BROWN
School of Engineering

# DIGITAL ELECTRONICS SYSTEM DESIGN

**FALL 2019**

**PROF. IRIS BAHAR**

SEPTEMBER 16, 2019

LECTURE 4: KARNAUGH MAPS & LOGIC MINIMIZATION

## LABORATORY ASSIGNMENTS

- Lab kits and lab manuals are now ready for pickup
  - See George Worth in B&H325 for pickup (remember your $60 --- check or cash)
  - If you haven't picked up your kits by the end of the week, I will assume you are not taking this course for credit
- Lab manuals can also be found on the course webpage
- TA lab schedule can be found on the course webpage
  - You are free to work on your lab outside of TA lab hours, but you need a TA to check off your lab.
- Join the Piazza discussion forum at piazza.com/brown/fall2019/engn1630

## DEFINITIONS

- Literals $x_i$ or $x_i'$
- Product Term $x_2x_1'$
- Sum Term $x_2 + x_1' + x_0$
- Minterm of *n* variables: A product of *n* variables in which every variable appears exactly once.
- Maxterm of *n* variables: A sum of *n* variables in which every variable appears exactly once.
- *Where do the name Minterm, Maxterms come from?*



## KARNAUGH MAPS (K-MAPS)

| A | B | C | Y |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 0 |

| Y, C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 0 | 0 |

| Y, C \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 0 | $\bar{A}\bar{B}\bar{C}$ | $\bar{A}B\bar{C}$ | $AB\bar{C}$ | $A\bar{B}\bar{C}$ |
| 1 | $\bar{A}\bar{B}C$ | $\bar{A}BC$ | $ABC$ | $A\bar{B}C$ |

- Boolean expressions can be minimized by combining terms
- Karnaugh maps (K-maps) minimize equations graphically
- $PA + P\overline{A} = P$

## 4-INPUT K-MAP

| A | B | C | D | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

| Y: CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | 1 | 1 | 0 | 0 |
| 10 | 1 | 1 | 0 | 1 |

## 4-INPUT K-MAP

| A | B | C | D | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 |

| Y: CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | 0 | 1 |
| 01 | 0 | 1 | 0 | 1 |
| 11 | 1 | 1 | 0 | 0 |
| 10 | 1 | 1 | 0 | 1 |

$Y = \bar{A}\bar{C} + \bar{A}BD + A\bar{B}\bar{C} + \bar{B}\bar{D}$

- How many product terms would you have if you expressed Y in its minterm canonical form?

## K-MAPS WITH DON'T CARES

| A | B | C | D | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | X |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | X |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | 0 | X |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |

| Y: CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | | | | |
| 01 | | | | |
| 11 | | | | |
| 10 | | | | |

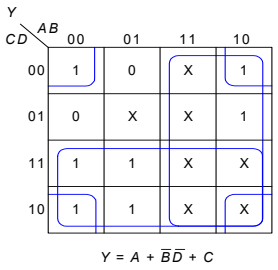
## K-MAPS WITH DON'T CARES

| A | B | C | D | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | X |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | X |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | 0 | X |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |

| Y: CD \ AB | 00 | 01 | 11 | 10 |
|---|---|---|---|---|
| 00 | 1 | 0 | X | 1 |
| 01 | 0 | X | X | 1 |
| 11 | 1 | 1 | X | X |
| 10 | 1 | 1 | X | X |

## K-MAPS WITH DON'T CARES

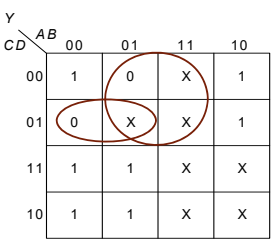
| A | B | C | D | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | X |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | X |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | 0 | X |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |




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

## K-MAPS WITH DON'T CARES USING MAXTERMS

| A | B | C | D | Y |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 1 |
| 0 | 0 | 0 | 1 | 0 |
| 0 | 0 | 1 | 0 | 1 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 0 |
| 0 | 1 | 0 | 1 | X |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 1 |
| 1 | 0 | 0 | 1 | 1 |
| 1 | 0 | 1 | 0 | X |
| 1 | 0 | 1 | 1 | X |
| 1 | 1 | 0 | 0 | X |
| 1 | 1 | 0 | 1 | X |
| 1 | 1 | 1 | 0 | X |
| 1 | 1 | 1 | 1 | X |




Y=(B'+C)(A+C+D')
= (AB'+AC+B'C+C+B'D'+CD')
= (AB' + C + B'D')

≠ (A + C + B'D')

## COMBINATIONAL LOGIC

- Universal Set of Gates
- Other Types of Gates
  1) XOR
  2) NAND / NOR

## UNIVERSAL SET

- Universal Set: A set of gates such that every Boolean function can be implemented with gates in this set.
- Examples:
  - {AND, OR, NOT}
  - {AND, NOT} OR can be implemented with AND and NOT gates: a+b = (a'b')'
  - {OR, NOT} AND can be implemented with OR and NOT gates: ab = (a'+b')'
  - {AND, OR} is not a universal set

## OTHER UNIVERSAL SETS

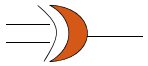
{NAND, NOR}

How do you make an inverter?

