09-Multi-Level Gate Circuits

Text: Unit 7

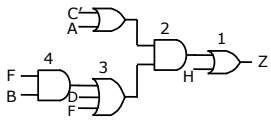
ECEGR/ISSC 201
Digital Operations and Computations
Winter 2011

Overview

- Multi-Level Gate Circuits
- Multi-Output Circuits
- Two-Level NAND and NOR Circuits
- Multi-Level NAND and NOR Circuits

Multi-Level Gate Circuits

- Levels: largest number of cascaded gates between an input and the output in a circuit
- How many levels?
  - four



Multi-Level Gate Circuits

- PoS or SoP forms result in a two level circuit
- Assume that inputs and their complements are available (inverters do not count)

Multi-Level Gate Circuits

- Notation
  - AND-OR: two level circuit with AND gates followed by OR gates
  - OR-AND: two level circuit with OR gates followed by AND gates
  - OR-AND-OR: three level circuit with OR gates followed by AND gates followed by OR gates
  - etc

Multi-Level Gate Circuits

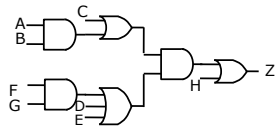
- Increasing or decreasing the number of levels can decrease the number of gates, but may also increase the overall cost
- Propagation delays may limit the number of levels
  - time period between input application and output response

Multi-Level Gate Circuits

- Number of gates, inputs and levels can be determined by inspecting the logic expression
- Example:
  - Z = (AB+C)(D+E+FG)+H

Multi-Level Gate Circuits

- Z = (AB+C)(D+E+FG)+H
- 4 levels
- 6 gates
- 13 inputs

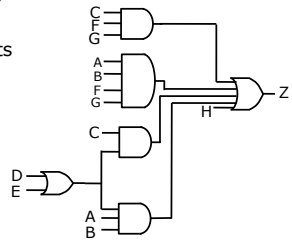


Multi-Level Gate Circuits

- Expression can be factored to yield different number of levels, gates, inputs
- Z = (AB+C)(D+E+FG)+H
  = AB(D+E) +C(D+E)+ABFG+CFG+H

Multi-Level Gate Circuits

- Z = AB(D+E) +C(D+E)+ABFG+CFG+H
- 3 Levels
- 6 Gates
- 19 inputs



Example

- Solution approach
  - Construct K-map
  - Derive the expression in AND-OR form
    - Reduce gates/inputs if possible by factoring
  - Derive the expression in OR-AND form
    - Reduce gates/inputs if possible by multiplying out

Example

- F(A, B, C, D) = Σm(1, 5, 6, 10, 13, 14)

CD \ AB	00	01	11	10
00	0	4	12	8
01	1	5	13	9
11	3	7	15	11
10	2	6	14	10

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

Example

- Derive the expression in AND-OR form:

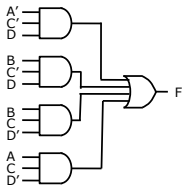
Example

- Derive the expression in AND-OR form:
- F = A'C'D + BC'D + BCD' + ACD'

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

Example

- F = A'C'D + BC'D + BCD' + ACD'
- 2 levels,5 gates, 16 gate inputs

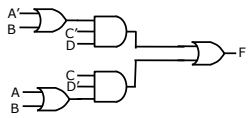


Example

- Reduce gates/inputs if possible
- F = A'C'D + BC'D + BCD' + ACD'
- F = C'D(A' +B) + CD'(A + B) [by factoring]

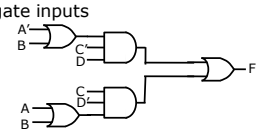
Example

- F = C'D(A' + B) + CD'(A + B)
- How many levels, gates and inputs?



Example

- F = C'D(A' + B) + CD'(A + B)
- How many levels, gates and inputs?
  - 3 levels
  - 5 gates
  - 12 gate inputs



Example

- Both realizations have an OR gate output
- A realization with an AND gate output might have fewer inputs or gates
- To do this find the OR-AND

Example

- Derive the expression in OR-AND form:

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

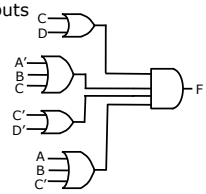
Example

- OR-AND gate realization
- F′ = C′D′ + AB′C′ + CD + A′B′C
- F = (C+D)(A′ + B + C)(C′ + D′)(A + B + C′)

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

Example

- F = (C+D)(A′ + B + C)(C′ + D′)(A + B + C′)
  - 2 levels
  - 5 gates
  - 14 gate inputs



