Topic 7: Combinational Circuits

Readings:
•Patterson & Hennessy Appendix B

Goals
•Boolean algebra
•Sum of products and products of sum form
•Karnaugh maps

Physical to Logical Level

4 Representations

•Circuits and gate diagrams correspond one to one
•We can map gate diagrams to TTs. How about TT's into gate diagrams?
Need a tool: boolean algebra
•How to manipulate TTs ?
Translate to boolean algebra and manipulate

Boolean Algebra

Algebra: Set of Elements, Set of Functions, Set of Axioms

•Elements: $\{0,1\}$
•Functions: AND (like multiply), OR (like add), NOT (bar)
•Identities:

$0X = 0 \quad 1 + X = 1$
$1X = X \quad 0 + X = X$

•Operator Axioms

$X\bar{X} = X \quad X + \bar{X} = X$
$X\bar{X} = 0 \quad X + \bar{X} = 1$

•AND and OR are commutative:

$A + B = B + A$, $AB = BA$

•AND and OR obey distributive law:

$A(B + C) = AB + AC$
$A + (BC) = (A + B)(A + C)$

•AND takes precedence over OR

$A + BC = A + (BC)$

•AND and OR obey associative law:

$A + (B + C) = (A + B) + C$
$A(BC) = (AB)C$

DeMorgan's laws

Proving Boolean Equations

Method #1: Algebraic method $(X + Y)(X + Z) = X + YZ$

Method #2: Truth tables $(X + Y)(X + Z) = X + YZ$

X	Y	Z	X+Y	X+Z	LHS	YZ	RHS
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

Generating Boolean Equations from Truth Tables

• Find all rows with F=1
• AND them together to make a product term. These are called minterms.
• OR the minterms together F = $\overline{A}BC + A\overline{B}C + ABC$

A	B	C	F	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{A}BC$
1	0	0	1	$A\overline{B}C$
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

Boolean Equations from Truth Tables (cont.)

Another Example F = $\overline{A}\overline{B}\overline{C}D + \overline{A}\overline{B}CD + \overline{A}B\overline{C}\overline{D} + \overline{A}BCD + A\overline{B}\overline{C}\overline{D} +$
$A\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}\overline{D} + ABCD$

A	B	C	D	F	
0	0	0	0	0	
0	0	0	1	1	$\overline{A}\overline{B}\overline{C}D$
0	0	1	0	0	
0	0	1	1	1	$\overline{A}\overline{B}CD$
0	1	0	0	1	$\overline{A}B\overline{C}\overline{D}$
0	1	0	1	0	
0	1	1	0	0	
0	1	1	1	1	$\overline{A}BCD$
1	0	0	0	1	$A\overline{B}\overline{C}\overline{D}$
1	0	0	1	0	
1	0	1	0	1	$A\overline{B}C\overline{D}$
1	0	1	1	1	$A\overline{B}CD$
1	1	0	0	1	$AB\overline{C}\overline{D}$
1	1	0	1	0	
1	1	1	0	0	
1	1	1	1	1	$ABCD$

Canonical Form

• Truth Table provides a signature for the boolean function.
    Is there an equivalent algebraic signature?

Sum of Products form (SOP).
    • Also known as disjoint normal form or minterm expansion.
    • This is what we just did in the last two examples.

Product of Sums form (POS). AKA maxterm expansion.
    • Find SOP for rows where F=0: Invert the entire expression, and apply De Morgan's law to get POS. Each sum is called a maxterm.

A	B	C	F	
0	0	0	0	$A+B+C$
0	0	1	0	$A+B+\overline{C}$
0	1	0	0	$A+\overline{B}+C$
0	1	1	1	
1	0	0	1	
1	0	1	0	$\overline{A}+B+\overline{C}$
1	1	0	0	$\overline{A}+\overline{B}+C$
1	1	1	1	

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

Canonical Form (cont.)

A	B	C	D	F	
0	0	0	0	0	$A+B+C+D$
0	0	0	1	1	
0	0	1	0	0	$A+B+\overline{C}+D$
0	0	1	1	1	
0	1	0	0	1	
0	1	0	1	0	$A+\overline{B}+C+\overline{D}$
0	1	1	0	0	$A+\overline{B}+C+D$
0	1	1	1	1	
1	0	0	0	1	
1	0	0	1	0	$\overline{A}+B+C+\overline{D}$
1	0	1	0	1	
1	0	1	1	1	
1	1	0	0	1	
1	1	0	1	0	$\overline{A}+\overline{B}+C+\overline{D}$
1	1	1	0	0	$\overline{A}+\overline{B}+\overline{C}+D$
1	1	1	1	1	

$(A+B+C+D)(A+B+\overline{C}+D)(A+\overline{B}+C+\overline{D})$
$(A+\overline{B}+C+D)(\overline{A}+B+C+\overline{D})(\overline{A}+\overline{B}+C+\overline{D})$
$(\overline{A}+\overline{B}+\overline{C}+D)$
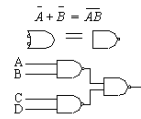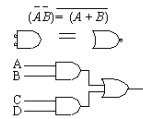
Implementing SOP using only NAND gates

Why just NAND gates?
- •AND gates are usually just NAND gates with an inverter.
- •INVERTER can be made by wiring together inputs of NAND gates
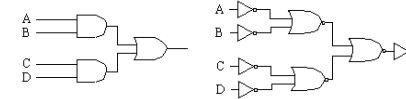
A —[>o— F
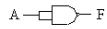
A	B	F
0	0	1
0	1	0
1	0	0
1	1	0

