Computer Systems and Networks

ECPE 170 – Jeff Shafer – University of the Pacific

State Machines & Karnaugh Maps

Upcoming Events

- **Homework 5 - Due Tuesday**
  - Paper submissions accepted for this assignment (since it involves drawing Karnaugh Maps…)

Upcoming Events

- **Quiz 2 - Tuesday**
  - Topics *may or may not* include:
    - Simplifying Boolean expressions with identities?
    - Sum-of-products or product-of-sum form?
    - Converting between a truth table and a circuit diagram (with logic gates)?
    - Common combinational circuits: decoders, multiplexers?
      - Basic operation of these devices, i.e. inputs and outputs
    - Sequential circuits: SR, JK, D flip-flops?
      - Basic operation of these devices, i.e. inputs and outputs

Recap from Last Class

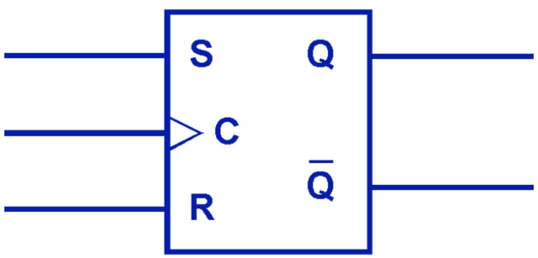
- **Why do real hardware devices used NAND / NOR gates instead of AND / OR / NOT gates?**
  - These are "universal" gates – any function can be made using only NAND or only NOR gates
  - Simplifies manufacturing to use the same gate type
- **What is the difference between combinational and sequential circuits?**
  - **Combinational** – output is based on input only
  - **Sequential** – output is based on input and current output (or "state")

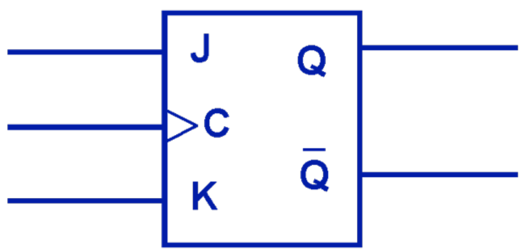
Recap from Last Class

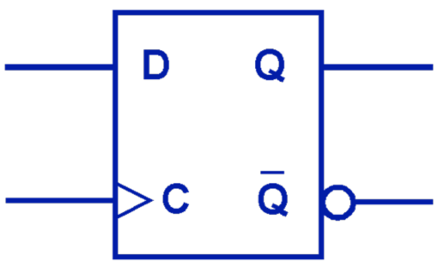
- **What is the difference between a half-adder and a full-adder?**
  - **Half adder** adds two inputs (x, y) and produces sum and carry-out
  - **Full adder** adds three inputs (x, y, carry-in) and produces sum and carry-out
    - We build it out of two half-adders!

Recap from Last Class

- **What are the outputs of these common flip-flops?**







S	R	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	undefined

J	K	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\bar{Q}$(t)

D	Q(t+1)
0	0
1	1

Discussion

- Engineering lab equipment and facilities
  - Partially paid for from your lab fee $$
  - Suggestions for improvement?

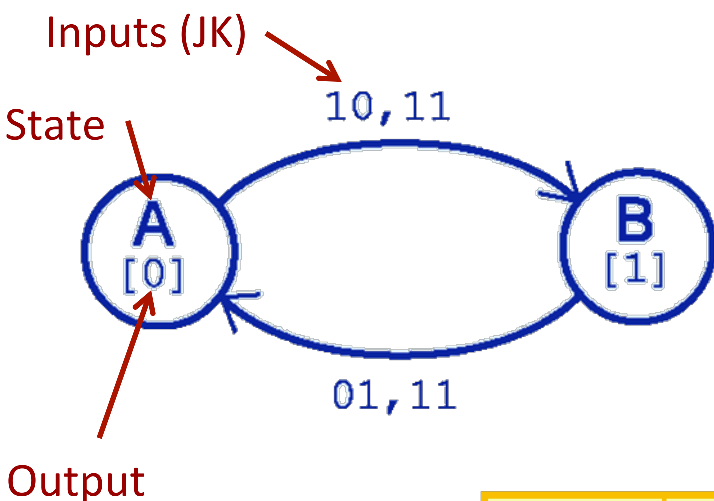
State Machines

State Machines

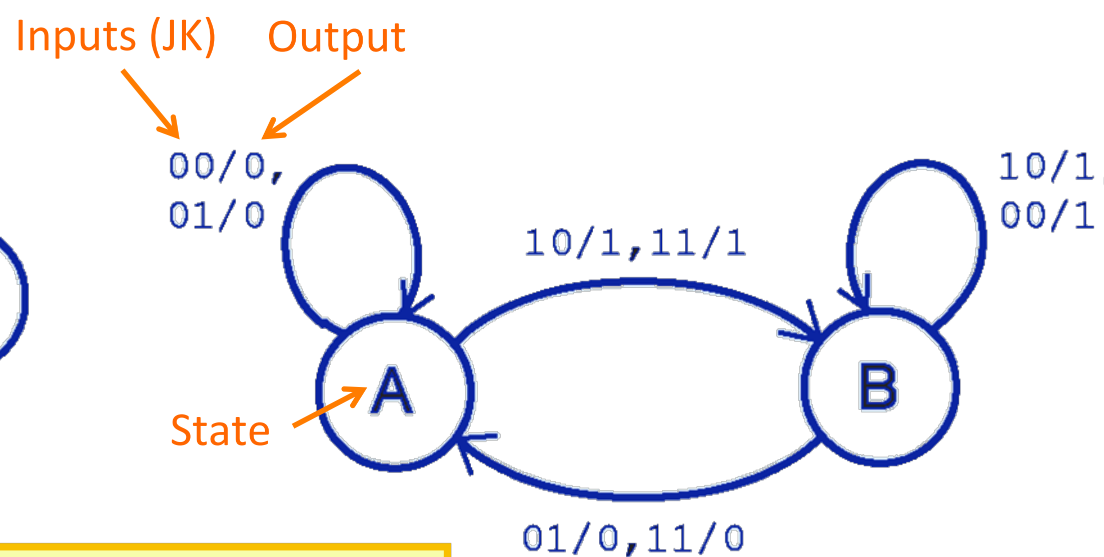
- How do we design complicated sequential systems?
  - **Finite State Machine** (FSM)
  - In visual form
    - A set of nodes that hold the states of the machine
    - A set of arcs that connect the states
- Two different types of state machines: **Moore** and **Mealy**
  - Both produce systems that produce the same output
  - Differ only in how the output of the machines are expressed
- Moore: place outputs on each node
- Mealy: present outputs on the transitions