Example

- We can find a 3 level expression by multiplying out
  - F = (C+D)(A′ + B + C)(C′ + D′)(A + B + C′)
  - F = (C + D(A′+B))(C′ + D′(A + B))
- How many gates and gate inputs does this have?
  - 7 gates
  - 16 gate inputs

Example

- Which should we construct?
  - F = A′C′D + BC′D + BCD′ + ACD′
    - 2 levels; 5 gate; 16 gate inputs
  - F = C′D(A′ + B) + CD′(A + B)
    - 3 levels; 5 gates; 12 gate inputs
  - F = (C+D)(A′ + B + C)(C′ + D′)(A + B + C′)
    - 2 levels; 5 gates; 14 gate inputs
  - F = (C + A′D+BD)(C′ + AD′ + BD′)
    - 3 levels; 7 gates; 16 gate inputs

Multi-Level Gate Circuits

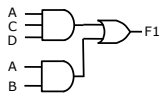
- AND-OR: SoP form
- OR-AND: PoS form
- In general find both the AND and OR gate output realizations for the expression
- Note: if F' has n-levels, then F will have n-levels
- Find an n-level expression for F' (with OR output gate) and then complement the expression to find the AND output circuit

Design of Two-Level, Multiple-Output Circuits

- Some logic design problems require multiple outputs based on the same input variables
- Example:
  - F1(A, B, C, D) = Σm(11, 12, 13, 14, 15)
  - F2(A, B, C, D) = Σm(3, 7, 11, 12, 13, 15)
  - F3(A, B, C, D) = Σm(3, 7, 12, 13, 14, 15)
- Basic approach:
  - Realize each function separately
  - Identify common gates
  - Eliminate common gates

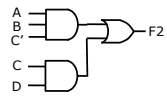
Design of Two-Level, Multiple-Output Circuits

- F1(A, B, C, D) = Σm(11, 12, 13, 14, 15)
  - F1 = AB + ACD
- F2(A, B, C, D) = Σm(3, 7, 11, 12, 13, 15)
- F3(A, B, C, D) = Σm(3, 7, 12, 13, 14, 15)



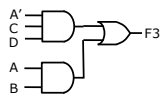
Design of Two-Level, Multiple-Output Circuits

- F1(A, B, C, D) = Σm(11, 12, 13, 14, 15)
  - F1 = AB + ACD
- F2(A, B, C, D) = Σm(3, 7, 11, 12, 13, 15)
  - F2 = ABC' + CD
- F3(A, B, C, D) = Σm(3, 7, 12, 13, 14, 15)



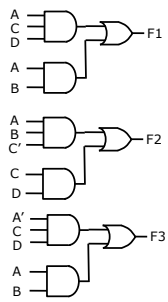
Design of Two-Level, Multiple-Output Circuits

- F1(A, B, C, D) = Σm(11, 12, 13, 14, 15)
  - F1 = AB + ACD
- F2(A, B, C, D) = Σm(3, 7, 11, 12, 13, 15)
  - F2 = ABC' + CD
- F3(A, B, C, D) = Σm(3, 7, 12, 13, 14, 15)
  - F3 = A'CD + AB



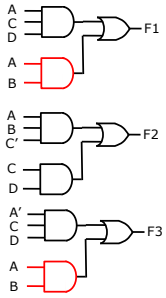
Design of Two-Level, Multiple-Output Circuits

- F1 = AB + ACD
- F2 = ABC' + CD
- F3 = A'CD + AB
- Realization:
  - 9 gates
  - 21 inputs
- How can we simplify this?



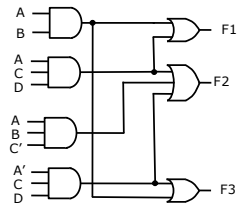
Design of Two-Level, Multiple-Output Circuits

- AB AND gate is redundant
- Also
  - ACD is needed for F1
  - A'CD is needed for F3
  - Replace CD with A'CD + ACD in F2



Design of Two-Level, Multiple-Output Circuits

- F1 = AB + ACD
- F2 = ACD + ABC' + A'CD
- F3 = A'CD + AB
- Realization:
  - 7 gates
  - 18 inputs



A Few Notes

- F2 is not in minimum SoP form
- Minimum sum of prime implicants does not necessarily lead to the minimal cost realization in multiple output circuits
- Gate minimization is the first priority
- Gate input minimization is the second priority
- K-maps can be used to for additional insight

