CSC258: Computer Organization

Digital Logic: Transistors and Gates

Pre-Class Review

1. What are the largest (positive) and smallest (negative) numbers that can be represented using 4-bit 2's complement?
2. Add the 6-bit 2's complement numbers 001011 and 111000. Please show your work but circle your result.
3. Demonstrate what is mean by a "minterm".

No Response to Email?

- Please make sure you're using my correct email address:

  *andrew.pepters**e**n @ utoronto.ca*

  - Andrew Peters**O**n is a student who is probably frustrated to be getting so much email for me.

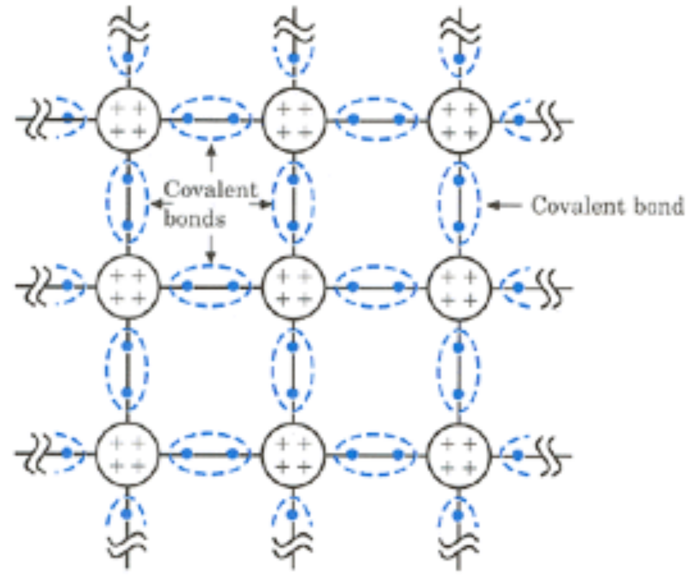
Transistors

Review of Electricity

- Electricity: the flow of electrons in a substance
- Voltage (V, voltage): electrical potential
- Current (I, amps): rate of flow
- Resistance (R, ohms): resistance to flow of electrons

Silicon as a Semiconductor

- Silicon (and germanium) is predominantly used
  - Grown as a 2D crystal
- Covalent bonds are strong ...
  - but energy can knock electrons loose ...
  - allowing current to flow

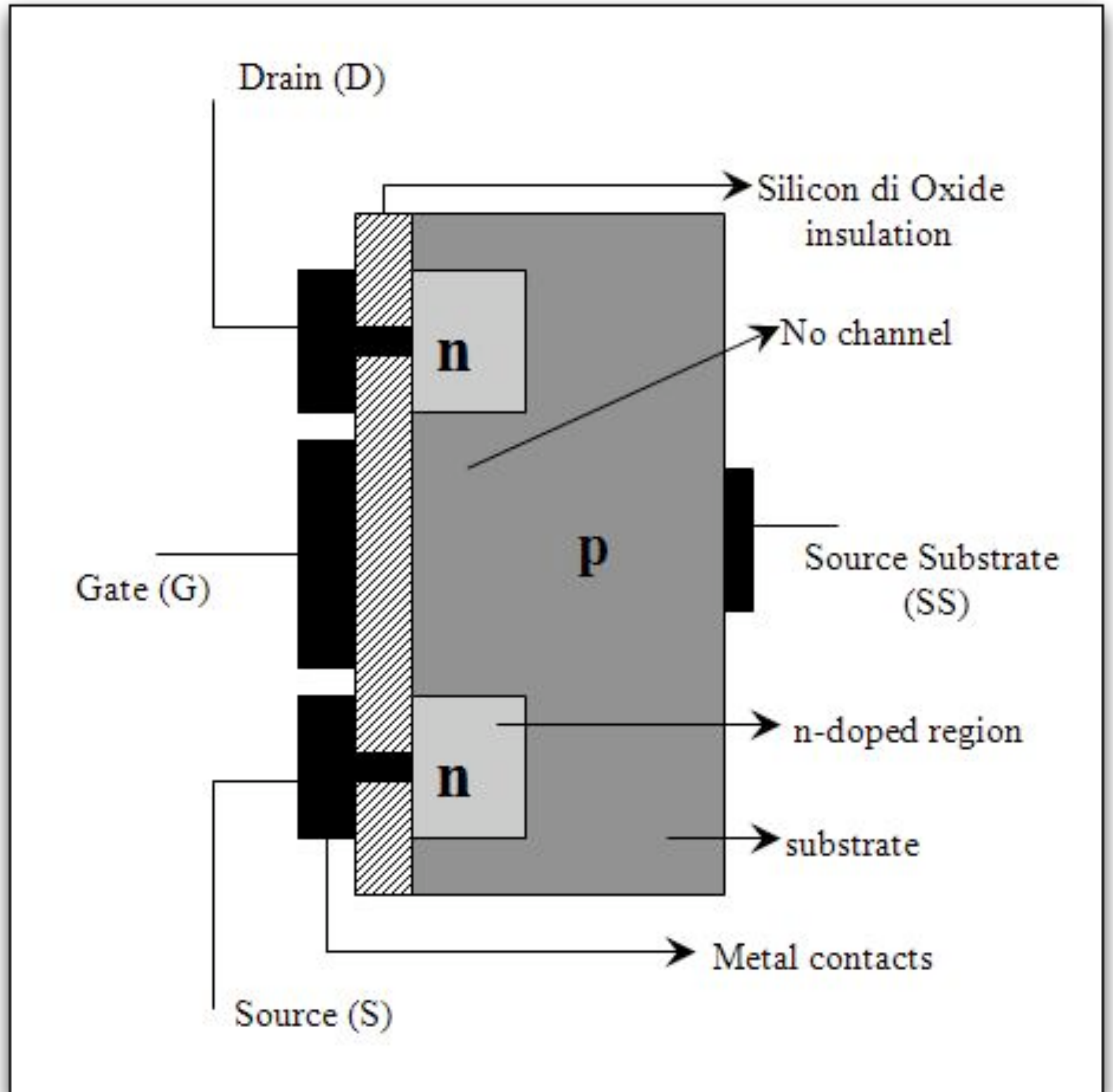


Doping (A Good Thing)

- Doping: introduction of impurities to a crystal
- Semiconductor properties can be altered by doping
  - n-type: Addition of phosphorous or arsenic, which have 5 valence electrons
    - Adds electrons (negative charge)
  - p-type: Addition of boron, which has 3 valence electrons
    - Adds “holes” (positive charge)