JK Flip-Flop in State Machine Form

Moore FSM



Mealy FSM

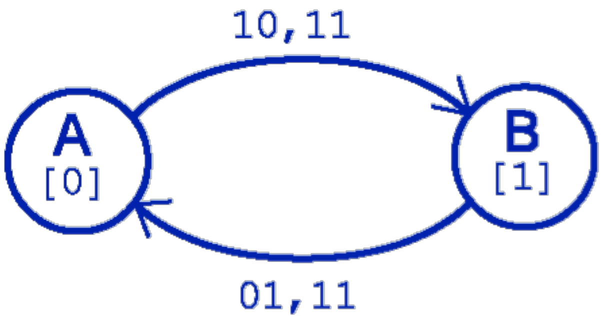


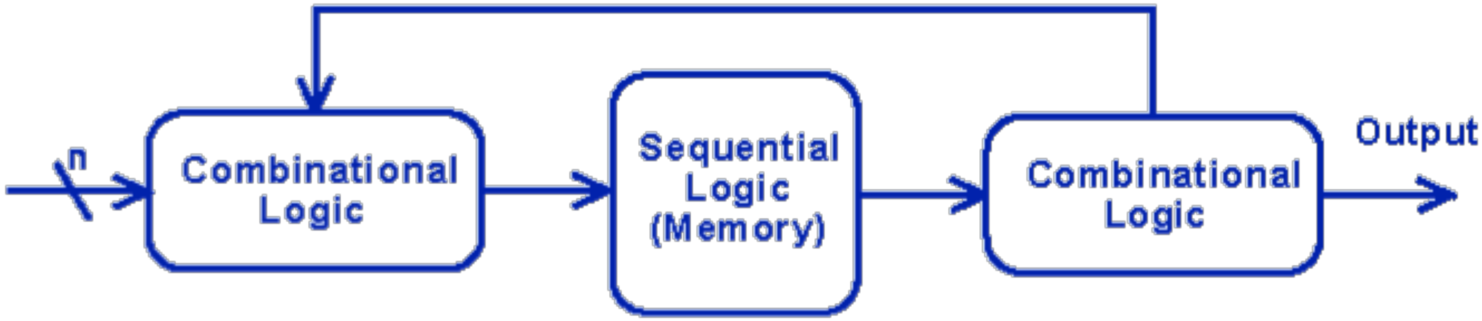
J	K	Q(t+1)
0	0	Q(t) (no change)
0	1	0 (reset to 0)
1	0	1 (set to 1)
1	1	$\overline{Q}$(t)

Different Implementations

- Although the behavior of Moore and Mealy machines is identical, their implementations differ:

Moore machine:

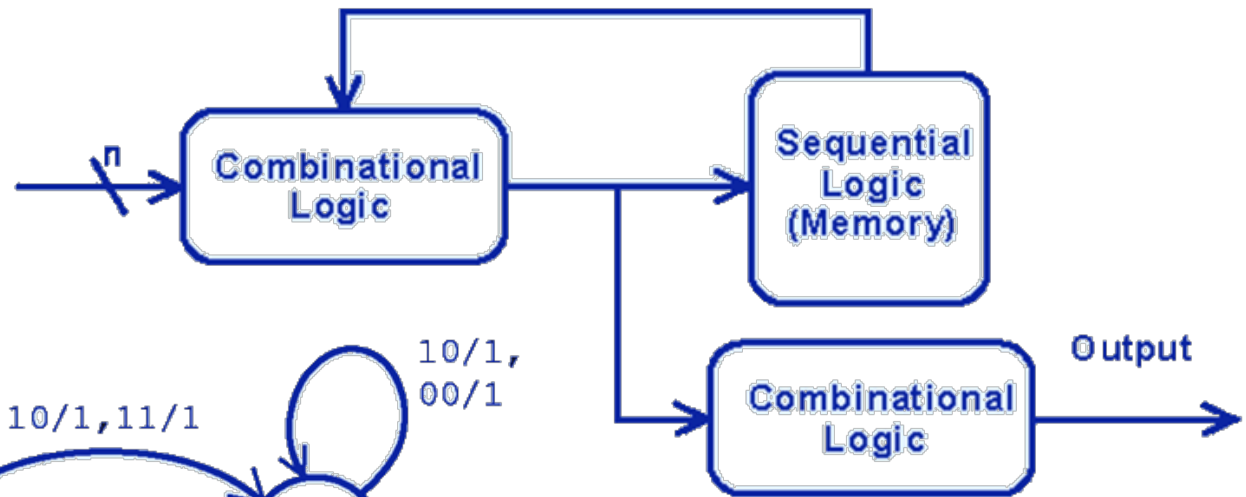


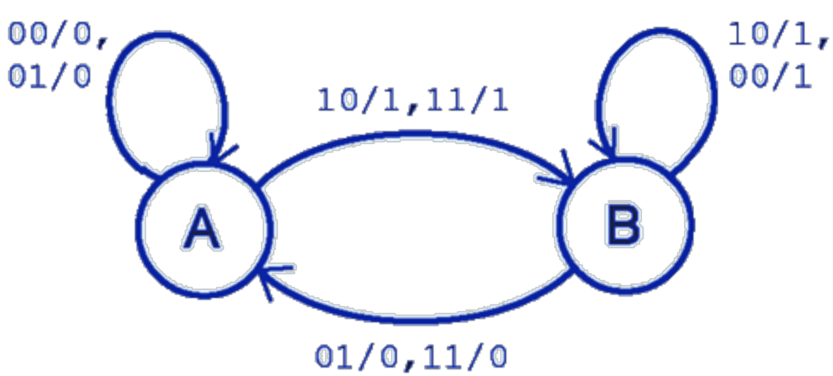


Different Implementations

- Although the behavior of Moore and Mealy machines is identical, their implementations differ:

