



MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L 01

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : I- BOOLEAN ALGEBRA AND LOGIC GATES

Date of Lecture:

Topic of Lecture: Review of Number systems, Number Representation

Introduction :

Number systems use different number bases. A number base indicates how many different digits are available when using a particular numbering system.

Prerequisite knowledge for Complete understanding and learning of Topic:

A basic idea regarding the initial concepts of Digital Electronics is enough to understand the topics covered.

If base or radix of a number system is 'r', then the numbers present in that number system are ranging from zero to r-1. The total numbers present in that number system is 'r'. So, we will get various number systems, by choosing the values of radix as greater than or equal to two.

In this chapter, let us discuss about the popular number systems and how to represent a number in the respective number system. The following number systems are the most commonly used.

- Decimal Number system
- Binary Number system
- Octal Number system
- Hexadecimal Number system

Decimal Number System

The base or radix of Decimal number system is 10. So, the numbers ranging from 0 to 9 are used in this number system..

In this number system, the successive positions to the left of the decimal point having weights of 100, 101, 102, 103 and so on. Similarly, the successive positions to the right of the decimal

point having weights of 10^{-1} , 10^{-2} , 10^{-3} and so on. That means, each position has specific weight, which is power of base 10

Example

Consider the decimal number 1358.246. Integer part of this number is 1358 and fractional part of this number is 0.246. The digits 8, 5, 3 and 1 have weights of 10^0 , 10^1 , 10^2 and 10^3 respectively. Similarly, the digits 2, 4 and 6 have weights of 10^{-1} , 10^{-2} and 10^{-3} respectively.

Mathematically, we can write it as

$$1358.246 = (1 \times 10^3) + (3 \times 10^2) + (5 \times 10^1) + (8 \times 10^0) + (2 \times 10^{-1}) + (4 \times 10^{-2}) + (6 \times 10^{-3})$$

After simplifying the right hand side terms, we will get the decimal number, which is on left hand side.

Binary Number System

All digital circuits and systems use this binary number system. The base or radix of this number system is 2. So, the numbers 0 and 1 are used in this number system.

The part of the number, which lies to the left of the binary point is known as integer part. Similarly, the part of the number, which lies to the right of the binary point is known as fractional part.

In this number system, the successive positions to the left of the binary point having weights of 2^0 , 2^1 , 2^2 , 2^3 and so on. Similarly, the successive positions to the right of the binary point having weights of 2^{-1} , 2^{-2} , 2^{-3} and so on. That means, each position has specific weight, which is power of base 2.

Example

Consider the binary number 1101.011. Integer part of this number is 1101 and fractional part of this number is 0.011. The digits 1, 0, 1 and 1 of integer part have weights of 2^0 , 2^1 , 2^2 , 2^3 respectively. Similarly, the digits 0, 1 and 1 of fractional part have weights of 2^{-1} , 2^{-2} , 2^{-3} respectively.

Mathematically, we can write it as

$$1101.011 = (1 \times 2^3) + (1 \times 2^2) + (0 \times 2^1) + (1 \times 2^0) + (0 \times 2^{-1}) + (1 \times 2^{-2}) + (1 \times 2^{-3})$$

After simplifying the right hand side terms, we will get a decimal number, which is an equivalent of binary number on left hand side.

Octal Number System

The base or radix of octal number system is 8. So, the numbers ranging from 0 to 7 are used in this number system.

In this number system, the successive positions to the left of the octal point having weights of 8^0 , 8^1 , 8^2 , 8^3 and so on. Similarly, the successive positions to the right of the octal point having weights of 8^{-1} , 8^{-2} , 8^{-3} and so on. That means, each position has specific weight, which is power of base 8.

Example

Consider the octal number 1457.236. Integer part of this number is 1457 and fractional part of this number is 0.236. The digits 7, 5, 4 and 1 have weights of 8^0 , 8^1 , 8^2 and 8^3 respectively. Similarly, the digits 2, 3 and 6 have weights of 8^{-1} , 8^{-2} , 8^{-3} respectively.

Mathematically, we can write it as

$$1457.236 = (1 \times 8^3) + (4 \times 8^2) + (5 \times 8^1) + (7 \times 8^0) + (2 \times 8^{-1}) + (3 \times 8^{-2}) + (6 \times 8^{-3})$$

After simplifying the right hand side terms, we will get a decimal number, which is an equivalent of octal number on left hand side.

Hexadecimal Number System

The base or radix of Hexa-decimal number system is 16. So, the numbers ranging from 0 to 9 and the letters from A to F are used in this number system. The decimal equivalent of Hexa-decimal digits from A to F are 10 to 15.

The part of the number, which lies to the left of the hexadecimal point is known as integer part. Similarly, the part of the number, which lies to the right of the Hexa-decimal point is known as fractional part.

In this number system, the successive positions to the left of the Hexa-decimal point having weights of 16^0 , 16^1 , 16^2 , 16^3 and so on. Similarly, the successive positions to the right of the Hexa-decimal point having weights of 16^{-1} , 16^{-2} , 16^{-3} and so on. That means, each position has specific weight, which is power of base 16.

Example

$$1A05.2C4 = (1 \times 16^3) + (10 \times 16^2) + (0 \times 16^1) + (5 \times 16^0) + (2 \times 16^{-1}) + (12 \times 16^{-2}) + (4 \times 16^{-3})$$

After simplifying the right hand side terms, we will get a decimal number, which is an equivalent of Hexa-decimal number on left hand side.

Video Content / Details of website for further learning (if any):

<https://www.khanacademy.org/math/algebra-home/alg-intro-to-algebra/algebra-alternate-number-bases/v/number-systems-introduction>

<https://www.youtube.com/watch?v=L2zsmYaI5ww>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (1)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 02

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : I- BOOLEAN ALGEBRA AND LOGIC GATES

Date of Lecture:

Topic of Lecture: Boolean algebra, Boolean postulates and laws

Introduction : Boolean Algebra is an algebra, which deals with binary numbers & binary variables. Hence, it is also called as Binary Algebra or logical Algebra. A mathematician, named George Boole had developed this algebra in 1854. The variables used in this algebra are also called as Boolean variables.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Decimal Number system
- Binary Number system
- Octal Number system
- Hexadecimal Number system

The range of voltages corresponding to Logic 'High' is represented with '1' and the range of voltages corresponding to logic 'Low' is represented with '0'.

Postulates and Basic Laws of Boolean Algebra

In this section, let us discuss about the Boolean postulates and basic laws that are used in Boolean algebra. These are useful in minimizing Boolean functions.

Boolean Postulates

Consider the binary numbers 0 and 1, Boolean variable x and its complement x' . Either the Boolean variable or complement of it is known as literal. The four possible logical OR operations among these literals and binary numbers are shown below.

$$x + 0 = x \quad x + 1 = 1 \quad x + x = x \quad x + x' = 1$$

Basic Laws of Boolean Algebra

Following are the three basic laws of Boolean Algebra.

- Commutative law
- Associative law

- Distributive law

Commutative Law

If any logical operation of two Boolean variables give the same result irrespective of the order of those two variables, then that logical operation is said to be Commutative. The logical OR & logical AND operations of two Boolean variables x & y are shown below

$$x + y = y + x$$

$$x.y = y.x$$

The symbol '+' indicates logical OR operation. Similarly, the symbol '.' indicates logical AND operation and it is optional to represent. Commutative law obeys for logical OR & logical AND operations.

Associative Law

The logical OR & logical AND operations of three Boolean variables x, y & z are shown below.

$$x + y + zy + z = x + yx + y + z$$

$$x.y.zy.z = x.yx.y.z$$

Associative law obeys for logical OR & logical AND operations.

Distributive Law

If any logical operation can be distributed to all the terms present in the Boolean function, then that logical operation is said to be Distributive. The distribution of logical OR & logical AND operations of three Boolean variables x, y & z are shown below.

$$x.y + zy + z = x.y + x.z$$

$$x + y.zy.z = x + yx + y.x + zx + z$$

Distributive law obeys for logical OR and logical AND operations.

These are the Basic laws of Boolean algebra. We can verify these laws easily, by substituting the Boolean variables with '0' or '1'.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=gj8QmRQtVao>

<https://www.youtube.com/watch?v=2U71nZYb990>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (38)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L 03

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : I- BOOLEAN ALGEBRA AND LOGIC GATES

Date of Lecture:

Topic of Lecture: De-Morgan's Theorem - Principle of Duality

Introduction :

One part may be obtained from the other if the binary operators and the identity elements are interchanged. This important property of Boolean algebra is called the duality principle.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Decimal Number system
- Binary Number system
- Octal Number system
- Hexadecimal Number system
- Boolean algebra
- Boolean postulates and laws

Theorems of Boolean Algebra

The following two theorems are used in Boolean algebra.

- Duality theorem
- DeMorgan's theorem

Duality Theorem

This theorem states that the dual of the Boolean function is obtained by interchanging the logical AND operator with logical OR operator and zeros with ones. For every Boolean function, there will be a corresponding Dual function.

Let us make the Boolean equations relations that we discussed in the section of Boolean postulates and basic laws into two groups. The following table shows these two groups.

Group1	Group2
$x + 0 = x$	$x.1 = x$
$x + 1 = 1$	$x.0 = 0$
$x + x = x$	$x.x = x$
$x + x' = 1$	$x.x' = 0$
$x + y = y + x$	$x.y = y.x$
$x + y+zy+z = x+yx+y + z$	$x.y.zy.z = x.yx.y.z$
$x.y+zy+z = x.y + x.z$	$x + y.zy.z = x+yx+y.x+zx+z$

In each row, there are two Boolean equations and they are dual to each other. We can verify all these Boolean equations of Group1 and Group2 by using duality theorem.

DeMorgan's Theorem

This theorem is useful in finding the complement of Boolean function. It states that the complement of logical OR of at least two Boolean variables is equal to the logical AND of each complemented variable.

DeMorgan's theorem with 2 Boolean variables x and y can be represented as

$$x+yx+y' = x'.y'$$

The dual of the above Boolean function is

$$x.yx.y' = x' + y'$$

Therefore, the complement of logical AND of two Boolean variables is equal to the logical OR of each complemented variable. Similarly, we can apply DeMorgan's theorem for more than 2 Boolean variables also.

Simplification of Boolean Functions

Till now, we discussed the postulates, basic laws and theorems of Boolean algebra. Now, let us simplify some Boolean functions.

Example 1

Let us simplify the Boolean function, $f = p'qr + pq'r + pqr' + pqr$

We can simplify this function in two methods.

Method 1

Given Boolean function, $f = p'qr + pq'r + pqr' + pqr$.

Step 1 – In first and second terms r is common and in third and fourth terms pq is common. So, take the common terms by using Distributive law.

$$\Rightarrow f = p'q + pq'p'q + pq'r + pqr' + rr' + r$$

Step 2 – The terms present in first parenthesis can be simplified to Ex-OR operation. The terms present in second parenthesis can be simplified to '1' using Boolean postulate

Method 2

Given Boolean function, $f = p'qr + pq'r + pqr' + pqr$.

Step 1 – Use the Boolean postulate, $x + x = x$. That means, the Logical OR operation with any Boolean variable 'n' times will be equal to the same variable. So, we can write the last term pqr two more times.

$$\Rightarrow f = p'qr + pq'r + pqr' + pqr + pqr + pqr$$

Step 2 – Use Distributive law for 1st and 4th terms, 2nd and 5th terms, 3rd and 6th terms.

$$\Rightarrow f = qrp' + pp' + p + prq' + qq' + q + pqr' + rr' + r$$

Step 3 – Use Boolean postulate, $x + x' = 1$ for simplifying the terms present in each parenthesis.

$$\Rightarrow f = qr11 + pr11 + pq11$$

Step 4 – Use Boolean postulate, $x.1 = x$ for simplifying the above three terms.

$$\Rightarrow f = qr + pr + pq$$

$$\Rightarrow f = pq + qr + pr$$

Therefore, the simplified Boolean function is $f = pq + qr + pr$.

So, we got two different Boolean functions after simplifying the given Boolean function in each method. Functionally, those two Boolean functions are same. So, based on the requirement, we can choose one of those two Boolean functions.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=yi6oxF8kgXk>

<https://www.youtube.com/watch?v=20bEwa8W4sQ>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (46)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)

Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L 04

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : I- BOOLEAN ALGEBRA AND LOGIC GATES

Date of Lecture:

Topic of Lecture: Simplification using Boolean algebra

Introduction :

Boolean algebra is an algebraic structure defined by a set of elements, B, together with two binary operators, + and #, provided that the following (Huntington) postulates are satisfied.

Prerequisite knowledge for Complete understanding and learning of Topic:

Binary operators

Basic laws and theorems of Boolean algebra are discussed. Now, let us simplify some Boolean functions.

Example 1

Let us simplify the Boolean function, $f = p'qr + pq'r + pqr' + pqr$

We can simplify this function in two methods.

Method 1

Given Boolean function, $f = p'qr + pq'r + pqr' + pqr$.

Step 1 – In first and second terms r is common and in third and fourth terms pq is common. So, take the common terms by using Distributive law.

$$\Rightarrow f = p'q + pq'p'q + pq'r + pqr' + rr'r$$

Step 2 – The terms present in first parenthesis can be simplified to Ex-OR operation. The terms present in second parenthesis can be simplified to '1' using Boolean postulate

$$\Rightarrow f = p \oplus qp \oplus qr + pq11$$

Step 3 – The first term can't be simplified further. But, the second term can be simplified to pq using Boolean postulate.

$$\Rightarrow f = p \oplus qp \oplus qr + pq$$

Therefore, the simplified Boolean function is $f = p \oplus qp \oplus qr + pq$

Method 2

Given Boolean function, $f = p'qr + pq'r + pqr' + pqr$.

Step 1 – Use the Boolean postulate, $x + x = x$. That means, the Logical OR operation with any Boolean variable 'n' times will be equal to the same variable. So, we can write the last term pqr two more times.

$$\Rightarrow f = p'qr + pq'r + pqr' + pqr + pqr + pqr$$

Step 2 – Use Distributive law for 1st and 4th terms, 2nd and 5th terms, 3rd and 6th terms.

$$\Rightarrow f = qrp' + pp' + p + prq' + qq' + q + pqr' + rr'r$$

Step 3 – Use Boolean postulate, $x + x' = 1$ for simplifying the terms present in each parenthesis.

$$\Rightarrow f = qr11 + pr11 + pq11$$

Step 4 – Use Boolean postulate, $x.1 = x$ for simplifying the above three terms.

$$\Rightarrow f = qr + pr + pq$$

$$\Rightarrow f = pq + qr + pr$$

Therefore, the simplified Boolean function is $f = pq + qr + pr$.

So, we got two different Boolean functions after simplifying the given Boolean function in each method. Functionally, those two Boolean functions are same. So, based on the requirement, we can choose one of those two Boolean functions.

Example 2

Let us find the complement of the Boolean function, $f = p'q + pq'$.

The complement of Boolean function is $f' = p'q + pq' + p'q + pq'$.

Step 1 – Use DeMorgan's theorem, $x+yx+y' = x'.y'$.

$$\Rightarrow f' = p'qp'q'.pq'pq''$$

Step 2 – Use DeMorgan's theorem, $x.yx.y' = x' + y'$

$$\Rightarrow f' = \{p'p'' + q'\}.\{p' + q'q''\}$$

Step 3 – Use the Boolean postulate, $x'x''=x$.

$$\Rightarrow f' = \{p + q'\}.\{p' + q\}$$

$$\Rightarrow f' = pp' + pq + p'q' + qq'$$

Step 4 – Use the Boolean postulate, $xx'=0$.

$$\Rightarrow f' = 0 + pq + p'q' + 0$$

$$\Rightarrow f' = pq + p'q'$$

Therefore, the complement of Boolean function, $p'q + pq'$ is $pq + p'q'$.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=0as464WmfCo>

<https://www.youtube.com/watch?v=59BbncMjL8I>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (43)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L 05

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : I- BOOLEAN ALGEBRA AND LOGIC GATES

Date of Lecture:

Topic of Lecture: Canonical forms Sum of product and Product of sum

Introduction :

A truth table consists of a set of inputs and outputs. If there are 'n' input variables, then there will be 2^n possible combinations with zeros and ones. So the value of each output variable depends on the combination of input variables. So, each output variable will have '1' for some combination of input variables and '0' for some other combination of input variables.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Boolean algebra
- Boolean postulates and laws

- Canonical SoP form
- Canonical PoS form

Canonical SoP form

Canonical SoP form means Canonical Sum of Products form. In this form, each product term contains all literals. So, these product terms are nothing but the min terms. Hence, canonical SoP form is also called as sum of min terms form.

First, identify the min terms for which, the output variable is one and then do the logical OR of those min terms in order to get the Boolean expression function corresponding to that output variable. This Boolean function will be in the form of sum of min terms.

Follow the same procedure for other output variables also, if there is more than one output variable.

Consider the following truth table.

Inputs		Output	
p	q	r	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Canonical PoS form

Canonical PoS form means Canonical Product of Sums form. In this form, each sum term contains all literals. So, these sum terms are nothing but the Max terms. Hence, canonical PoS form is also called as product of Max terms form.

First, identify the Max terms for which, the output variable is zero and then do the logical AND of those Max terms in order to get the Boolean expression function corresponding to that output variable. This Boolean function will be in the form of product of Max terms.

Standard SoP and PoS forms

We discussed two canonical forms of representing the Boolean outputs. Similarly, there are two standard forms of representing the Boolean outputs. These are the simplified version of canonical forms.

- Standard SoP form
- Standard PoS form

Standard SoP form

Standard SoP form means Standard Sum of Products form. In this form, each product term need not contain all literals. So, the product terms may or may not be the min terms. Therefore, the Standard SoP form is the simplified form of canonical SoP form.

We will get Standard SoP form of output variable in two steps.

- Get the canonical SoP form of output variable
- Simplify the above Boolean function, which is in canonical SoP form.

Standard PoS form

Standard PoS form means Standard Product of Sums form. In this form, each sum term need not contain all literals. So, the sum terms may or may not be the Max terms. Therefore, the Standard PoS form is the simplified form of canonical PoS form.

We will get Standard PoS form of output variable in two steps.

- Get the canonical PoS form of output variable
- Simplify the above Boolean function, which is in canonical PoS form.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=iilBvuv1QU4>

<https://www.youtube.com/watch?v=OaLUco7KL1Q>

<https://www.youtube.com/watch?v=Dv0KZ6E3tc4>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (51)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L 06

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : I - BOOLEAN ALGEBRA AND LOGIC GATES

Date of Lecture:

Topic of Lecture: Minimization using Karnaugh map

Introduction :

The procedure of minimization is awkward because it lacks specific rules to predict each succeeding step in the manipulative process. The map method presented here provides a simple, straightforward procedure for minimizing Boolean functions. This method may be regarded as a pictorial form of a truth table. The map method is also known as the Karnaugh map or K-map .

Prerequisite knowledge for Complete understanding and learning of Topic:

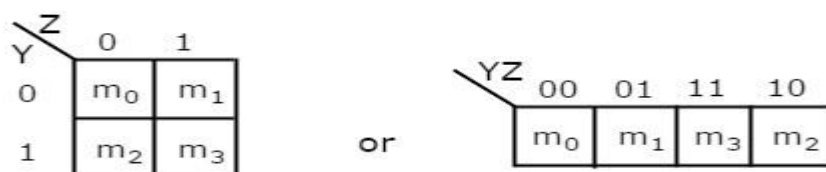
- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum

K-Maps for 2 to 5 Variables

K-Map method is most suitable for minimizing Boolean functions of 2 variables to 5 variables. Now, let us discuss about the K-Maps for 2 to 5 variables one by one.

2 Variable K-Map

The number of cells in 2 variable K-map is four, since the number of variables is two. The following figure shows 2 variable K-Map.



- There is only one possibility of grouping 4 adjacent min terms.
- The possible combinations of grouping 2 adjacent min terms are {(m0, m1), (m2, m3), (m0, m2) and (m1, m3)}.

3 Variable K-Map

The number of cells in 3 variable K-map is eight, since the number of variables is three. The following figure shows 3 variable K-Map.

		YZ			
		00	01	11	10
X	0	m ₀	m ₁	m ₃	m ₂
	1	m ₄	m ₅	m ₇	m ₆

- There is only one possibility of grouping 8 adjacent min terms.
- The possible combinations of grouping 4 adjacent min terms are {(m₀, m₁, m₃, m₂), (m₄, m₅, m₇, m₆), (m₀, m₁, m₄, m₅), (m₁, m₃, m₅, m₇), (m₃, m₂, m₇, m₆) and (m₂, m₀, m₆, m₄}.
- The possible combinations of grouping 2 adjacent min terms are {(m₀, m₁), (m₁, m₃), (m₃, m₂), (m₂, m₀), (m₄, m₅), (m₅, m₇), (m₇, m₆), (m₆, m₄), (m₀, m₄), (m₁, m₅), (m₃, m₇) and (m₂, m₆}.
- If x=0, then 3 variable K-map becomes 2 variable K-map.