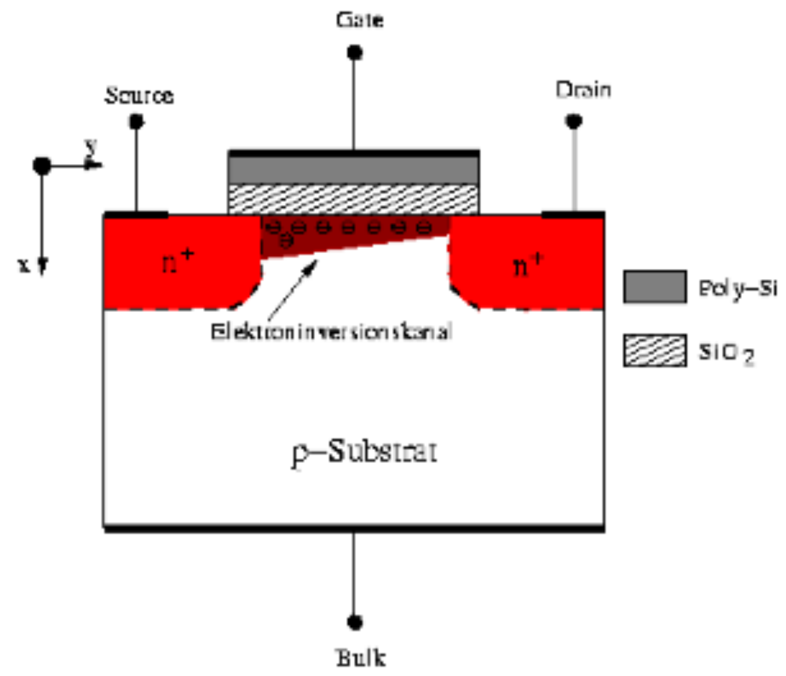
MOSFETs

- Idea: Produce a path between the source and drain
- Four contact points
  - Source and Drain
  - Gate
  - Body (bulk)



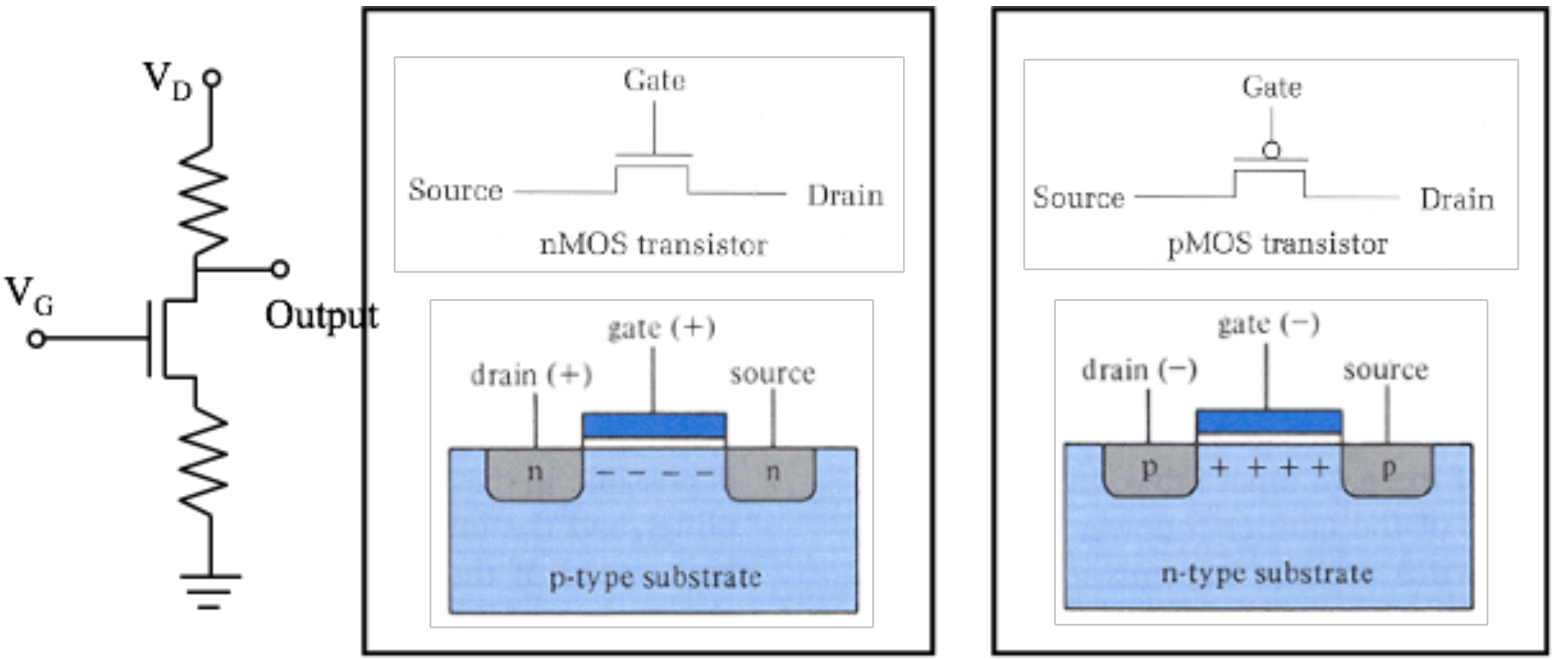
MOSFET Operation

- A voltage is applied across the source and drain
  - Normally, little current will pass
- Applying a voltage between the source and gate creates a channel



NPN and PNP

- Two kinds of transistors exist
  - The circle denotes “inversion” -- we’ll see this often

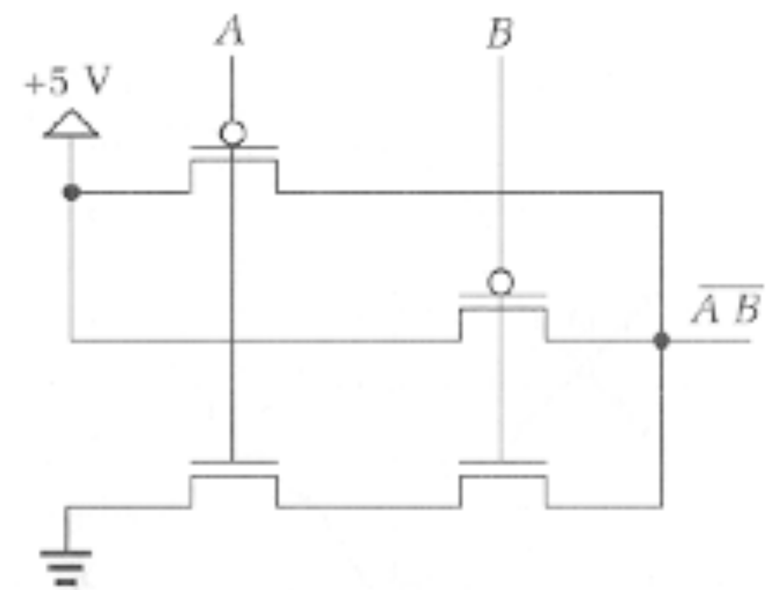


Creating Logic (Gates)

- We need to turn transistors into logic
  - i.e., Current into bits
- Transistors are more complicated than switches
  - They are *conditional switches*

Building Gates using CMOS

- Multiple transistors are required to build the logic functions we know
  - Both P and N types are used for power savings
- Example: NAND



Discussion Questions

- Why do we use CMOS transistors instead of other forms of logic devices?

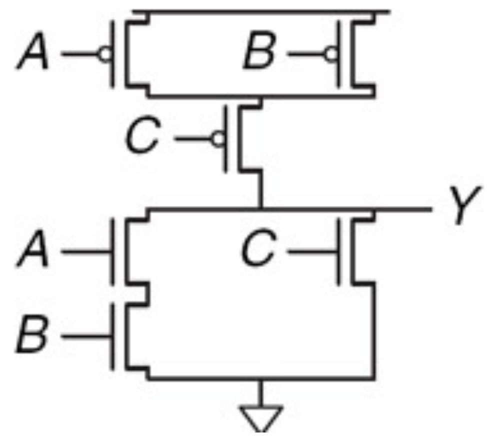
Discussion Questions

- Why do we use CMOS transistors instead of other forms of logic devices?