Mealy machine

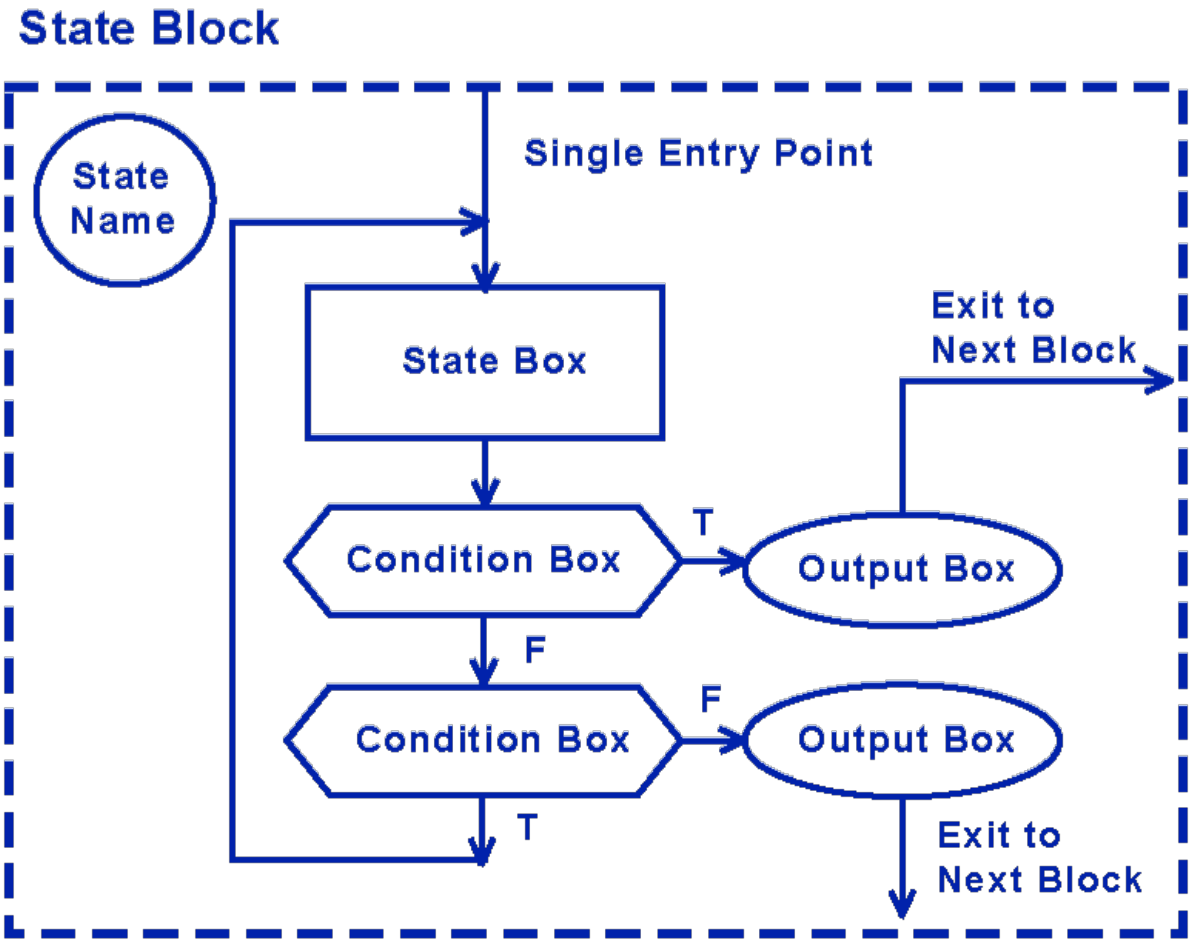




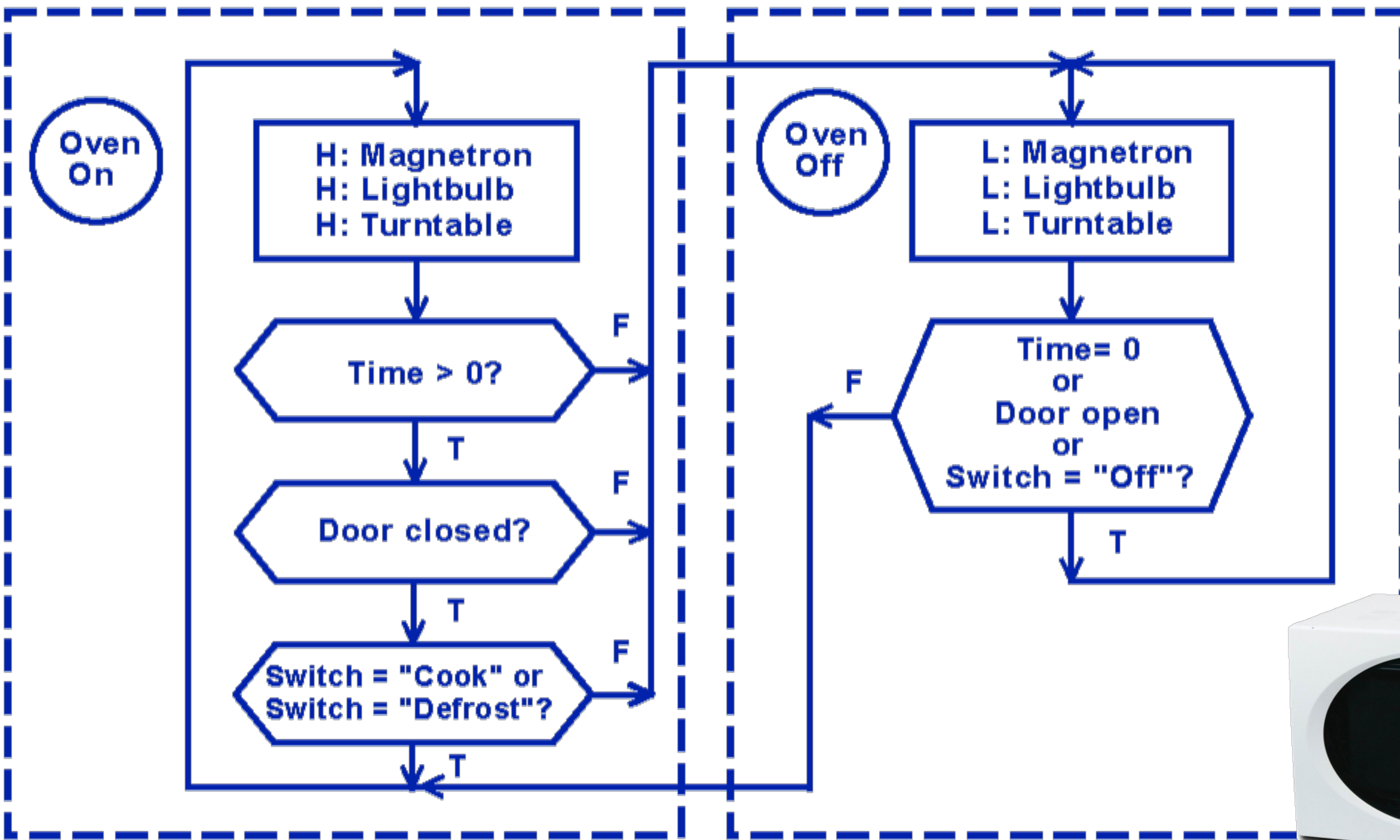
Algorithmic State Machine

- Moore and Mealy machines are challenging to draw for complex designs
  - An interaction of numerous signals is required to advance a machine from one state to the next
- Alternate approach: **Algorithmic State Machine**
  - A block diagram approach to describing digital systems

