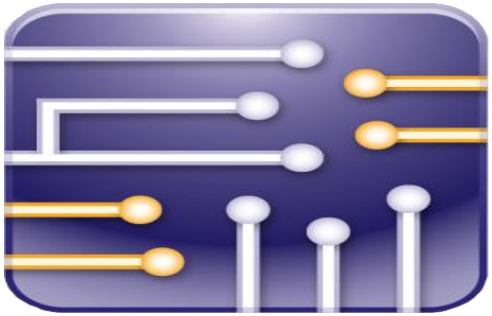
X =

K- Map

	C'	C'	C	C	
A'					B'
A'					B
A					B
A					B'
	D'	D	D	D'	

X =

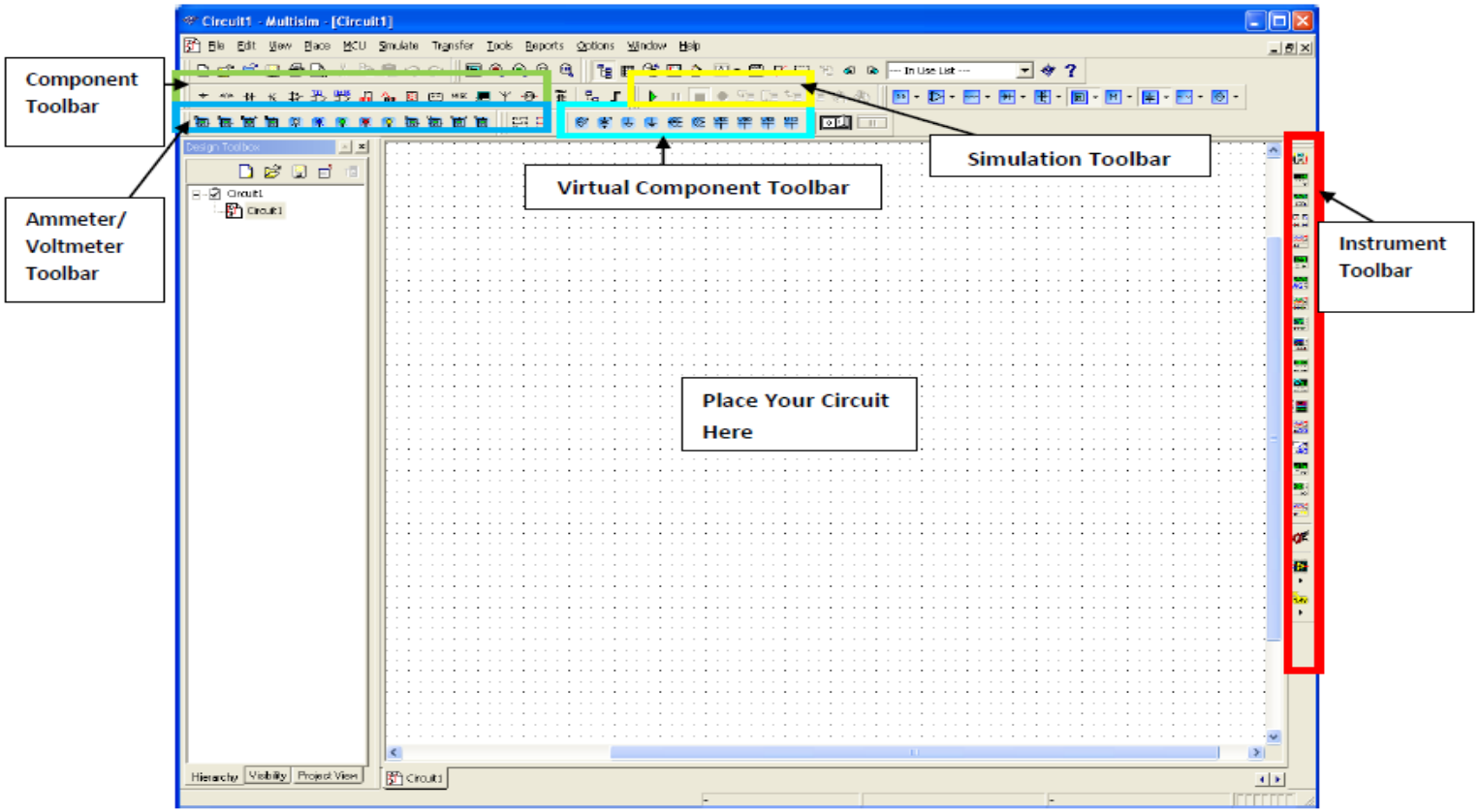




Multisim

Start

Click on Start ⟶ All Programs ⟶ National Instruments ⟶ Circuit Design Suite 10.0 ⟶ Multisim.

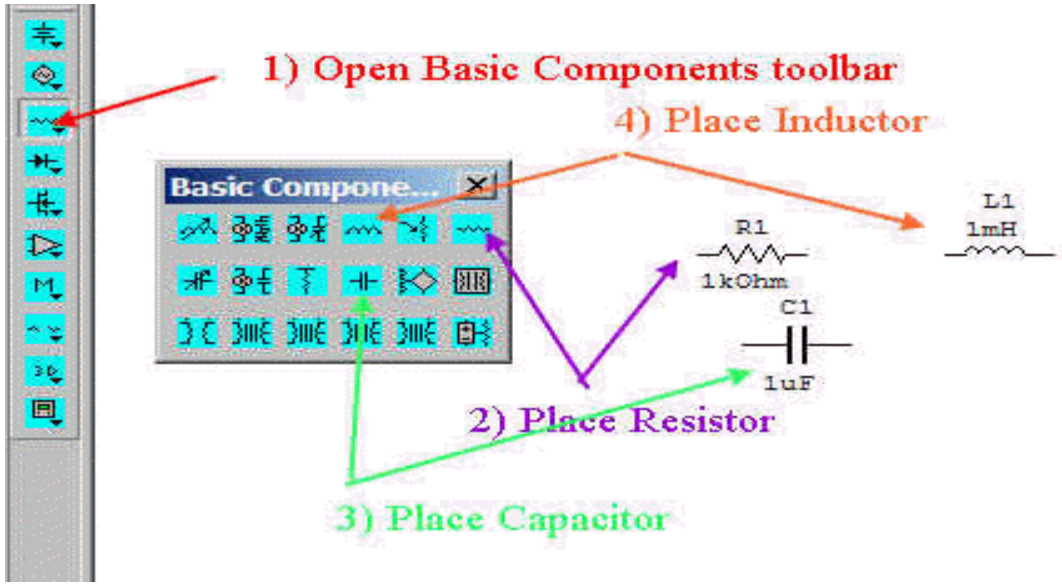


MultiSim

- **1. Open/Create Schematic**
- A blank schematic Circuit 1 is automatically created. To create a new schematic click on File – New – Schematic Capture. To save the schematic click on File /Save As.
- To open an existing file click on File/ Open in the toolbar.
- **2. Place Components**
- To Place Components click on Place/Components. On the Select Component Window click on Group to select the components needed for the circuit.
- Click OK to place the component on the schematic