4 Variable K-Map

- The number of cells in 4 variable K-map is sixteen, since the number of variables is four. The following figure shows 4 variable K-Map.

		YZ			
		00	01	11	10
WX	00	m ₀	m ₁	m ₃	m ₂
	01	m ₄	m ₅	m ₇	m ₆
	11	m ₁₂	m ₁₃	m ₁₅	m ₁₄
	10	m ₈	m ₉	m ₁₁	m ₁₀

- There is only one possibility of grouping 16 adjacent min terms.
- Let R₁, R₂, R₃ and R₄ represents the min terms of first row, second row, third row and fourth row respectively. Similarly, C₁, C₂, C₃ and C₄ represents the min terms of first column, second column, third column and fourth column respectively. The possible combinations of grouping 8 adjacent min terms are {(R₁, R₂), (R₂, R₃), (R₃, R₄), (R₄, R₁), (C₁, C₂), (C₂, C₃), (C₃, C₄), (C₄, C₁}.
- If w=0, then 4 variable K-map becomes 3 variable K-map.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=wjM2RDG5yTI>

<https://www.youtube.com/watch?v=CpsJoAwreqo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (73)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L 07

CSE

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : I- BOOLEAN ALGEBRA AND LOGIC GATES

Date of Lecture:

Topic of Lecture: Minimization using Karnaugh map

Introduction :

The procedure of minimization is awkward because it lacks specific rules to predict each succeeding step in the manipulative process. The map method presented here provides a simple, straightforward procedure for minimizing Boolean functions. This method may be regarded as a pictorial form of a truth table. The map method is also known as the Karnaugh map or K-map .

Prerequisite knowledge for Complete understanding and learning of Topic:

- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum

Minimization of Boolean Functions using K-Maps

Follow these rules for simplifying K-maps in order to get standard sum of products form.

- Select the respective K-map based on the number of variables present in the Boolean function.
- If the Boolean function is given as sum of min terms form, then place the ones at respective min term cells in the K-map.
- Check for the possibilities of grouping maximum number of adjacent ones. It should be powers of two.
- Each grouping will give either a literal or one product term. It is known as prime implicant. The prime implicant is said to be essential prime implicant, if atleast single '1' is not covered with any other groupings but only that grouping covers.
- Note down all the prime implicants and essential prime implicants. The simplified Boolean function contains all essential prime implicants and only the required prime implicants.

Example

Let us simplify the following Boolean function, $f(W,X,Y,Z) = WX'Y' + WY + W'YZ'$ using K-map.

The given Boolean function is in sum of products form. It is having 4 variables W, X, Y & Z. So, we require 4 variable K-map. The 4 variable K-map with ones corresponding to the given product terms is shown in the following figure.

WX \ YZ	00	01	11	10
00				1
01				1
11			1	1
10	1	1	1	1

Here, 1s are placed in the following cells of K-map.

- The cells, which are common to the intersection of Row 4 and columns 1 & 2 are corresponding to the product term, $WX'Y'$.
- The cells, which are common to the intersection of Rows 3 & 4 and columns 3 & 4 are corresponding to the product term, WY .
- The cells, which are common to the intersection of Rows 1 & 2 and column 4 are corresponding to the product term, $W'YZ'$.

WX \ YZ	00	01	11	10
00				1
01				1
11			1	1
10	1	1	1	1

Groupings are shown with dashed lines: a red box around the 1s in the 10 row (labeled WX'), a blue box around the 1s in the 11 and 10 rows for columns 11 and 10 (labeled WY), and a red box around the 1s in the 00 and 01 rows for column 10 (labeled YZ').

Here, we got three prime implicants WX' , WY & YZ' . All these prime implicants are essential because of following reasons.

- Two ones (m8 & m9) of fourth row grouping are not covered by any other groupings. Only fourth row grouping covers those two ones.
- Single one (m15) of square shape grouping is not covered by any other groupings. Only the square shape grouping covers that one.
- Two ones (m2 & m6) of fourth column grouping are not covered by any other groupings. Only fourth column grouping covers those two ones.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=wjM2RDG5yTI>

<https://www.youtube.com/watch?v=CpsJoAwreqo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (78)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L 08

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : I- BOOLEAN ALGEBRA AND LOGIC GATES

Date of Lecture:

Topic of Lecture: Minimization using Tabulation method

Introduction :

Quine-McClukey tabular method is a tabular method based on the concept of prime implicants. We know that prime implicant is a product orsumorsum term, which can't be further reduced by combining with any other product orsumorsum terms of the given Boolean function.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

:

Procedure of Quine-McCluskey Tabular Method

Follow these steps for simplifying Boolean functions using Quine-McClukey tabular method.

Step 1 – Arrange the given min terms in an ascending order and make the groups based on the number of ones present in their binary representations. So, there will be at most 'n+1' groups if there are 'n' Boolean variables in a Boolean function or 'n' bits in the binary equivalent of min terms.

Step 2 – Compare the min terms present in successive groups. If there is a change in only one-bit position, then take the pair of those two min terms. Place this symbol '_' in the differed bit position and keep the remaining bits as it is.

Step 3 – Repeat step2 with newly formed terms till we get all prime implicants.

Step 4 – Formulate the prime implicant table. It consists of set of rows and columns. Prime implicants can be placed in row wise and min terms can be placed in column wise. Place '1' in the cells corresponding to the min terms that are covered in each prime implicant.

Step 5 – Find the essential prime implicants by observing each column. If the min term is covered only by one prime implicant, then it is essential prime implicant. Those essential prime implicants will be part of the simplified Boolean function.

Step 6 – Reduce the prime implicant table by removing the row of each essential prime implicant and the columns corresponding to the min terms that are covered in that essential prime implicant. Repeat step 5 for Reduced prime implicant table. Stop this process when all min terms of given Boolean function are over.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=l1jgq0R5EwQ>

<https://www.youtube.com/watch?v=POULm0cbIhE>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design”, Pearson Education- V Edition,2013. Page No (97)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE

(An Autonomous Institution)

(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu



LECTURE HANDOUTS

L 09

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : I- BOOLEAN ALGEBRA AND LOGIC GATES

Date of Lecture:

Topic of Lecture: Minimization using Tabulation method

Introduction :

Quine-McClukey tabular method is a tabular method based on the concept of prime implicants. We know that prime implicant is a product or sum term, which can't be further reduced by combining with any other product or sum terms of the given Boolean function.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Example

Let us simplify the following Boolean function, $f(W,X,Y,Z)=\sum m(2,6,8,9,10,11,14,15)$ using Quine-McClukey tabular method.

The given Boolean function is in sum of min terms form. It is having 4 variables W, X, Y & Z. The given min terms are 2, 6, 8, 9, 10, 11, 14 and 15. The ascending order of these min terms based on the number of ones present in their binary equivalent is 2, 8, 6, 9, 10, 11, 14 and 15. The following table shows these min terms and their equivalent binary representations.

Group Name	Min terms	W	X	Y	Z
GA1	2	0	0	1	0
	8	1	0	0	0
GA2	6	0	1	1	0
	9	1	0	0	1

	10	1	0	1	0
GA3	11	1	0	1	1
	14	1	1	1	0
GA4	15	1	1	1	1

The given min terms are arranged into 4 groups based on the number of ones present in their binary equivalents. The following table shows the possible merging of min terms from adjacent groups.

Group Name	Min terms	W	X	Y	Z
GB1	2,6	0	-	1	0
	2,10	-	0	1	0
	8,9	1	0	0	-
	8,10	1	0	-	0
GB2	6,14	-	1	1	0
	9,11	1	0	-	1
	10,11	1	0	1	-
	10,14	1	-	1	0
GB3	11,15	1	-	1	1
	14,15	1	1	1	-

The min terms, which are differed in only one-bit position from adjacent groups are merged. That differed bit is represented with this symbol, '-'. In this case, there are three groups and each group contains combinations of two min terms. The following table shows the possible merging of min term pairs from adjacent groups.

Group Name	Min terms	W	X	Y	Z
GB1	2,6,10,14	-	-	1	0
	2,10,6,14	-	-	1	0
	8,9,10,11	1	0	-	-
	8,10,9,11	1	0	-	-
GB2	10,11,14,15	1	-	1	-
	10,14,11,15	1	-	1	-

The successive groups of min term pairs, which are differed in only one-bit position are merged. That differed bit is represented with this symbol, '-'. In this case, there are two groups and each group contains combinations of four min terms. Here, these combinations of 4 min terms are available in two rows. So, we can remove the repeated rows. The reduced table after removing the redundant rows is shown below.

Group Name	Min terms	W	X	Y	Z
GC1	2,6,10,14	-	-	1	0
	8,9,10,11	1	0	-	-
GC2	10,11,14,15	1	-	1	-

Further merging of the combinations of min terms from adjacent groups is not possible, since they are differed in more than one-bit position. There are three rows in the above table. So, each row will give one prime implicant. Therefore, the prime implicants are YZ' , WX' & WY .

The prime implicant table is shown below.

Min terms / Prime Implicants	2	6	8	9	10	11	14	15
YZ'	1	1			1		1	
WX'			1	1	1	1		

WY					1	1	1	1
----	--	--	--	--	---	---	---	---

The prime implicants are placed in row wise and min terms are placed in column wise. 1s are placed in the common cells of prime implicant rows and the corresponding min term columns.

The min terms 2 and 6 are covered only by one prime implicant YZ' . So, it is an essential prime implicant. This will be part of simplified Boolean function. Now, remove this prime implicant row and the corresponding min term columns. The reduced prime implicant table is shown below.

Min terms / Prime Implicants	8	9	11	15
WX'	1	1	1	
WY			1	1

The min terms 8 and 9 are covered only by one prime implicant WX' . So, it is an essential prime implicant. This will be part of simplified Boolean function. Now, remove this prime implicant row and the corresponding min term columns. The reduced prime implicant table is shown below.

Min terms / Prime Implicants	15
WY	1

The min term 15 is covered only by one prime implicant WY . So, it is an essential prime implicant. This will be part of simplified Boolean function.

In this example problem, we got three prime implicants and all the three are essential. Therefore, the simplified Boolean function is

$$f_{W,X,Y,Z} = YZ' + WX' + WY.$$

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=l1jgq0R5EwQ>
<https://www.youtube.com/watch?v=POULm0cbIhE>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (103)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 15

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : II- COMBINATIONAL LOGIC Date of Lecture:

Topic of Lecture:

Encoder, Decoder

Introduction :

Encoder is a combinational circuit that has '2n' input lines and maximum of n output lines.

Decoder is a combinational circuit that has 'n' input lines and maximum of 2n output lines. One of these outputs will be active High based on the combination of inputs present, when the decoder is enabled. That means decoder detects a particular code. The outputs of the decoder are nothing but the min terms of 'n' input variables lines, when it is enabled

Prerequisite knowledge for Complete understanding and learning of Topic:

- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

ENCODER

An encoder is a digital circuit that performs the inverse operation of a decoder. An encoder has 2n (or fewer) input lines and n output lines.

The encoder can be implemented with OR gates whose inputs are determined directly from the truth table. Output z is equal to 1 when the input octal digit is 1, 3, 5, or 7. Output y is 1 for octal digits 2, 3, 6, or 7, and output x is 1 for digits 4, 5, 6, or 7. These conditions can be expressed by the following Boolean output functions:

$$z = D1 + D3 + D5 + D7$$

$$y = D2 + D3 + D6 + D7$$

$$x = D4 + D5 + D6 + D7$$

The encoder can be implemented with three OR gates.

Truth Table of an Octal-to-Binary Encoder

Inputs								Outputs		
D_0	D_1	D_2	D_3	D_4	D_5	D_6	D_7	x	y	z
1	0	0	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0
0	0	0	1	0	0	0	0	0	1	1
0	0	0	0	1	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	1	0
0	0	0	0	0	0	0	1	1	1	1

The encoder defined in Table 4.7 has the limitation that only one input can be active at any given time. If two inputs are active simultaneously, the output produces an undefined combination. For example, if D_3 and D_6 are 1 simultaneously, the output of the encoder will be 111 because all three outputs are equal to 1. The output 111 does not present either binary 3 or binary 6.

Decoder

Decoder is a combinational circuit that has 'n' input lines and maximum of 2^n output lines. One of these outputs will be active High based on the combination of inputs present, when the decoder is enabled. That means decoder detects a particular code. The outputs of the decoder are nothing but the min terms of 'n' input variables lines, when it is enabled.

2 to 4 Decoder

Let 2 to 4 Decoder has two inputs A_1 & A_0 and four outputs Y_3 , Y_2 , Y_1 & Y_0 . The block diagram of 2 to 4 decoder is shown in the following figure.

Enable	Inputs		Outputs			
E	A_1	A_0	Y_3	Y_2	Y_1	Y_0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

From Truth table, we can write the Boolean functions for each output as

$$Y_3 = E \cdot A_1 \cdot A_0$$

$$Y_2 = E \cdot A_1 \cdot A_0'$$

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=t3Ed13z9uz8>

<https://www.youtube.com/watch?v=4kgPMT9k3bg>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design” , Pearson Education- V Edition,2013. Page No (150)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 16

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : II- COMBINATIONAL LOGIC Date of Lecture:

Topic of Lecture:

Multiplexer/ Demultiplexer

Introduction :

A multiplexer also known as a data selector, is a device that selects between several analog or digital input signals and forwards it to a single output line.

A demultiplexer (or demux) is a device that takes a single input line and routes it to one of several digital output lines. A demultiplexer of 2^n outputs has n select lines, which are used to select which output line to send the input.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Multiplexer

A multiplexer is a combinational circuit that selects binary information from one of many input lines and directs it to a single output line. The selection of a particular input line is controlled by a set of selection lines. Normally, there are 2^n input lines and n selection lines whose bit combinations determine which input is selected.

A two-to-one-line multiplexer connects one of two 1-bit sources to a common destination, as shown in Fig. 4.24 .

The outputs of the AND gates are applied to a single OR gate that provides the one-line output. The function table lists the input that is passed to the output for each combination of the binary selection values. To demonstrate the operation of the circuit, consider the case when $S_1S_0 = 10$. The AND gate associated with input I_2 has two of its inputs equal to 1 and the third input connected to I_2 . The other three AND gates have at least one input equal to 0, which makes

their outputs equal to 0. The output of the OR gate is now equal to the value of I_2 , providing a path

from the selected input to the output. A multiplexer is also called a data selector, since it selects one of many inputs and steers the binary information to the output line. The size of a multiplexer is specified by

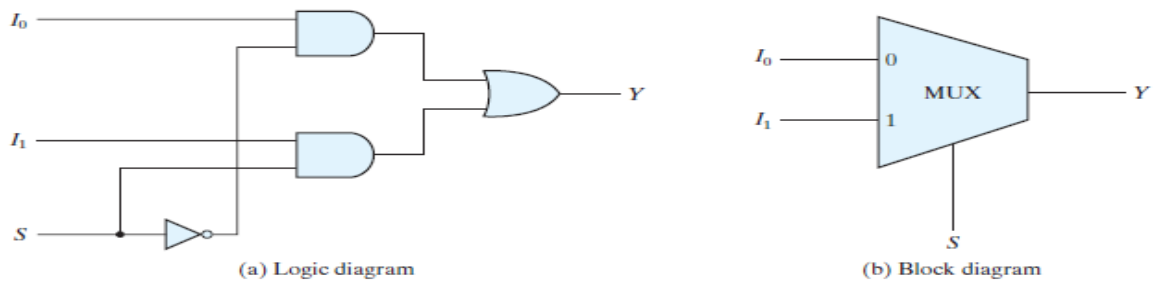


FIGURE 4.24
Two-to-one-line multiplexer

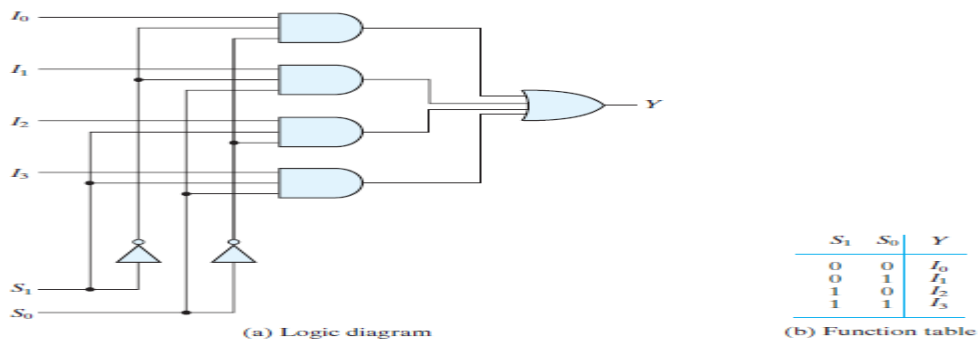


FIGURE 4.25
Four-to-one-line multiplexer

the number 2^n of its data input lines and the single output line. The n selection lines are implied from the 2^n data lines. As in decoders, multiplexers may have an enable input to control the operation of the unit. When the enable input is in the inactive state, the outputs are disabled, and when it is in the active state, the circuit functions as a normal multiplexer.

Video Content / Details of website for further learning (if any):

https://www.youtube.com/watch?v=HU31Wh-4_K8

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (158)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 17

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : II - COMBINATIONAL LOGIC

Date of Lecture:

Topic of Lecture:

Introduction to HDL

Introduction :

A hardware description language (HDL) is a specialized computer language used to describe the structure and behavior of electronic circuits, and most commonly, digital logic circuits.

It also allows for the synthesis of an HDL description into a netlist which can then be placed and routed to produce the set of masks used to create an integrated circuit.

A hardware description language looks much like a programming language such as C or ALGOL; it is a textual description consisting of expressions, statements and control structures.

Prerequisite knowledge for Complete understanding and learning of Topic:

- VLSI

In computer engineering, a hardware description language (**HDL**) is a specialized computer language used to describe the structure and behavior of electronic circuits, and most commonly, digital logic circuits.

HDL is a language that describes the hardware of digital systems in a textual form. It resembles a programming language, but is specifically oriented to describing hardware structures and behaviors.

The main difference with the traditional programming languages is HDL's representation of extensive parallel operations whereas traditional ones represents mostly serial operations. The most common use of a HDL is to provide an alternative to schematics.

When a language is used for the above purpose (i.e. to provide an alternative to schematics), it is referred to as a structural description in which the language describes an interconnection of components.

Models for each of the primitive components are required. If an HDL is used, then these models can also be written in the HDL providing a more uniform, portable representation for simulation input.

HDL can be used to represent logic diagrams, Boolean expressions, and other more complex digital circuits.

There are two applications of HDL processing: Simulation and Synthesis

- To simulate a digital system
- Design is first described in HDL

There are two standard HDL's that are supported by IEEE.

- VHDL (Very-High-Speed Integrated Circuits Hardware Description Language)
- Verilog HDL

A module is the building block in Verilog. It is declared by the keyword module and is always terminated by the keyword end module. Each statement is terminated with a semicolon, but there is no semi-colon after end module

HDL Example

```
module smpl_circuit(A,B,C,x,y);  
input A,B,C;  
output x,y;  
wire e;  
and g1(e,A,B);  
not g2(y,C);  
org3(x,e,y);  
end module
```

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=yFRICQtxrj4>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (154)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : II -COMBINATIONAL LOGIC Date of Lecture:

Topic of Lecture:

HDL models of combinational circuits

Introduction :

The Verilog HDL model of a combinational circuit can be described in any one of the following modeling styles, ü Gate level modeling-using instantiations of predefined and user defined primitive gates. ü Dataflow modeling using continuous assignment with the keyword assign.

Prerequisite knowledge for Complete understanding and learning of Topic:

- VLSI

Logic circuits for digital systems may be combinational or sequential. n A combinational circuit consists of input variables, logic gates, and output variables.



Fig. 4-1 Block Diagram of Combinational Circuit

A module can be described in any one (or a combination) of the following modeling techniques.

Gate-level modeling using instantiation of primitive gates and user defined modules. This describes the circuit by specifying the gates and how they are connected with each other.

Dataflow modeling using continuous assignment statements with the keyword assign. This is mostly used for describing combinational circuits.

Behavioral modeling using procedural assignment statements with keyword always. This is used to describe digital systems at a higher level of abstraction.

There are two basic types of design methodologies.

Top down: In top-down design, the top level block is defined and then sub-blocks necessary to build the top level block are identified.

Bottom up: Here the building blocks are first identified and then combine to build the top level block.