Algorithmic State Machine



Algorithmic State Machine – Microwave Oven

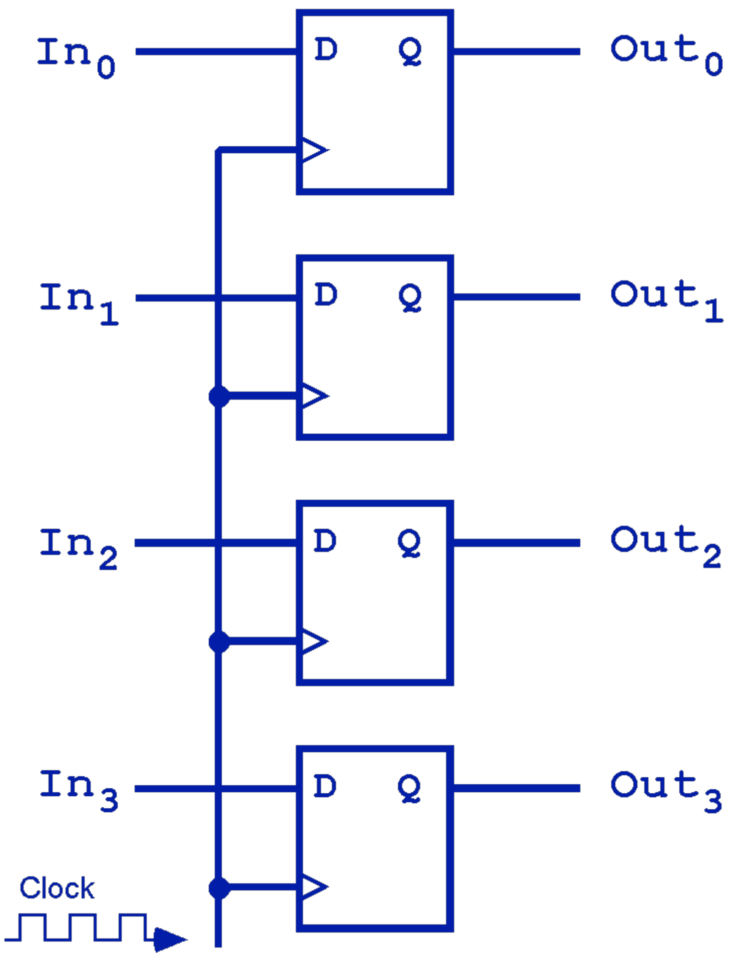


Sequential Circuit Applications

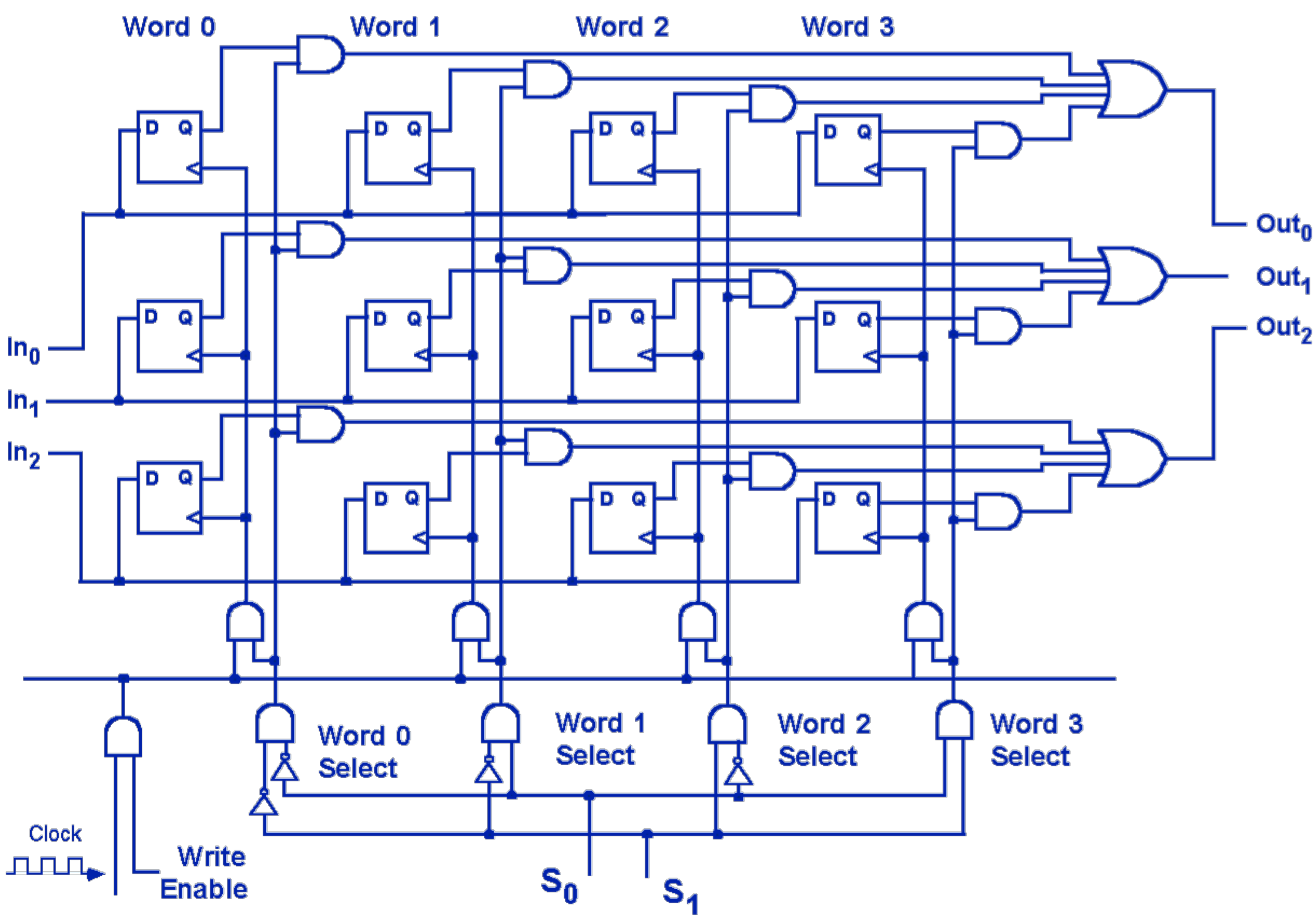
- When do I use sequential circuits?
  - Whenever the application is “**stateful**”
  - The next state of the machine depends on the **current state** of the machine and the input
- Stateful applications requires both combinational and sequential logic
- Examples: Register, Memory, Counters, …

Sequential Circuits – Register