Example

- Consider
  - F1(A, B, C, D) = Σm(2, 3, 5, 7, 8 ,9, 10, 11, 13, 15)
  - F2(A, B, C, D) = Σm(2, 3, 5, 6, 7, 10, 11, 14, 15)
  - F3(A, B, C, D) = Σm(6, 7, 8, 9, 13, 14, 15)
- First create a K-map for each output

Example

- F1(A, B, C, D) = Σm(2, 3, 5, 7, 8 ,9, 10, 11, 13, 15)

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

Example

- F2(A, B, C, D) = Σm(2, 3, 5, 6, 7, 10, 11, 14, 15)

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

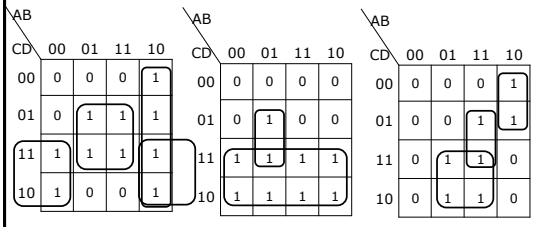
Example

- F3(A, B, C, D) = Σm(6, 7, 8, 9, 13, 14, 15)

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

Example

- If each are minimized separately



F1 = BD + B'C +AB' F2 = C + A'BD F3 = BC + AB'C' + ABD

Example

- If each are minimized separately

  F1 = BD + B'C +AB'

  F2 = C + A'BD

  F3 = BC + AB'C' + ABD

- This results in how many gate and gate inputs?

Example

- If each are minimized separately

  F1 = BD + B'C +AB'

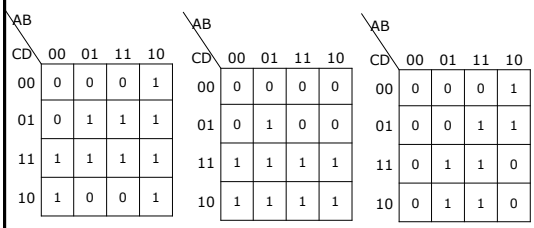
  F2 = C + A'BD

  F3 = BC + AB'C' + ABD

- This results in how many gate and gate inputs?
  - 10 gates
  - 25 gate inputs

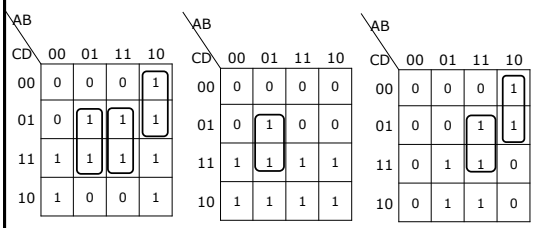
Example

- Instead, try looking for common groupings



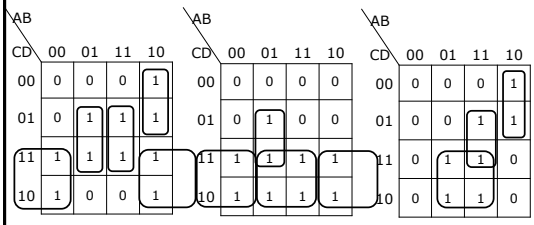
Example

- Instead, try looking for common groupings



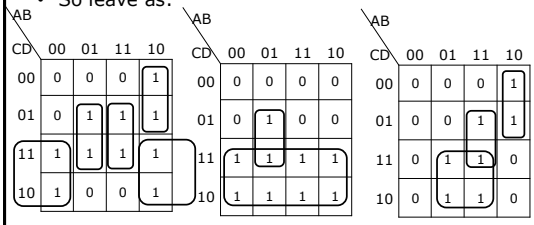
Example

- Does this grouping help?



Example

- No! Since C does not require a gate anyway
- So leave as:



Example

- The resulting functions are:
  - F1 = A'BD + ABD + AB'C' + B'C
  - F2 = C + A'BD
  - F3 = BC + AB'C' + ABD
- This results in
  - 8 gates
  - 22 gate inputs
- This is superior to the previous expressions
  - 10 gates
  - 25 gate inputs

Example

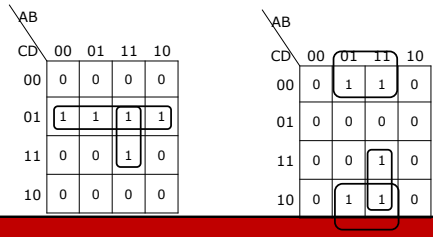
- How many terms can be reused?
  - F1 = A'BD + ABD + AB'C' + B'C
  - F2 = C + A'BD
  - F3 = BC + AB'C' + ABD