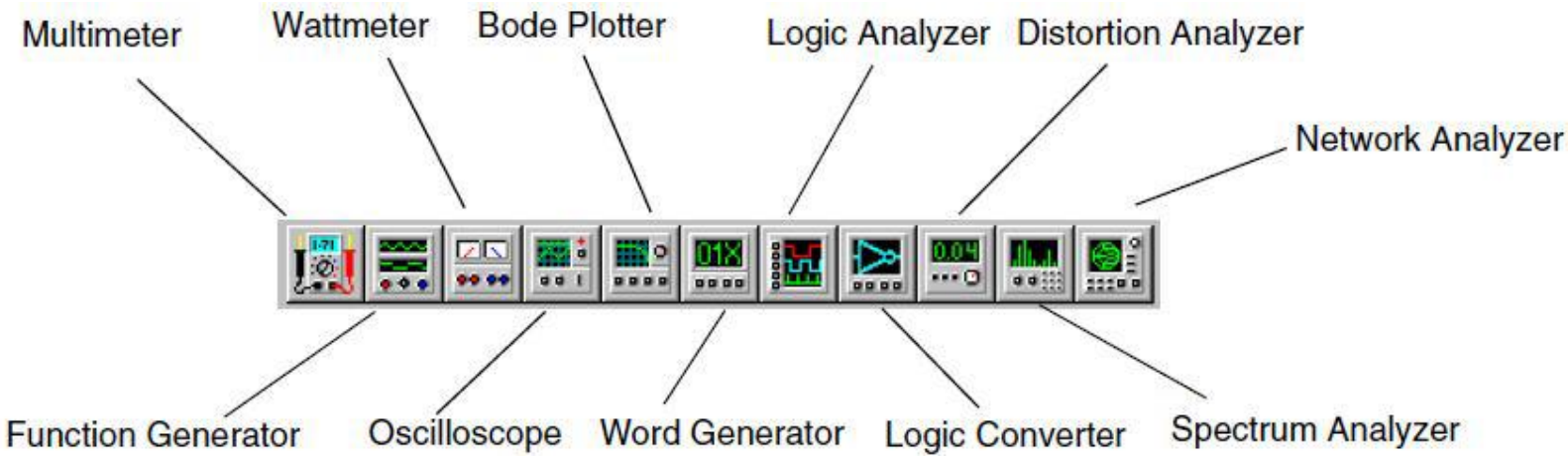
MultiSim - Components

- There are two kinds of component models used in MultiSim: Those modeled after **actual components** and those modeled after “ideal” components. Those modeled after ideal components are referred to as **“virtual” components**. There is a broad selection of virtual components available, as shown in Figure



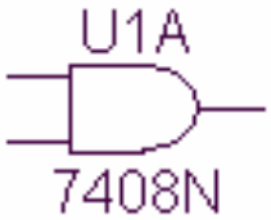
Measurement Devices

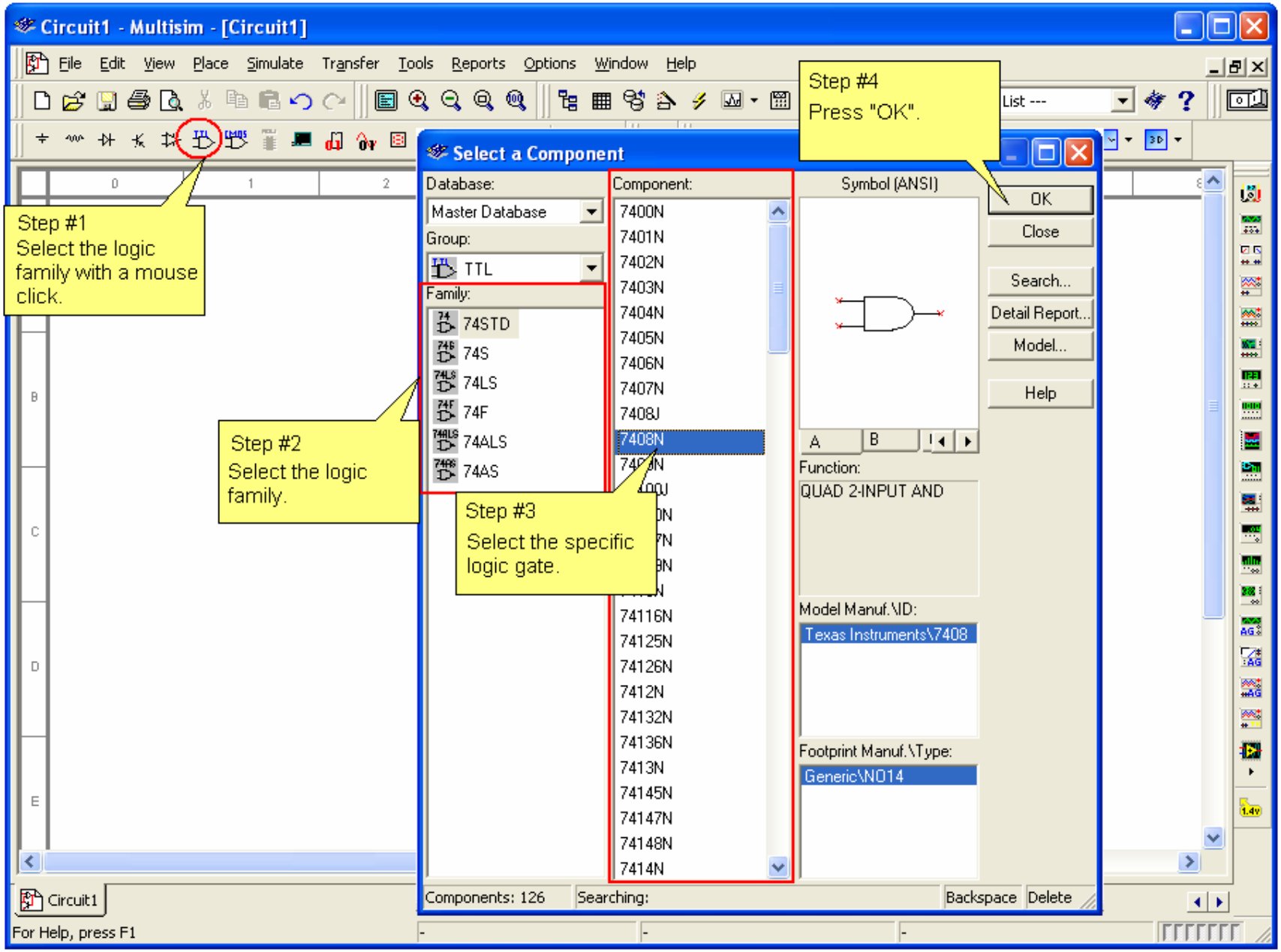
- Multimeter
- Function Generator
- Oscilloscope and etc..