- This illustration shows a 4-bit register consisting of D flip-flops. You will usually see its block diagram (below) instead.

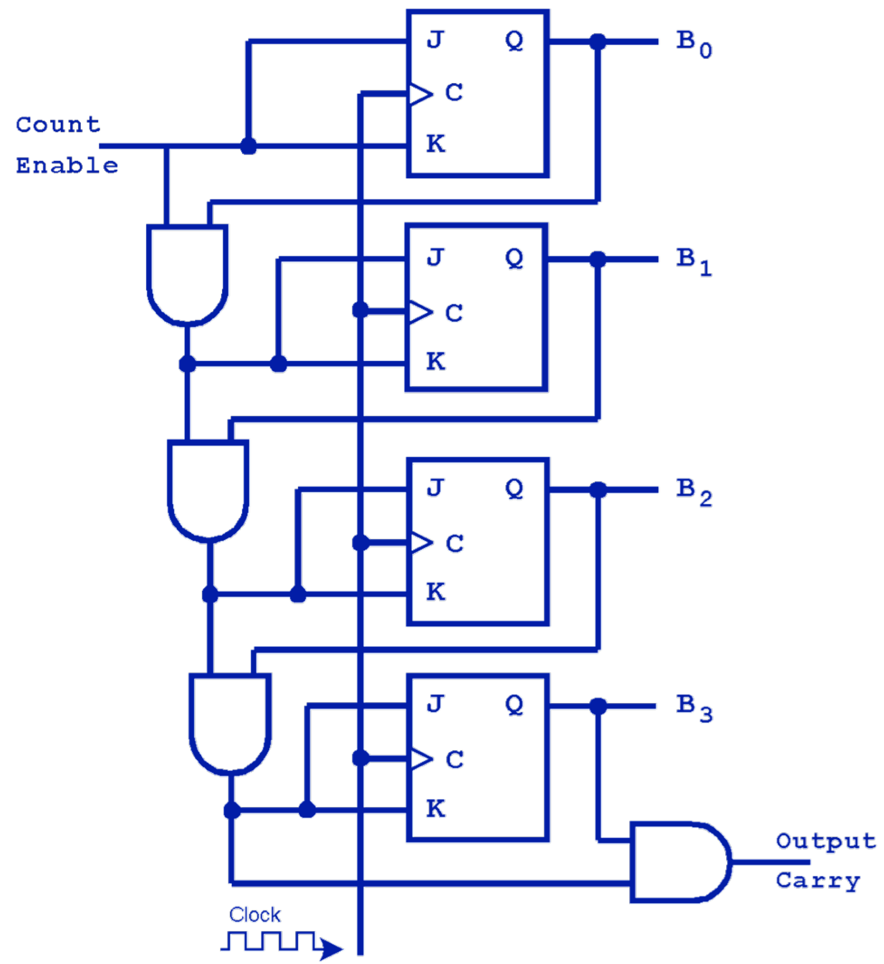


Sequential Circuits – Group of Registers



Sequential Circuits – Binary Counter

- Binary counter operation
  - JK flip-flops toggle when J=K=1
  - Low-order bit is complemented at each clock pulse
  - Whenever low order bit changes from 0 to 1, the next bit is complemented, and so on through the other flip-flops