{NOR}

How do you make a NAND?

{XOR, AND}



X⊕Y = XY' + X'Y

How do you make an inverter?

$X \oplus 1 = X*1' + X'*1 = X'$

## SHANNON'S EXPANSION (FOR SWITCHING FUNCTIONS)

**Formula:** $f(x,Y) = x \cdot f(1, Y) + x' \cdot f(0, Y)$

Proof by enumeration:

If $x = 1, f(x,Y) = f(1, Y) : 1 \cdot f(1, Y) + 1' \cdot f(0,Y) = f(1, Y)$

If $x = 0, f(x,Y) = f(0, Y) : 0 \cdot f(1, Y) + 0' \cdot f(0,Y) = f(0, Y)$

Useful for evaluating function and generating canonical form of a function

## EVALUATE USING SHANNON'S EXPANSION

Shannon's Expansion

$X \oplus XY' \oplus X'Y \oplus (X + Y) \oplus X = ?$

$\Rightarrow X \oplus (XY') \oplus (X'Y) \oplus (X + Y) \oplus X = f(X, Y)$

If $X = 1, f(1, Y) = 1 \oplus Y' \oplus 0 \oplus 1 \oplus 1 = Y \oplus 1 \oplus 1 = Y' \oplus 1 = Y$
If $X = 0, f(0, Y) = 0 \oplus 0 \oplus Y \oplus Y \oplus 0 = 0$

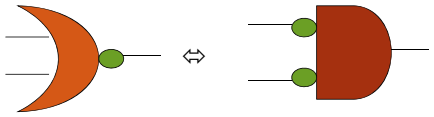
Thus, $f(X, Y) = X \cdot f(1,Y) + X' \cdot f(0,Y) = XY$

## A NOTE ON NAND/NOR

NAND, NOR gates

- NAND, NOR gates are not associative
- Let a | b = (ab)'
- (a | b) | c ≠ a | (b | c)
- (a | b) | c = (a' + b')' + c' = ab + c'
- a | (b | c) = a' + (b' + c')' = a' + bc

## DEMORGAN'S LAW

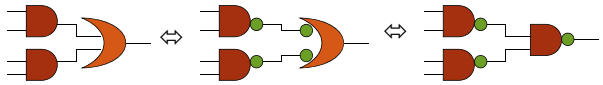


(a+b)' = a'b'

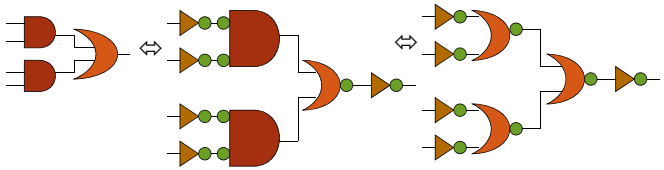
(ab)' = a'+b'

## TRANSLATING SUM OF PRODUCTS TO GATES

Sum of Products (Using only NAND gates)

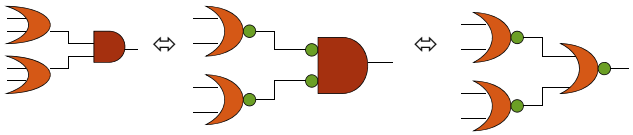


Sum of Products (Using only NOR gates)



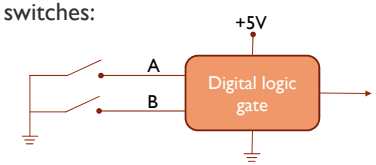
## TRANSLATING PRODUCT OF SUMS TO GATES

Product of Sums (NOR gates only)



## WHY DO WE NEED PULL-UP RESISTORS?

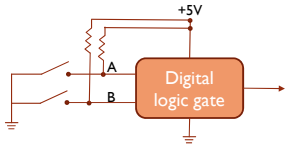
- Consider a generic logic gate with 2 inputs, connected to 2 switches:




- Say the gate implements a NOR function.
  - What happens when the input switches are closed?
  - What happens when the input switches are open?

## WHY DO WE NEED PULL-UP RESISTORS?

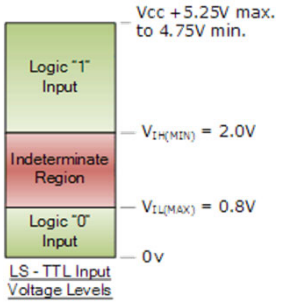
- Never leave inputs floating
- Connect them directly or indirectly to Vcc (+5V) or GND
  - Direct connection: when inputs only take on one value
  - Indirect connection (via a resistor): when inputs conditionally take on a specific value




How large should these resistors be?
Depends on input current $I_{IH}$ and tolerable voltage drop

## DIGITAL VS. ANALOG SIGNALS

- We use Vcc to represent logic "1" and Gnd to represent logic "0"
- Within each of these states there is a range of voltages that define upper and lower voltages of these binary states
- Example:
  - $V_{IH}$ = 2.0V is the minimum input voltage guaranteed to be recognized as logic "1"
  - $V_{IL}$ = 0.8V is the maximum input voltage guaranteed to be recognized as logic "0"




## PULL-UP VS. PULL-DOWN

- What about a pull-down resistor?
- When would you need a pull-up/pull-down at the output?