MultiSim - Logic Gates

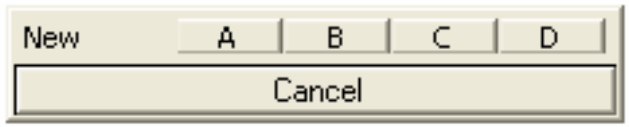
- The logic families are represented for both TTL and CMOS integrated circuits (IC).
- The logic gates are shown as a single gate, not the entire IC. If an IC contains more that one logic gate, the schematic symbol for the gate will contain a letter designation.
- For example, the **7408N is a quad 2-input AND gate.**
- The logic gates located inside the 7408N are designated A, B, C, and D.
- The user selects either the TTL or CMOS family of integrated circuits. Then the IC family is selected and finally the specific IC.



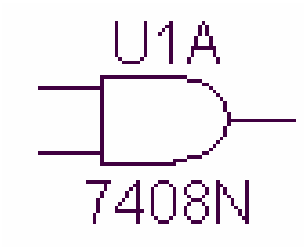

Circuit1 - Multisim - [Circuit1]
File Edit View Place Simulate Transfer Tools Reports Options Window Help
Step #1
Select the logic family with a mouse click.
Step #2
Select the logic family.
Step #3
Select the specific logic gate.
Step #4
Press "OK".
Select a Component
Database:
Master Database
Group:
TTL
Family:
74STD
74S
74LS
74F
74ALS
74AS
Component:
7400N
7401N
7402N
7403N
7404N
7405N
7406N
7407N
7408J
7408N
74116N
74125N
74126N
7412N
74132N
74136N
7413N
74145N
74147N
74148N
7414N
Symbol (ANSI)
A
B
Function:
QUAD 2-INPUT AND
Model Manuf.\ID:
Texas Instruments\7408
Footprint Manuf.\Type:
Generic\NO14
OK
Close
Search...
Detail Report...
Model...
Help
Components: 126
Searching:
Backspace
Delete
Circuit1
For Help, press F1

MultiSim - Logic Gates

- Once the IC is selected, MultiSim will prompt the user as to which logic gate inside the IC is to be used.
- The 7408N was selected. The 7408N is a quad 2-input AND gate. Therefore the user will have to choose from gates A through D as shown in Figure

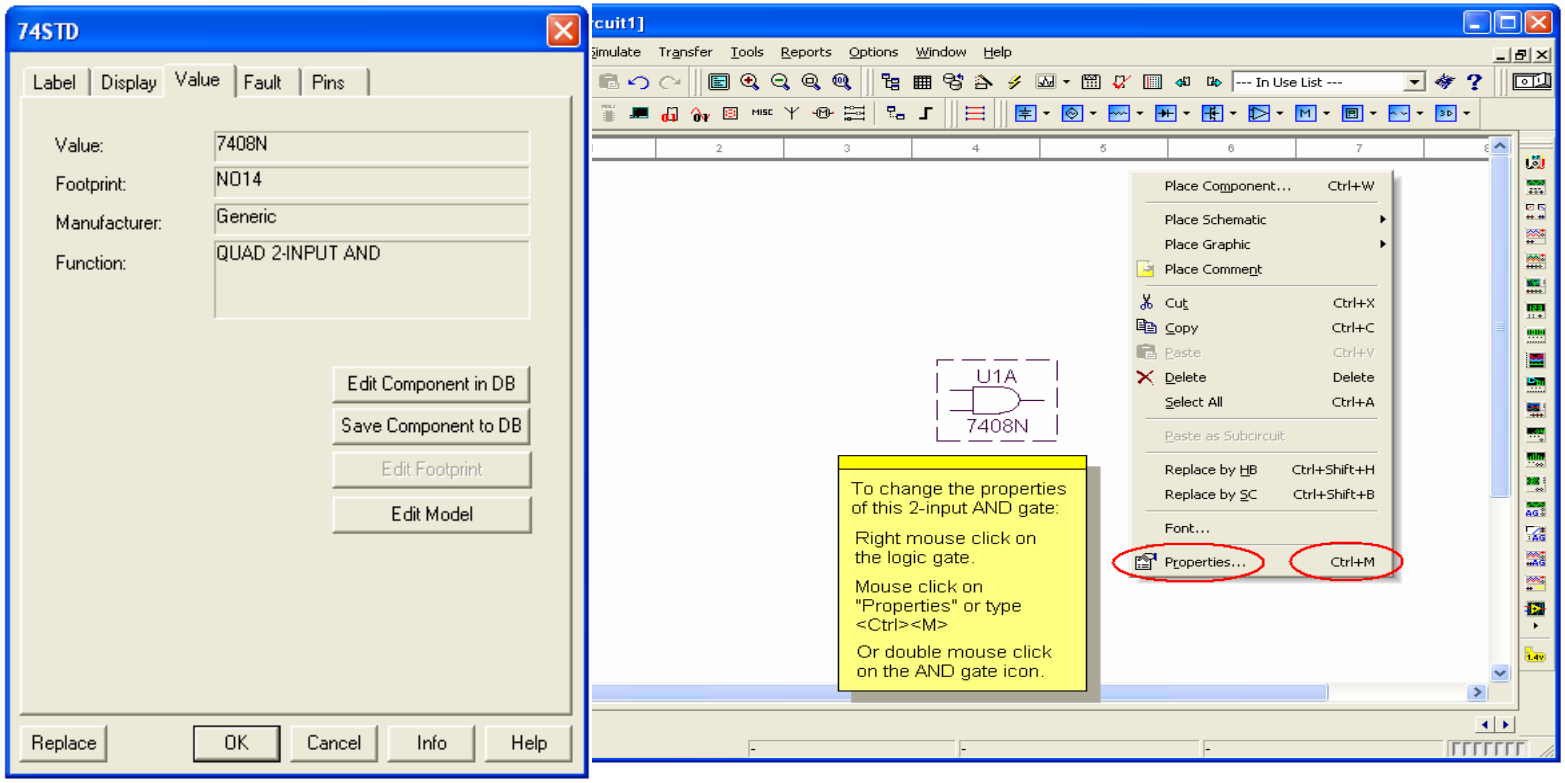


- The schematic symbol for the 2-input AND gate will contain two important pieces of information. The IC number will have a “U” prefix. The IC number will follow, with an A through D suffix designating the specific gate within the IC.