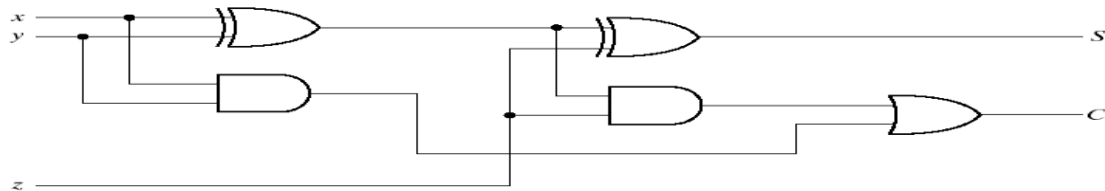


Fig. 4-8 Implementation of Full Adder with Two Half Adders and an OR Gate

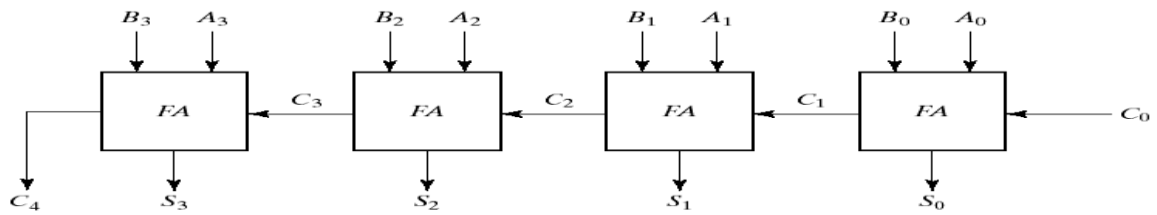


Fig. 4-9 4-Bit Adder

//Gate-level hierarchical description of 4-bit adder

```
module halfadder (S,C,x,y);
```

```
input x,y;
```

```
output S,C;
```

```
//Instantiate primitive gates xor (S,x,y);
```

```
and (C,x,y);
```

```
endmodule module fulladder (S,C,x,y,z);
```

```
input x,y,z; output S,C;
```

```
wire S1,D1,D2;
```

```
//Outputs of first XOR and two AND gates
```

```
//Instantiate the half adders halfadder HA1(S1,D1,x,y), HA2(S,D2,S1,z); or g1(C,D2,D1);
```

```
endmodule
```

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=CZT7dgX6CTo>

<https://www.youtube.com/watch?v=kgL5UaSVuro>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (130)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : II- COMBINATIONAL LOGIC Date of Lecture:

Topic of Lecture:

Realization of combinational logic using gates

Introduction :

Digital circuits are frequently constructed with NAND or NOR gates rather than with AND and OR gates. NAND and NOR gates are easier to fabricate with electronic components and are the basic gates used in all IC digital logic families. Because of the prominence of NAND and NOR gates in the design of digital circuits, rules and procedures have been developed for the conversion from Boolean functions given in terms of AND, OR, and NOT into equivalent NAND and NOR logic diagrams.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Implement the following Boolean function with NAND gates:

$$F(x, y, z) = (1, 2, 3, 4, 5, 7)$$

The first step is to simplify the function into sum-of-products form. This is done by means of the map of Fig. 3.19 (a), from which the simplified function is obtained:

$$F = xy_ + x_ y + z$$

The two-level NAND implementation is shown in Fig. 3.19 (b) in mixed notation.

Note that input z must have a one-input NAND gate (an inverter) to compensate for the bubble in the second-level gate. An alternative way of drawing the logic diagram is given in Fig. 3.19 (c). Here, all the NAND gates are drawn with the same graphic symbol. The inverter with input z has been removed, but the input variable is complemented and denoted by z_.

The procedure described in the previous example indicates that a Boolean function can be implemented with two levels of NAND gates. The procedure for obtaining the logic diagram from a Boolean function is as follows:

1. Simplify the function and express it in sum-of-products form.
2. Draw a NAND gate for each product term of the expression that has at least two literals. The inputs to each NAND gate are the literals of the term. This procedure produces a group of first-level gates.

3. Draw a single gate using the AND-invert or the invert-OR graphic symbol in the second level, with inputs coming from outputs of first-level gates.

4. A term with a single literal requires an inverter in the first level. However, if the single literal is

complemented, it can be connected directly to an input of the second level NAND gate.

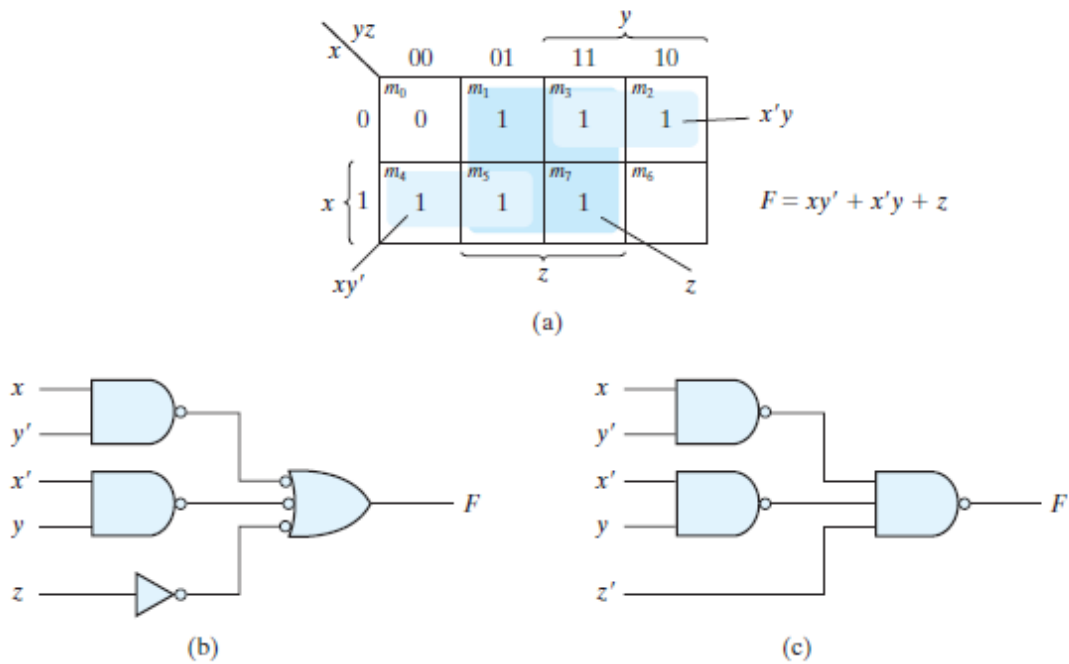


FIGURE 3.19
Solution to Example 3.9

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=wAGoIaF5qWM&vl=hi>

https://www.youtube.com/watch?v=ykfGh4Ye_Lc

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (92)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 11

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : II- COMBINATIONAL LOGIC Date of Lecture:

Topic of Lecture:

Design of combinational circuits, Adder , Subtractor

Introduction :

The basic arithmetic circuits like Binary adder and Binary subtractor. These circuits can be operated with binary values 0 and 1.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Binary Adder

The most basic arithmetic operation is addition. The circuit, which performs the addition of two binary numbers is known as Binary adder. First, let us implement an adder, which performs the addition of two bits.

Half Adder

Half adder is a combinational circuit, which performs the addition of two binary numbers A and B are of single bit. It produces two outputs sum, S & carry, C.

The Truth table of Half adder is shown below.

Inputs		Outputs	
A	B	C	S
0	0	0	0
0	1	0	1
1	0	0	1

1	1	1	0
---	---	---	---

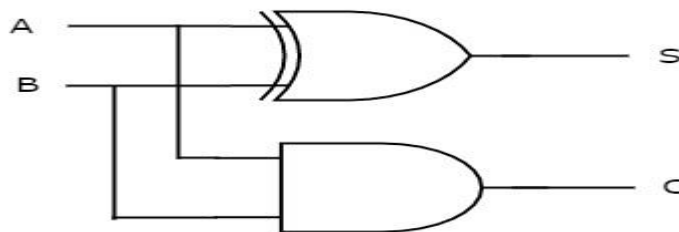
Let, sum, S is the Least significant bit and carry, C is the Most significant bit of the resultant sum. For first three combinations of inputs, carry, C is zero and the value of S will be either zero or one based on the number of ones present at the inputs. But, for last combination of inputs, carry, C is one and sum, S is zero, since the resultant sum is two.

From Truth table, we can directly write the Boolean functions for each output as

$$S = A \oplus B$$

$$C = A \cdot B$$

We can implement the above functions with 2-input Ex-OR gate & 2-input AND gate. The circuit diagram of Half adder is shown in the following figure.



In the above circuit, a two input Ex-OR gate & two input AND gate produces sum, S & carry, C respectively. Therefore, Half-adder performs the addition of two bits.

Binary Subtractor

The circuit, which performs the subtraction of two binary numbers is known as Binary subtractor. We can implement Binary subtractor in following two methods.

- Cascade Full subtractors
- 2's complement method

In first method, we will get an n-bit binary subtractor by cascading 'n' Full subtractors. So, first you can implement Half subtractor and Full subtractor, similar to Half adder & Full adder. Then, you can implement an n-bit binary subtractor, by cascading 'n' Full subtractors. So, we will be having two separate circuits for binary addition and subtraction of two binary numbers.

In second method, we can use same binary adder for subtracting two binary numbers just by doing some modifications in the second input. So, internally binary addition operation takes place but, the output is resultant subtraction.

We know that the subtraction of two binary numbers A & B can be written as,

$$A - B = A + (2's \text{ complement of } B)$$

$$\Rightarrow A - B = A + (1's \text{ complement of } B) + 1$$

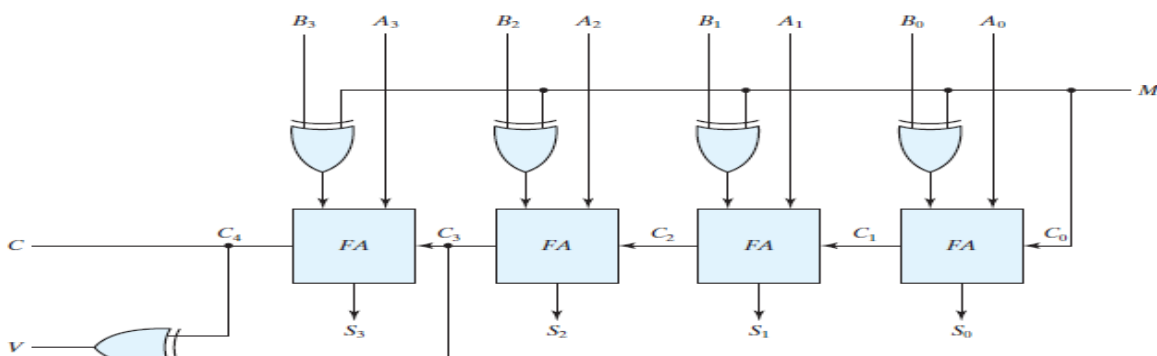


FIGURE 4.13 Four-bit adder-subtractor (with overflow detection)

The circuit for subtracting $A - B$ consists of an adder with inverters placed between each data input B and the corresponding input of the full adder. The input carry C_0 must be equal to 1 when subtraction is performed. The operation thus performed becomes A , plus the 1's complement of B , plus 1. This is equal to A plus the 2's complement of B . For unsigned numbers, that gives $A - B$ if $A \geq B$ or the 2's complement of $B - A$ if $A < B$. For signed numbers, the result is $A - B$, provided that there is no overflow.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=c1CQ8J-yk7I>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (133)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 12

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : II- COMBINATIONAL LOGIC

Date of Lecture:

Topic of Lecture:

Parallel adder Subtractor, Carry look ahead adder

Introduction :

Addition of n-bit numbers requires a chain of n full adders or a chain of one-half adder and n-1 full adders. In the former case, the input carry to the least significant position is fixed at 0. The interconnection of four full-adder (FA) circuits to provide a four-bit binary ripple carry adder

Prerequisite knowledge for Complete understanding and learning of Topic:

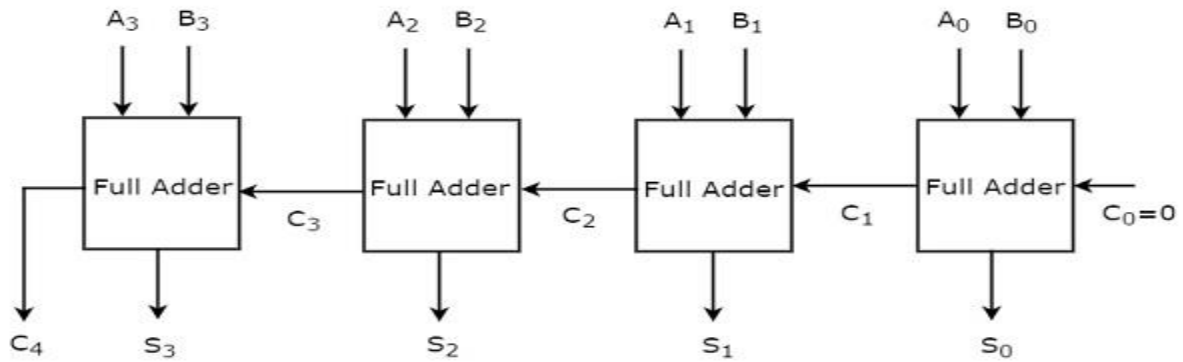
- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

4-bit Binary Adder

The 4-bit binary adder performs the addition of two 4-bit numbers. Let the 4-bit binary numbers, $A=A_3A_2A_1A_0$ and $B=B_3B_2B_1B_0$. We can implement 4-bit binary adder in one of the two following ways.

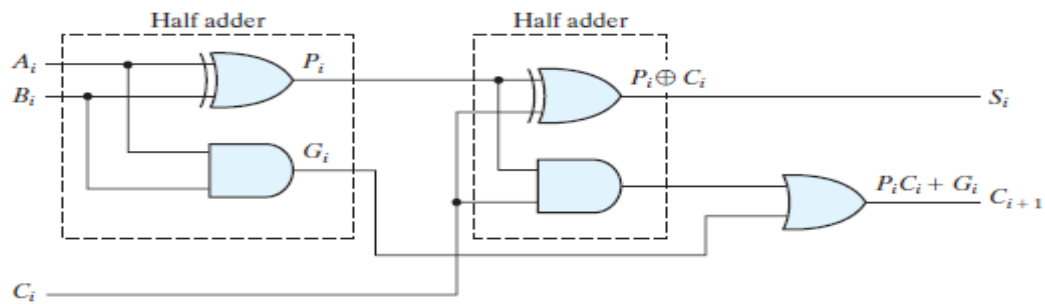
- Use one Half adder for doing the addition of two Least significant bits and three Full adders for doing the addition of three higher significant bits.
- Use four Full adders for uniformity. Since, initial carry C_{in} is zero, the Full adder which is used for adding the least significant bits becomes Half adder.

The block diagram of 4-bit binary adder is



Carry Propagation

The addition of two binary numbers in parallel implies that all the bits of the augend and addend are available for computation at the same time. As in any combinational circuit, the signal must propagate through the gates before the correct output sum is available in the output terminals. The total propagation time is equal to the propagation delay of a typical gate, times the number of gate levels in the circuit.



$$P_i = A_i \oplus B_i$$

$$G_i = A_i B_i$$

the output sum and carry can respectively be expressed as

$$S_i = P_i \oplus C_i$$

$$C_{i+1} = G_i + P_i C_i$$

G_i is called a carry generate, and it produces a carry of 1 when both A_i and B_i are 1, regardless of the input carry C_i . P_i is called a carry propagate, because it determines whether a carry into stage i will propagate into stage $i + 1$ (i.e., whether an assertion of C_i will propagate to an assertion of C_{i+1}).

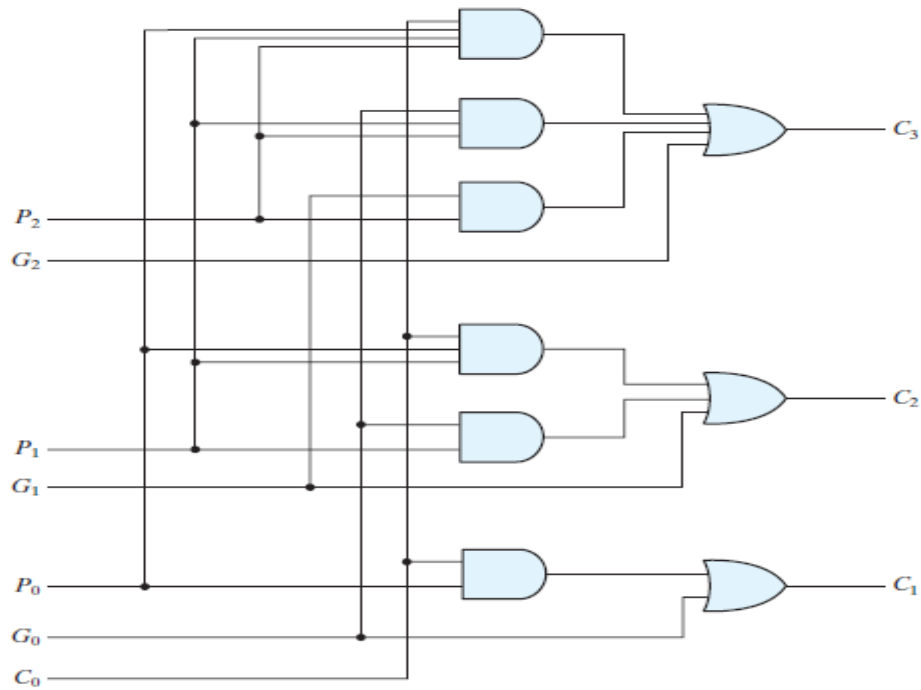
We now write the Boolean functions for the carry outputs of each stage and substitute the value of each C_i from the previous equations:

$$C_0 = \text{input carry}$$

$$C_1 = G_0 + P_0 C_0$$

$$C_2 = G_1 + P_1 C_1 = G_1 + P_1 G_0 + P_1 P_0 C_0 = G_1 + P_1 G_0 + P_1 P_0 C_0$$

$$C_3 = G_2 + P_2 C_2 = G_2 + P_2 G_1 + P_2 P_1 G_0 = P_2 P_1 P_0 C_0$$



Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=XcPwcpFEiy4>

<https://www.youtube.com/watch?v=bJ53XErKY8>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (138)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 13

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : II- COMBINATIONAL LOGIC Date of Lecture:

Topic of Lecture:

Magnitude Comparator, Code conversion

Introduction :

The comparison of two numbers is an operation that determines whether one number is greater than, less than, or equal to the other number. A magnitude comparator is a combinational circuit that compares two numbers A and B and determines their relative magnitudes.

To convert from binary code A to binary code B, the input lines must supply the bit combination of elements as specified by code A and the output lines must generate the corresponding bit combination of code B.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Magnitude Comparator :

The comparison of two numbers is an operation that determines whether one number is greater than, less than, or equal to the other number. A magnitude comparator is a combinational circuit that compares two numbers A and B and determines their relative magnitudes. The outcome of the comparison is specified by three binary variables that indicate whether $A > B$, $A = B$, or $A < B$.

The algorithm is a direct application of the procedure a person uses to compare the relative magnitudes of two numbers. Consider two numbers, A and B, with four digits each. Write the coefficients of the numbers in descending order of significance:

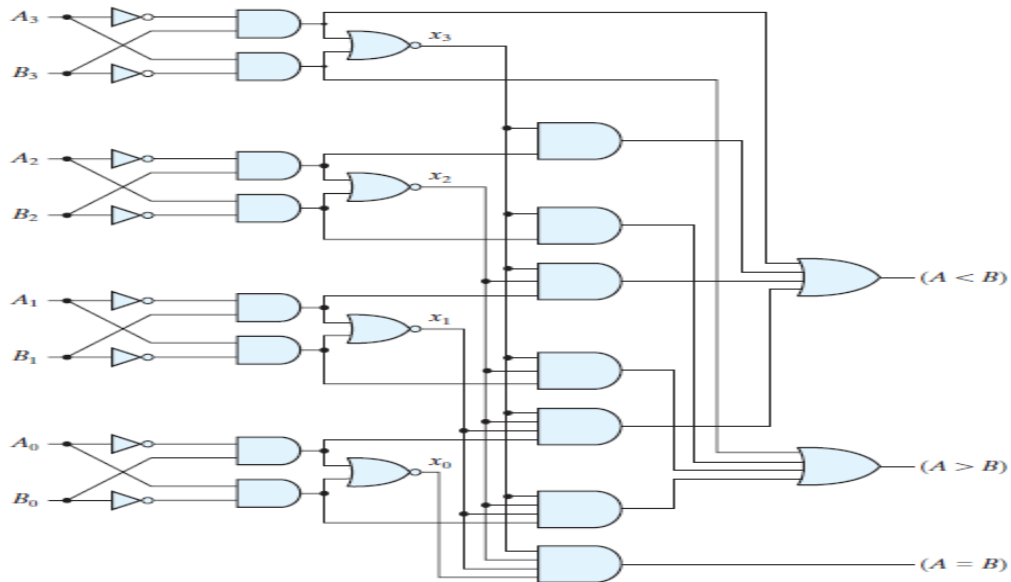
$$A = A_3 A_2 A_1 A_0$$

$$B = B_3 B_2 B_1 B_0$$

the numbers are binary, the digits are either 1 or 0, and the equality of each pair of bits can be expressed logically with an exclusive-NOR function as

$$x_i = A_i B_i + \bar{A}_i \bar{B}_i \text{ for } i = 0, 1, 2, 3$$

where $x_i = 1$ only if the pair of bits in position i are equal (i.e., if both are 1 or both are 0).