- Does our technology fundamentally limit the kind of computation we *can* do?

Quiz-like Questions

The schematic on the left is from Ex.1.88 in your textbook.



1. Provide the truth table and the sum-of-products and product-of-sums boolean equations for the function implemented in the schematic.

2. Draw a CMOS circuit that implements a NOT gate.

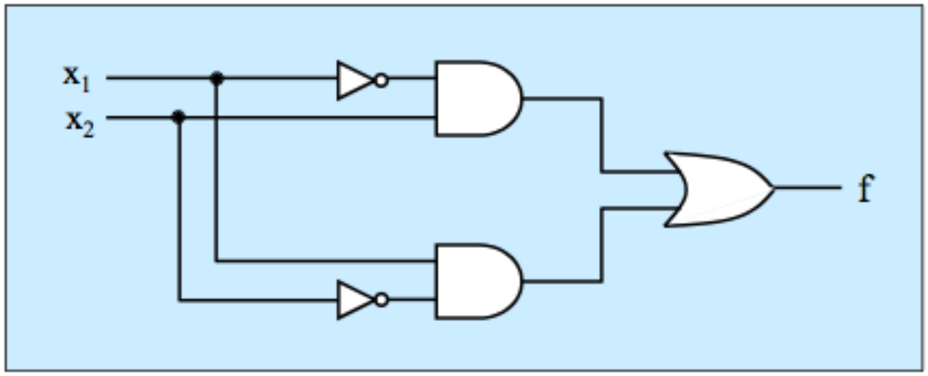
3. Draw a CMOS circuit that implements an AND gate.

4. What is the drawback of using pseudo-NMOS circuitry?

Gates

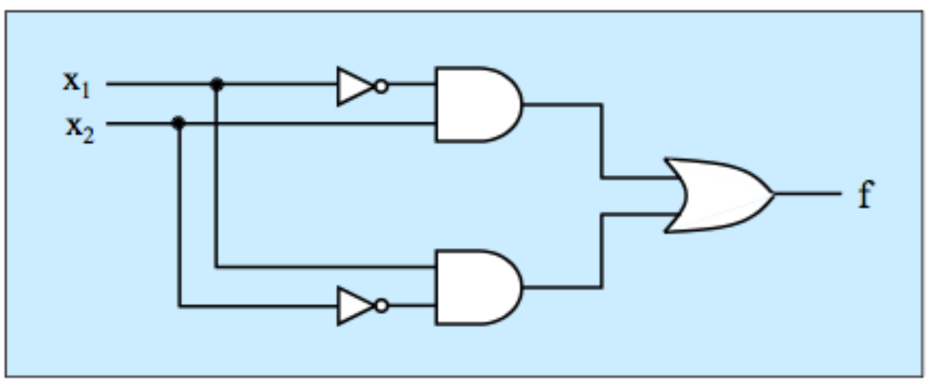
Intro Circuits

- Gates are a physical representation of boolean logic
- Can convert between a circuit and a logic expression
  - Order of operations is important



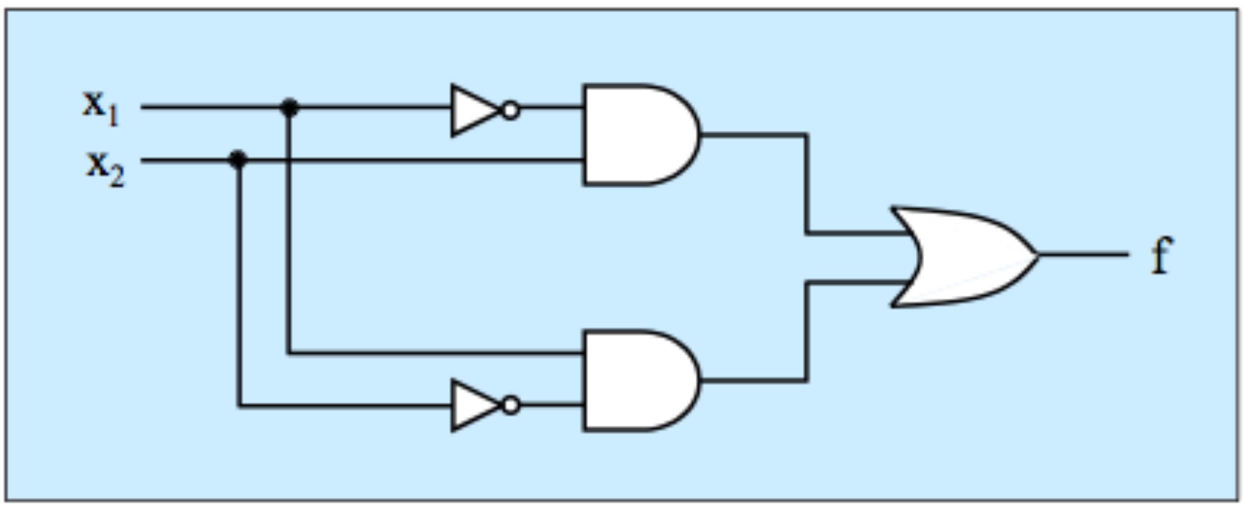
Circuits and Boolean Logic

- Gates are a physical representation of boolean logic
- What boolean logic expression is represented by the circuit below?



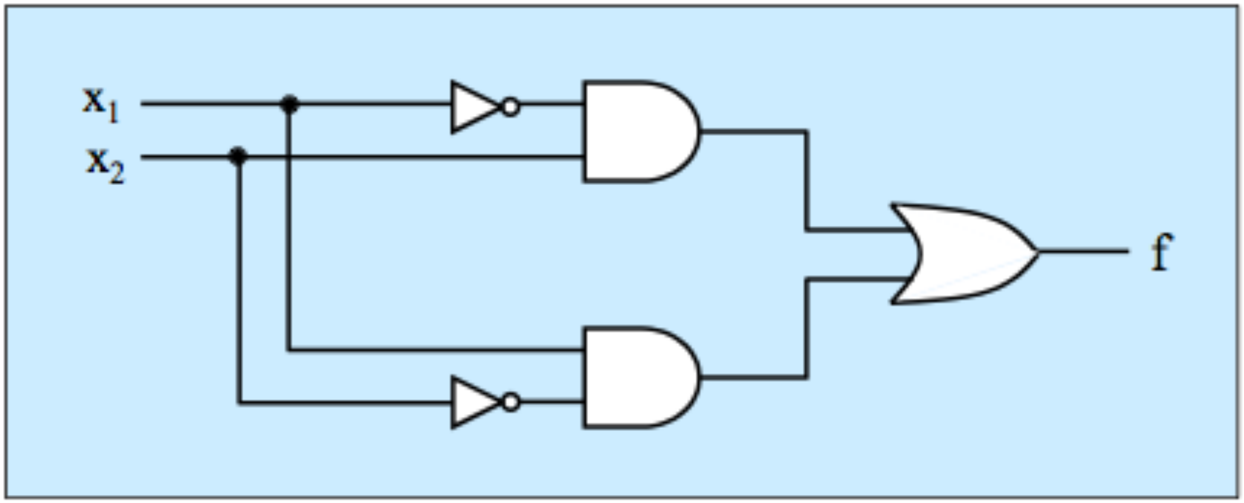
Circuit to Logic Conversion

- Start from the output. Put in placeholders for every input you encounter and fill them in one by one.



