DIGITAL LOGIC DESIGN Course Code: 328356(28)

Mr. Manjeet Singh Sonwani
Assistant Professor
Department of Electronics & Telecomm.
Government Engineering College Raipur

Course Contents

UNIT-I NUMBER SYSTEMS, CODES AND BOOLEAN ALGEBRA

UNIT-II MINIMIZATIONTECHNIQUES

UNIT-III COMBINATIONAL CIRCUITS

UNIT-IV SEQUENTIAL CIRCUITS

UNIT-V DIGITAL LOGIC FAMILIES

Text Books:

1. Fundamentals of Digital Circuits: A. Anand Kumar, PHI.(Unit – I to V)
2. Digital Electronics-Principles and Integrated Circuits, A.K. Maini, 1st Edition, Wiley India.

Reference Books:

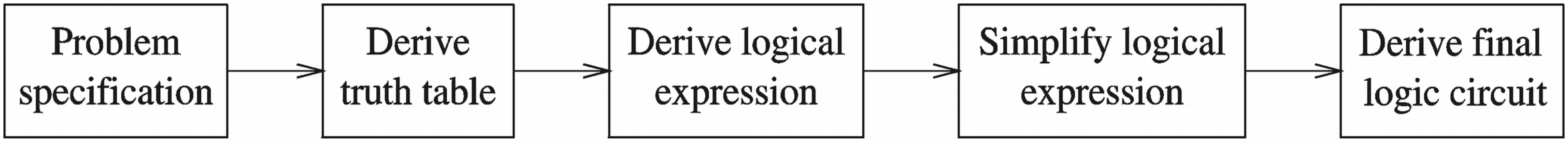
1. Digital Fundamentals: Floyd & Jain: Pearson Education.
2. Digital Electronics: A. P. Malvino: Tata McGraw Hill. 3. Digital Circuits & Logic Design–LEE, PHI.

UNIT-I MINIMIZATIONTECHNIQUES

- Expansion of a Boolean expression to SOP form
- Expansion of a Boolean expression to POS form
- Two, Three & Four variable K-Map: Mapping and minimization of SOP and POS expressions
- Completely and Incompletely Specified Functions- Concept of Don't Care Terms
- Quine – McClusky Method (Up to 5 variable)
- Synthesis using AND-OR, NAND-NOR and XOR forms; Design Examples;
- Programmable Logic Devices: PAL, PLA’s & PROMS.

Logic Circuit Design Process

- A simple logic design process involves
  - Problem specification
  - Truth table derivation
  - Derivation of logical expression
  - Simplification of logical expression
  - Implementation



Deriving Logical Expressions

- Derivation of logical expressions from truth tables
  - sum-of-products (SOP) form
  - product-of-sums (POS) form
- SOP form
  - Write an AND term for each input combination that produces a 1 output
    - Write the variable if its value is 1; complement otherwise
  - OR the AND terms to get the final expression
- POS form
  - Dual of the SOP form

Deriving Logical Expressions (cont.)

- 3-input majority function

A	B	C	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

- SOP logical expression
- Four product terms
  - Because there are 4 rows with a 1 output

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

Deriving Logical Expressions (cont.)

- 3-input majority function

A	B	C	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

- POS logical expression
- Four sum terms
  - Because there are 4 rows with a 0 output

$$F = (A + B + C)\ (A + B + \overline{C})\ (A + \overline{B} + C)\ (\overline{A} + B + C)$$

Logical Expression Simplification

- Algebraic manipulation
  - Use Boolean laws to simplify the expression
    - Difficult to use
    - Don’t know if you have the simplified form