Designing Circuits

- **Do designers usually lay out circuits by hand?**
  - No – designers today rely on specialized software to create efficient circuits
  - Software is an enabler for the construction of better hardware!
- Many challenges in modern hardware designs
  - Sheer number of gates to implement!
    - Create "building blocks" (modules) that can be quickly assembled
  - Timing constraints – Result is **correct**, but **when** is it correct?
    - Propagation delays occur between the time when a circuit's inputs are energized and when the output is accurate and stable

K-Maps

Introduction to Karnaugh Maps

- **Chapter 3A in textbook**
- Simplification of Boolean functions is good…
  - Produces simpler (and usually faster) digital circuits
- … but also time-consuming and error-prone
  - Easy to mis-use identities

Introduction to Karnaugh Maps

- K-Maps are an easy, systematic method for reducing Boolean expressions
  - Named after Maurice Karnaugh (engineer at Bell Labs in 1950's)
  - Invented a graphical way of visualizing and then simplifying Boolean expressions

Introduction to Karnaugh Maps

- A Kmap is a matrix representing a Boolean function
  - Rows and column headers represent the input values
  - Cells represent corresponding output values
- Input values are formatted as *minterms*
  - Minterm is a product term that contains all of the function's variables exactly once, either complemented or not complemented

Minterms

- For example, the minterms for a function having the inputs x and y are: $\bar{x}\bar{y}$, $\bar{x}y$, $x\bar{y}$, and xy
- Consider the Boolean function,

$$F(x,y) = xy + x\bar{y}$$

- Its minterms are:

Minterm	X	Y
$\bar{X}\bar{Y}$	0	0
$\bar{X}Y$	0	1
$X\bar{Y}$	1	0
XY	1	1

Minterms

- Function with three inputs?
  - Minterms are similar…
  - Just imagine counting in binary to find all the minterms…

Minterm	X	Y	Z
$\overline{X}\,\overline{Y}\,\overline{Z}$	0	0	0
$\overline{X}\,\overline{Y}\,Z$	0	0	1
$\overline{X}\,Y\,\overline{Z}$	0	1	0
$\overline{X}\,Y\,Z$	0	1	1
$X\,\overline{Y}\,\overline{Z}$	1	0	0
$X\,\overline{Y}\,Z$	1	0	1
$X\,Y\,\overline{Z}$	1	1	0
$X\,Y\,Z$	1	1	1

Introduction to Karnaugh Maps

- A Kmap has a cell for each minterm
  - Cell for each line for the truth table of a function
- The truth table for the function F(x,y) = xy is shown along with its corresponding Kmap

F(X,Y) = XY

X	Y	XY
0	0	0
0	1	0
1	0	0
1	1	1

X \ Y	0	1
0	0	0
1	0	1

Introduction to Karnaugh Maps

- Truth table and Kmap for the function F(x,y) = x + y
- This function is equivalent to the OR of all of the minterms that have a value of 1

$$F(x,y) = x + y = \bar{x}y + x\bar{y} + xy$$

F(X,Y) = X+Y

X	Y	X+Y
0	0	0
0	1	1
1	0	1
1	1	1

X \ Y	0	1
0	0	1
1	1	1

Introduction to Karnaugh Maps

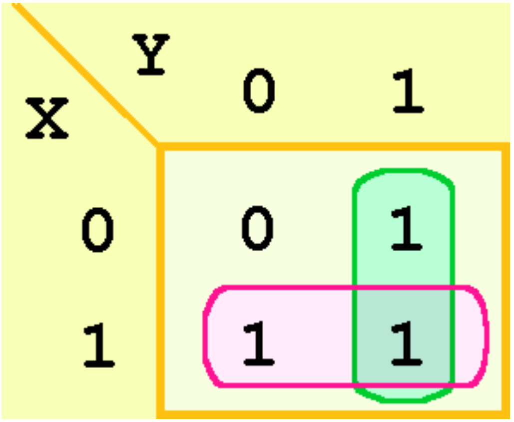
- Minterm function derived from Kmap was not in simplest terms
- Use Kmap to reduce expression to simplest terms
  - Find **adjacent 1's** in the Kmap that can be collected into groups that are **powers of two**

Two groups in this example:

X \ Y	0	1
0	0	1
1	1	1

Introduction to Karnaugh Maps

- Selected groups shown below
  - Groups are powers of two
  - Overlapping is OK!



Rules for Simplification

- Groupings can contain only 1’s; no 0’s
- Groups can be formed only at right angles
  - Diagonal groups are not allowed
- The number of 1’s in a group must be a power of 2
  - A single 1 is OK then, but not three 1’s!
- Groups must be made as large as possible
  - Otherwise simplification is incomplete
- Groups can overlap
- Groups can wrap around the sides of the Kmap

Kmap – Three Variables

- Extend to three variables? Easy!
- Note that the values for the yz combination at the top of the matrix form a pattern that is not a normal binary sequence
  - **Each position can only differ by 1 variable**

X \ YZ	00	01	11	10
0	$\overline{X}\,\overline{Y}\,\overline{Z}$	$\overline{X}\,\overline{Y}Z$	$\overline{X}YZ$	$\overline{X}Y\overline{Z}$
1	$X\overline{Y}\,\overline{Z}$	$X\overline{Y}Z$	XYZ	$XY\overline{Z}$

Kmap – Three Variables

- What do the values look like?
  - First row contains all minterms where x has a value of zero.
  - First column contains all minterms where y and z both have a value of zero

X \ YZ	00	01	11	10
0	$\overline{X}\overline{Y}\overline{Z}$	$\overline{X}\overline{Y}Z$	$\overline{X}YZ$	$\overline{X}Y\overline{Z}$
1	$X\overline{Y}\overline{Z}$	$X\overline{Y}Z$	XYZ	$XY\overline{Z}$

Kmap – Three Variables

- Example:

$$F(X,Y) = \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}Z + XYZ$$

- Kmap:

X \ YZ	00	01	11	10
0	0	1	1	0
1	0	1	1	0

- What is the largest group of 1's that is a power of 2?