Code Converters:

The bit combinations assigned to the BCD and excess-3 codes are listed in Table 1.5. Since each code uses four bits to represent a decimal digit, there must be four input variables and four output variables. We designate the four input binary variables by the symbols A, B, C, and D, and the four output variables by w, x, y, and z. The truth table relating the input and output variables is shown in Table 4.2

Table 4.2
Truth Table for Code Conversion Example

Input BCD				Output Excess-3 Code			
A	B	C	D	w	x	y	z
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

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=BhUUmzb76P0>

<https://www.youtube.com/watch?v=3BiSCqXXGo0>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (148)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : II- COMBINATIONAL LOGIC

Date of Lecture:

Topic of Lecture:

Parity generator and checker

Introduction :

The parity generating technique is one of the most widely used error detection techniques for the data transmission. In digital systems, when binary data is transmitted and processed , data may be subjected to noise so that such noise can alter 0s (of data bits) to 1s and 1s to 0s.

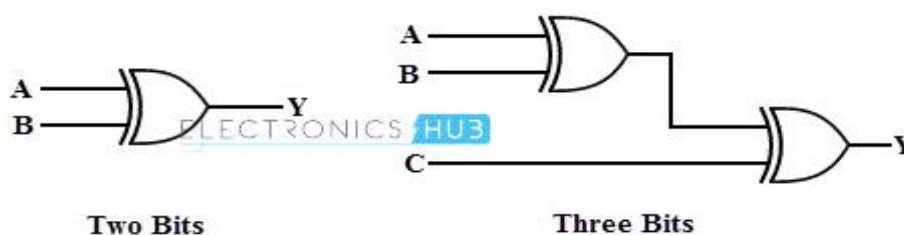
Prerequisite knowledge for Complete understanding and learning of Topic:

- Boolean algebra
- Boolean postulates and laws
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Parity generator and checker

A parity generator is a combinational logic circuit that generates the parity bit in the transmitter. On the other hand, a circuit that checks the parity in the receiver is called parity checker. A combined circuit or devices of parity generators and parity checkers are commonly used in digital systems to detect the single bit errors in the transmitted data word.

To produce two bits sum, one Ex-OR gate is sufficient whereas for adding three bits two Ex-OR gates are required as shown in below figure.

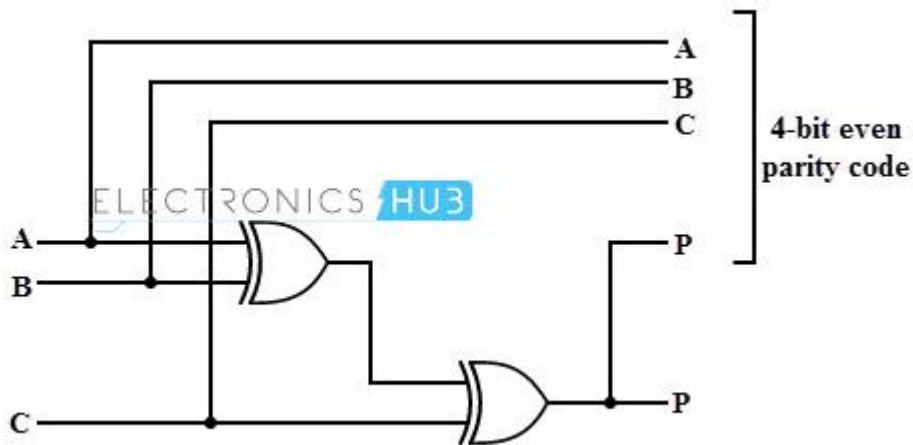


Even Parity Generator

Let us assume that a 3-bit message is to be transmitted with an even parity bit. Let the three inputs A, B and C are applied to the circuits and output bit is the parity bit P. The total number of 1s must be even, to generate the even parity bit P.

The figure below shows the truth table of even parity generator in which 1 is placed as parity bit in order to make all 1s as even when the number of 1s in the truth table is odd.

3-bit message			Even parity bit generator (P)
A	B	C	Y
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



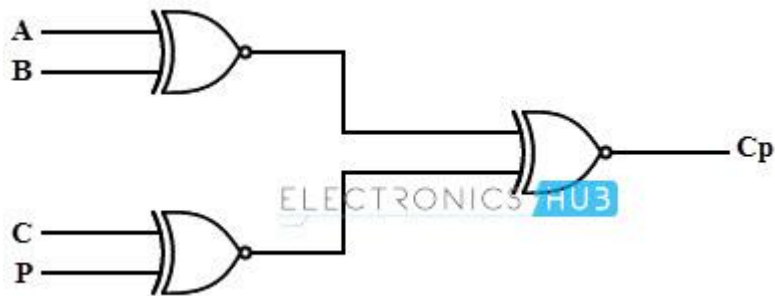
Parity Check

When a parity error occurs, the 'sum even' output goes low and 'sum odd' output goes high. If this logic circuit is used as an odd parity checker, the number of input bits should be odd, but if an error occurs the 'sum odd' output goes low and 'sum even' output goes high.

Odd Parity Checker

The below figure shows the truth table for odd parity generator where PEC = 1 if the 4-bit message received consists of even number of 1s (hence the error occurred) and PEC = 0 if the message contains odd number of 1s (that means no error).

4-bit received message				Parity error check C_p
A	B	C	P	
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	0
1	0	0	1	1
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1



Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=ke-kpusTSvs>
<https://www.youtube.com/watch?v=789NN7c1RdI>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (156)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : III- SYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Shift registers - Types – Universal shift registers

Introduction :

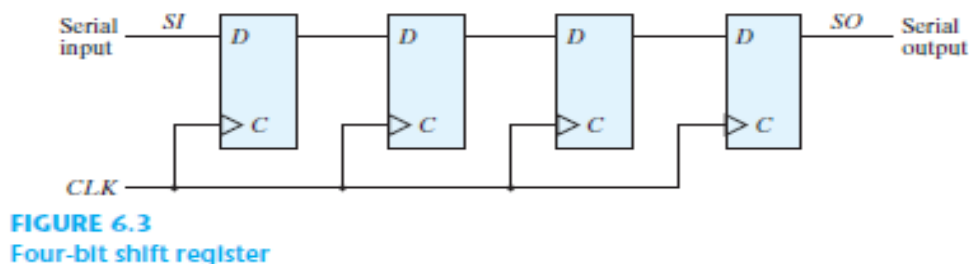
A register is a group of flip-flops, each one of which shares a common clock and is capable of storing one bit of information. An n -bit register consists of a group of n flip-flops capable of storing n bits of binary information.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Shift registers

A register capable of shifting the binary information held in each cell to its neighboring cell, in a selected direction, is called a shift register.



The simplest possible shift register is one that uses only flip-flops, as shown in Fig. 6.3 . The output of a given flip-flop is connected to the D input of the flip-flop at its right. This shift register is unidirectional (left-to-right). Each clock pulse shifts the contents of the register one bit position to the right. The configuration does not support a left shift. The serial input determines what goes into the leftmost flip-flop during the shift. The serial output is taken from the output of the rightmost flip-flop. Sometimes it is necessary to control the shift so that it occurs only with certain pulses, but not with others.

Universal Shift Register

If the flip-flop outputs of a shift register are accessible, then information entered serially by shifting can be taken out in parallel from the outputs of the flip-flops. If a parallel load capability is added to a shift register, then data entered in parallel can be taken out in serial fashion by shifting the data stored in the register.

1. A clear control to clear the register to 0.
2. A clock input to synchronize the operations.
3. A shift-right control to enable the shift-right operation and the serial input and output lines associated with the shift right.
4. A shift-left control to enable the shift-left operation and the serial input and output lines associated with the shift left.
5. A parallel-load control to enable a parallel transfer and the n input lines associated with the parallel transfer.
6. n parallel output lines.
7. A control state that leaves the information in the register unchanged in response to the clock. Other shift registers may have only some of the preceding functions, with at least one shift operation.

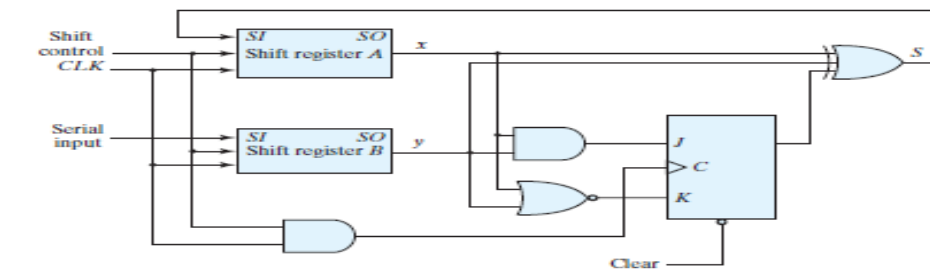


FIGURE 6.6
Second form of serial adder

Video Content / Details of website for further learning (if any):

- <https://www.youtube.com/watch?v=unorn9n-UpE>
- <https://www.youtube.com/watch?v=AEGzpMIOsvc>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design”, Pearson Education- V Edition,2013. Page No (258)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 25

CS

II / III

Course Name with Code : 19GES34/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : III- SYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Ring counter

Introduction :

Counters can be designed to generate any desired sequence of states. A divide-by- N counter (also known as a modulo- N counter) is a counter that goes through a repeated sequence of N states. The sequence may follow the binary count or may be any other arbitrary sequence. Counters are used to generate timing signals to control the sequence of operations in a digital system. Counters can also be constructed by means of shift registers.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Ring Counter

Timing signals that control the sequence of operations in a digital system can be generated by a shift register or by a counter with a decoder. A ring counter is a circular shift register with only one flip-flop being set at any particular time; all others are cleared. The single bit is shifted from one flip-flop to the next to produce the sequence of timing signals. Figure 6.17 (a) shows a four-bit shift register connected as a ring counter. The initial value of the register is 1000 and requires Preset/Clear flip-flops. The single bit is shifted right with every clock pulse and circulates back from T3 to T0. Each flip-flop is in the 1 state once every four clock cycles and produces one of the four timing signals shown in Fig. 6.17 (b). Each output becomes a 1 after the negative-edge transition of a clock pulse and remains 1 during the next clock cycle.

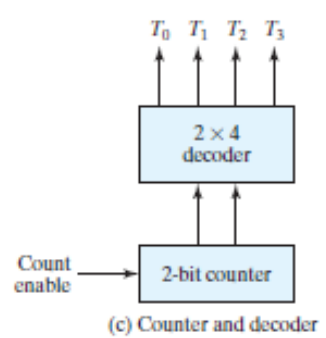
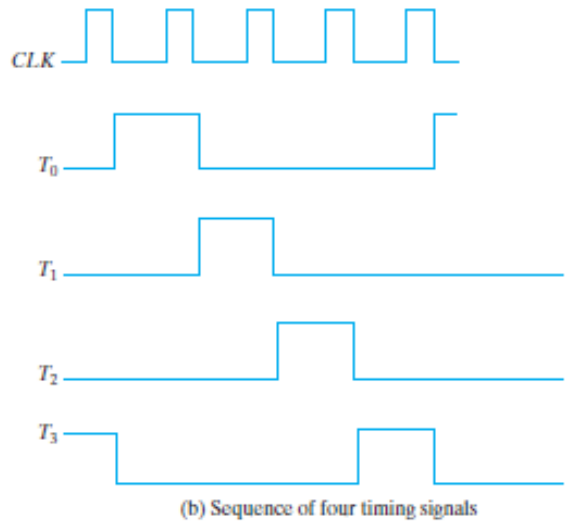
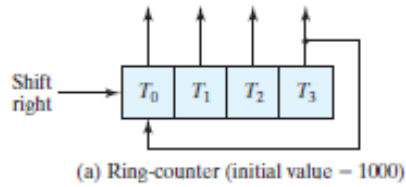


FIGURE 6.17
Generation of timing signals

Video Content / Details of website for further learning (if any):
<https://www.youtube.com/watch?v=yOW-JsJL1Ks>

Important Books/Journals for further learning including the page nos.:
 Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (280)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 26

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : III- SYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Johnson Counter

Introduction :

Counters can be designed to generate any desired sequence of states. A divide-by- N counter (also known as a modulo- N counter) is a counter that goes through a repeated sequence of N states. The sequence may follow the binary count or may be any other arbitrary sequence. Counters are used to generate timing signals to control the sequence of operations in a digital system. Counters can also be constructed by means of shift registers.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Johnson Counter

A k -bit ring counter circulates a single bit among the flip-flops to provide k distinguishable states. The number of states can be doubled if the shift register is connected as a switch-tail ring counter. A switch-tail ring counter is a circular shift register with the complemented output of the last flip-flop connected to the input of the first flip-flop. Figure 6.18 (a) shows such a shift register. The circular connection is made from the complemented output of the rightmost flip-flop to the input of the leftmost flip-flop.

A Johnson counter is a k -bit switch-tail ring counter with 2 k decoding gates to provide outputs for 2 k timing signals. The decoding gates are not shown in Fig. 6.18 , but are specified in the last column of the table. The eight AND gates listed in the table, when connected to the circuit, will complete the construction of the Johnson counter. Since each gate is enabled during one particular state sequence, the outputs of the gates generate eight timing signals in succession.

The decoding of a k -bit switch-tail ring counter to obtain 2 k timing signals follows a regular pattern. The all-0's state is decoded by taking the complement of the two extreme flip-flop outputs. The all-1's state is decoded by taking the normal outputs of the two extreme flip-flops.

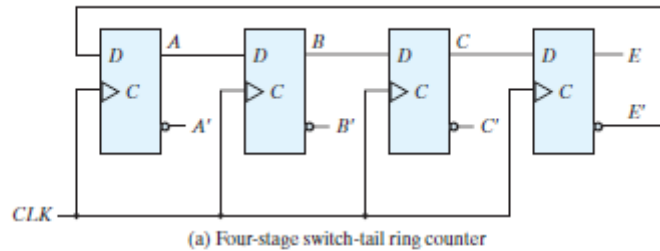
All other states are decoded from an adjacent 1, 0 or 0, 1 pattern in the sequence. For example, sequence 7 has an adjacent 0, 1 pattern in flip-flops B and C . The decoded output is then obtained by

taking the complement of B and the normal output of C, or B'C.

One disadvantage of the circuit in Fig. 6.18 (a) is that if it finds itself in an unused state, it will persist in moving from one invalid state to another and never find its way to a valid state. The difficulty can be corrected by modifying the circuit to avoid this undesirable condition. One correcting procedure is to disconnect the output from flip-flop B that goes to the D input of flip-flop C and instead enable the input of flip-flop C by the function

$$DC = (A + C)B$$

where DC is the flip-flop input equation for the D input of flip-flop C.



Sequence number	Flip-flop outputs				AND gate required for output
	A	B	C	E	
1	0	0	0	0	A'E'
2	1	0	0	0	AB'
3	1	1	0	0	BC'
4	1	1	1	0	CE'
5	1	1	1	1	AE
6	0	1	1	1	A'B
7	0	0	1	1	B'C
8	0	0	0	1	C'E

(b) Count sequence and required decoding

FIGURE 6.18
Construction of a Johnson counter

Video Content / Details of website for further learning (if any):

- <https://www.youtube.com/watch?v=X4mx7J1ckyU>
- <https://www.youtube.com/watch?v=SXbXnfgY6jk>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (282)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 27

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : III- SYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

HDL for Sequential Logic circuits

Introduction :

A Sequential circuit combinational logic circuit that consists of inputs variable (X), logic gates (Computational circuit), and output variable (Z). Combinational circuit produces an output based on input variable only, but Sequential circuit produces an output based on current input and previous input variables

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Structural Description – This is directly equivalent to the schematic of a circuit and is specifically oriented to describing hardware structures using the components of a circuit.

Dataflow Description – This describes a circuit in terms of function rather than structure and is made up of concurrent assignment statements or their equivalent. Concurrent assignments statements are executed concurrently, i.e. in parallel whenever one of the values on the right hand side of the statement changes.

Hierarchical Description – Descriptions that represent circuits using hierarchy have multiple entities, one for each element of the Hierarchy.

Behavioral Description – This refers to a description of a circuit at a level higher than the logic level. This type of description is also referred to as the register transfer's level.

Behavioral Modeling

```
//Behavioral description of 4-to-1 line mux
module mux4x1_bh (i0,i1,i2,i3,select,y);
input i0,i1,i2,i3;
input [1:0] select;
output y;
reg y;
always @(i0 or i1 or i2 or i3 or select)
case (select) 2'b00: y = i0; 2'b01: y = i1; 2'b10: y = i2; 2'b11: y = i3;
end case
end module
```

Video Content / Details of website for further learning (if any):

<https://www.isabekov.pro/four-bit-serial-adder-subtractor/>

<https://unacademy.com/lesson/serial-adder-serial-subtractor-and-serial-adder-subtractor/ZXQER68L>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design”, Pearson Education- V Edition,2013. Page No (261)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 19

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : III- SYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:
Sequential Circuits

Introduction :

A Sequential combinational logic circuit that consists of inputs variable (X), logic gates (Computational circuit), and output variable (Z). Combinational circuit produces an output based on input variable only, but Sequential circuit produces an output based on current input and previous input variables

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Combinational circuit produces an output based on input variable only, but Sequential circuit produces an output based on current input and previous input variables. That means sequential circuits include memory elements which are capable of storing binary information. That binary information defines the state of the sequential circuit at that time. A latch capable of storing one bit of information.

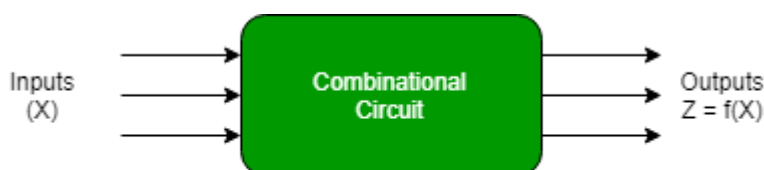


Figure: Combinational Circuits

Types of Sequential Circuits – There are two types of sequential circuit :

1. **Asynchronous sequential circuit** – These circuit **do not use a clock signal** but uses the pulses of the inputs. These circuits are **faster** than synchronous sequential circuits because there is clock pulse and change their state immediately when there is a change in the input signal. We use asynchronous sequential circuits when speed of operation is important and **independent** of internal clock pulse.

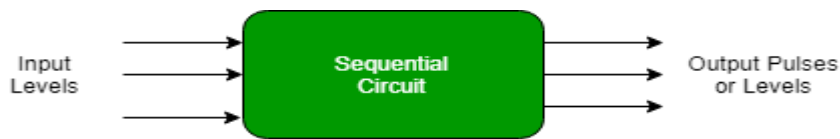


Figure: Asynchronous Sequential Circuit

2. **Synchronous sequential circuit** – These circuit **uses clock signal** and level inputs (or pulsed) (with restrictions on pulse width and circuit propagation). The output pulse is the same duration as the clock pulse for the clocked sequential circuits. Since they wait for the next clock pulse to arrive to perform the next operation, so these circuits are bit **slower** compared to asynchronous. Level output changes state at the start of an input pulse and remains in that until the next input or clock pulse.