Algebraic Manipulation

- Majority function example

Added extra

$$\overline{A}BC + A\overline{B}C + AB\overline{C} + ABC =$$

$$\overline{A}BC + ABC + A\overline{B}C + ABC + AB\overline{C} + ABC$$

- We can now simplify this expression as

$$BC + AC + AB$$

- A difficult method to use for complex expressions

Implementation Using NAND Gates

- Using NAND gates
  - Get an equivalent expression

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

  - Using de Morgan's law

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

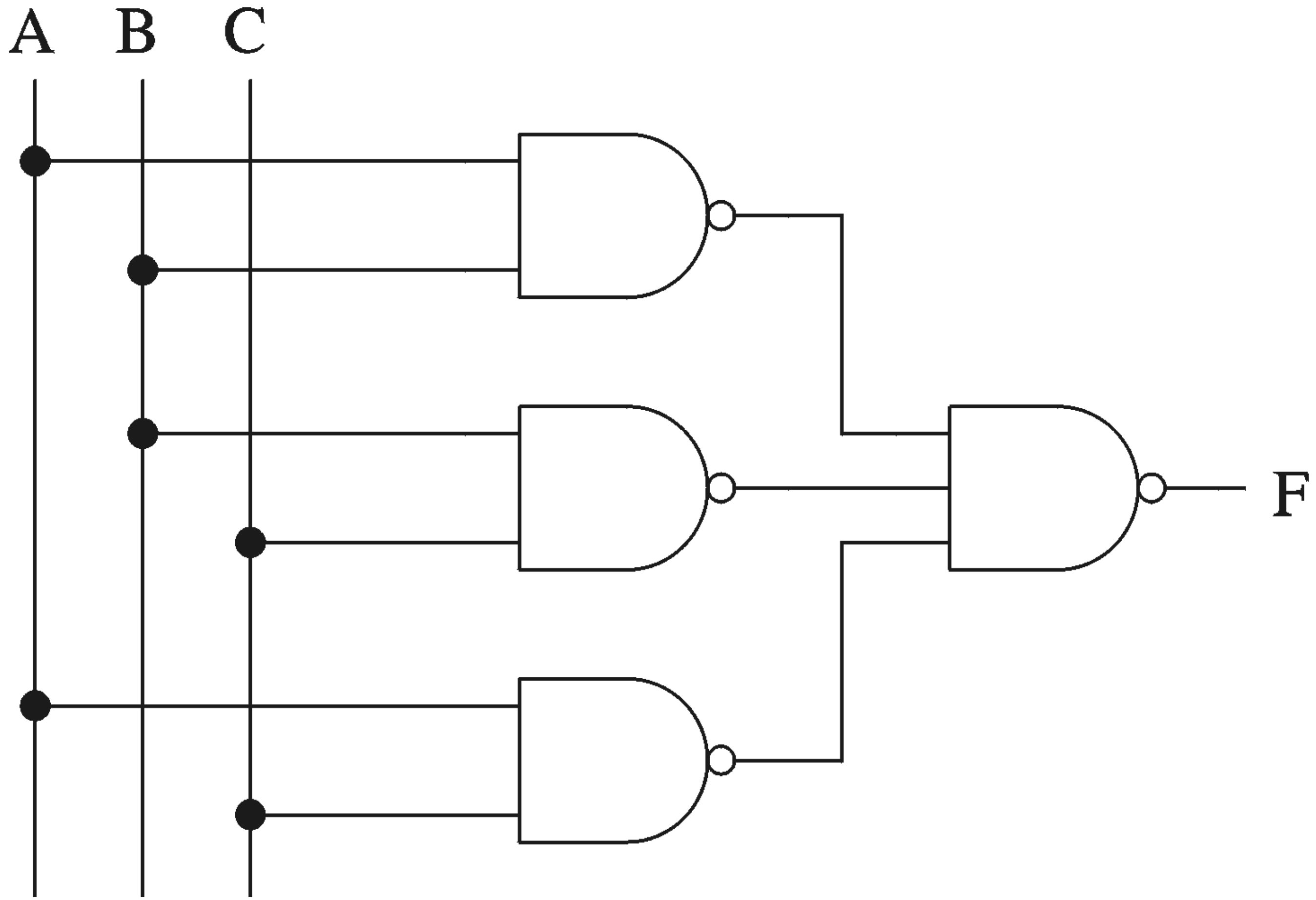
  - Can be generalized
    - Majority function

$$A\,B + B\,C + AC = \overline{\overline{A\,B} \cdot \overline{BC} \cdot \overline{AC}}$$

Idea: NAND Gates: Sum-of-Products, NOR Gates: Product-of-Sums

Implementation Using NAND Gates (cont.)

- Majority function



Minimization procedure

- 1.Draw the K-map with 2^n cells, where n is the number of variables in a Boolean function.
- 2.Fill in the K-map with 1s and 0s as per the function given in the algebraic form (SOP or POS) or truth-table form.
- 3. Determine the set of prime implicants that consist of all the essential prime implicants as per the criteria: All the 1-entered or 0-entered cells are covered by a set of implicants, while making the number of cells covered by each implicant as large as possible. Eliminate the redundant implicants. Identify all the essential prime implicants. Whenever there is a choice among the prime implicants select the prime implicant with the smaller number of literals.