Circuit to Logic Conversion

- f = _ + _
- f = (_ · _) + _
- f = ($\neg x_1$ · _) + _
- f = ($\neg x_1 \cdot x_2$) + _
- … f = ($\neg x_1 \cdot x_2$) + ($x_1 \cdot \neg x_2$)



Quiz-like Question

Draw the circuit that results from this expression:

$$f = AB + \neg(AB)C$$

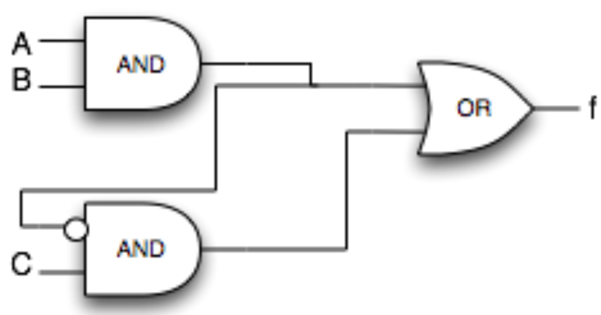
Hint: Make sure the expression is fully parenthesized, then draw from the innermost terms out.

Logic to Circuit Conversions

Draw the circuit that results from this expression:

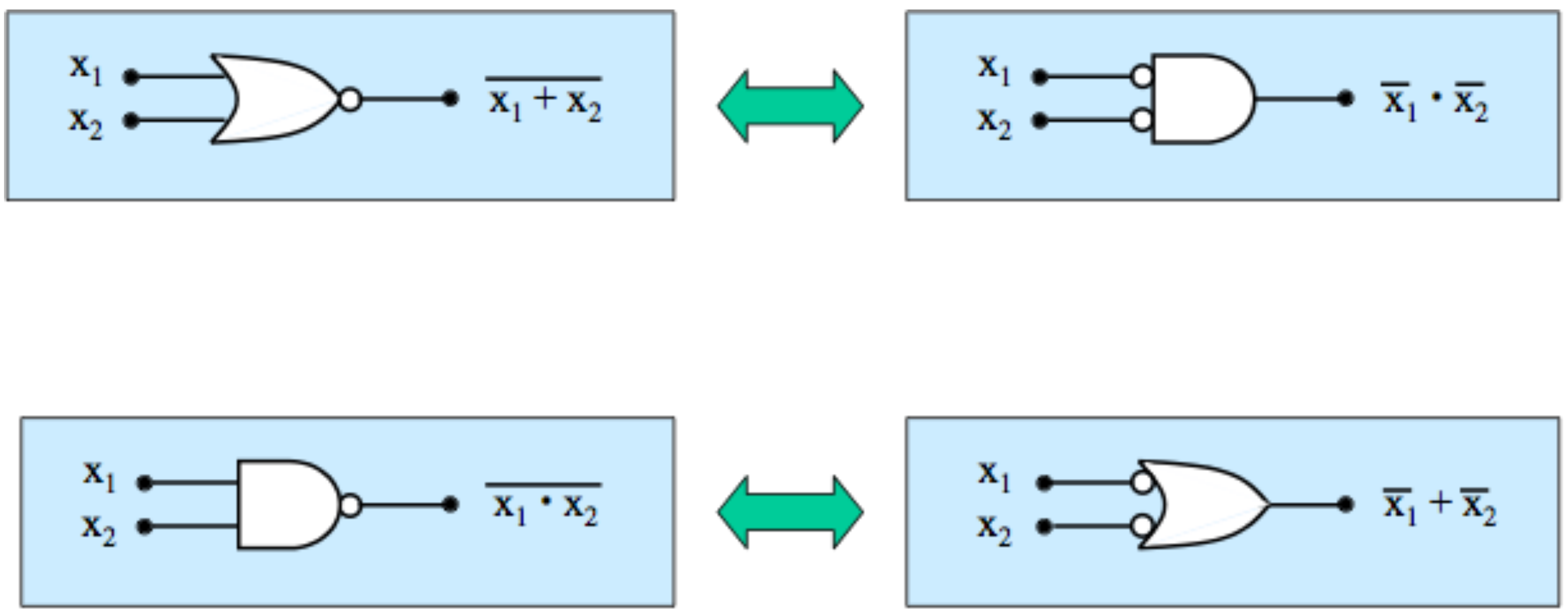
$$f = AB + \neg(AB)C$$

Hint: Make sure the expression is fully parenthesized, then draw from the innermost terms out.



de Morgan’s Theorem

- Technique for moving negations
- Essentially “bubble pushing”



Simplifying Circuits

Why Karnaugh Maps?

- Applying algebraic simplification rules to a formula is an awful way to find a minimum circuit
  - Simplification takes trial and error.
  - There's no guarantee that you've found the minimal circuit.
  - Checking your simplification requires enumeration on a truth table.

- The (partial) solution: Karnaugh maps

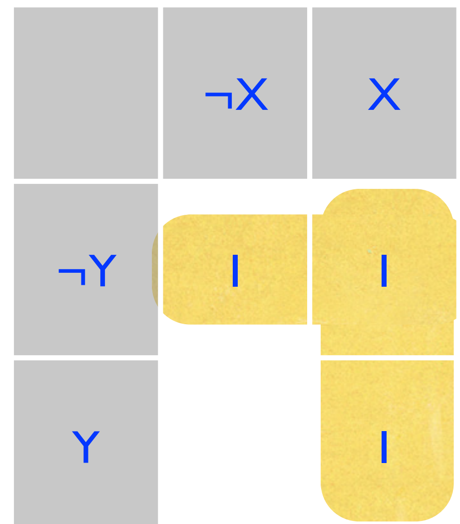
Gray Codes

- “Gray codes” were developed for error-prone systems
  - Named after Frank Gray, one of the developers of the television
- The idea: encoding in which only *one bit* changes at a time

2-bit	3-bit
00	000
01	001
11	011
10	010
	110
	111
	101

Karnaugh Maps

- Karnaugh maps represent logic formula
- To optimize a formula, draw the biggest rectangles you can!
- Results in the *optimal* formula