Design of Two-Level, Multiple-Output Circuits

- It is not always best to combine 1s
- However, the solution with the maximum number of common terms is not always the best

Design of Two-Level, Multiple-Output Circuits

- Try combining as many 1s as possible
- F1 = C'D + ABD
- F2 = D'B + ABC



Design of Two-Level, Multiple-Output Circuits

- How many gates and gate inputs are required?
  - F1 = C'D + ABD
  - F2 = D'B + ABC
  - 6 gates
  - 14 gate inputs

Design of Two-Level, Multiple-Output Circuits

- Next, we will see if we can reduce the number of gate/gate inputs using essential prime implicants
- Determination of essential prime implicants for multiple output circuits
  - Essential prime implications to one output may not be essential to the multiple output realization

Design of Two-Level, Multiple-Output Circuits

- Similar to procedure for single output, but
  - Check 1s to see if they are essential if the 1s do not appear on the other K-maps

Design of Two-Level, Multiple-Output Circuits

- In F1, the 1s to be considered are

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

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

Design of Two-Level, Multiple-Output Circuits

- In F2, the 1s to be considered are

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

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

Design of Two-Level, Multiple-Output Circuits

- F1 = C'D + ABCD
- F2 = D'B + ABCD

Not looped because they appear on both maps

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

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

Design of Two-Level, Multiple-Output Circuits

- How many gates and gate inputs are required?
  - F1 = C'D + ABCD
  - F2 = D'B + ABCD
  - 5 gates
  - 12 gate inputs
- Recall if all 1s were combined:
  - F1 = C'D + ABD
  - F2 = D'B + ABC
  - 6 gates
  - 14 gate inputs

Design of Two-Level, Multiple-Output Circuits

- Next, we show that the solution that combines the most common 1s is not necessarily the best

Design of Two-Level, Multiple-Output Circuits

- The common terms are:

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

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

Design of Two-Level, Multiple-Output Circuits

- Looping the remaining 1s

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

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

Design of Two-Level, Multiple-Output Circuits

- F1 = A'C'D' + A'BC' + CD'A' + CD'B
- F2 = A'C'D' + A'B'C' + C'D'B' + CD'B
- 8 gates, 26 gate inputs

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

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

Design of Two-Level, Multiple-Output Circuits

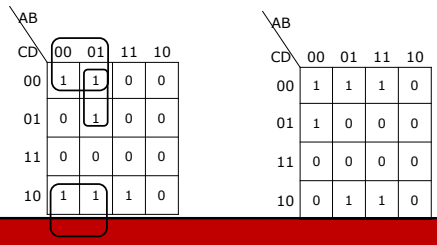
- Now use the essential implicant method
- In F1, the 1s to be considered are

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

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

Design of Two-Level, Multiple-Output Circuits

- The essential prime implicants that cover these 1s are:

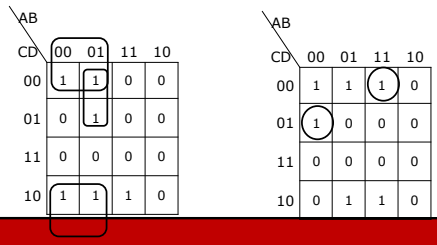


CD \ AB	00	01	11	10
00	1	1	0	0
01	0	1	0	0
11	0	0	0	0
10	1	1	1	0

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

Design of Two-Level, Multiple-Output Circuits

- In F2, the 1s to be considered are

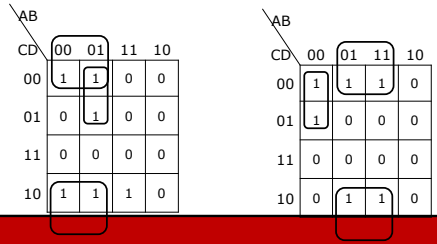


CD \ AB	00	01	11	10
00	1	1	0	0
01	0	1	0	0
11	0	0	0	0
10	1	1	1	0

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

Design of Two-Level, Multiple-Output Circuits

- The essential prime implicants are

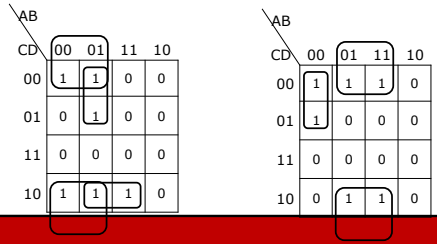


CD \ AB	00	01	11	10
00	1	1	0	0
01	0	1	0	0
11	0	0	0	0
10	1	1	1	0

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

Design of Two-Level, Multiple-Output Circuits

- Looping the remaining 1s
  - 7 gates
  - 18 inputs



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

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

Design of Two-Level, Multiple-Output Circuits

- Combining common 1s resulted in
  - 8 gates
  - 26 gate inputs
- Essential implicant method
  - 7 gates
  - 18 gates inputs