Kmap – Three Variables

- Look at the grouping closely
  - Changes in the variables x and y have no influence upon the value of the function
  - Thus, the function

$$F(X,Y) = \bar{X}\bar{Y}Z + \bar{X}YZ + X\bar{Y}Z + XYZ$$

  - **reduces to F(x) = z**

You could verify this reduction with identities or a truth table

X \ YZ	00	01	11	10
0	0	1	1	0
1	0	1	1	0

Kmap – Three Variables

- Example:

$$F(X,Y,Z) = \bar{X}\bar{Y}\bar{Z} + \bar{X}\bar{Y}Z + \bar{X}YZ + \bar{X}Y\bar{Z} + X\bar{Y}\bar{Z} + XY\bar{Z}$$

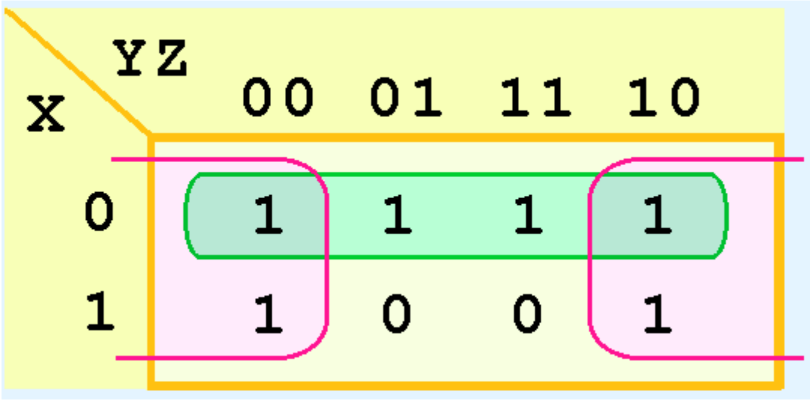
- Kmap:

X \ YZ	00	01	11	10
0	1	1	1	1
1	1	0	0	1

- **What are the largest groups of 1's that are a power of 2?**
  - **How many groups do you see?**

Kmap – Three Variables

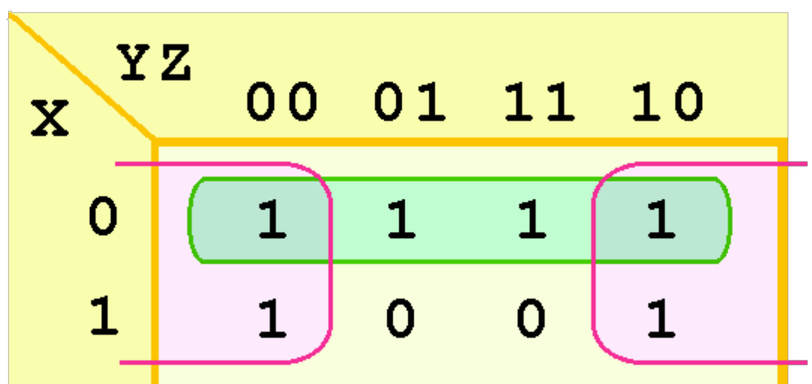
- To make the **largest groups possible**, wrap around the sides
- **How do we interpret results?**
  - Green row?
  - Pink square?



Kmap – Three Variables

- Green group – only the value of x is significant
  - Thus, $\overline{X}$
- Pink group – only the value of z is significant
- Our reduced function is: $F(X,Y,Z) = \overline{X} + \overline{Z}$

Recall that we had six minterms in our original function!



Kmap – Four Variables

- Model can be extended to accommodate a four-input function
  - 16 minterms produced

WX \ YZ	00	01	11	10
00	$\overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z}$	$\overline{W}\,\overline{X}\,\overline{Y}Z$	$\overline{W}\,\overline{X}YZ$	$\overline{W}\,\overline{X}Y\overline{Z}$
01	$\overline{W}X\overline{Y}\,\overline{Z}$	$\overline{W}X\overline{Y}Z$	$\overline{W}XYZ$	$\overline{W}XY\overline{Z}$
11	$WX\overline{Y}\,\overline{Z}$	$WX\overline{Y}Z$	$WXYZ$	$WXY\overline{Z}$
10	$W\overline{X}\,\overline{Y}\,\overline{Z}$	$W\overline{X}\,\overline{Y}Z$	$W\overline{X}YZ$	$W\overline{X}Y\overline{Z}$

Kmap – Four Variables

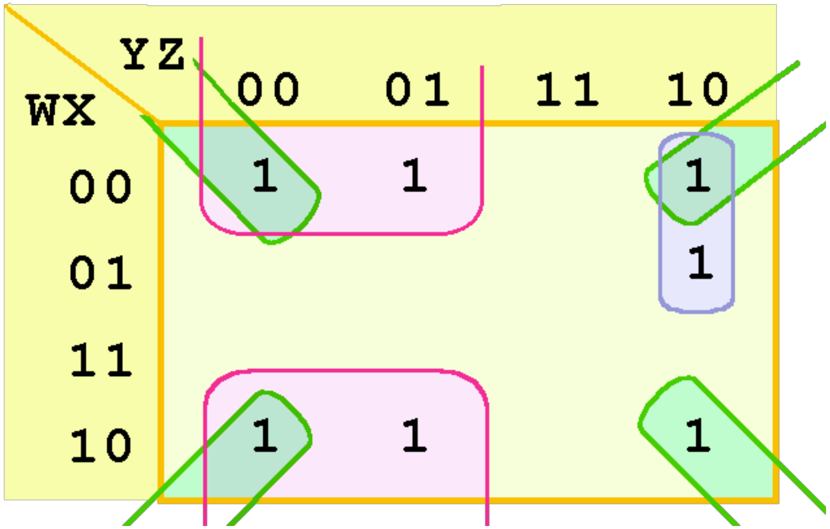
- Example: $F(W,X,Y,Z) = \overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z} + \overline{W}\,\overline{X}\,\overline{Y}Z + \overline{W}\,\overline{X}Y\overline{Z}$
  $+ \overline{W}XY\overline{Z} + W\overline{X}\,\overline{Y}\,\overline{Z} + W\overline{X}\,\overline{Y}Z + W\overline{X}Y\overline{Z}$

- Kmap (showing non-zero terms)
- **What largest groups should we select?**
  - Groups can overlap!
  - Groups can wrap!