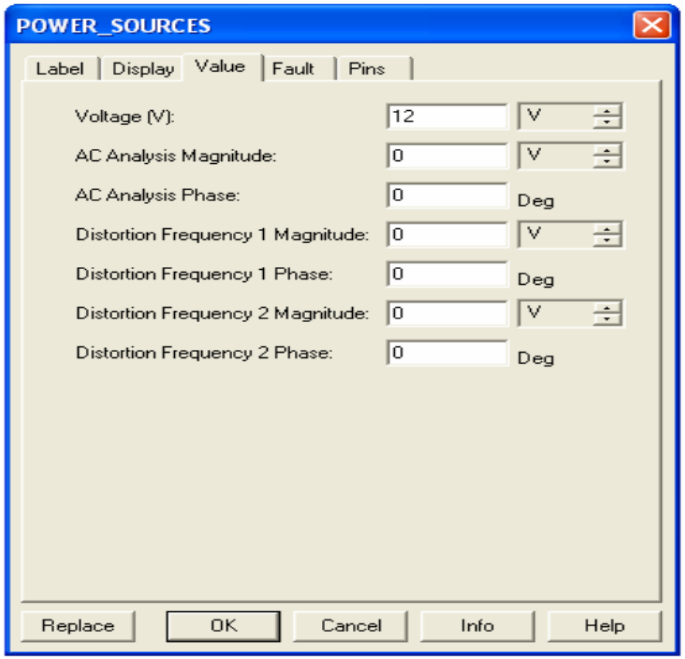
MultiSim - Logic Gates

- To change the properties of the logic gate, right mouse click on the gate and either select “Properties” from the pop-up menu or type <Ctrl><M> or simply double mouse click on the logic gate. This will cause the configuration screen to pop up.



Sources

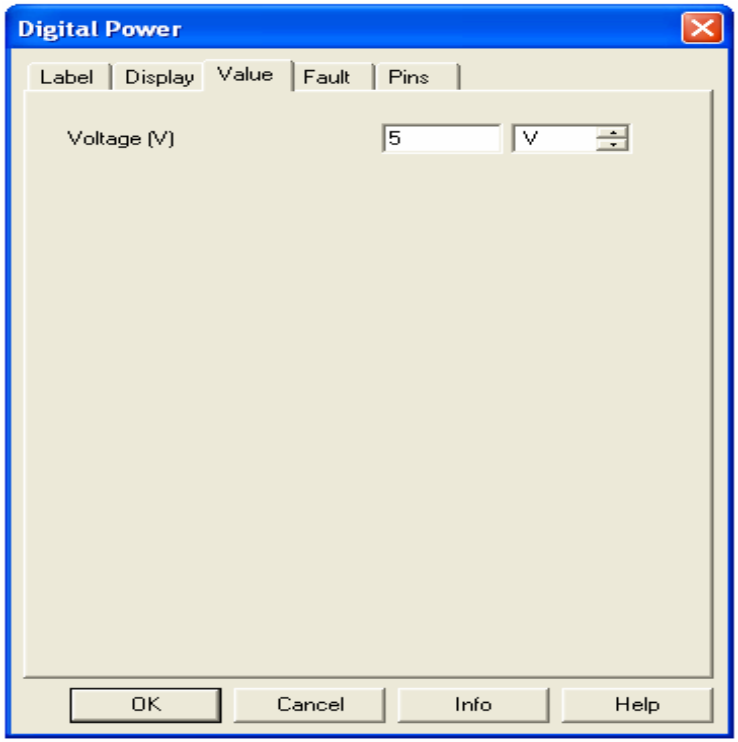
- In the study of Digital Electronics, all of the circuits contain a DC voltage source to supply power to the integrated circuits.
- A significant number of circuits also require a clock signal.
- The DC voltage source can be represented two ways: As a battery and as a voltage supply.



The voltage rating is fully adjustable.
The default value is 12 VDC.
If the component is double clicked, the configuration screen will pop up and the voltage value can be changed.

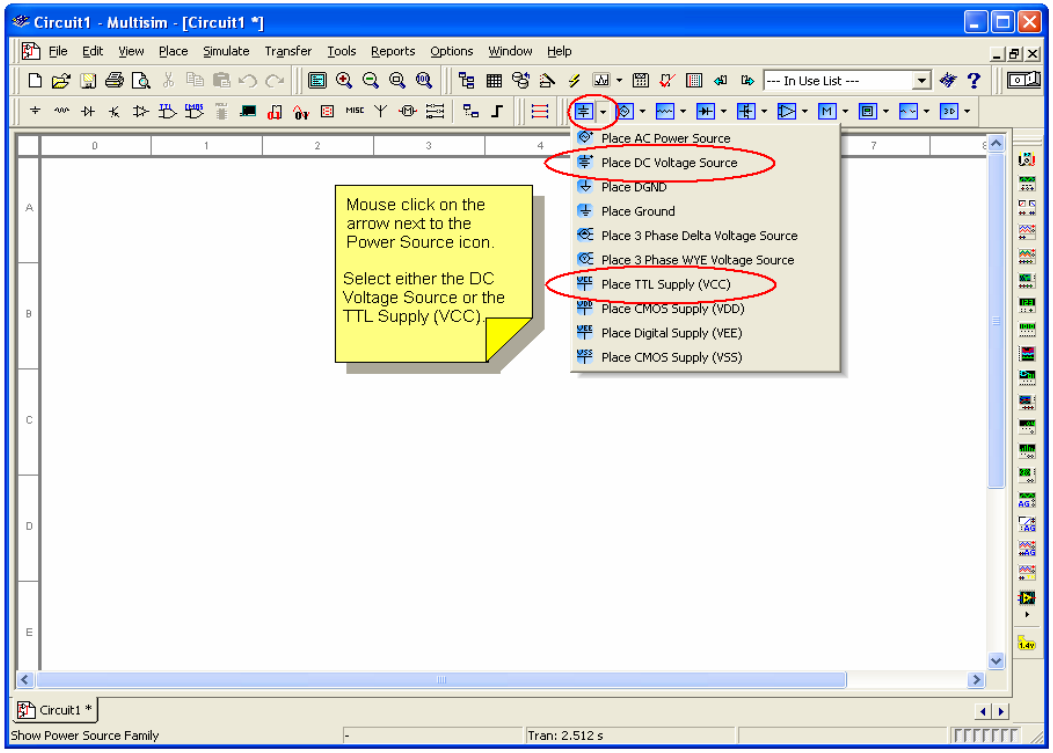
Sources

- The voltage rating is fully adjustable for the VCC voltage supply. The default value is +5 VDC.
- If the voltage supply is double clicked, the configuration screen will pop up and the voltage value can be changed.