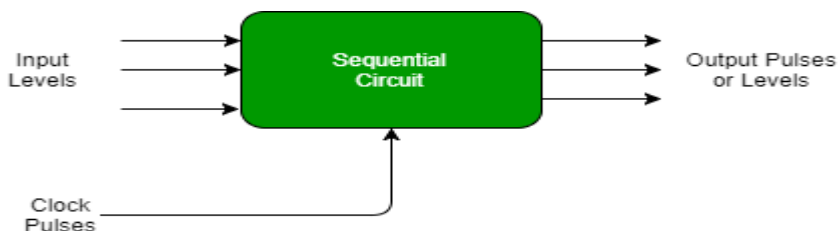


Figure: Synchronous Sequential Circuit

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=xQgLFK0Jv2g>

<https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design”, Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : III- SYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture: Latches, Flip-flops

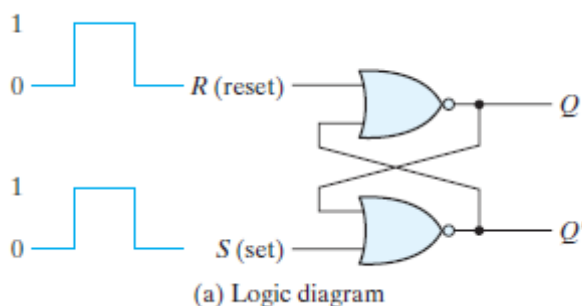
Introduction: Storage elements that operate with signal levels (rather than signal transitions) are referred to as latches; those controlled by a clock transition are flip-flops. Latches are said to be level sensitive devices; flip-flops are edge-sensitive devices.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

SR Latch

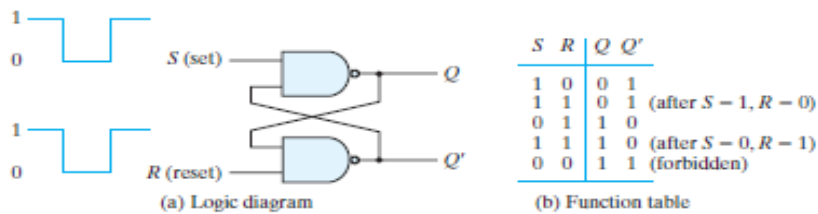
The SR latch is a circuit with two cross-coupled NOR gates or two cross-coupled NAND gates, and two inputs labeled S for set and R for reset. The SR latch constructed with two cross-coupled NOR gates is shown in Figure. The latch has two useful states. When output $Q = 1$ and $Q' = 0$, the latch is said to be in the set state. When $Q = 0$ and $Q' = 1$, it is in the reset state. Outputs Q and Q' are normally the complement of each other. However, when both inputs are equal to 1 at the same time, a condition in which both outputs are equal to 0 (rather than be mutually complementary) occurs. If both inputs are then switched to 0 simultaneously, the device will enter an unpredictable or undefined state or a metastable state. Consequently, in practical applications, setting both inputs to 1 is forbidden.



S	R	Q	Q'
1	0	1	0
0	0	1	0 (after $S = 1, R = 0$)
0	1	0	1
0	0	0	1 (after $S = 0, R = 1$)
1	1	0	0 (forbidden)

(b) Function table

Under normal conditions, both inputs of the latch remain at 0 unless the state has to be changed. The application of a momentary 1 to the S input causes the latch to go to the set state. The S input must go back to 0 before any other changes take place, in order to avoid the occurrence of an undefined next state that results from the forbidden input condition. As shown in the function table of Figure, two input conditions cause the circuit to be in

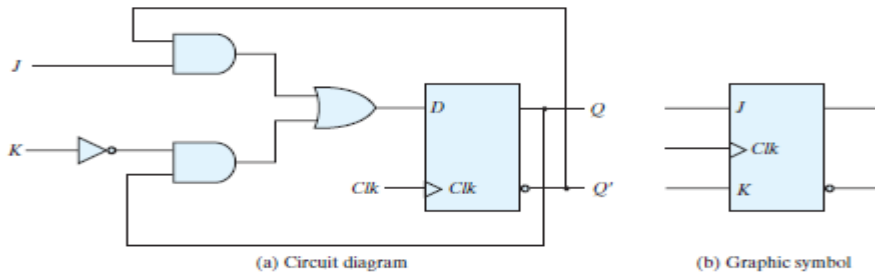


the set state. The first condition ($S = 1, R = 0$) is the action that must be taken by input S to bring the circuit to the set state. Removing the active input from S leaves the circuit in the same state. After both inputs return to 0, it is then possible to shift to the reset state by momentarily applying a 1 to the R input. The 1 can then be removed from R, whereupon the circuit remains in the reset state. Thus, when both inputs S and R are equal to 0, the latch can be in either the set or the reset state, depending on which input was most recently a 1.

SR Latch

Synchronized by a clock signal, the JK flip-flop has two inputs and performs all three operations. The circuit diagram of a JK flip-flop constructed with a D flip-flop and gates is shown in Fig. 5.12 (a). The J input sets the flip-flop to 1, the K input resets it to 0, and when both inputs are enabled, the output is complemented. This can be verified by investigating the circuit applied to the D input:

$$D = JQ' + K'Q$$



When $J = 1$ and $K = 0$, $D = Q' + Q = 1$, so the next clock edge sets the output to 1. When $J = 0$ and $K = 1$, $D = 0$, so the next clock edge resets the output to 0. When both $J = K = 1$ and $D = Q'$, the next clock edge complements the output. When both $J = K = 0$ and $D = Q$, the clock edge leaves the output unchanged.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=tSti91b6qec>
https://www.youtube.com/watch?v=V_JJ_YrPU6Q

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (193, 200)

Course Teacher

Verified by HOD



LECTURE HANDOUTS

L 21

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : III- SYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

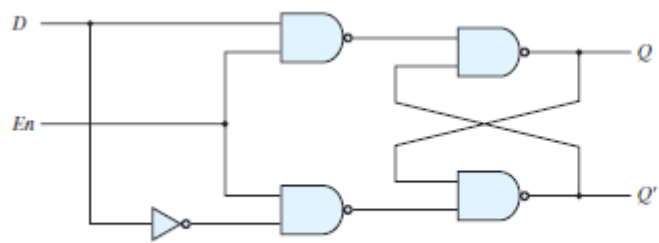
Topic of Lecture:
Flip-flops - D and T- Master-Slave

Introduction : Storage elements that operate with signal levels (rather than signal transitions) are referred to as latches; those controlled by a clock transition are flip-flops. Latches are said to be level sensitive devices; flip-flops are edge-sensitive devices.

- Prerequisite knowledge for Complete understanding and learning of Topic:**
- Latches
 - Logic Gates
 - Canonical forms Sum of product and Product of sum
 - Minimization using Karnaugh map

D Latch

One way to eliminate the undesirable condition of the indeterminate state in the SR latch is to ensure that inputs S and R are never equal to 1 at the same time. This is done in the D latch, shown in Fig. 5.6 . This latch has only two inputs: D (data) and



(a) Logic diagram

En	D	Next state of Q
0	X	No change
1	0	Q = 0; reset state
1	1	Q = 1; set state

(b) Function table

En (enable). The D input goes directly to the S input, and its complement is applied to the R input. As long as the enable input is at 0, the cross-coupled SR latch has both inputs at the 1 level and the circuit cannot change state regardless of the value of D . The D input is sampled when En = 1. If D = 1, the Q output goes to 1, placing the circuit in the set state. If D = 0, output Q goes to 0, placing the circuit in the reset state.

T- Master-Slave

The T (toggle) flip-flop is a complementing flip-flop and can be obtained from a JK flip-flop when

inputs J and K are tied together. This is shown in Fig. 5.13 (a). When $T = 0$ ($J = K = 0$), a clock edge does not change the output. When $T = 1$ ($J = K = 1$), a clock edge complements the output. The complementing flip-flop is useful for designing binary counters.

The T flip-flop can be constructed with a D flip-flop and an exclusive-OR gate as shown in Fig. 5.13 (b). The expression for the D input is

$$D = T \oplus Q = TQ' + T'Q$$

When $T = 0$, $D = Q$ and there is no change in the output. When $T = 1$, $D = Q'$ and the output complements. The graphic symbol for this flip-flop has a T symbol in the input.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=kQ9WICIFWnU>

<https://www.youtube.com/watch?v=PyWQGZDx-aY>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design”, Pearson Education- V Edition,2013. Page No (195, 201)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : III- SYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Analysis and design Procedure

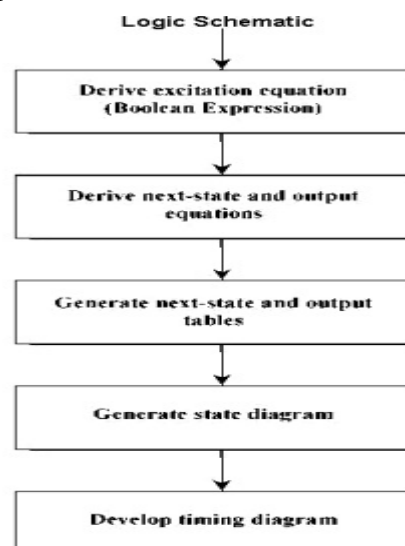
Introduction :

A Sequential circuit combinational logic circuit that consists of inputs variable (X), logic gates (Computational circuit), and output variable (Z). Combinational circuit produces an output based on input variable only, but Sequential circuit produces an output based on current input and previous input variables

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

The behaviour of a sequential circuit is determined from the inputs, the outputs and the states of its flip-flops. Both the output and the next state are a function of the inputs and the present state.



Derive the state table and state diagram for the sequential circuit shown in Figure 7.

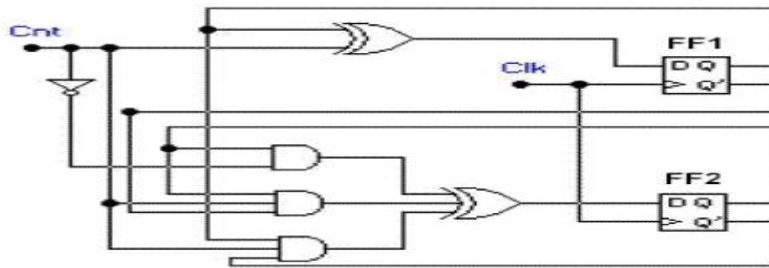


Figure 7. Logic schematic of a sequential circuit.

STEP 1: First we derive the Boolean expressions for the inputs of each flip-flops in the schematic, in terms of external input Cnt and the flip-flop outputs Q1 and Q0.

STEP 2: Derive the next-state equations by converting these excitation equations into flip-flop characteristic equations. In the case of D flip-flops, $Q(\text{next}) = D$.

STEP 3: Now convert these next-state equations into tabular form called the next-state table.

Present State Q1Q0	Next State	
	Cnt = 0	Cnt = 1
00	00	01
01	01	10
10	10	11
11	11	00

STEP 4: The state diagram is generated directly from the next-state table

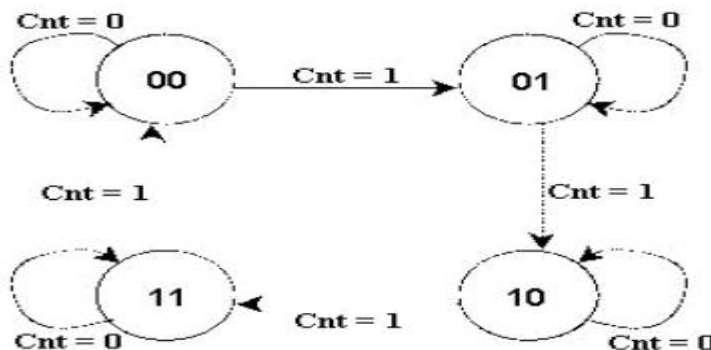


Figure 8. State diagram

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=xQgLFK0Jv2g>

<https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : III- SYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Analysis and design Procedure

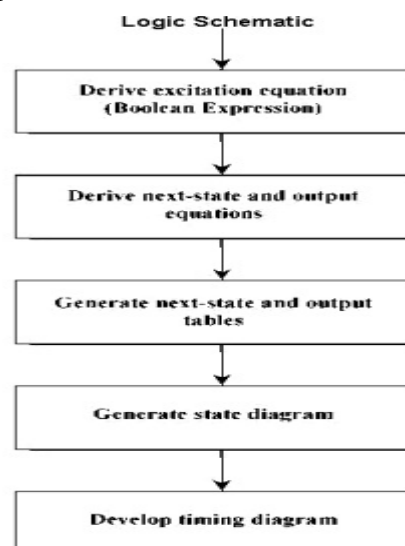
Introduction :

A Sequential circuit combinational logic circuit that consists of inputs variable (X), logic gates (Computational circuit), and output variable (Z). Combinational circuit produces an output based on input variable only, but Sequential circuit produces an output based on current input and previous input variables

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

The behaviour of a sequential circuit is determined from the inputs, the outputs and the states of its flip-flops. Both the output and the next state are a function of the inputs and the present state.



Derive the state table and state diagram for the sequential circuit shown in Figure 7.

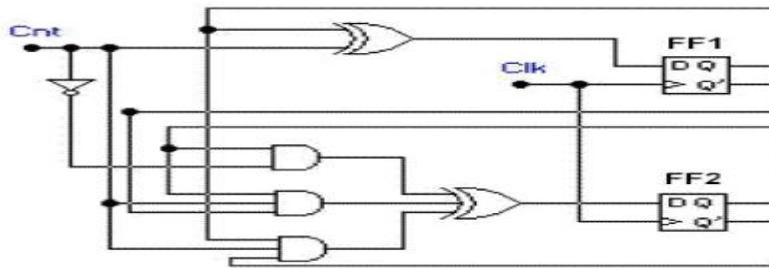


Figure 7. Logic schematic of a sequential circuit.

STEP 1: First we derive the Boolean expressions for the inputs of each flip-flops in the schematic, in terms of external input Cnt and the flip-flop outputs Q1 and Q0.

STEP 2: Derive the next-state equations by converting these excitation equations into flip-flop characteristic equations. In the case of D flip-flops, $Q(\text{next}) = D$.

STEP 3: Now convert these next-state equations into tabular form called the next-state table.

Present State Q1Q0	Next State	
	Cnt = 0	Cnt = 1
00	00	01
01	01	10
10	10	11
11	11	00

STEP 4: The state diagram is generated directly from the next-state table

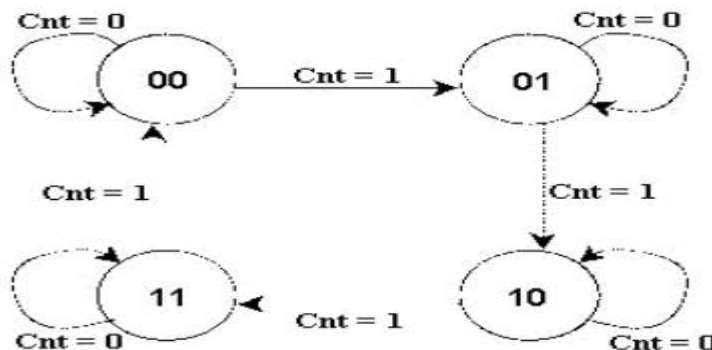


Figure 8. State diagram

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=xQgLFK0Jv2g>

<https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : IV- ASYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Hazards in Sequential circuits

Introduction :

Asynchronous sequential circuits have state that is not synchronized with a clock. Like the synchronous sequential circuits we have studied up to this point they are realized by adding state feedback to combinational logic that implements a next-state function.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Hazards in Sequential Circuits

In normal combinational-circuit design associated with synchronous sequential circuits, hazards are of no concern, since momentary erroneous signals are not generally troublesome. However, if a momentary incorrect signal is fed back in an asynchronous sequential circuit, it may cause the circuit to go to the wrong stable state.

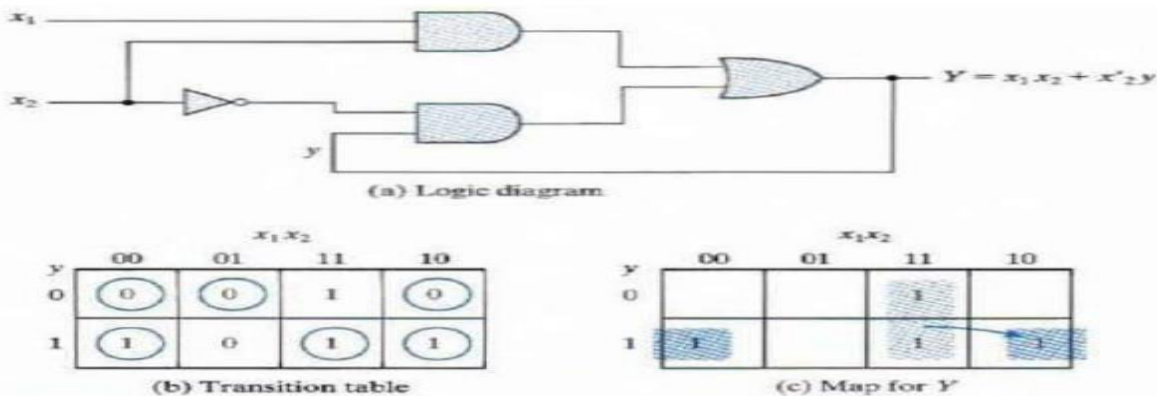


Fig: Hazard in an Asynchronous sequential circuit

If the circuit is in total stable state $y_{x_1x_2} = 111$ and input x_2 changes from 1 to 0, the next total stable state should be 110. However, because of the hazard, output Y may go to 0 momentarily. If this false signal feeds back into gate 2 before the output of the inverter goes to 1, the output of gate 2 will remain at 0 and the circuit will switch to the incorrect total stable state 010. This malfunction can be eliminated by adding an extra gate.

ü **Essential Hazards**

Another type of hazard that may occur in asynchronous sequential circuits is called an **essential hazard**. This type of hazard is caused by unequal delays along two or more paths that originate from the same input. An excessive delay through an inverter circuit in comparison to the delay associated with the feedback path may cause such a hazard.

Essential hazards cannot be corrected by adding redundant gates as in static hazards. The problem that they impose can be corrected by adjusting the amount of delay in the affected path. To avoid essential hazards, each feedback loop must be handled with individual care to ensure that the delay in the feedback path is long enough compared with delays of other signals that originate from the input terminals.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=xQgLFK0Jv2g>

<https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (196)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 34

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : IV- ASYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Problems-Asynchronous counter Analysis

Introduction :

Asynchronous sequential circuits have state that is not synchronized with a clock. Like the synchronous sequential circuits we have studied up to this point they are realized by adding state feedback to combinational logic that implements a next-state function.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

EXAMPLE : Asynchronous Counter Analysis

Problem: Analyze the operation of the counter in Figure (4).

Solution:

By observing the circuit in Figure (4), it can be seen that it is a 4-bit positive edge-triggered asynchronous counter. Begin the waveform analysis by drawing an input clock waveform Assume a 50% duty cycle unless otherwise known. Then graph the output waveforms from each flip-flop stage of the counter. Note that only the LSB stage is triggered by the input clock; all other stages are triggered by an output from the previous stage. The waveforms are shown in Figure (8). The state transition diagram, derived from the waveforms, is shown in Figure (9).

The counter classification is completed by summarizing the characteristics displayed in the state transition diagram:

- MOD 16.
- Divide by 16.

Asynchronous

- Four-bit.
- Positive edge-triggered.
- Binary down counter.

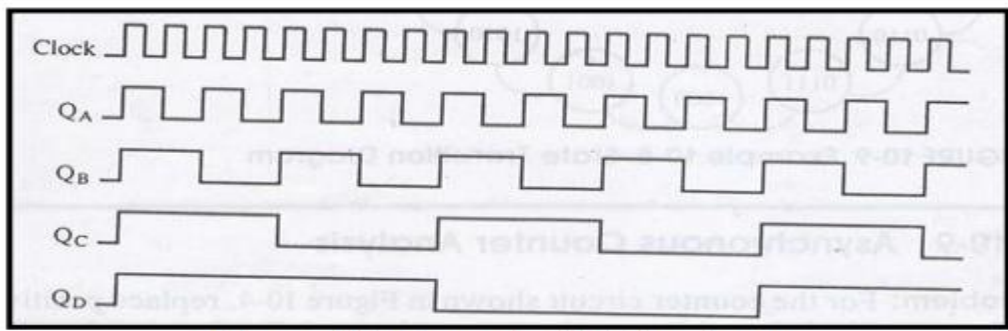


Figure (8): Example (8) Waveforms.