Minimization procedure

- 4. If the final expression is to be generated in SOP form, the prime implicants should be identified by suitably grouping the positionally adjacent 1-entered cells, and converting each of the prime implicant into a product term. The final SOP expression is the OR of all the product terms.
- 5. If the final simplified expression is to be given in the POS form, the prime implicants should be identified by suitably grouping the positionally adjacent 0-entered cells, and converting each of the prime implicant into a sum term. The final POS expression is the AND of all sum terms.

Common Terms for 2-Level Minimization

- Literal – A variable in complemented or uncomplemented form
- Product – The disjunction (AND) of a set of literals; also represents a cube
- Support Set – Set of all variables that define the domain of a switching function
- Minterm – A disjunction (AND) containing an instance of each literal corresponding to a variable in the support set that is in the on-set, f_{on},of a function
- Don't Care – The absence of a supporting variable in a product term
- Implicant – A product term that covers one or more minterms in the on-set, f_{on}, of a function
- Prime Implicant – An implicant in the on-set, f_{on}, of a function such that it is not a subproduct of any other possible implicant in the set.
- Essential Prime Implicant – A prime implicant that covers at least one minterm ***NOT*** covered by any other implicant in the on-set, f_{on}.

Minimization

- Minimization can be done using
    - To combine terms
    - Or equivalently
    - Visual identification of terms that can be combined
    - – Karnaugh maps

$$B\bar{C} + B C = B(\bar{C} + C) = B$$

Truth table to K-Map

A	B	P
0	0	1
0	1	1
1	0	0
1	1	1

A \ B	0	1
0	1	1
1		1

minterms are represented by a 1 in the corresponding location in the K map.

The expression is:

$$\overline{A}.\overline{B} + \overline{A}.B + A.B$$

K-Maps

- Adjacent 1’s can be “paired off”
- Any variable which is both a 1 and a zero in this pairing can be eliminated
- Pairs may be adjacent horizontally or vertically

B

A 0 1

0	1	1
1		1

a pair

B is eliminated,
leaving A as the term

another pair

A is eliminated,
leaving B as the term

The expression becomes A + B

An example

- Two Variable K-Map

A	B	C	P
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

$.C \quad + A.\bar{B}.C$

A ↓ \ BC →	00	01	11	10
0				1
1	1			1

One square filled in for each minterm.

Grouping the Pairs

equates to B.$\overline{C}$ as A is eliminated.

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

Our truth table simplifies to

A.C + B.C

Here, we can "wrap around" and this pair equates to A.$\overline{C}$ as B is eliminated.

Groups of 4

Groups of 4 in a block can be used to eliminate two variables:

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

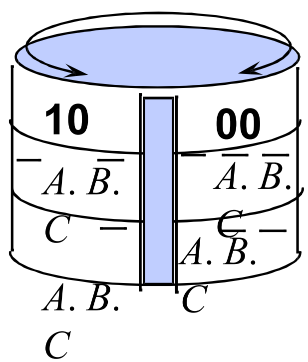
The solution is B because it is a 1 over the whole

block (vertical pairs) = BC + BC = B(C + C) = B.

Karnaugh Maps

- Three Variable K-M a

BC / A	00	01	11	10
0	$\bar{A}.\bar{B}.\bar{C}$	$\bar{A}.\bar{B}.C$	$\bar{A}.B.C$	$\bar{A}.B.\bar{C}$
1				$A\ B$



