## CDA 3200 Digital Systems

Instructor: Dr. Janusz Zalewski
Developed by: Dr. Dahai Guo Spring 2012

## Outine

- Multi-Level Gate Circuits
- NAND and NOR Gates
- Design of Two-Level Circuits Using NAND and NOR Gates


## Multi-Level Gate Circuits (1/3)

- The maximum number of gates cascaded in series between a circuit input and the output is referred to as the number of levels of gates.
- sum-of-products: two levels
- product-of-sums: two levels
- Inverters are not counted.


## Multi-Level Gate Circuits (2/3)



## Multi-Level Gate Circuits (3/3)

- When the input of a gate is changed, there is a finite time before the output changes.
- The number of gates which can be cascaded is limited by gate delays.
- When several gates are cascaded, the gate delay may become excessive and slow down the operation of the digital system.


## NAND and NOR Gates (1/5)

- NAND gate

(a) 3-input NAND gate

(b) NAND gate equivalent

(c) $n$-input NAND gate
$-F(A B C)=(A B C)^{\prime}=A^{\prime}+B^{\prime}+C^{\prime}$
- NOR gate

(a) 3-input NOR gate

(b) NOR gate equivalent

(c) $n$-input NOR gate
$-F(A B C)=(A+B+C)^{\prime}=A^{\prime} B^{\prime} C^{\prime}$
- Any logic function can be implemented using only NAND or NOR gates.


## NAND and NOR Gates (2/5)

- If we can use NAND or NOR gates to implement AND, OR, and inverter, then we can prove that any logic function can be expressed using only NAND or NOR gates.
- A set of logic operations is said to be functionally complete if any Boolean function can be expressed in terms of this set of operations. The set AND, OR, and NOT is obviously functionally complete.


## NAND and NOR Gates (3/5)


$X^{\prime}=X$ nand $X$
$A B=(A$ nand $B)$ nand ( $A$ nand $B$ )
$A+B=(A$ nand $A)$ nand $(B$ nand $B)$

## NAND and NOR Gates (4/5)

- Actually, as long as we could show NAND can express OR and NOT (AND and NOT), we can show NAND is functionally complete.
$-X Y=\left(X^{\prime}+Y^{\prime}\right)^{\prime}$
$-X+Y=\left(X^{\prime} Y^{\prime}\right)^{\prime}$


## NAND and NOR Gates (5/5)

- Can you prove that NOR is functionally complete?


## Design of Two-Level Circuits Using NAND and NOR Gates (1/7)

- The conversion from circuits composed of AND and OR gates to circuits composed of NAND or NOR gates is carried out by using $F=\left(F^{\prime}\right)$ ' and then applying DeMorgan's laws:
$-\left(X_{1}+X_{2}+\ldots+X_{n}\right)^{\prime}=X_{1}{ }^{\prime} X_{2}^{\prime} \ldots X_{n}{ }^{\prime}$
$-\left(X_{1} X_{2} \ldots X_{n}\right)^{\prime}=X_{1}^{\prime}+X_{2}^{\prime}+\ldots+X_{n}^{\prime}$


## Design of Two-Level Circuits Using NAND and NOR Gates (2/7)

- $F=A+B C^{\prime}+B^{\prime} C D$
- $=\left[\left(A+B C^{\prime}+B^{\prime} C D\right)^{\prime}\right]^{\prime}$
(F')'
- $=\left[A^{\prime}\left(B C^{\prime}\right)^{\prime}\left(B^{\prime} C D\right)^{\prime}\right]^{\prime}$
- $=\left[A^{\prime}\left(B^{\prime}+C\right)\left(B+C^{\prime}+D^{\prime}\right)\right]^{\prime}$
- $=A+\left(B^{\prime}+C\right)^{\prime}+\left(B+C^{\prime}+D^{\prime}\right)^{\prime}$

NAND-NAND
OR-NAND

- We started with the minimum sum-of-products expression.


## Design of Two-Level Circuits Using NAND and NOR Gates (3/7)

- Procedure for designing a minimum two-level NAND-NAND circuit
- Find a minimum sum-of-products expression for F: Karnaugh maps, Quinne-McCluskey and Petrick methods
- Draw the corresponding two-level AND-OR circuit
- Replace all gates with NAND gates leaving the gate interconnections unchanged.
- If the output gate has any single literals as inputs, complement these literals.