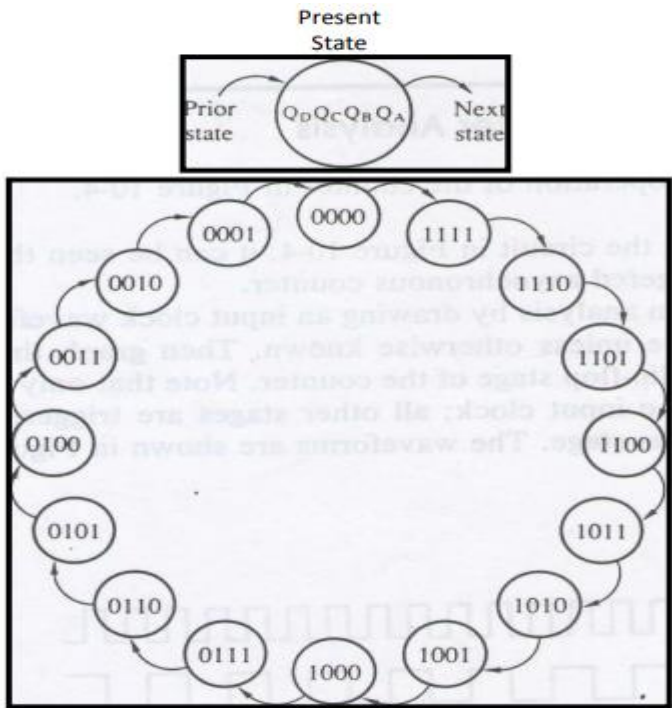


Figure (9): Example (8) State Transition Diagram.

Video Content / Details of website for further learning (if any):

- <https://www.youtube.com/watch?v=xQgLFK0Jv2g>
- <https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design”, Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : IV- ASYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Problems-Asynchronous counter Analysis

Introduction :

Asynchronous sequential circuits have state that is not synchronized with a clock. Like the synchronous sequential circuits we have studied up to this point they are realized by adding state feedback to combinational logic that implements a next-state function.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

EXAMPLE : Asynchronous Counter Analysis

Problem: For the counter circuit shown in Figure (4), replace positive edge-triggered flip-flops with negative edge-triggered flip-flops and analyze the operation of the circuit.

Solution:

The output waveforms are graphed in Figure (10) to determine the operation of the counter.

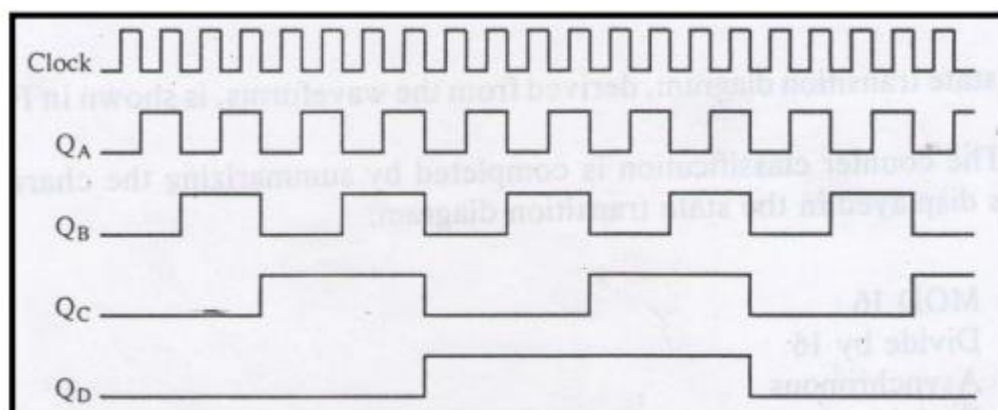


Figure (10): Example (9) Waveforms.

By observing the output waveforms, it can be seen that the count direction has changed just by changing the triggering characteristic of the flip-flop forming the counter. The state transition diagram for this circuit is shown in Figure (11).

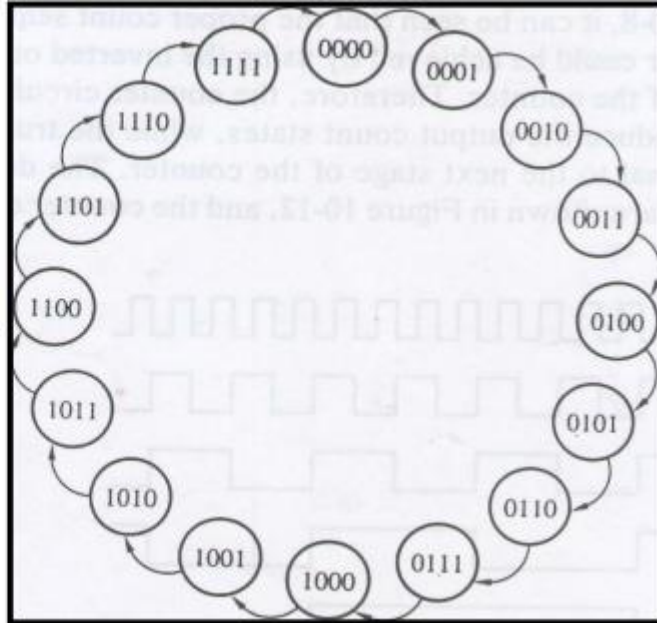


Figure (11): Example (9) State Transition Diagram.

The counter classification is summarized here: • MOD 16. • Divide by 16. • Asynchronous. • Four-bit. • Negative edge-triggered. • Binary up counter.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=xQgLfk0Jv2g>
<https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design”, Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 36

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : IV- ASYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Problems-Asynchronous counter Analysis

Introduction :

Asynchronous sequential circuits have state that is not synchronized with a clock. Like the synchronous sequential circuits we have studied up to this point they are realized by adding state feedback to combinational logic that implements a next-state function.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

EXAMPLE : Asynchronous Counter Design

Problem:

Design a 4-bit, MOD-16, asynchronous binary up counter using positive edge-triggered toggle flip-flops.

Solution:

By observing the output waveforms for a binary up counter, such as those shown in Figure (8), it can be seen that the proper count sequence for a 4-bit binary up counter could be achieved by using the inverted outputs from the flip-flops stages of the counter. Therefore, the counter circuit uses the inverted outputs to produce the output count states, while the true outputs provide the clock signal to the next stage of the counter. The desired counter output waveforms are shown in Figure (12), and the counter circuit is shown in Figure (13).

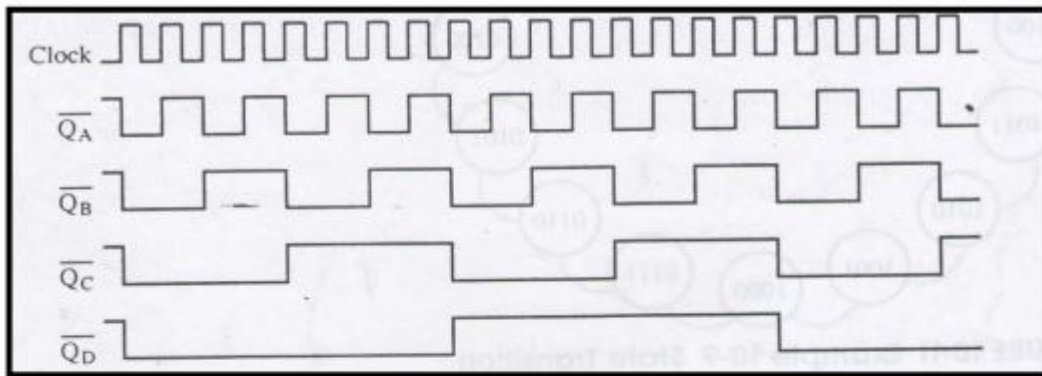


Figure (12): Example (10) Waveforms.

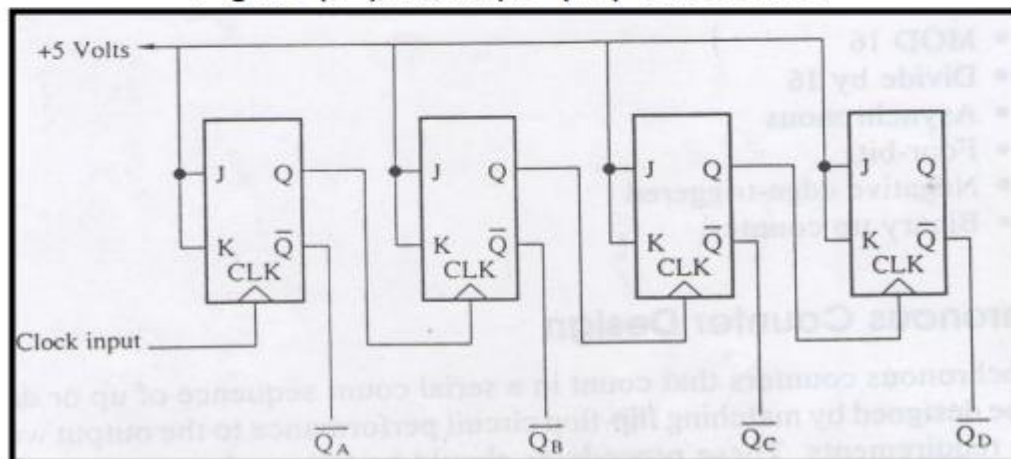


Figure (13): Example (10) Asynchronous 4-Bit Binary Up Counter.

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=xQgLfk0Jv2g>

<https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : IV- ASYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:
Analysis Procedure

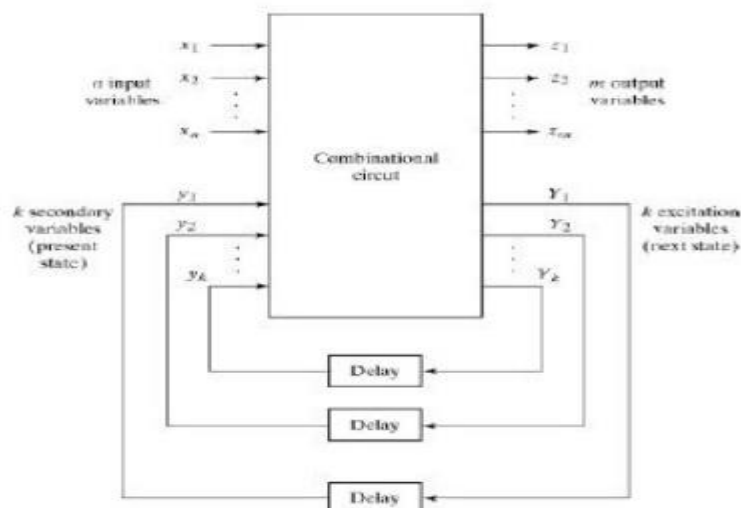
Introduction :

Asynchronous sequential circuits have state that is not synchronized with a clock. Like the synchronous sequential circuits we have studied up to this point they are realized by adding state feedback to combinational logic that implements a next-state function.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

The analysis of asynchronous sequential circuits proceeds in much the same way as that of clocked Synchronous sequential circuits. From a logic diagram, Boolean expressions are written and then transferred into tabular form.



Transition Table

The state table of the circuit is shown below:

Present State		Next State			
		$x = 0$		$x = 1$	
0	0	0	0	0	1
0	1	1	1	0	1
1	0	0	0	1	0
1	1	1	1	1	0

CIRCUITS WITH SR LATCHES

The SR latch is used as a time-delay element in asynchronous sequential circuits. The NOR gate SR latch and its truth table are:

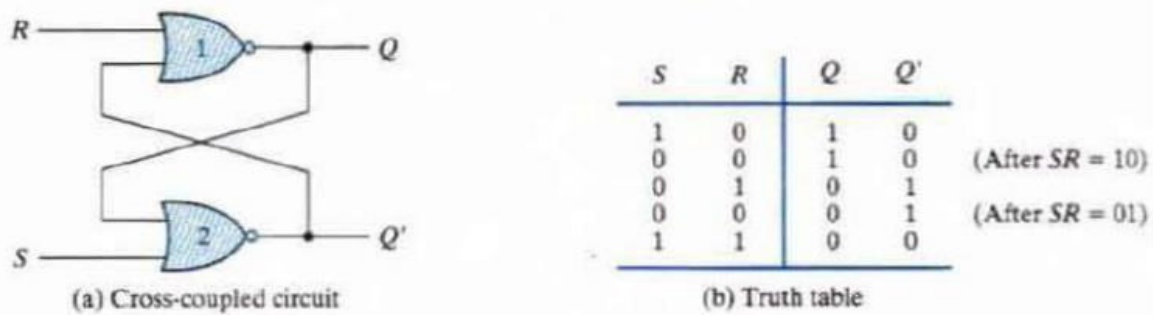
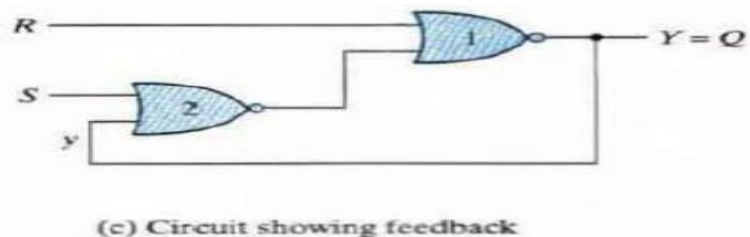


Fig: SR latch with NOR gates

Fig: SR latch with NOR gates

The feedback is more visible when the circuit is redrawn as:



The Boolean function of the output is:

$$S\bar{R} + SR = S(\bar{R} + R) = S$$

$$Y = S\bar{R} + \bar{R}y$$

The reduced excitation function,

$$Y = S + \bar{R}y \text{ when } SR=0$$

$$Y = [S(\bar{R}y)]' = \bar{S} + Ry$$

and the transition table for the circuit is:

y		SR			
		00	01	11	10
0	0	0	0	0	1
1	1	1	0	0	1

$Y = SR' + R'y$
 $Y = S + R'y \text{ when } SR = 0$

(d) Transition table

The behavior of the SR latch can be investigated from the transition table. The condition to be avoided is that both S and R inputs must not be 1 simultaneously. This condition is avoided when $SR = 0$ (i.e., ANDing of S and R must always result in 0). When $SR = 0$ holds at all times, the excitation function

derived previously:

can be expressed as:

$$Y = S\bar{R} + \bar{R}y$$

$$Y = S + \bar{R}y$$

The NAND gate SR latch and its truth table are:

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=xQgLFK0Jv2g>

<https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design”, Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 29

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : IV- ASYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:
Design Procedure

Introduction :

Asynchronous sequential circuits have state that is not synchronized with a clock. Like the synchronous sequential circuits we have studied up to this point they are realized by adding state feedback to combinational logic that implements a next-state function.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

There are a number of steps that must be carried out in order to minimize the circuit complexity and to produce a stable circuit without critical races. Briefly, the design steps are as follows:

- Ø Obtain a primitive flow table from the given specification.
- Ø Reduce the flow table by merging rows in the primitive flow table.
- Ø Assign binary states variables to each row of the reduced flow table to obtain the transition table.
- Ø Assign output values to the dashes associated with the unstable states to obtain the output maps.

- Ø Simplify the Boolean functions of the excitation and output variables and draw the logic diagram.

The design process will be demonstrated by going through a specific example:

Design Example – Specification

Step 1: Primitive Flow Table

A primitive flow table is a flow table with only one stable total state in each row. The total state consists of the internal state combined with the input.

To derive the primitive flow table, first a table with all possible total states in the system is needed:

State	Inputs		Output	Comments
	D	G	Q	
<i>a</i>	0	1	0	$D = Q$ because $G = 1$
<i>b</i>	1	1	1	$D = Q$ because $G = 1$
<i>c</i>	0	0	0	After state <i>a</i> or <i>d</i>
<i>d</i>	1	0	0	After state <i>c</i>
<i>e</i>	1	0	1	After state <i>b</i> or <i>f</i>
<i>f</i>	0	0	1	After state <i>e</i>

Step 2: Reduction of the Primitive Flow Table

The primitive flow table can be reduced to a smaller number of rows if two or more stable states are placed in the same row of the flow table. The simplified merging rules are as follows:

- Ø Two or more rows in the primitive flow table can be merged into one if there are non- conflicting states and outputs in each of the columns.
- Ø Whenever, one state symbol and don't care entries are encountered in the same state column, is listed in the merged row.
- Ø If the state is circled in one of the rows, it is also circled in the merged row.
- Ø The output state is included with each stable state in the merged row.

Step 3: Transition table and logic diagram

The assignment of distinct binary value to each state converts the flow table into a transition table. In the general case, a binary state assignment must be made to ensure that the circuit will be free of critical races.

$$Y = DG + \bar{G}y$$

Step 4: Assigning Outputs to Unstable States

The stable states in a flow table have specific output values associated with them. The unstable states have unspecified output entries designated by a dash. The output values for the unstable states must be chosen so that no momentary false outputs occur when the circuit switches between stable states. This means that if an output variable is not supposed to change as the result of a transition, then an unstable state that is a transient state between two stable states must have the same output value as the stable states.

a	$\textcircled{a}, 0$	$b, -$
b	$c, -$	$\textcircled{b}, 0$
c	$\textcircled{c}, 1$	$d, -$
d	$a, -$	$\textcircled{d}, 1$

(a) Flow table

	0	0
	X	0
	1	1
	X	1

(b) Output assignment

Fig: Assigning output values to unstable states

	DG			
	00	01	11	10
y				
0	0	0	1	0
1	X	0	X	X

(a) $S = DG$

	DG			
	00	01	11	10
y				
0	X	X	0	X
1	0	1	0	0

$R = D'G$

(a) Maps for S and R

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=xQgLFK0Jv2g>
<https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 30

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : IV- ASYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Reduction of State Tables

Introduction :

Asynchronous sequential circuits have state that is not synchronized with a clock. Like the synchronous sequential circuits we have studied up to this point they are realized by adding state feedback to combinational logic that implements a next-state function.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

The procedure for reducing the number of internal states in an asynchronous sequential circuit resembles the procedure that is used for synchronous circuits.

Implication Table and Implied State

The state-reduction procedure for completely specified state tables is based on an algorithm that combines two states in a state table into one as long as they can be shown to be equivalent. Two states are equivalent if, for each possible input, they give exactly the same output and go to the same next states or to equivalent next states.

State Table to Demonstrate Equivalent States

Present State	Next State		Output	
	$x = 0$	$x = 1$	$x = 0$	$x = 1$
<i>a</i>	<i>c</i>	<i>b</i>	0	1
<i>b</i>	<i>d</i>	<i>a</i>	0	1
<i>c</i>	<i>a</i>	<i>d</i>	1	0
<i>d</i>	<i>b</i>	<i>d</i>	1	0

State Table to Be Reduced

Present State	Next State		Output	
	x = 0	x = 1	x = 0	x = 1
a	d	b	0	0
b	e	a	0	0
c	g	f	0	1
d	a	d	1	0
e	a	d	1	0
f	c	b	0	0
g	a	e	1	0

Two states that are not equivalent are marked with a cross [X] in the corresponding square whereas their equivalence is recorded with a check mark. Some of the squares have entries of implied states that must be investigated further to determine whether they are equivalent. Thus table can be reduced from seven states to four one for each member of the preceding partition. The reduced state table is obtained by replacing state b by a and states e and g by d and it is shown below,

Reduced State Table

Present State	Next State		Output	
	x = 0	x = 1	x = 0	x = 1
a	d	a	0	0
c	d	f	0	1
d	a	d	1	0
f	c	a	0	0

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=xQgLFK0Jv2g>

<https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : IV- ASYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Reduction of Flow Tables

Introduction :

Asynchronous sequential circuits have state that is not synchronized with a clock. Like the synchronous sequential circuits we have studied up to this point they are realized by adding state feedback to combinational logic that implements a next-state function.

Prerequisite knowledge for Complete understanding and learning of Topic:

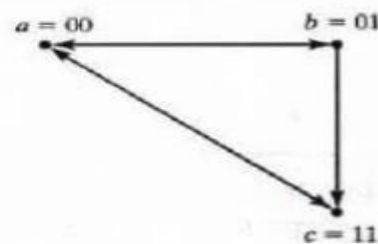
- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Once a reduced flow table has been derived for an asynchronous sequential circuit, the next step in the design is to assign binary variables to each stable state. This assignment results in the transformation of the flow table into its equivalent transition table. The primary objective in choosing a proper binary state assignment is the prevention of critical races. Critical races can be avoided by making a binary state assignment in such a way that only one variable changes at any given time when a state transition occurs in the flow table.

Three-Row Flow-Table Example

	$x_1 x_2$			
	00	01	11	10
a	a	b	c	a
b	a	b	b	c
c	a	c	c	c

(a) Flow table



(b) Transition diagram

Fig: Three row flow table example

To avoid critical races, we must find a binary state assignment such that only one binary variable changes during each state transition. An attempt to find such an assignment is shown in the transition diagram. State a is assigned binary 00, and state c is assigned binary 11. This assignment will cause a critical race during the transition from a to c because there are two changes in the binary state variables and the transition from a to c may occur directly or pass through b. Note that the transition from c to a also causes a race condition, but it is noncritical because the transition does not pass through other states.