- Extreme ends of same row are *adjacent*

The Block of 4, again

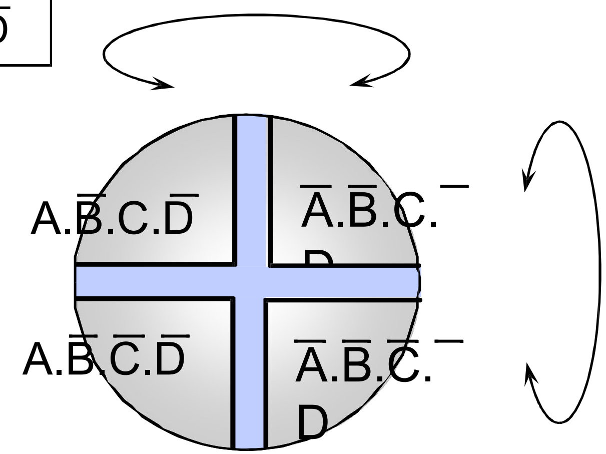
BC / A	00	01	11	10
0	1			1
1	1			1

$X = \overline{C}$

4-variable Karnaugh Maps

- Four Variable K-Map

CD / AB	00	01	11	10
00	$\bar{A}.\bar{B}.\bar{C}.\bar{D}$	$\bar{A}.\bar{B}.\bar{C}.D$	$\bar{A}.\bar{B}.C.D$	$\bar{A}.\bar{B}.C.\bar{D}$
01	$\bar{A}.B.\bar{C}.\bar{D}$	$\bar{A}.B.\bar{C}.D$	$\bar{A}.B.C.D$	$\bar{A}.B.C.\bar{D}$
11	$A.B.\bar{C}.\bar{D}$	$A.B.\bar{C}.D$	$A.B.C.D$	$A.B.C.\bar{D}$
10	$A.\bar{B}.\bar{C}.\bar{D}$	$A.\bar{B}.\bar{C}.D$	$A.\bar{B}.C.D$	$A.\bar{B}.C.\bar{D}$



- Four corners adjacent

Karnaugh Maps

- Four Variable K-Map example

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

AB \ CD	00	01	11	10
00				
01				
11				
10				

F =

Karnaugh Maps

- Four Variable K-Map solution

F = A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C.D + A.B.C .D

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

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

Incompletely specified functions

- All Boolean functions are not always completely specified Consider the BCD decoder,
- Only 10 outputs are decoded from 16 possible input combinations.
- The six invalid combinations of the inputs never occur
- We dont-care what the output is for any of these combinations that should never occur
- These dont-care situations can be used advantageously in generating a simpler Boolean expression
- Such dont-care combinations of the variables are represented by an "X" in the appropriate cell of the K-map

Truth-table and K-map with don’t cares Using the three dont care conditions

A	**B**	**C**	**F**
0	0	0	X
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	X
1	1	1	X

Quine-McClusky

- Map methods unsuitable if the number of variables is more than six
- Quine formulated the concept of tabular minimisation in 1952
- Improved by McClusky in 1956
- Quine-McClusky method :
- Can be performed by hand, but tedious, time-consuming and subject to error
- Better suited to implementation on a digital computer

Principle of Quine-McCusky Method

- Quine-McClusky method is a two stage simplification process
- Step 1: Prime implicants are generated by a special tabulation process
- Step 2: A minimal set of implicants is determined

Tabulation

- List the specified minterms for the 1s of a function and dont-cares
- Generate all the prime implicants using logical adjacency (AB/ + AB = A)
- One can work with the equivalent binary number of the product terms.
- Example: A'BCD' and A'BC'D' are entered as 0110 and 0100 Combined to form a term 01-0

Creation of Prime Implicant Table

- Selected prime implicants are combined and arranged in a table
    - Selection of minimal set of implicants
- Determine essential prime implicants These are the minterms not covered by any other prime implicant Identified by columns that have only one asterisk Columns 2 and 14 have only one asterisk each The associated row, CD/, is an essential prime implicant. CD/ is selected as a member of the minimal set (mark it by an asterisk) Remove the corresponding columns, 2, 6, 10, 14, from the prime implicant table A new table is prepared.

Dominating Prime Implicants

- Identified by the rows that have more asterisks than others Choose Row A/BD Includes the minterm 7, which is the only one included in the row represented by A/BC A/BD is dominant implicant over A/BC A/BC can be eliminated Mark A/BD by an asterisk Check off the columns 5 and 7
- Choose AB/D Dominates over the row AB/C Mark the row AB/D by an asterisk Eliminate the row AB/C Check off columns 9 and 11 Select A/C/D Dominates over B/C/D. B/C/D also dominates over A/C/D Either B/C/D or A/C/D can be chosen as the dominant prime implicant