WX \ YZ	00	01	11	10
00	1	1		1
01				1
11				
10	1	1		1

Kmap – Four Variables

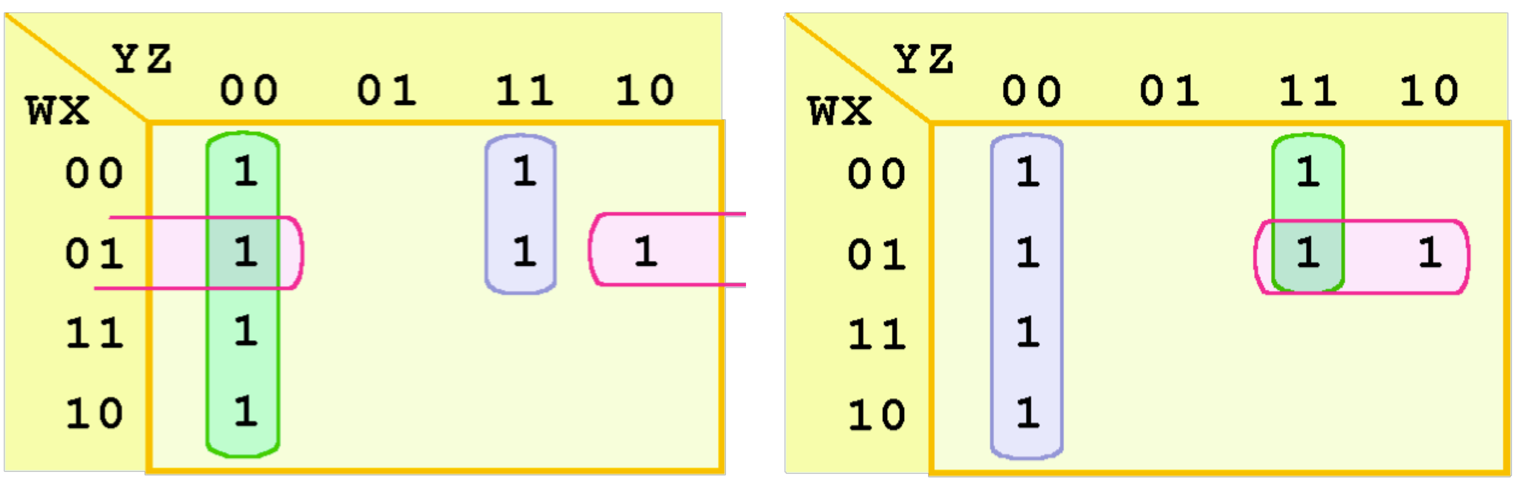
- Three groups
  1. Pink group that wraps top and bottom
  2. Green group that spans the corners
  3. Purple group entirely within the Kmap at the right



$$F(W,X,Y,Z) = \overline{X}\,\overline{Y} + \overline{X}\,\overline{Z} + \overline{W}Y\overline{Z}$$

Kmap – Four Variables

- Kmap simplification may not be unique
  - Possible to have different largest possible groups…
- The (different) functions that result from the groupings below are logically equivalent



Don’t Care Conditions

- Real circuits don’t always need to have an output defined for every possible input
  - Example: Calculator displays have 7-segment LEDs. These LEDs can display 2^7-1 patterns, but only ten of them are useful
- If a circuit is designed so that a particular set of inputs can never happen, we call this set of inputs a don’t care condition
  - Helpful for Kmap circuit simplification

Don’t Care Conditions

- Represent a don’t care condition with an X
- Free to include or ignore the X’s when choosing groups

WX \ YZ	00	01	11	10
00	X	1	1	X
01		X	1	
11	X		1	
10			1	

Don’t Care Conditions

- Grouping option #1:

WX \ YZ	00	01	11	10
00	X	1	1	X
01		X	1	
11	X		1	
10			1	

$$F(W,X,Y,Z) = \overline{W}\,\overline{X} + YZ$$

Don't Care Conditions

- Grouping option #2:

WX \ YZ	00	01	11	10
00	×	1	1	×
01		×	1	
11	×		1	
10			1	

$$F(W,X,Y,Z) = \bar{W}Z + YZ$$

Don't Care Conditions

- The truth table of

$$F(W,X,Y,Z) = \overline{W}\,\overline{X} + YZ$$

- differs from the truth table of

$$F(W,X,Y,Z) = \overline{W}Z + YZ$$

- However, the values for which they differ are the inputs for which we have don't care conditions
  - **Either is an acceptable solution**

Homework #1 Review

- **Grades** and **solutions** posted on **Sakai**
  - Papers available after class (for those with Sakai issues…)
- 50-word sentence - **Describe why the "Von Neumann bottleneck" constrains CPU performance**
  - *The Von Neumann Bottleneck is a constraint on stored program machines in which the computer is limited to a single path between the main memory and the CPU, which forces the CPU to alternate between fetching and processing data, thereby limiting efficiency and performance.*
    - 44 words (< 50 word limit), 1 sentence

Quiz #1 Review

- **Grades** and **solutions** posted on **Sakai**
- Problem 4 - **Why were transistors a huge technology improvement over vacuum tubes?**
  - Cooler, more reliable, cheaper, smaller, faster, …
- Problem 5 - **What does Moore's Law "promise"? As of 2011, is the law still in effect?**
  - Number of transistors you can buy (for fixed $$ / size) doubles ~2 years
  - Not "performance"!

Quiz #1 Review

- Problem 6 - **Memory is large and contains many instructions and data. How does the hardware know which instruction should be executed next?**
  - Program counter has next address in memory
- Problem 6 - **What function does the ALU perform?**
  - Mathematical operations! (Add, sub, mul, div, compare, …)