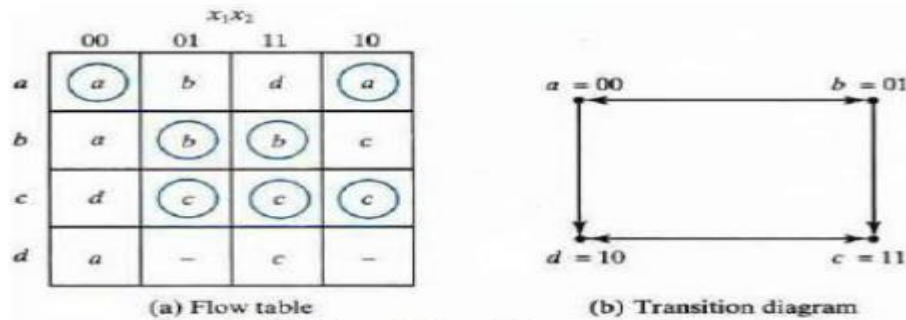


Fig: Flow table with an extra row

A race-free assignment can be obtained if we add an extra row to the flow table. The use of a fourth row does not increase the number of binary state variables, but it allows the formation of cycles between two stable states.

The transition table corresponding to the flow table with the indicated binary state assignment is shown in Fig. The two dashes in row d represent unspecified states that can be considered don't-care conditions. However, care must be taken not to assign 10 to these squares, in order to avoid the possibility of an unwanted stable state being established in the fourth row.

	x_1x_2			
	00	01	11	10
$a = 00$	00	01	10	00
$b = 01$	00	01	01	11
$c = 11$	10	11	11	11
$d = 10$	00	-	11	-

Fig: Transition table

Video Content / Details of website for further learning (if any):

- <https://www.youtube.com/watch?v=xQgLFK0Jv2g>
- <https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : IV- ASYNCHRONOUS SEQUENTIAL LOGIC Date of Lecture:

Topic of Lecture:

Hazards In combinational circuits

Introduction :

Asynchronous sequential circuits have state that is not synchronized with a clock. Like the synchronous sequential circuits we have studied up to this point they are realized by adding state feedback to combinational logic that implements a next-state function.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Latches
- Logic Gates
- Canonical forms Sum of product and Product of sum
- Minimization using Karnaugh map

Hazards are unwanted switching transients that may appear at the output of a circuit because different paths exhibit different propagation delays. Hazards occur in combinational circuits, where they may cause a temporary false output value. When they occur in asynchronous sequential circuits hazards may result in a transition to a wrong stable state.

Hazards In Combinational Circuits

A hazard is a condition in which a change in a single variable produces a momentary change in output when no change in output should occur.

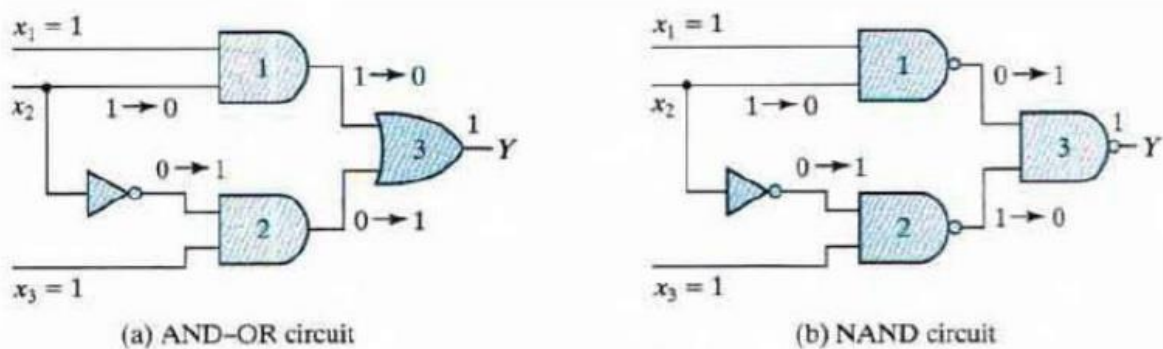


Fig: Circuits with Hazards

Assume that all three inputs are initially equal to 1. This causes the output of gate 1 to be 1, that of gate 2 to be 0 and that of the circuit to be 1. Now consider a change in x_2 from 1 to 0. Then the output of gate 1 changes to 0 and that of gate 2 changes to 1, leaving the output at 1.

The two circuits shown in Fig implement the Boolean function in sum-of-products form:

$$Y = x_1x_2 + \bar{x}_2x_3$$

This type of implementation may cause the output to go to 0 when it should remain a 1. If however, the circuit is implemented instead in product-of-sums form namely,

$$Y = (x_1 + \bar{x}_2)(x_2 + x_3)$$

then the output may momentarily go to 1 when it should remain 0. The first case is referred to as **static 1-hazard** and the second case as **static 0-hazard**.

A third type of hazard, known as **dynamic hazard**, causes the output to change three or more times when it should change from 1 to 0 or from 0 to 1.



Fig: Types of hazards

The hazard-free circuit obtained by such a configuration is shown in figure below. The extra gate in the circuit generates the product term x_1x_3 . In general, hazards in combinational circuits can be removed by covering any two minterms that may produce a hazard with a product term common to both. The removal of hazards requires the addition of redundant gates to the circuit.

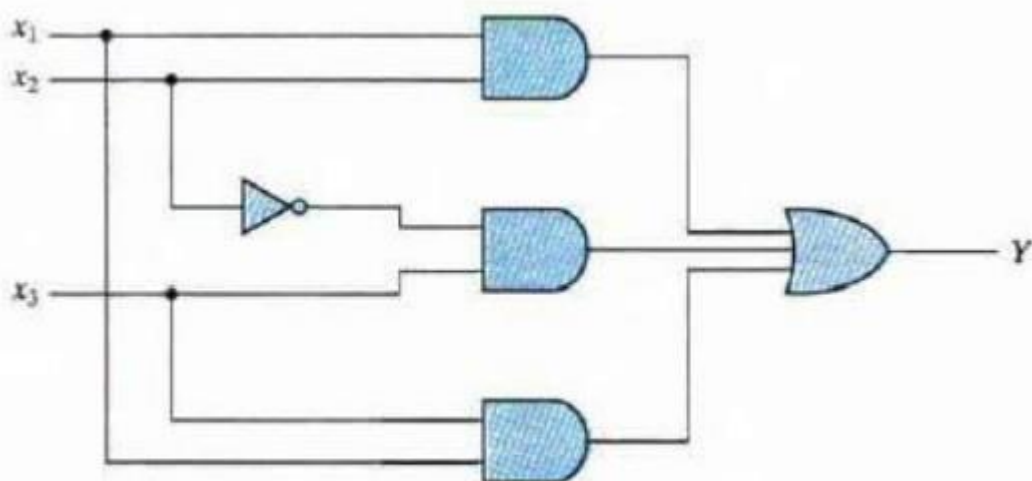


Fig: Hazard free circuit

Video Content / Details of website for further learning (if any):

<https://www.youtube.com/watch?v=xQgLFK0Jv2g>

<https://www.youtube.com/watch?v=WE2wbOAVzdo>

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design” , Pearson Education- V Edition,2013. Page No (196)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : V-MEMORY AND PROGRAMMABLE LOGIC

Date of Lecture:

Topic of Lecture:

Sequential Programmable Devices

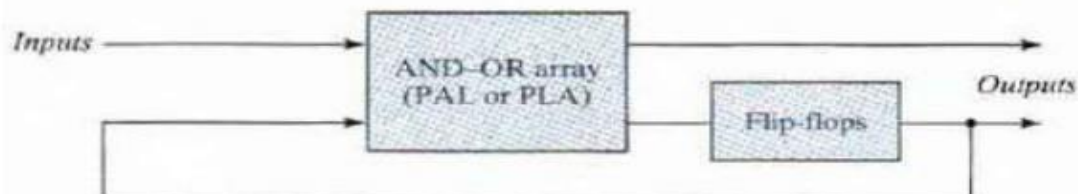
Introduction :

Digital systems are designed with flip-flops and gates. Since the combinational PLD consists of only gates, it is necessary to include external flip-flops when they are used in the design. Sequential programmable devices include both gates and flip-flops.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory devices

A programmable logic device (PLD) is an electronic component used to build reconfigurable digital circuits. Unlike integrated circuits (IC) which consist of logic gates and have a fixed function, a PLD has an undefined function at the time of manufacture. Before the PLD can be used in a circuit it must be programmed (reconfigured) by using a specialized program.



1. Sequential (or simple) programmable logic device (SPLD)
2. Complex programmable logic device (CPLD)
3. Field-programmable gate array (FPGA)

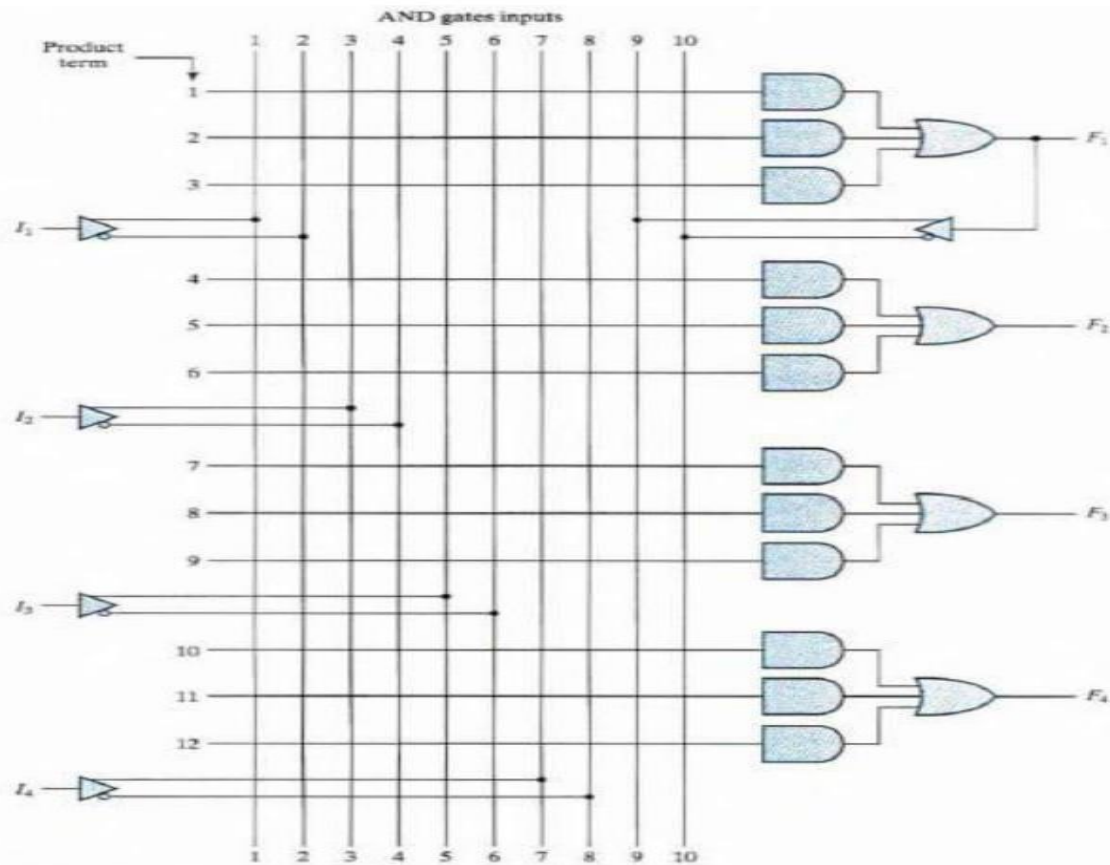


Fig: PAL with four inputs, four outputs and three AND-OR structure

The ever-increasing capacity of integrated circuits created an opportunity for IC manufacturers to design larger PLDs for larger digital-design applications. Instead, IC manufacturers devised complex PLD (CPLD) architectures to achieve the required scale. A typical CPLD is merely a collection of multiple PLDs and an interconnection structure, all on the same chip. In addition to the individual PLDs, the on-chip interconnection structure is also programmable, providing a rich variety of design possibilities. CPLDs can be scaled to larger sizes by increasing the number of individual PLDs and the richness of the interconnection structure on the CPLD chip.

Video Content / Details of website for further learning (if any):

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (92)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 43

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : V-MEMORY AND PROGRAMMABLE LOGIC

Date of Lecture:

Topic of Lecture:

Simple Programmable Devices, Complex Programmable Logic Devices

Introduction :

SPLDs are the simplest, smallest and least-expensive type of programmable logic device. These devices typically have logic gates laid out in arrays where the interconnection between these arrays is configurable by the user.

CPLDs contain several logic blocks, each of which includes eight to 500 macrocells. For most practical purposes, CPLDs can be thought of as multiple SPLDs (plus some programmable interconnect) in a single chip.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory devices

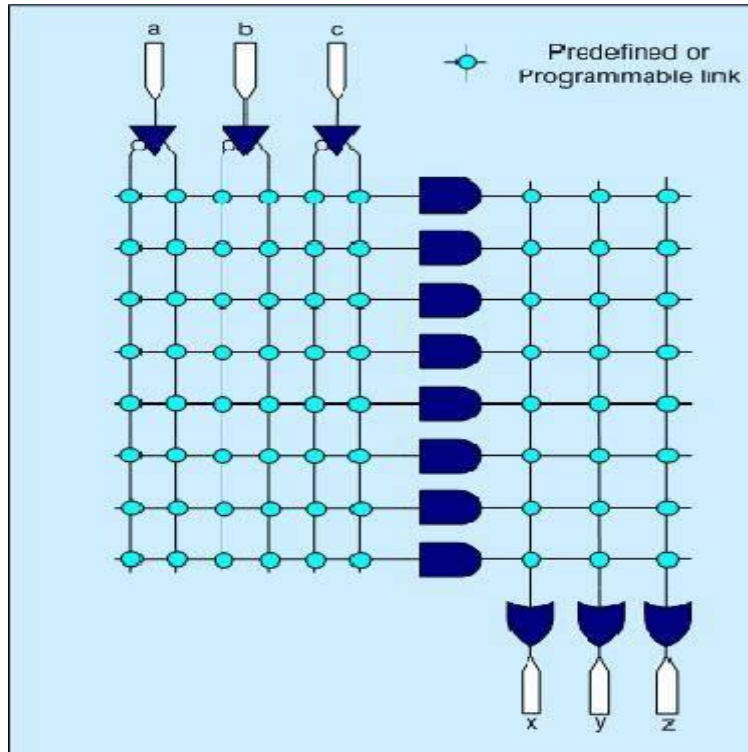
SPLD and CPLD:

The term SPLD covers several types of device:

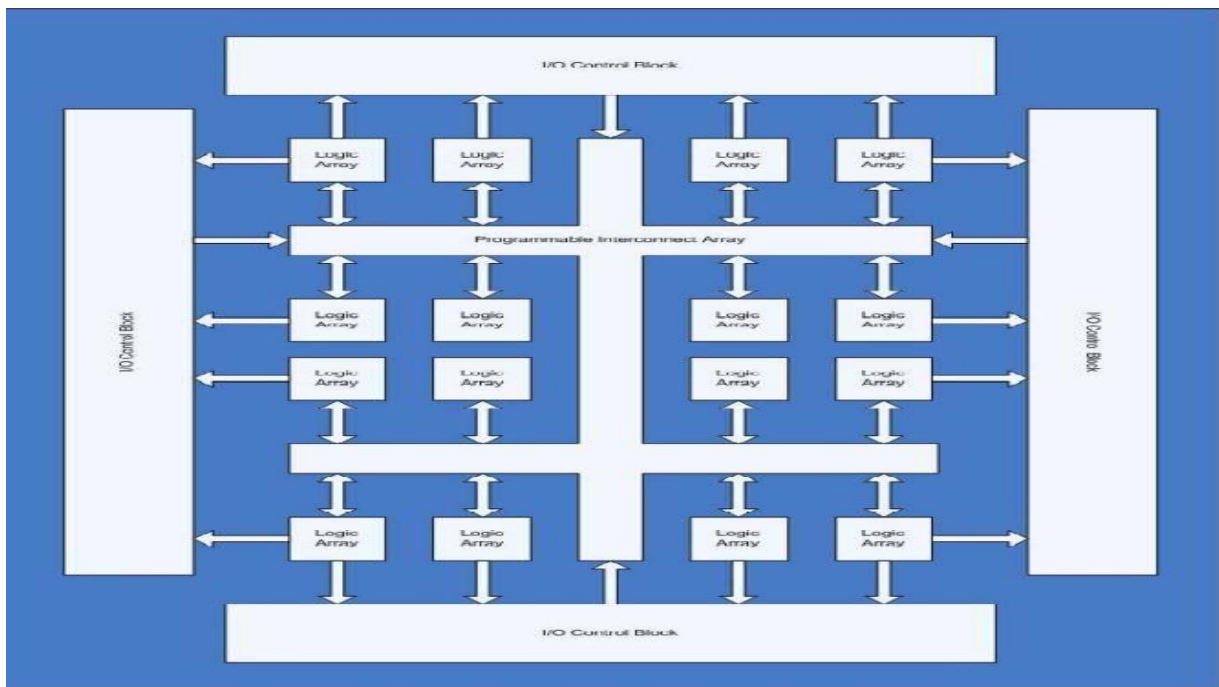
- Programmable Logic Array (PLA) – This device has both programmable AND and OR planes.
- Field Programmable Logic Array (FPLA) – Same as PLA but can be erased and reprogrammed.
- Programmable Array Logic (PAL) – This device has a programmable AND plane and a fixed OR plane.
- GAL – This device has the same logical properties as the PAL but can be erased and reprogrammed

The figure shows a general structure of an SPLD. The connection link across two wires can either be

predefined or programmable depending on the type of SPLD.



CPLDs contain several logic blocks, each of which includes eight to 500 macrocells. For most practical purposes, CPLDs can be thought of as multiple SPLDs (plus some programmable interconnect) in a single chip. The typical structure of a CPLD is shown in Fig 5. Each of the 16 logic array blocks shown is the equivalent of one SPLD. However, in an actual CPLD, there may be more (or less) than 16 logic array blocks. Also, each of these logic array blocks are themselves comprised of macrocells and interconnect wiring, just like an ordinary SPLD.



The larger size of a CPLD allows you to implement either more logic equations or a more complicated design. Most complex programmable logic devices contain macro cells with a sum-of-product combinatorial logic function and an optional flip-flop. Complex programmable logic devices feature predictable timing characteristics that make them ideal for critical, high-performance control applications.

CPLDs can hold larger designs than SPLDs, their potential uses are more varied. They are still sometimes used for simple applications like address decoding, but more often contain high-performance control-logic or complex finite state machines. At the high-end (in terms of numbers of gates), there is also a lot of overlap in potential applications with FPGAs.

Video Content / Details of website for further learning (if any):

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design” , Pearson Education- V Edition,2013. Page No (92)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 44

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : V-MEMORY AND PROGRAMMABLE LOGIC

Date of Lecture:

Topic of Lecture:

Field Programmable Logic Arrays

Introduction :

Field Programmable Gate Arrays are a two-dimensional array of logic blocks and flip-flops with electrically programmable interconnections between logic blocks. The interconnections consist of electrically programmable switches which are why FPGA differs from Custom ICs, as Custom IC is programmed using integrated circuit fabrication technology to form metal interconnections between logic blocks.

Prerequisite knowledge for Complete understanding and learning of Topic:

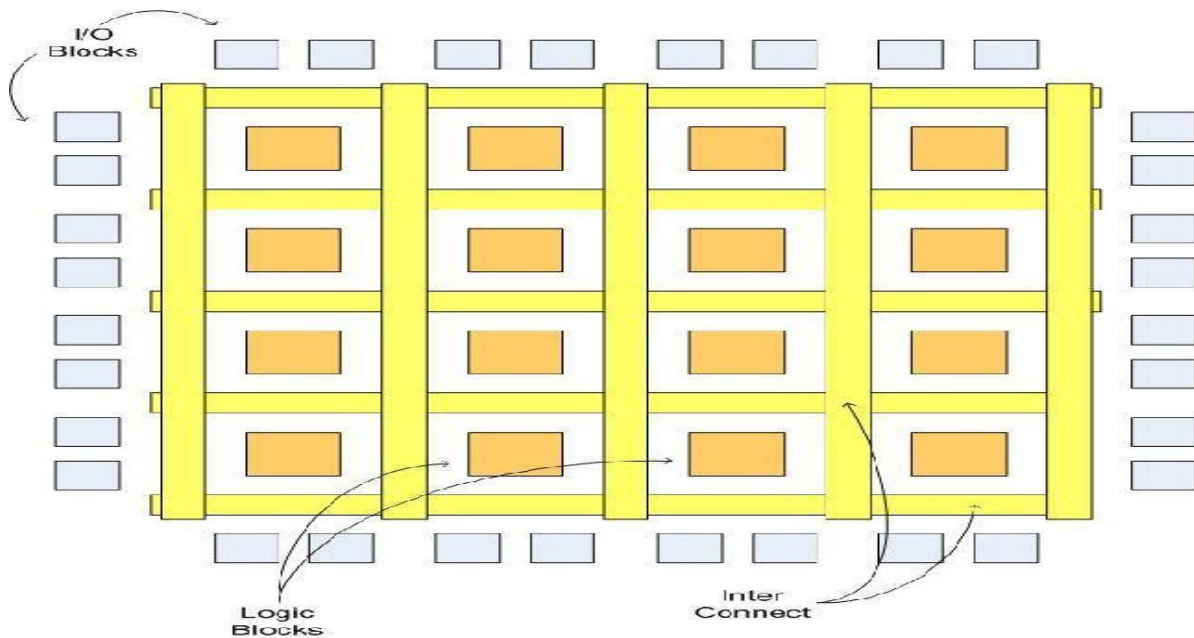
- Memory devices

FPGA

FPGAs can be used to implement just about any hardware design. One common use is to prototype a system that will eventually find its way into an ASIC. However, especially if the product has to be available as soon as possible then there is no reason why the FPGA can't be in the final product. Whether it does or not depends on the balance between development time and costs, and cost of the final device (number of required parts).