Process of simplification

- All columns have two asterisks
- There are no essential prime implicants.
- Choose any one of the prime implicants to start with Start with prime implicant a (mark with asterisk)
- Delete corresponding columns, 0 and 1
- Row c becomes dominant over row b, delete row b Delete columns 3 and 7
- Row e dominates row d, and row d can be eliminated
- Delete columns 14 and 15 Choose row g it covers the remaining asterisks associated with rows h and f

STEP 1 - EXAMPLE

$f^{on} = \{m_0, m_1, m_2, m_3, m_5, m_8, m_{10}, m_{11}, m_{13}, m_{15}\} = \Sigma\,(0, 1, 2, 3, 5, 8, 10, 11, 13, 15)$

Minterm	Cube				
0	0	0	0	0	✓
1	0	0	0	1	✓
2	0	0	1	0	✓
8	1	0	0	0	✓
3	0	0	1	1	✓
5	0	1	0	1	✓
10	1	0	1	0	✓
11	1	0	1	1	✓
13	1	1	0	1	✓
15	1	1	1	1	✓

Minterm	Cube				
0,1	0	0	0	-	✓
0,2	0	0	-	0	✓
0,8	-	0	0	0	✓
1,3	0	0	-	1	✓
1,5	0	-	0	1	PI=D
2,3	0	0	1	-	✓
2,10	-	0	1	0	✓
8,10	1	0	-	0	✓
3,11	-	0	1	1	✓
5,13	-	1	0	1	PI=E
10,11	1	0	1	-	✓
11,15	1	-	1	1	PI=F
13,15	1	1	-	1	PI=G

Minterm	Cube				
0,1,2,3	0	0	-	-	PI=A
0,8,2,10	-	0	-	0	PI=C
2,3,10,11	-	0	1	-	PI=B

***Question:* Can this be done on a CCM? How modified?**

f^{on} = {A,B,C,D,E,F,G} = {00--, -01-, -0-0, 0-01, -101, 1-11, 11-1}

STEP 2 – Construct Cover Table

- PIs Along Vertical Axis (in order of # of literals)
- Minterms Along Horizontal Axis

	0	1	2	3	5	8	10	11	13	15
A	x	x	x	x						
B			x	x			x	x		
C	x		x			x	x			
D		x			x					
E					x				x	
F								x		x
G									x	x

NOTE: Table 4.2 in book is incomplete

STEP 2 – Finding the Minimum Cover

- Extract All Essential Prime Implicants, EPI
- EPIs are the PI for which a Single x Appears in a Column

	0	1	2	3	5	8	10	11	13	15
A	x	x	x	x						
B			x	x			x	x		
C	x		x			x	x			
D		x			x					
E					x				x	
F								x		x
G									x	x

- *C* is an EPI so: $f^{on}=\{C, ...\}$
- Row C and Columns 0, 2, 8, and 10 can be Eliminated Giving Reduced Cover Table
- Examine Reduced Table for New EPIs

STEP 2 – Reduced Table

Distinguished Column

	0	1	2	3	5	8	10	11	13	15
A	x	x	x	x						
B			x	x			x	x		
C	x		x			x	x			
D		x			x					
E					x				x	
F								x		x
G									x	x

Essential row

	1	3	5	11	13	15
A	x	x				
B		x		x		
D	x		x			
E			x		x	
F				x		x
G					x	x

- The Row of an EPI is an *Essential row*
- The Column of the Single x in the *Essential Row* is a *Distinguished Column*

Row and Column Dominance

- If Row P has x's Everywhere Row Q Does
  Then Q Dominates P if P has fewer x's

- If Column *i* has x's Everywhere *j* Does
  Then *j* Dominates *i* if *i* has fewer x's

- If Row P is equal to Row Q and Row Q does not cost more than Row P, eliminate Row P, or if Row P is dominated by Row Q and Row Q Does not cost more than Row P, eliminate Row P

- If Column *i* is equal to Column *j*, eliminate Column *i* or if Column *i* dominates Column *j*, eliminate Column *i*

STEP 3 – The Reduced Cover Table

Initially, Columns 0, 2, 8 and 10 Removed

	1	3	5	11	13	15
A	x	x				
B		x		x		
D	x		x			
E			x		x	
F				x		x
G					x	x

- No EPIs are Present
- No Row Dominance Exists
- No Column Dominance Exists
- This is *Cyclic Cover* Table
- Must Solve Exactly OR Use a Heuristic