Two-Level NAND and NOR Gate Circuits

- PoS and SoP forms utilize AND and OR gates
- How do we convert these to circuits using NAND and NOR?
- Recall DeMorgan's Laws
  - $(X_1 + X_2 + \ldots + X_n)' = X_1'X_2'\ldots X_n'$
  - $(X_1X_2\ldots X_n)' = X_1' + X_2' + \ldots + X_n'$

Two-Level NAND and NOR Gate Circuits

- Consider building F = A + BC' + B'CD using NAND and/or NOR gate(s)
- In NAND-NAND form:
  - F = A + BC' + B'CD = [(A + BC' + B'CD)']'
  - Using F = [(F')]'
  - F = [A'(BC')'(B'CD)']'
- In OR-NAND
  - F = [A'(BC')'(B'CD)']' = [A'(B'+C)(B+C'+D')]'
- In NOR-OR form:
  - F = A + (B'+C)' + (B+C'+D')'

Two-Level NAND and NOR Gate Circuits

- To obtain the other forms, start with a PoS form
  - F = A + BC' + B'CD =(A+B+C)(A+B'+C')(A+C'+D)
- NOR-NOR
  - F ={[(A+B+C)(A+B'+C')(A+C'+D)]'}'
  - Using F ={[F]'}'
  - F={(A+B+C)'+(A+B'+C')'+(A+C'+D)'}'

Two-Level NAND and NOR Gate Circuits

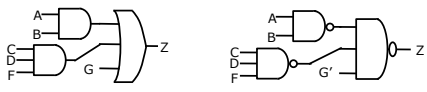
- AND-NOR
  - F=(A'B'C' + A'BC + A'CD')'
- NAND-AND
  - F = (A'B'C')'(A'BC)'(A'CD')'

Two-Level NAND and NOR Gate Circuits

- Procedure for obtaining two-level NAND-NAND circuits
  1. Find minimum SoP form of F
  2. Draw the two-level AND-OR circuit
  3. Replace all gates with NAND gates. If the output has any single literal inputs, complement those inputs

Two-Level NAND and NOR Gate Circuits

- Example:



Two-Level NAND and NOR Gate Circuits

- Procedure for obtaining two-level NOR-NOR circuits
  1. Find minimum PoS form of F
  2. Draw the two-level OR-AND circuit
  3. Replace all gates with NOR gates. If the output has any single literal inputs, complement those inputs

Example

- Find the NAND-NAND representation of F = ABC + B'C' + D

Example

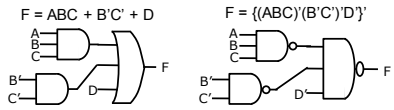
- Find the NAND-NAND representation of F = ABC + B'C' + D
  F = {(ABC + B'C' + D)'}'
  F = {(ABC)'(B'C')'D'}'

Example

- Draw the NAND-NAND circuit for F = ABC + B'C' + D

Example

- Draw the NAND-NAND circuit for F = ABC + B'C' + D



Design of Multi-Level NAND Circuits

1. Simplify switching function
2. Design multi-level circuit of AND and OR gates
   - Output gate must be OR
   - AND gate outputs cannot be input into other AND gates
   - OR gate outputs cannot be input into other OR gates
3. Replace all gates with NAND gates
4. Invert any literals that are inputs to gates in odd numbered levels (1, 3, 5, …)

Design of Multi-Level NOR Circuits

1. Simplify switching function
2. Design multi-level circuit of AND and OR gates
   - Output gate must be AND
   - AND gate outputs cannot be input into other AND gates
   - OR gate outputs cannot be input into other OR gates
3. Replace all gates with NOR gates
4. Invert any literals that are inputs to gates in odd numbered levels (1, 3, 5, …)

Multiple Output NAND and NOR Circuits

- Design procedure for single output NAND and NOR circuits also applies to multiple output circuits

Alternative Logic Symbols

- Read Section 7.2 of the text