Unit 7: Multi-Level Gate Circuits NAND and NOR Gates

7.1 Multi-Level Gate Circuits

Levels: maximum number of gates cascaded in series between a circuit input and an output

i.e. Two-level gate circuits
SOP and POS

Note: Normally invertors connected directly to input variables are NOT counted as a level in a circuit

Terminology

1. *AND-OR circuits:* means a two-level circuit composed of a level of AND gates followed by an OR gate at the output

2. *OR-AND circuit:* means a two-level circuit composed of a level of OR gates followed by an AND gate at the output

3. *OR-AND-OR circuits:* means a three-level circuit composed of a level of OR gates followed by a level of AND gates followed by an OR gate at the output

4. Circuit of AND and OR gates implies no particular ordering of the gates; the outputs gate may be either AND or OR.

Design Notes:

- Sometimes increasing the number of levels will reduce the number of gates thus reducing costs

- Increasing the number of levels may increase costs, (gate delay slowing the operation)

Tree Diagram

Each node represents a gate with the number of inputs written besides it

Mutiplying out to yield a 3-level

Ex. Multi-Level design using AND and OR gates

$$f(a,b,c,d) = \Sigma\, m\,(1, 5, 6, 10, 13, 14)$$

cd \ ab	00	01	11	10
00	0	0	0	0
01	1	1	1	0
11	0	0	0	0
10	0	1	1	1

Factoring:

POS from the Karnaugh:

Multiplying – using $(X + Y)(X + Z) = X + YZ$

Notes:

- In general one must find both the AND and OR gate solutions to obtain the minimum solution.

- If an expression f' has n-levels, the complement of the expression is an n-level expression or f

7.2 NAND and NOR Gates

Frequently used by designers because they are faster an use fewer components

NAND

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

n-input

$$F = (X_1 X_{2\ldots} X_n)' = X_1' + X_2' + \ldots + X_n'$$

Note: The output of F is 1 iff one or more of the inputs are 0

NOR

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

n-input

$$F = (X_1 + X_2 + \ldots + X_n)' = X_1'X_2'\ldots X_n'$$

Functionally Complete: name given to a set of logical operators where any Boolean funciton can be expressed in terms of this set of operators (AND, OR and NOT)

If a gate forms a functionally complete set by itself, then any switching function can be realized using only gates of that type

NAND and NOR gates are two such types

7.3 Design of Two-Level Circuits Using NAND and NOR Gates

Two-level AND and OR circuits convert to NAND and NOR circuits by

$$F = (F')'$$

Minimum sum of products:

Minimum product of sums

See Figure 7-11 on page 198 *Eight Basic Forms for Two-Level Circuits*

Because NAND and NOR gates are readily available in circuit form, two of the most common circuits are NAND-NAND and NOR-NOR

Procedure for designing a minimum two-level NAND-NAND circuit:

1. Find a minimum sum-of-products expression for F

2. Draw the corresponding two-level AND-OR circuit

3. Replace all gates with NAND gates, leaving the gate interconnections unchanged. If the output gate has any single literals as inputs, complement these literals.

Procedure for designing a minimum two-level NOR-NOR circuit:

1. Find a minimum product-of-sums expression for F

2. Draw the corresponding two-level OR-AND circuit

3. Replace all gates with NOR gates, leaving the gate interconnections unchanged. If the output gate has any single literals as inputs, complement these literals.

7.4 Design of Multi-Level NAND- and NOR-gate Circuits

Procedure for designing multi-level NAND circuits:

1. Simplify the switching function to be realized

2. Design a multi-level circuit of AND and OR gates. The output gate must be OR. AND-gate outputs cannot be used as AND gate inputs; OR-gate outputs cannot be used as OR-gate inputs.

3. Number the levels starting with the output gate as level 1. Replace all gates with NAND gates leaving the gate interconnections unchanged. Leave the inputs to levels 2, 4, 6… unchanged. Invert any literals which appear as inputs to levels 1, 3, 5….

Procedure for designing multi-level NOR circuits:

Same as the procedure for NAND gate circuits except the output gate in step 2 must be an AND gate.

7.5 Circuit Conversion Using Alternative Gate Symbols

Alternate ways of writing common gate symbols

These symbols can often facilitate analysis and design of NAND and NOR circuits

Ex. NAND gate circuit conversion

Procedure to convert to a NAND or NOR circuit

1. Convert all AND gates to NAND gates by adding an inversion bubble at the outputs. Convert all OR gates to NAND gates by adding an inversion bubble at the inputs. (To convert to NOR, add inversion bubbles at all OR gate outputs and all AND gate inputs)

2. Whenever an inverter output drives an inverted input, no further action s needed because the two inversions cancel.

3. Whenever a noninverted gate output drives an inverted gate output or vice versa, insert an inverter so the bubbles will cancel. (Choose an inverted with the bubble at the input or output as required.)

4. Whenever a variable drives an inverted input, complement the variable (or add an inverter) so the complementation cancels the inversion at the input.