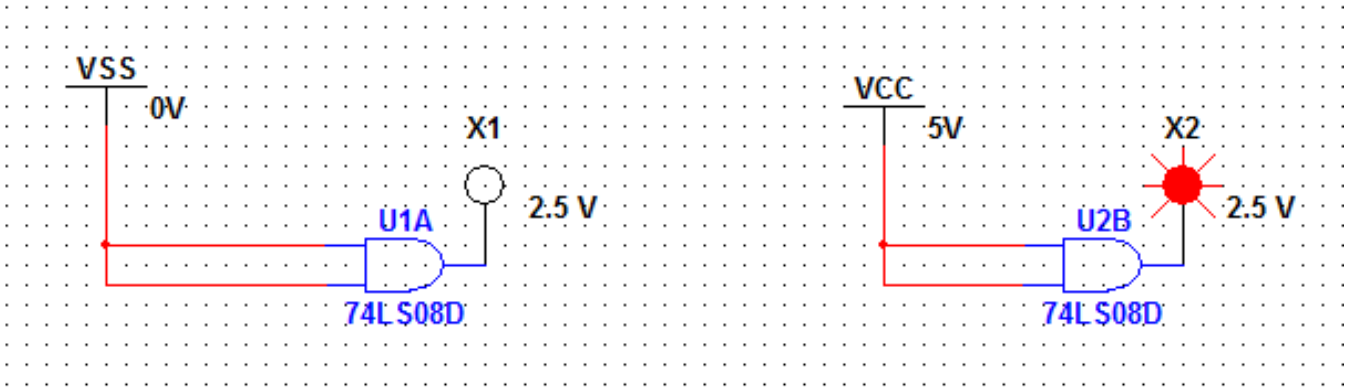
Sources

- The DC voltage sources are located on the drop down menu under the Power Source icon. To view the Power Source drop down menu, mouse click on the arrow next to the Power Source icon. The DC voltage sources can then be selected as shown in Figure



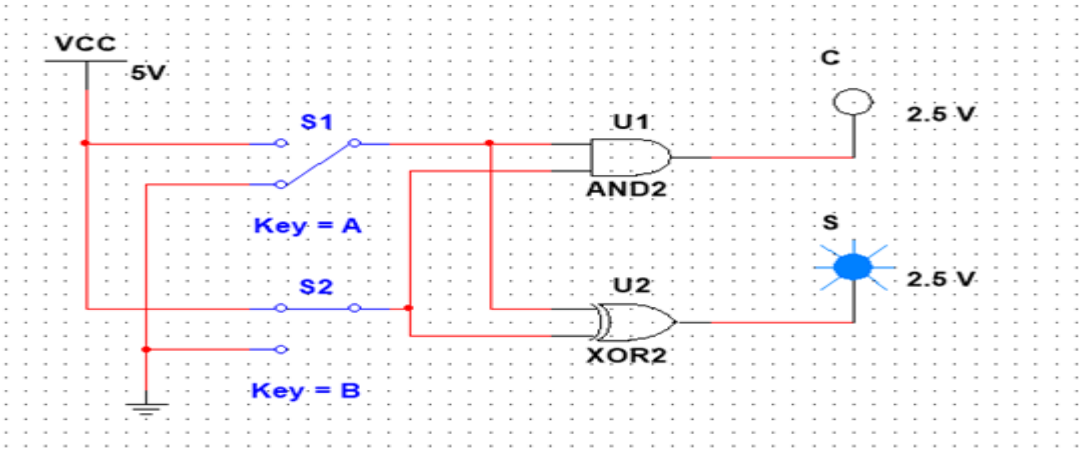
Example

- Construct the circuits as shown in the figure below and observe the response using the simulator.
- Replace the 74LS08D gate with a 74LS03D in both the above circuits and observe the response.



SPST switch

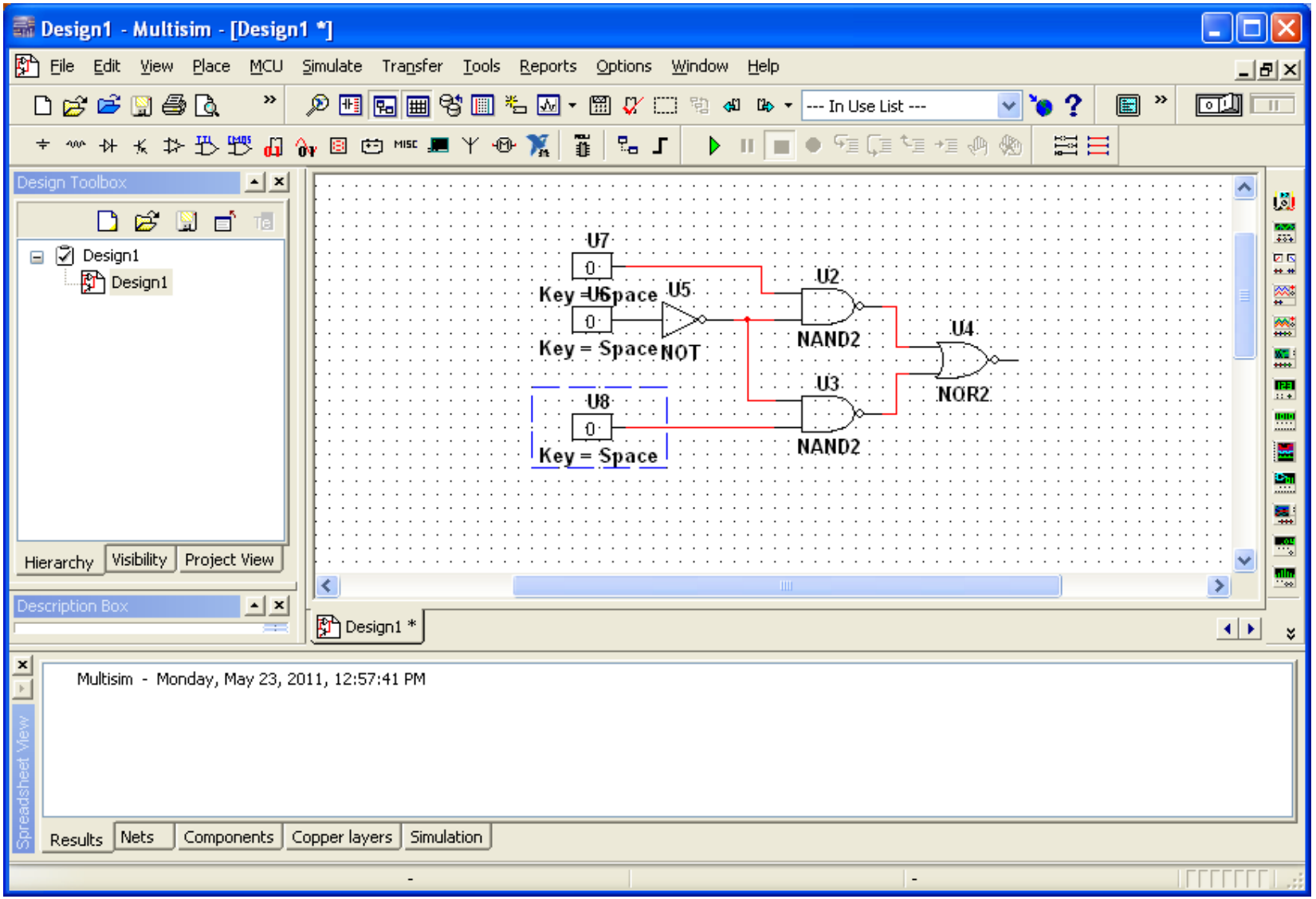
- This **single-pole double-throw switch** can be toggled on and off as described in Interactive components.
- To toggle the switch open or closed using the **keyboard, press the key that you entered in Key for toggle.** To toggle the switch open or closed using the mouse, hover the cursor over the switch's push button and click when the push button takes on a thickened appearance.