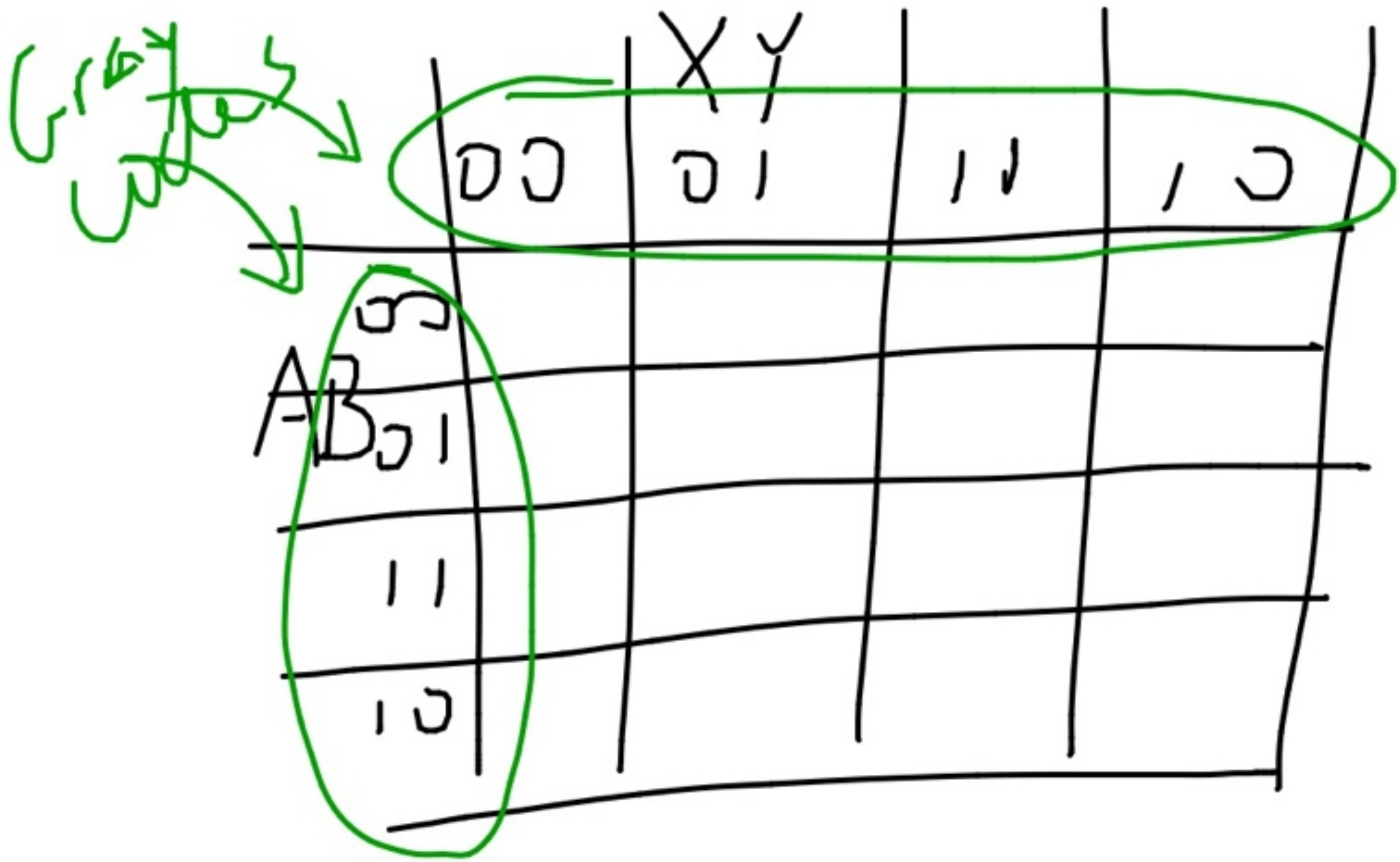
What is “Optimal”?

- K-maps don’t necessarily create the smallest circuits.
- Instead, they create small circuits that minimize *latency*:

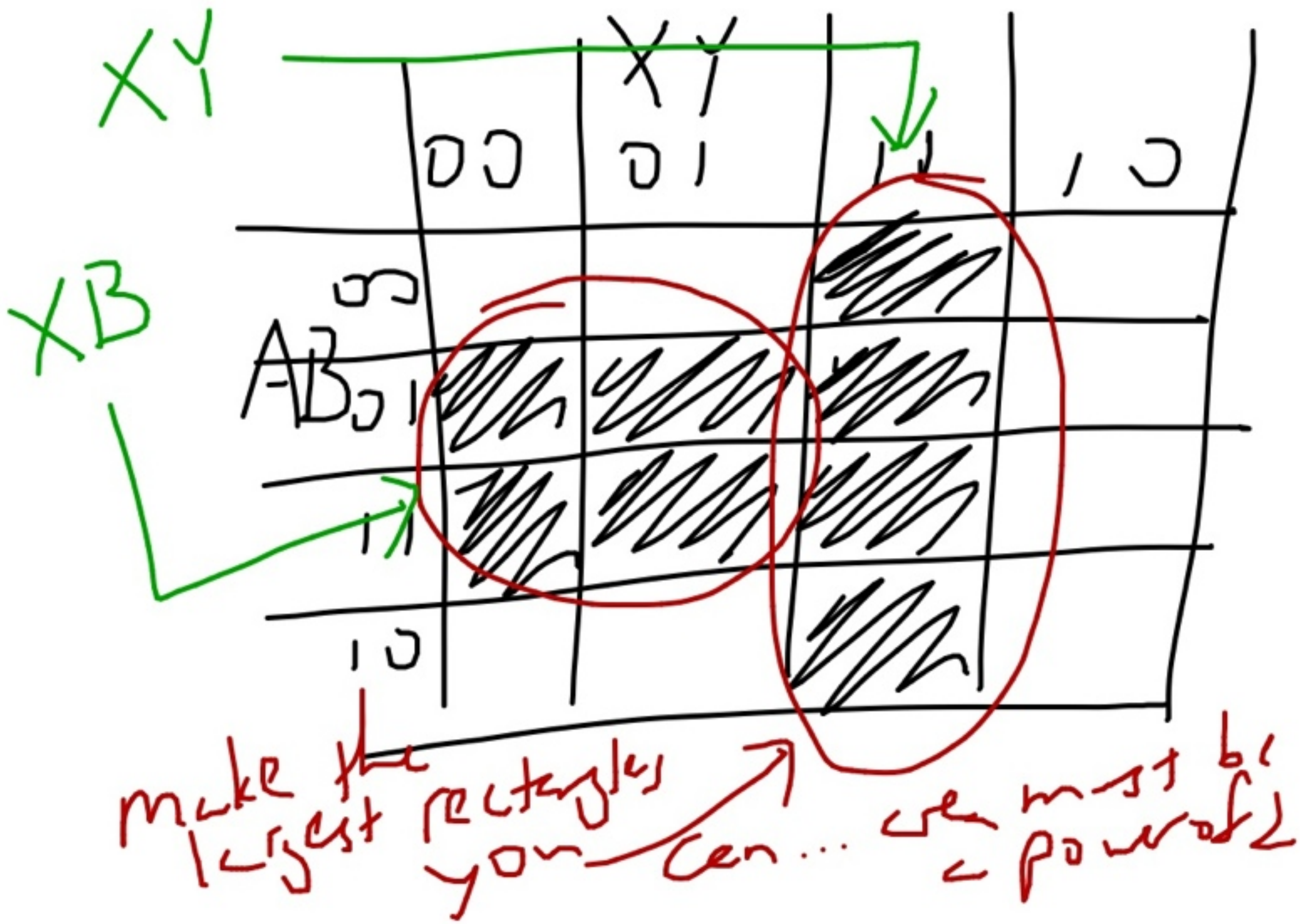
  “How long must we wait for a circuit to complete its computation?”

- Each transistor adds latency, since it takes a small amount of time to switch.
  - Hence, each gate in series adds delay, so deeper circuits are slower.
  - 2-level circuits, like those corresponding to sum-of-products or product-of-sums forms, are fast circuits.