Covert to NOR gates

Convert AND-OR to NAND

7.6 Design of Two-Level, Multiple-Output Circuits

Solutions of digital design often require several functions of the same variables which can be realized separately but ca be done so more economically.

Ex. Circuit 4-inputs with 3-outputs

$F_1 (A,B,C,D) = \Sigma m (11, 12, 13, 14, 15)$

$F_2 (A,B,C,D) = \Sigma m (3, 7, 11, 12, 13, 15)$

$F_3 (A,B,C,D) = \Sigma m (3, 7, 12, 13, 14, 15)$

Individual solutions realized using Karnaugh Maps

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

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

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

An obvious savings can be realized by using the same AB gate for F_1 and F_3 but…

By ACD from F_1 and A'CD from F_3, CD in F_2 is unnecessary if we OR the two together.

Note: When designing multi-output circuits try to minimize the number of gates for the solution. If the number of gates are equal then go for the minimum number of gate inputs.

Ex. Circuit 4-inputs with 3-outputs

$F_1 (A,B,C,D) = \Sigma\, m\,(2, 3, 5, 7, 8, 9, 10, 11, 13, 15)$

$F_2 (A,B,C,D) = \Sigma\, m\,(2, 3, 5, 6, 7, 10, 11, 14, 15)$

$F_3 (A,B,C,D) = \Sigma\, m\,(6, 7, 8, 9, 13, 14, 15)$

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

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

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

By inspection

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

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

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

A'BD can be used for F_1 and F_2 ABD and AB'C' can be used for F_1 and F_3

Determination of Essential Prime Implicants for Multiple-Output Realization

An essential prime implicant for a single function may not be an essential prime implicant to a multi-output function

To determine, check each 1 on the map to see if it is covered by only one prime implicant, but only those 1's which do not appear on the other functions maps.

Individual solution

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

F1

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

F2

Best solution

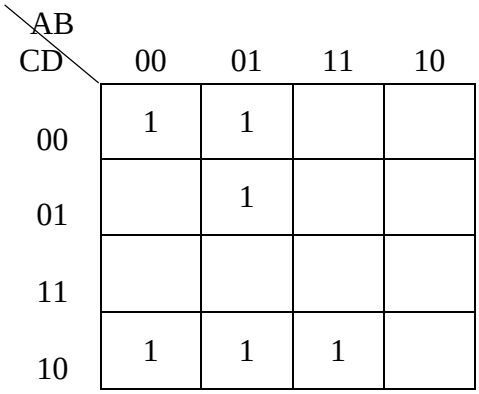
AB / CD	00	01	11	10
00				
01	1	1	1	1
11			1	
10				

F1

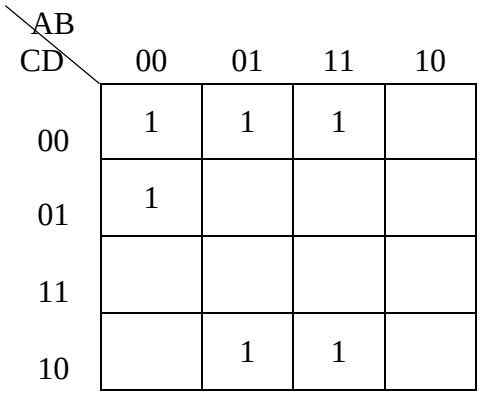
AB / CD	00	01	11	10
00		1	1	
01				
11			1	
10		1	1	

F2

Individual solution

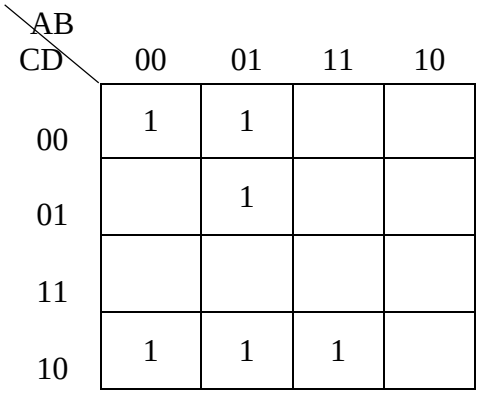


F1

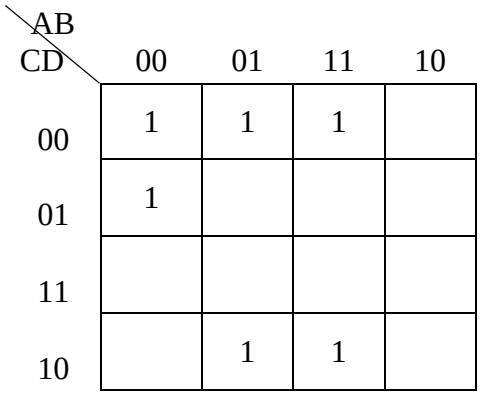


F2

Best solution



F1



F2

7.7 Multiple-Output NAND and N0R Circuits

The procedure for design of multi-output systems is the same as in section 7.4 for single output functions.

If all outputs are OR's then directly convert to NANDs

If all outputs are AND's then directly convert to NORs

Ex. Multi-level Circuit converted to NOR gates