Digital Sources

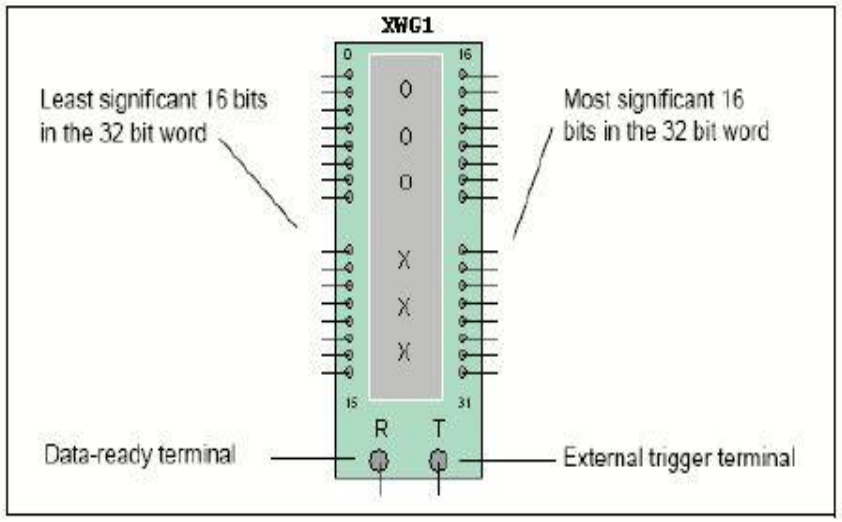
- There are three basic digital sources:
- **1. DIGITAL_CONSTANT** – this is a box with a **constant logic 1 or 0 output,** and would be used where the logic values is not to be changed during simulation. To change the output value, right click on the box, select Properties, select the desired value on the Value tab, and click the OK button.
- **2. INTERACTIVE_DIGITAL_CONSTANT** – this is a clickable box that can be connected to a circuit input. Clicking on the box toggles its output between 0 and 1. This can be used to interactively change a circuit input during simulation.
- **3. DIGITAL_CLOCK –** this is a box that produces a **repeating pulse train (square waveform), oscillating between 0 and 1 at a specified frequency.** To set the frequency and duty cycle, right click on the box, select Properties, select the desired frequency and duty cycle value on the Value tab, and click the OK button.

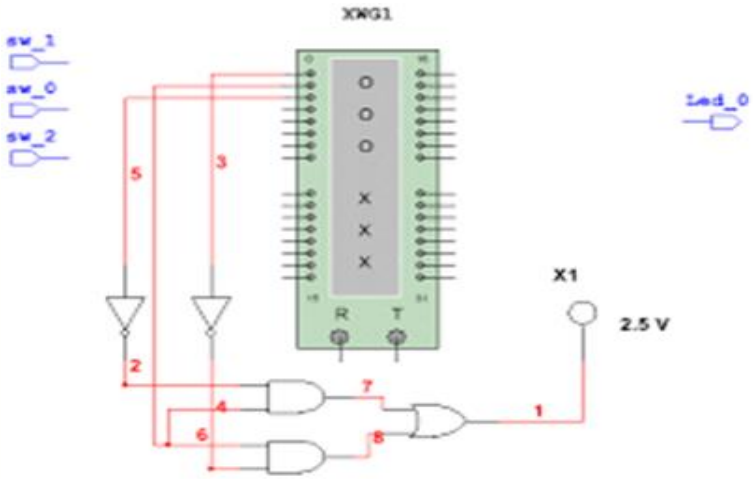
Digital Sources



Word Generator

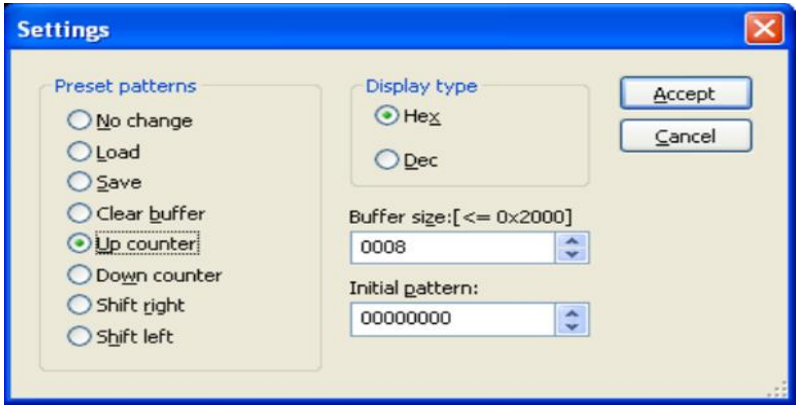
- One additional mechanism for generating digital patterns is the **"Word Generator" instrument**, which can be added to the circuit via the shortcut icon on the right side of the main window, or via the menu bar Simulate>Instruments>Word Generator.
- The Word Generator produces a sequence of patterns, each containing from 1 to 32 bits.