A 4-variable K-map

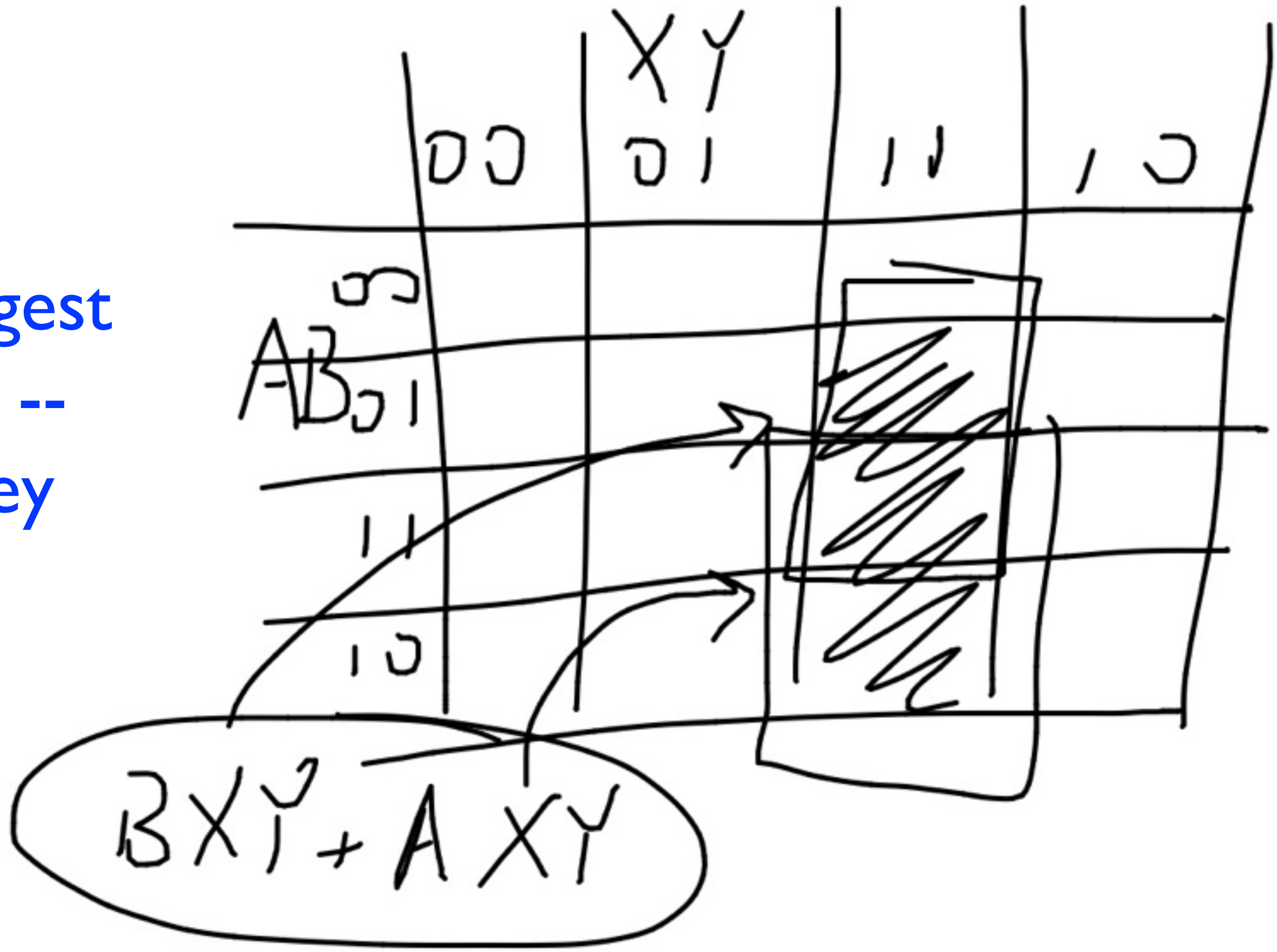


K-map Example: Reading XY + XB

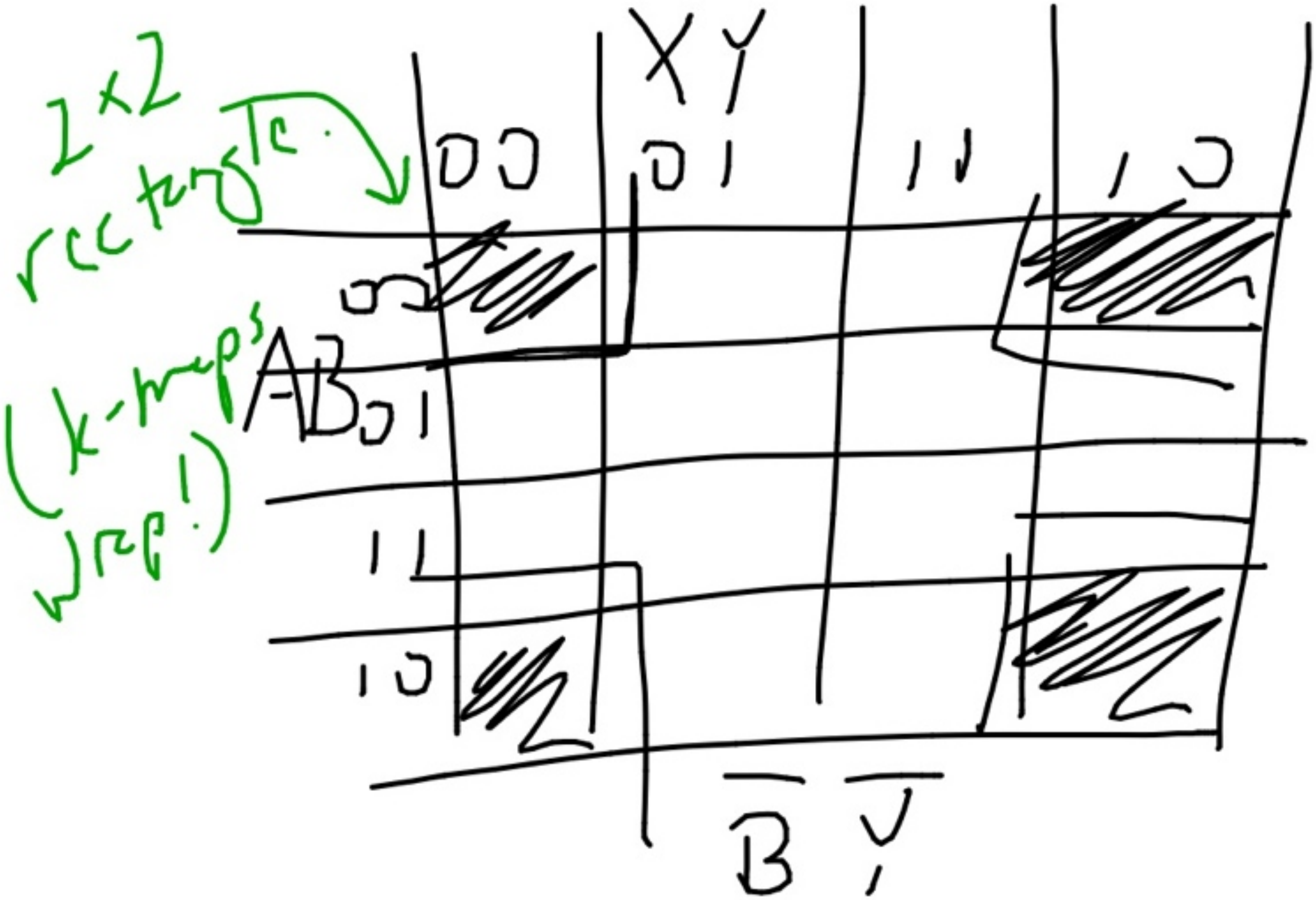


Overlapping

Pick the largest rectangles -- even if they overlap



K-Maps Wrap Around



An Old Lab Problem …

An old logic puzzle describes a (F)armer who must cross a river with a (W)olf, (G)oose, and sack of (C)orn. The farmer has a rowboat, but it's only big enough to carry him and one other item across. Furthermore, if he leaves the wolf with the goose or the goose with the corn, something will get eaten. The farmer must transport all three possessions across the river.