Figure 6 illustrates a typical FPGA architecture. There are three key parts of its structure: logic blocks, interconnect, and I/O blocks. The I/O blocks form a ring around the outer edge of the part. Each of these provides individually selectable input, output, or bi-directional access to one of the general-purpose I/O pins on the exterior of the FPGA package. Inside the ring of I/O blocks lies a rectangular array of logic blocks.

And connecting logic blocks to logic blocks and I/O blocks to logic blocks is the programmable interconnect wiring.



The logic blocks within a FPGA can be as small and simple as the macro cells in a PLD (a so-called fine-grained architecture) or larger and more complex (coarse-grained). The logic blocks in a FPGA are generally nothing more than a couple of logic gates or a look-up table and a flip-flop. Because of all the extra flip-flops, the architecture of a FPGA is much more flexible than that of a CPLD. This makes FPGAs better in register-heavy and pipelined applications. They are also often used in place of a processor-plus-software solution, particularly where the processing of input data streams must be performed at a very fast pace.

Video Content / Details of website for further learning (if any):

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (92)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 45

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : V-MEMORY AND PROGRAMMABLE LOGIC

Date of Lecture:

Topic of Lecture:

Application Specific Integrated Circuits

Introduction :

ASIC full form is **Application Specific Integrated Circuit**. These circuits are application specific .i.e. tailored made ICs for a particular application. These are usually designed from root level based on the requirement of the particular application. Some of the basic **application-specific integrated circuit examples** are chips used in toys, the chip used for interfacing of memory and microprocessor

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory devices
- Microprocessor

ASIC

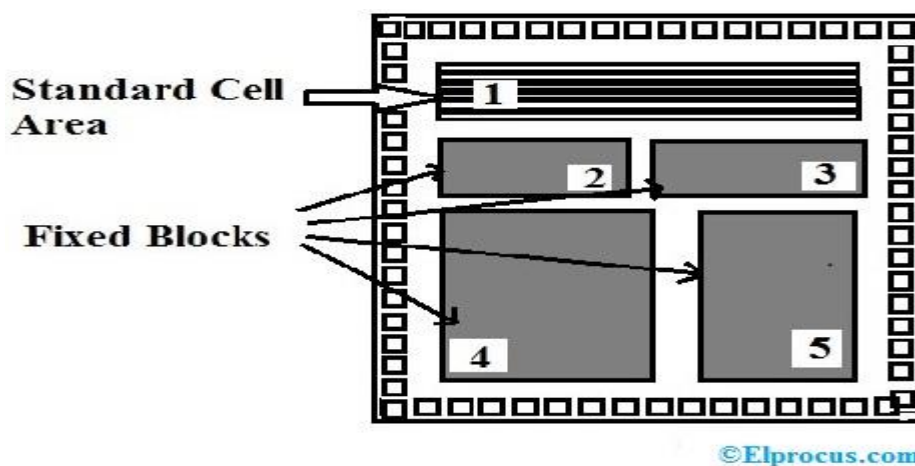
As ASICs are designed from the root level they have high cost and are recommended only for high volume productions. The main advantage of ASIC is reduced chip size as a large number of functional units of a circuit are constructed over a single chip. Modern ASIC generally includes a 32-bit microprocessor, memory blocks, network circuits etc . Such type of ASICs is known as System on Chip. With the development in manufacturing technology and increased research in design methods, ASICs with different levels of customization are developed.

Types of ASIC

ASICs are categorized based on the amount of customization a programmer is allowed to do on a chip.



In this type of design all the logic cells are tailored made for specific application .i.e. designer has to specially make the logic cells for the circuits. All the mask layers for interconnection are customized. So programmer can't change interconnections of the chip and while programming he has to be aware of the circuit layout. One of the best examples of Full custom ASIC is a microprocessor. This type of customization allows designers to built various analog circuits, optimized memory cells, or mechanical structures on a single IC. This ASIC is costly and very time consuming to manufacture and design. The time is taken to design these ICs is around eight weeks.



Video Content / Details of website for further learning (if any):

Important Books/Journals for further learning including the page nos.:

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 37

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : V-MEMORY AND PROGRAMMABLE LOGIC

Date of Lecture:

Topic of Lecture:

RAM and ROM

Introduction :

RAM (Random Access Memory) is the hardware in a computing device where the operating system (OS), application programs and data in current use are kept so they can be quickly reached by the device's processor. RAM is the main memory in a computer. Random Access Memory is volatile. That means data is retained in RAM as long as the computer is on, but it is lost when the computer is turned off.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory devices

The term random access as applied to RAM comes from the fact that any storage location, also known as any memory address, can be accessed directly. Originally, the term Random Access Memory was used to distinguish regular core memory from offline memory.

Types of RAM:

RAM comes in two primary forms:

Dynamic Random Access Memory (DRAM) makes up the typical computing device's RAM, and as was previously noted, it needs that power to be on to retain stored data.

Static Random Access Memory (SRAM) also needs constant power to hold on to data, but it doesn't need to be continually refreshed the way DRAM does.

ROM stands for Read Only Memory. The memory from which we can only read but cannot write on it.

This type of memory is non-volatile. The information is stored permanently in such memories

during manufacture. A ROM stores such instructions that are required to start a computer. This operation is referred to as bootstrap.

Types of ROM:

(i) PROM (Programmable Read Only Memory) is read-only memory that can be modified only once by a user. The user buys a blank PROM and enters the desired contents using a PROM program. Inside the PROM chip, there are small fuses which are burnt open during programming. It can be programmed only once and is not erasable.

(ii) EPROM (Erasable and Programmable Read Only Memory) can be erased by exposing it to ultra-violet light for a duration of up to 40 minutes. For erasing this charge, ultra-violet light is passed through a quartz crystal window (lid). This exposure to ultra-violet light dissipates the charge. During normal use, the quartz lid is sealed with a sticker.

(iii) EEPROM (Electrically Erasable and Programmable Read Only Memory) is programmed and erased electrically. It can be erased and reprogrammed about ten thousand times. Both erasing and programming take about 4 to 10 ms (millisecond). EEPROMs can be erased one byte at a time, rather than erasing the entire chip. Hence, the process of reprogramming is flexible but slow.

The advantages of ROM are as follows –

- Non-volatile in nature
- Cannot be accidentally changed
- Cheaper than RAMs
- Easy to test
- More reliable than RAMs

Video Content / Details of website for further learning (if any):**Important Books/Journals for further learning including the page nos.:**

Morris Mano M. and Michael D. Ciletti “Digital Design”, Pearson Education- V Edition, 2013. Page No (92)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 38

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : V-MEMORY AND PROGRAMMABLE LOGIC

Date of Lecture:

Topic of Lecture: Memory Decoding

Introduction :

The Memory IC used in a digital system is selected or enabled only for the range of addresses assigned to it and this process is called memory decoding. It decodes the memory to be selected for a specific address.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory devices

To require storage components in a memory unit, there is a need for decoding circuits to select the memory word specified by the input address.

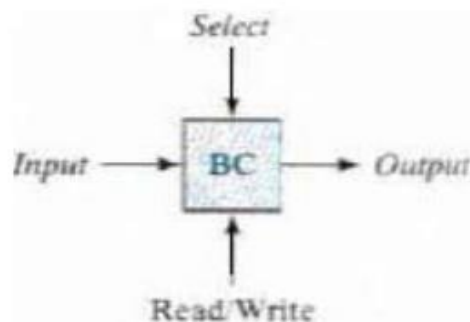


Fig: Block diagram of a memory cell

The storage part of the cell is modeled by an SR latch with associated gate s to form a D latch. A 1 in the read/write input provides the read operation by fanning a path from the latch to the output terminal. A 0 in the read/write input provides the write operation by forming a path from the input terminal to the latch.

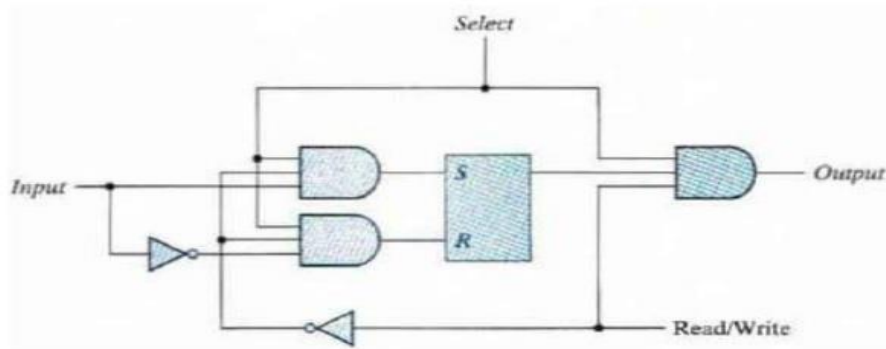


Fig: Logic diagram of a memory cell

- The logical construction of a small RAM consists of four words of four bits each and has a total of 16 binary cells.
- The small blocks labeled BC represent the binary cell with its three inputs and one output. A memory with four words needs two address lines.
- When the memory enable is 0, all outputs of the decoder are 0 and none of the memory words are selected. With the memory select at 1, one of the four words is selected, dictated by the value in the two address lines.
- Once a word has been selected, the read/write input determines the operation. During the read operation the four bits of the selected word go through OR gates to the output terminals.
- During the write operation, the data available in the input lines are transferred into the four binary cells of the selected word.
- The binary cells that are not selected are disabled and their previous binary values remain unchanged.
- When the memory select input that goes into the decoder is equal to 0 none of the words are selected and the contents of all cells remain unchanged regardless of the value of the read/write input.

Video Content / Details of website for further learning (if any):

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (92)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 39

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : V-MEMORY AND PROGRAMMABLE LOGIC

Date of Lecture:

Topic of Lecture:

Error Detection and Correction

Introduction :

Error is a condition when the output information does not match with the input information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from one system to other. That means a 0 bit may change to 1 or a 1 bit may change to 0.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory devices

Error-Detecting codes

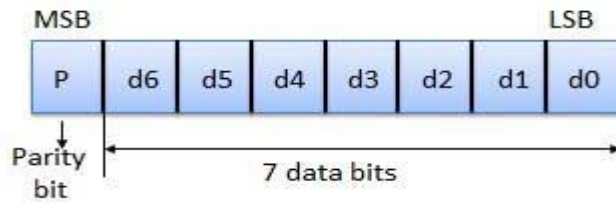
Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if an error occurred during transmission of the message. A simple example of error-detecting code is **parity check**.

Error-Correcting codes

Along with error-detecting code, we can also pass some data to figure out the original message from the corrupt message that we received. This type of code is called an error-correcting code.

Parity Checking of Error Detection

It is the simplest technique for detecting and correcting errors. The MSB of an 8-bits word is used as the parity bit and the remaining 7 bits are used as data or message bits. The parity of 8-bits transmitted word can be either even parity or odd parity.



Even parity -- Even parity means the number of 1's in the given word including the parity bit should be even (2,4,6,....).

Odd parity -- Odd parity means the number of 1's in the given word including the parity bit should be odd (1,3,5,....).

Use of Parity Bit

The parity bit can be set to 0 and 1 depending on the type of the parity required.

- For even parity, this bit is set to 1 or 0 such that the no. of "1 bits" in the entire word is even. Shown in fig. (a).
- For odd parity, this bit is set to 1 or 0 such that the no. of "1 bits" in the entire word is odd. Shown in fig. (b).

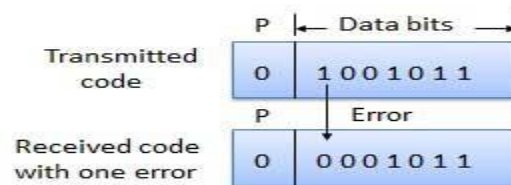


Fig. (a)



Fig. (b)

Parity checking at the receiver can detect the presence of an error if the parity of the receiver signal is different from the expected parity. If an error is detected, then the receiver will ignore the received byte and request for retransmission of the same byte to the transmitter.



Video Content / Details of website for further learning (if any):

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition,2013. Page No (92)

Course Teacher

Verified by HOD



MUTHAYAMMAL ENGINEERING COLLEGE
(An Autonomous Institution)



(Approved by AICTE, New Delhi, Accredited by NAAC & Affiliated to
Anna University)
Rasipuram - 637 408, Namakkal Dist., Tamil Nadu

LECTURE HANDOUTS

L 40

CS

II / III

Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : V-MEMORY AND PROGRAMMABLE LOGIC

Date of Lecture:

Topic of Lecture:

Programmable Logic Array

Introduction :

A programmable logic array (PLA) is a kind of programmable logic device used to implement combinational logic circuits. The PLA has a set of programmable AND gate planes, which link to a set of programmable OR gate planes, which can then be conditionally complemented to produce an output. It has 2^N AND Gates for N input variables, and for M outputs from PLA, there should be M OR Gates, each with programmable inputs from all of the AND gates.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory devices

PLA

The definition of term PLA presents the Boolean function in the form of a sum of product (SOP). The designing of this programmable logic array can be done using the logic gates like AND, OR, and NOT by fabricating on the chip, that makes every input as well as its compliment obtainable toward every AND gate. An every AND gate's output is connected to the every OR gate. Finally, the output of the OR gate generates the output of the chip.

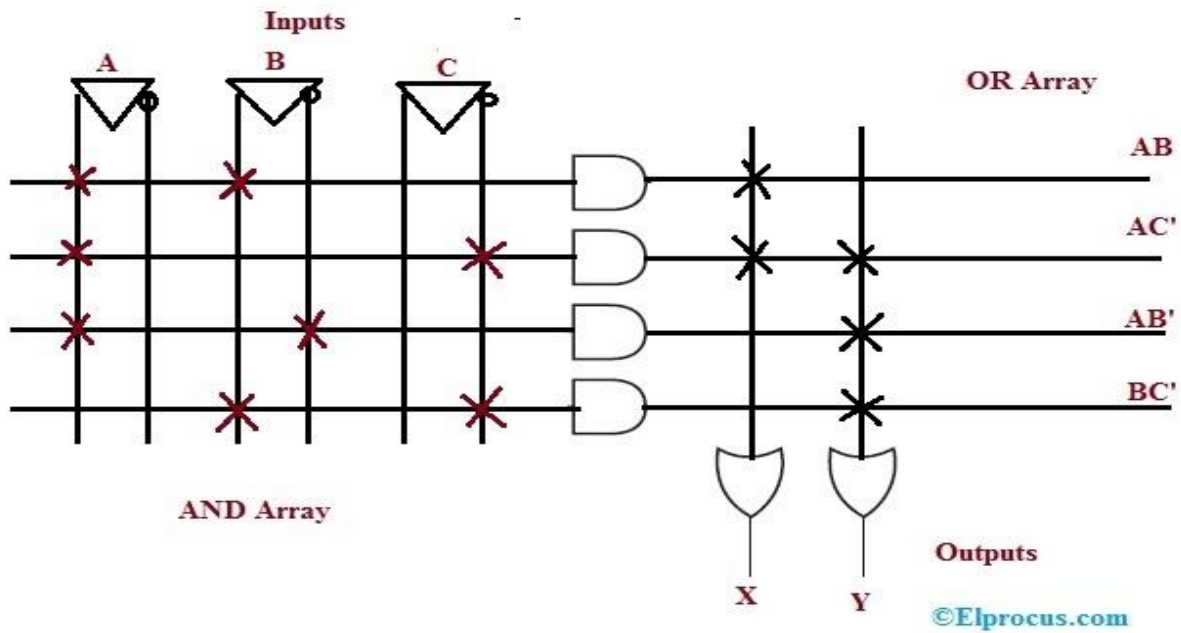
Example of PLA

Implement the following Boolean expression with the help of programmable logic array (PLA)

$$X=AB+AC'$$

$$Y = AB' +BC + AC'$$

The above given two Boolean functions are in the form of SOP (sum of products). The product terms present in the Boolean expressions are X & Y, and one product term that is AC' is common in every equation. So, the total required logic gates for generating the above two equations is AND gates-4, OR programmable OR gates-2.



The AND gates which are programmable have the right of entry for normal as well as complemented variable inputs. In the above logic diagram, the available inputs for each AND gate are A, A', B, B', C, C'. So, in order to generate a single product term with every AND gate, the program is required. All the product terms are obtainable at the inputs of each OR gate. Here, the programmable connections on the logic gate can be denoted with the symbol 'X'.

Video Content / Details of website for further learning (if any):

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti "Digital Design", Pearson Education- V Edition, 2013. Page No (92)

Course Teacher

Verified by HOD



Course Name with Code : 19GES24/Digital Principles and System Design

Course Teacher : Ms.S.Priya, AP/ECE

Unit : V-MEMORY AND PROGRAMMABLE LOGIC

Date of Lecture:

Topic of Lecture:

Programmable Array Logic

Introduction :

Programmable Array Logic (PAL) is a type of Programmable Logic Device (PLD) used to realize a particular logical function. PALs comprise of an AND gate array followed by an OR gate array. However it is to be noted that here only the AND gate array is programmable unlike the OR gate array which has a fixed logic.

Prerequisite knowledge for Complete understanding and learning of Topic:

- Memory devices

PAL

Programmable-AND and fixed-OR structure of PALs make them less flexible from programming point of view when compared with Programmable Logic Arrays (PLAs). However due to the same reason PALs are less expensive than PLAs.

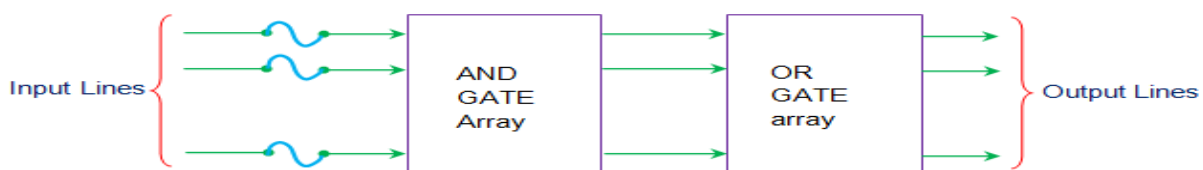


Figure 1 Programmable Array Logic (PAL)

PALs comprise of an AND gate array followed by an OR gate array as shown by Figure 1. Further this connection matrix is programmable (red box in Figure 2) which lets the user to decide the connection between the input lines and the AND gates.

This means that one has to connect each and every input line to either single or multiple AND gate(s), depending on the logic. This causes one to realize the logical ‘and’ functionality between the input lines. Further the outputs of the AND gate array are fed as inputs to the OR gates via hard-wired connections (shown by blue box in Figure 2), which are fixed and hence unalterable.

Moreover it is to be noted that the output of every AND gate is not fed to every OR gate. For example, OR gate 1 (O_1) is has multiple inputs including the outputs of AND gate 1 (A_1), AND gate 2 (A_2) and AND gate p (A_p).

However OR gate n (O_n) has only two inputs which are the outputs of AND gates A_1 and A_p . As these connections are fixed, one has to pay attention while establishing the connection to realize the logical ‘or’ functionality of the product-terms obtained as outputs from AND gate array.

Finally there are n output lines of the OR gate array resulting in n output PAL realizing the required logic in sum-of-products (SOP) form. The PAL shown in Figure 2 can be addressed as m-input, p-product-term, n-output PAL. However it is to be noted that the number of inputs, AND gates and OR gates present in the PAL are all independent i.e. one PAL can have 3 inputs, 8 AND gates and 4 outputs (and thus 4 OR gates).

Video Content / Details of website for further learning (if any):

Important Books/Journals for further learning including the page nos.:

Morris Mano M. and Michael D. Ciletti “Digital Design” , Pearson Education- V Edition,2013. Page No (92)

Course Teacher

Verified by HOD