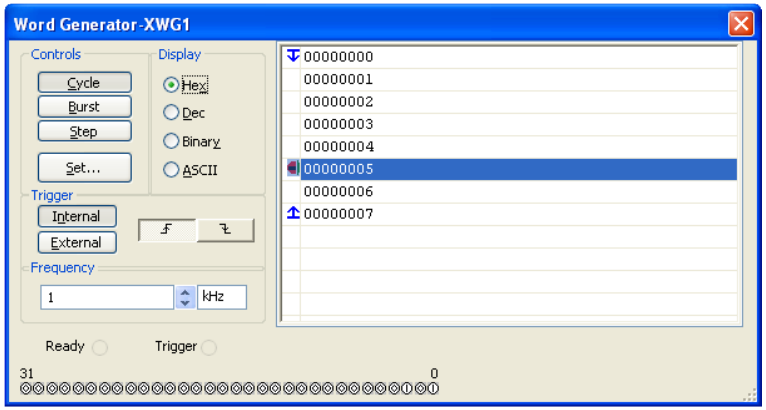
Word Generator

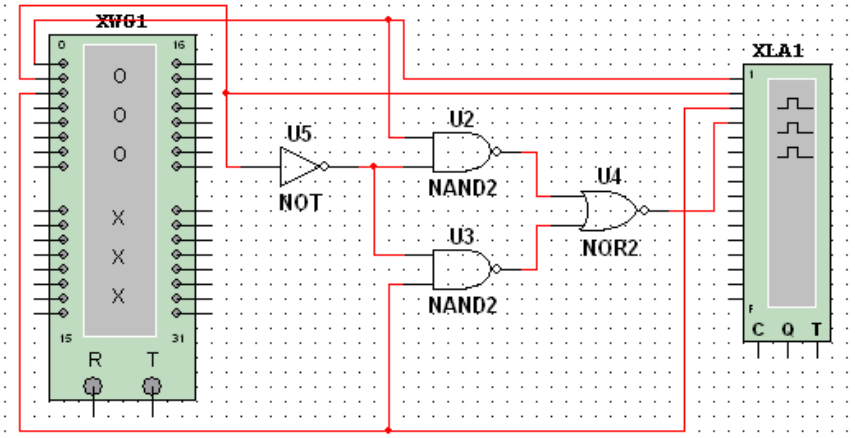
- To specify the patterns and the rate at which they should be produced, double click on the Word Generator symbol, producing the window.
- The Control buttons on the left of this window allow patterns (1) to be continuously applied and repeated (Cycle button), (2) a single set of patterns to be applied (Burst button), or (3) a single pattern to be applied (Step button).
- The patterns displayed in this window will be applied in the order listed, and can be entered manually or generated automatically.
- For automatically-generated patterns, press the Set button, producing the Settings window.
- In this example, “Up Counter” is selected to produce a sequence of binary numbers, “Buffer Size” is set to 8 to limit the sequence to 8 numbers, and “Initial pattern” is set to 0.



Word Generator

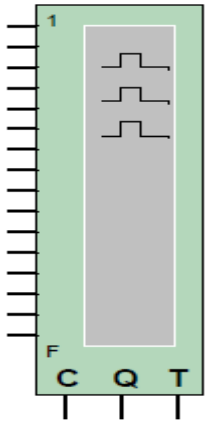
- Note that the 8 binary numbers are displayed in the Word Generator window.
- The rate at which the patterns are to be applied to the circuit is specified via the Frequency box of the Word Generator window. The frequency has been set to 1KHz, which means that 1000 patterns will be generated each second, or one pattern every 1msec.





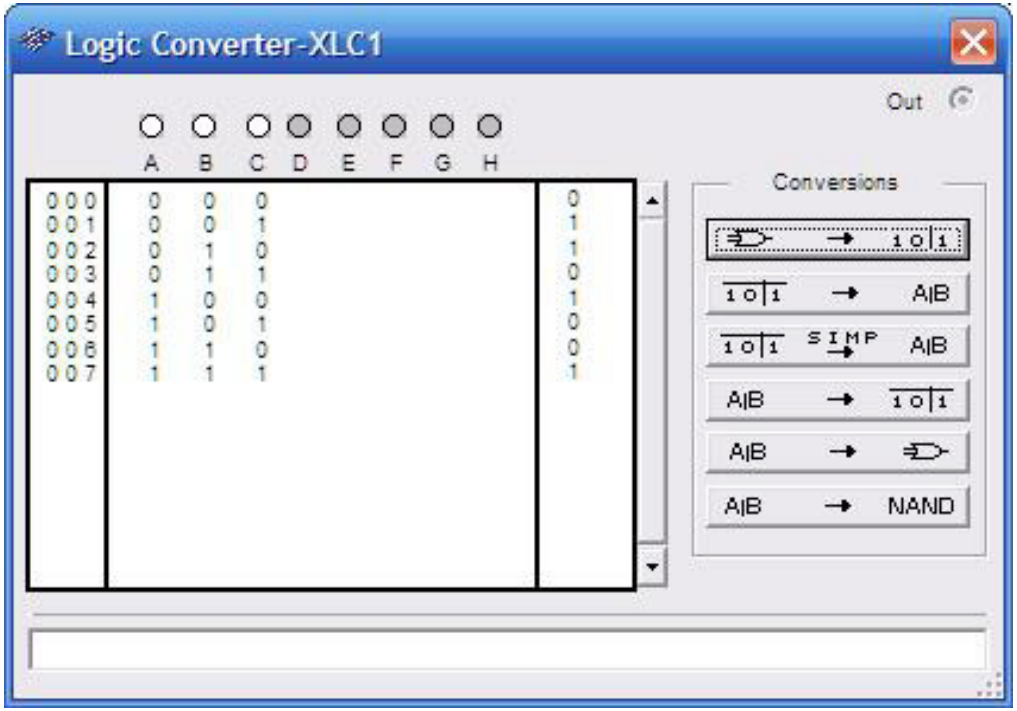
Logic Analyzer

- This instrument can show up to 16 input or output digital waveforms.
- Its inputs are on the left side and in the bottom there is a clock input, a clock qualifier and a trigger qualifier
- Trigger functions allow us to save data in a buffer when a trigger condition is met.
- This buffer will save data before and after the trigger condition according to the size specified in the properties dialog window.