Let the variables F, W, G, and C represent the locations of the various actors. If a variable as a value 0, the object is on the west bank. If a variable has the value 1, it is on the east bank.

Your Task

- Derive the truth table for a function on F, W, G, and C that outputs 1 if the goose or corn are about to be eaten.

- Use a K-map to simplify the function as much as possible. Provide both the K-map and the function extracted from it.

Specify the Function on a Truthtable

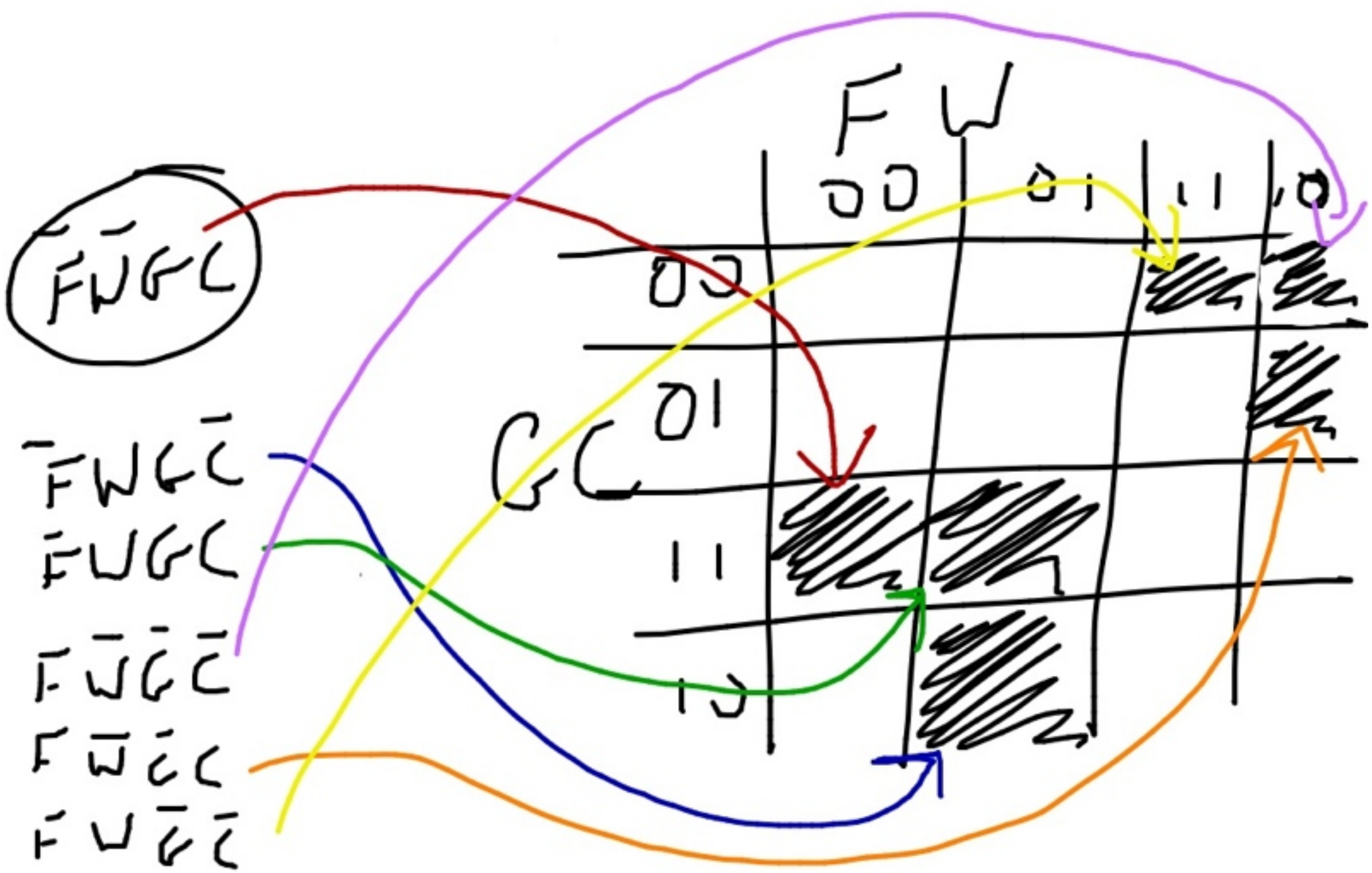
	F	W	G	C	f	
	0	0	0	0	0	
	0	0	0	1	0	
	0	0	1	0	0	
$\bar{F}\bar{W}GC$	0	0	1	1	1	✓
	0	1	0	0	0	
	0	1	0	1	0	
$\bar{F}WG\bar{C}$	0	1	1	0	1	✓
$\bar{F}WGC$	0	1	1	1	1	✓
	1					
	⋮	symmetric				