"Pushing a Bubble" through an AND changes it to an OR, and vice versa

$(\overline{AB}) = (\overline{A} + \overline{B})$

$\overline{A} + \overline{B} = \overline{AB}$

Implementing SOP using only NOR gates

What goes for NAND gates, goes for NOR gates:
- •Inverters can be made by wiring gates
- •OR gate is sometimes implemented as NOR plus INVERTER
- •Unfortunately, circuits are not very clean

What about POS using only NOR? $F = (B + \overline{C})(\overline{B} + C)(A + C)$
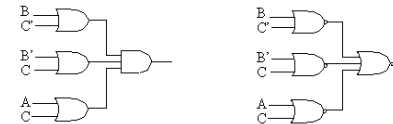
Canonical Form is not minimal form

Example:

```
A B C F
0 0 0 0
0 0 1 0
0 1 0 0
0 1 1 1
1 0 0 1
1 0 1 1
1 1 0 1
1 1 1 1
```

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

$F = A + BC$

Question: how to find minimal form?

Finding the minimal form

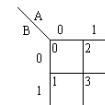Goals is to reduce the number of literals in a boolean equation.
A literal is a variable and its complement in an equation.
- •A1: Use Boolean Algebra. Hard to know when you're "done."
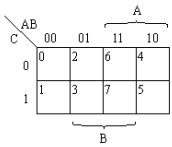- •A2: Use Karnaugh maps

Karnaugh Maps

Graphical way to unify terms

Example #1: $F = A\overline{B} + AB$

More Karnaugh Maps

Example #2: $F = \bar{A}BC + A\bar{B}\bar{C} + A\bar{B}C + AB\bar{C} + ABC$

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

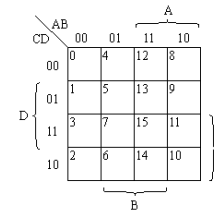(AB across top labeled A, C down side, B bracket)

Goal:
- circle as few rectangles of 1's as possible, covering all 1's
- but: rectangle sides must be power-of-two in size
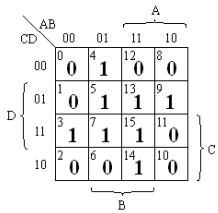(e.g., 1x1, 1x2, 2x2, 1x4, 2x4, 4x4)

More Karnaugh Maps

```
A B C D F
0 0 0 0 0
0 0 0 1 1
0 0 1 0 0
0 0 1 1 1
0 1 0 0 1
0 1 0 1 0
0 1 1 0 0
0 1 1 1 1
1 0 0 0 1
1 0 0 1 0
1 0 1 0 1
1 0 1 1 1
1 1 0 0 1
1 1 0 1 0
1 1 1 0 0
1 1 1 1 1
```

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

Rules of thumb for finding minimal expressions

Suppose I have this K-Map

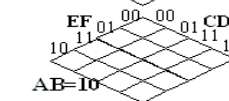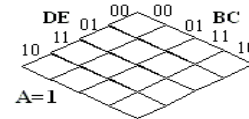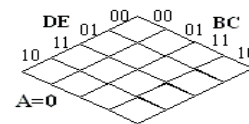| CD \ AB	00	01	11	10
00	0	1	0	0
01	0	1	1	1
11	1	1	1	0
10	0	0	1	0

(cells numbered 0,4,12,8 / 1,5,13,9 / 3,7,15,11 / 2,6,14,10)

- Largest subcube to smallest?
- Smallest subcube to largest?

---

5 and 6 variable K-maps

Just a stack of 4-var K-maps

A=0, A=1 (DE/BC maps); AB=00, AB=01, AB=11, AB=10 (EF/CD maps)

Mapping real problems to boolean equations

"Design a circuit for a digital clock that computes the next hour on its output, given the current hour as input."
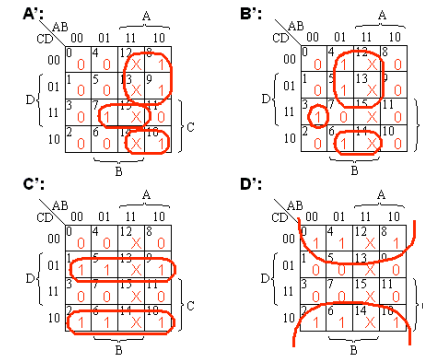
Assumption: Hours are represented as 4 bit binary number

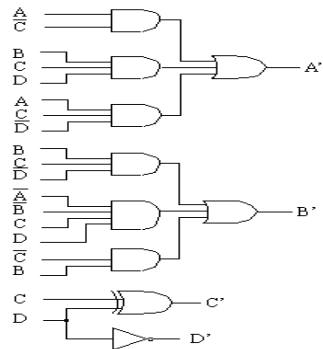| A	B	C	D	A'	B'	C'	D'
0	0	0	0	0	0	0	1
0	0	0	1	0	0	1	0
0	0	1	0	0	0	1	1
0	0	1	1	0	1	0	0
0	1	0	0	0	1	0	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	1	1
0	1	1	1	1	0	0	0
1	0	0	0	1	0	0	1
1	0	0	1	1	0	1	0
1	0	1	0	1	0	1	1
1	0	1	1	0	0	0	0
1	1	0	0	X	X	X	X
1	1	0	1	X	X	X	X
1	1	1	0	X	X	X	X
1	1	1	1	X	X	X	X

Digital Clock Example

Karnaugh Maps:

not the best!

Digital Clock Circuit

Summary