## Design of Two-Level Circuits Using NAND and NOR Gates (4/7)


(a) Before transformation

(b) After transformation

## Design of Two-Level Circuits Using NAND and NOR Gates (5/7)

- $F=(A+B+C)\left(A+B^{\prime}+C^{\prime}\right)\left(A+C^{\prime}+D\right)$
- $=\left\{\left[(A+B+C)\left(A+B^{\prime}+C^{\prime}\right)\left(A+C^{\prime}+D\right)\right]^{\prime}\right\}^{\prime}$
( $\mathrm{F}^{\prime}$ )'
- $=\left[(A+B+C)^{\prime}+\left(A+B^{\prime}+C^{\prime}\right)^{\prime}+\left(A+C^{\prime}+D\right)^{\prime}\right]^{\prime} \quad$ NOR-NOR
- $=\left(A^{\prime} B^{\prime} C^{\prime}+A^{\prime} B C+A^{\prime} C D\right)^{\prime}$
- $=\left(A^{\prime} B^{\prime} C^{\prime}\right)^{\prime}\left(A^{\prime} B C\right)^{\prime}\left(A^{\prime} C D\right)^{\prime}$

AND-NOR
NAND-AND

- We started with the minimum product-of-sums expression.


## Design of Two-Level Circuits Using NAND and NOR Gates (6/7)

- Procedure for designing a minimum twolevel NOR-NOR circuit
- Find a minimum product-of-sums expression for F
- Draw the corresponding two-level OR-AND for F
- Replace all gates with NOR gates leaving the gate interconnections unchanged.
- If the output gate has any single literals as inputs, complement these literals.


## Design of Two-Level Circuits Using NAND and NOR Gates (7/7)


(a) Before Transformation

(b) After Transformation

## Design of Two-Level, MultipleOutput Circuits (1/10)

- Given a logic function, we can simplify it using some methods.
- But if we need to design a circuit to implement several functions, it may not be enough to simplify each function separately.
- Example:
- F1 (A,B,C,D)=sum[m(11,12,13,14,15)]
- F2 $(A, B, C, D)=\operatorname{sum}[m(3,7,11,12,13,15)]$
- F3(A,B,C,D)=sum[m(3,7,12,13,14,15)]


## Design of Two-Level, MultipleOutput Circuits (2/10)


$F 1=A B+A C D$

$F 2=A B C^{\prime}+C D$

| $C D$ |  |  |  |  |
| :---: | :---: | :---: | :---: | :---: |
|  | 00 | 01 | 11 | 10 |
| 00 |  |  | 1 |  |
| 01 |  |  | 1 |  |
| 11 | 1 | 1. | 1 |  |
| 10 |  |  | 1 |  |
|  |  |  |  |  |

$F 3=A^{\prime} C D+A B$

## Design of Two-Level, Multiple-Output Circuits (3/10)



## Design of Two-Level, MultipleOutput Circuits (4/10)



## Design of Two-Level, MultipleOutput Circuits (5/10)

- Determination of Essential Prime Implicants for Multiple-Output Realization
- Some of the prime implicants essential to an individual function may not be essential to the multiple-output realization.


## Design of Two-Level, MultipleOutput Circuits (6/10)


ab' and bd are essential to F1, but not to the multipleoutput system.

## Design of Two-Level, MultipleOutput Circuits (7/10)

- Determination of Essential Prime Implicants for Multiple-Output Realization
- Some of the prime implicants essential to an individual function may not be essential to the multiple-output realization.
- When checking 1's for essential prime implicants, we only check those 1's which do not appear on the other function maps.


## Design of Two-Level, MultipleOutput Circuits (8/10)


(a) Best solution

(b) Solution requires and extra gate

## Design of Two-Level, MultipleOutput Circuits (9/10)



This 1 appears on both maps.
When checking 1's for essential prime implicants, we only check those 1's which do not appear on the other function maps.

## Design of Two-Level, MultipleOutput Circuits (10/10)



- In this picture, no minterm only appears in one Karnaugh map.
- How to find essential prime implicants for a multi-output system is not discussed in the textbook.