$F\bar{W}\bar{G}\bar{C}$

$F\bar{W}\bar{G}C$

$FW\bar{G}\bar{C}$

Mapping Minterms to K-Map



Extracting (Latency-Minimized) Formula

$\bar{F}GC + \bar{F}WG + F\bar{G}\bar{C} + F\bar{W}\bar{F}$

$LC = 12$

$GC = 22$

$\bar{F}\bar{W}GC$

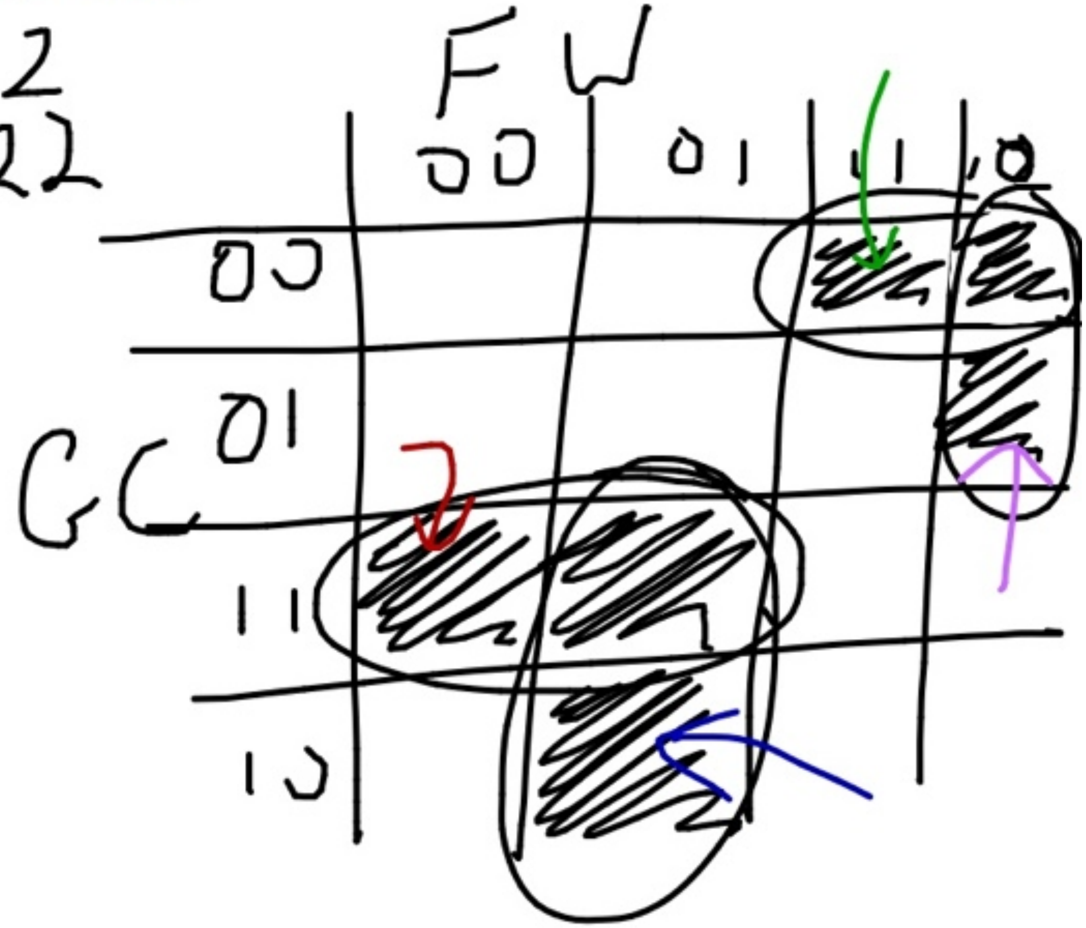
$\bar{F}WG\bar{C}$

$\bar{F}WGC$

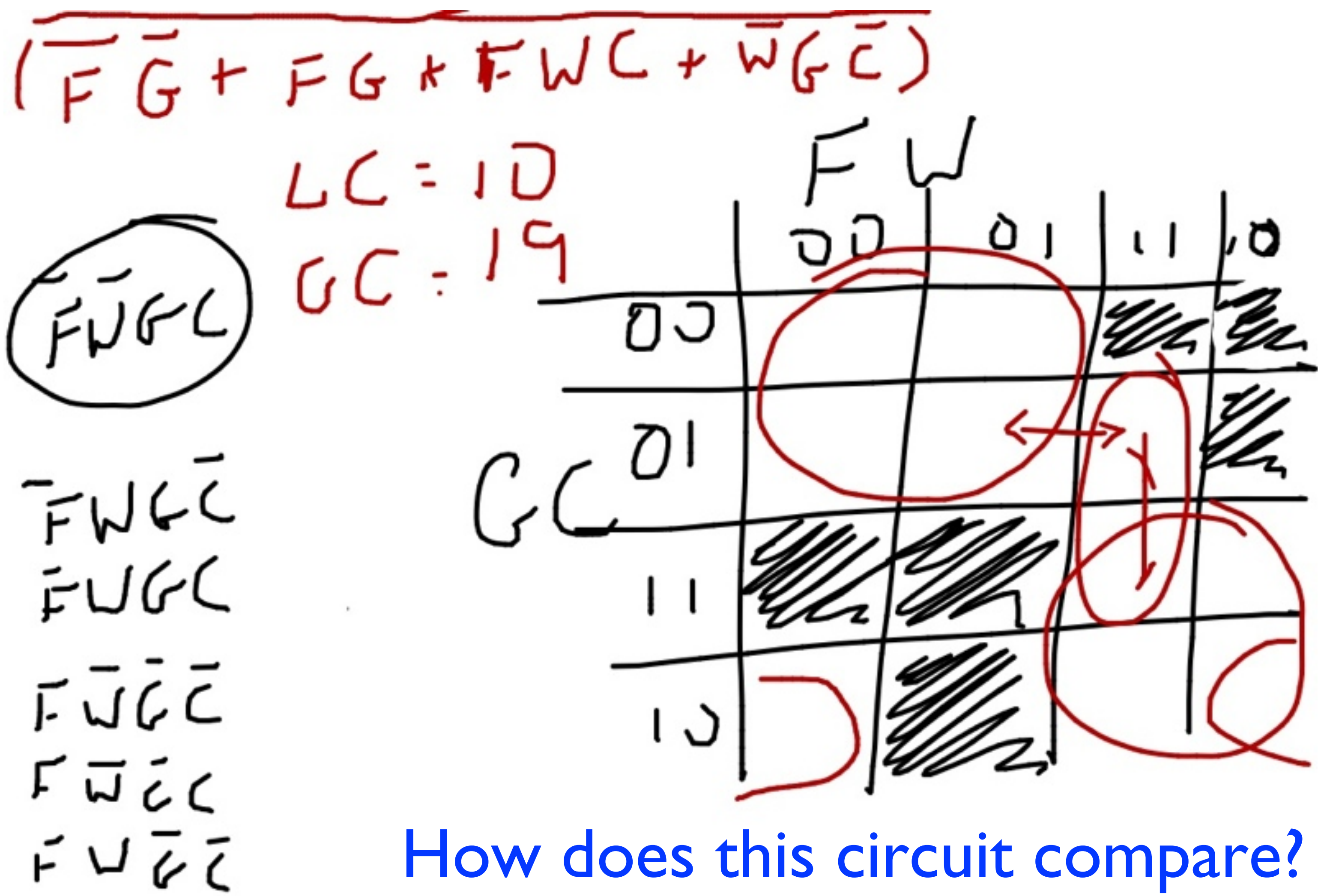
$F\bar{W}\bar{G}\bar{C}$

$F\bar{W}\bar{G}C$

$FW\bar{G}\bar{C}$



Alternately: Map the 0's and Negate



How does this circuit compare?

Limits to Karnaugh Maps

- Any design with more than 4 variables is difficult to visualize
- Karnaugh maps provide an optimized 2-level circuit, but ...
  - If we care about other factors like space efficiency or heat, we need to perform other optimizations.

The Grim Bottom Line

- Very few constraints can be optimized automatically
- Many hardware design problems can only be fully solved by enumeration -- by trying every possibility
  - Wire and gate layout
  - Modularization

Quiz-like Questions

A	B	C	OUT
0	0	0	1
0	0	1	0
0	1	0	1
0	1	1	1
1	0	0	X
1	0	1	X
1	1	0	1
1	1	1	0

1. Find a minimal boolean equation for the function in the table to the right. Remember to take advantage of don't-care entries. Please show your work.

2. Provide a truth table for this boolean equation:

$$Y = AB + \neg B$$

Simplify the equation using a k-map.

Summary of Logic Gates

- The current dominant technology for building circuits is CMOS.
- Circuits are modeled with Boolean logic
  - Functions are represented as formulas or truth tables
  - Each logical operator has a corresponding gate
- Our tools for optimizing circuits don't scale, so we have to program and test.
- Logic design at modern scales is a difficult and expensive proposition