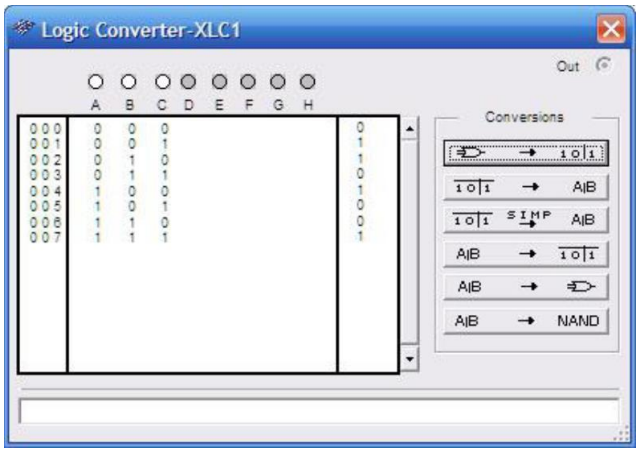
Logic Converter

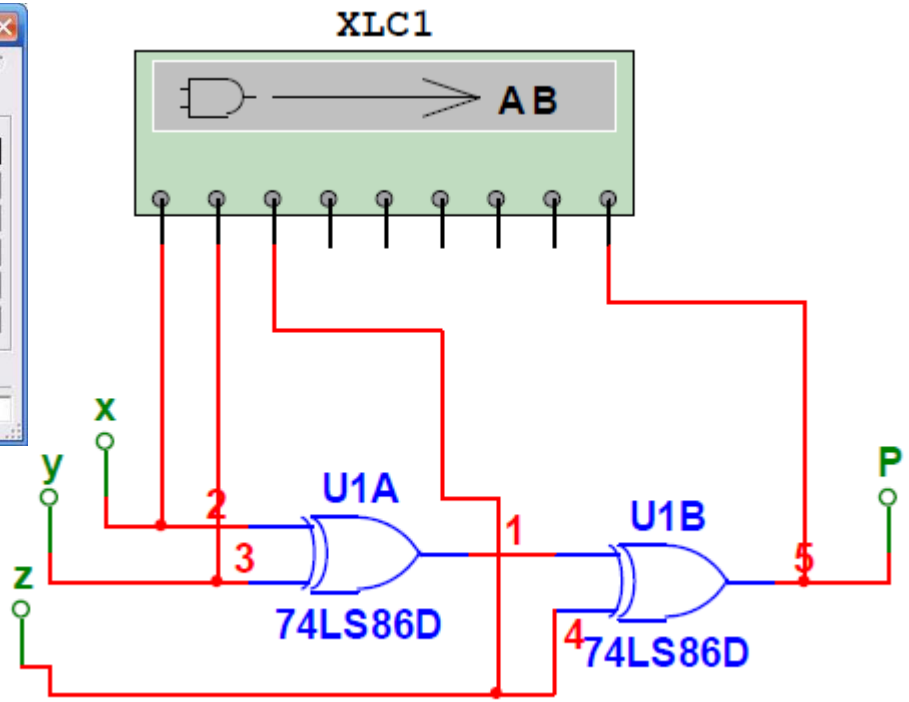
- A logic converter is a special tool in Multisim. This device does not have a real word version of itself.
- It is a tool for working with digital circuits, logic expressions and / or truth tables.



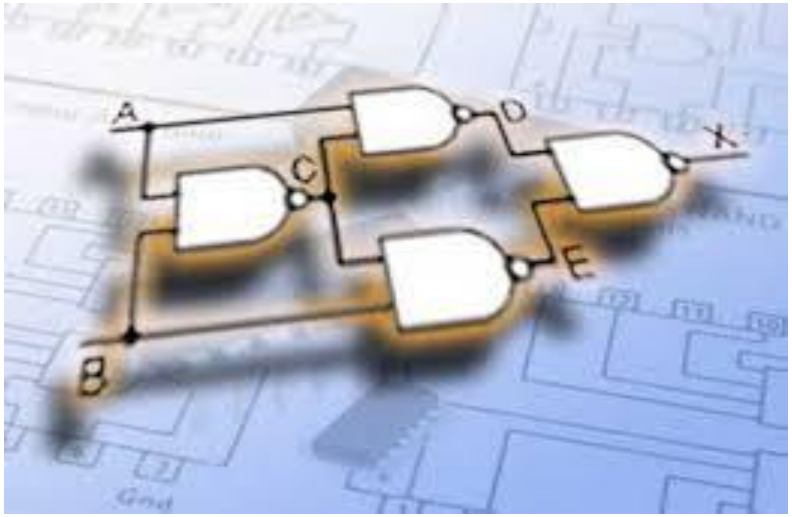
Logic Converter

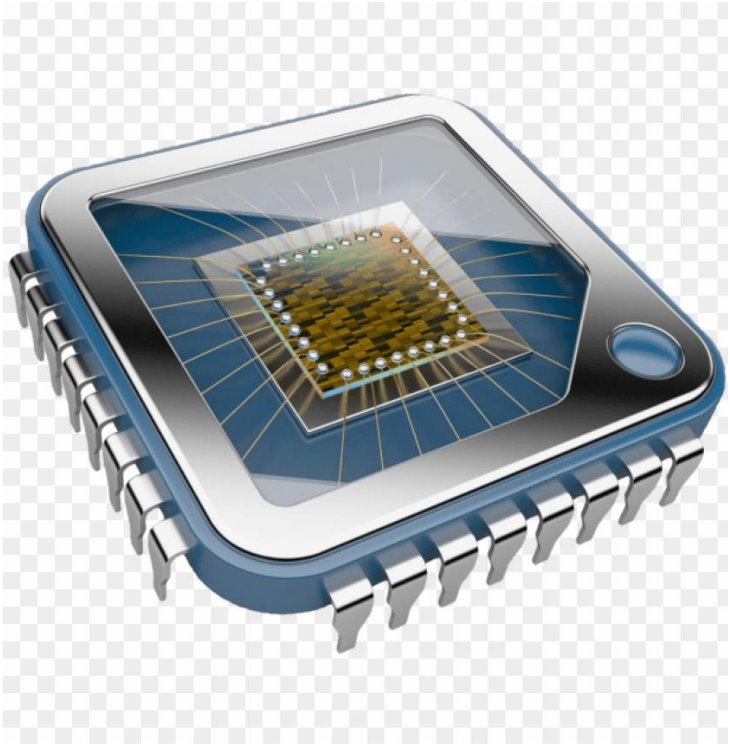
- This is a list of its functions:
- From gate implementation to truth tables.
- From truth table to a Boolean logic function.





Thank You





Dr. Hatem Yousry
E-mail: Hyousry@nctu.edu.eg