Announcements

Homework 10 (LAST!!) due TODAY by 5 pm

Final Project : Card Game

- Due December 12 – In-class Demos

Intermediate Deadlines

- Wed (11/30): Find project partner -- TODAY
  - Google Doc to find others (email to cs202-tas@cs.wisc.edu)
- Fri (12/2): Project proposal
  - At least 1 sentence email to cs202-tas@cs.wisc.edu (cc partner)
  - Writing more is useful
- Wed (12/7): Project draft to Learn@UW dropbox
  - Whatever you have completed

EXTRA Instructor Office Hours

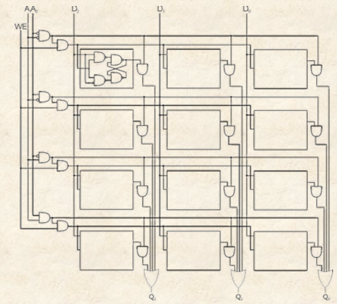
Tuesday and Thursday -- 1:30 – 4:30

UNIVERSITY of WISCONSIN-MADISON
Computer Sciences Department

CS 202: Introduction to Computation

Professor Andrea Arpaci-Dusseau

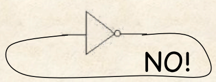
How does a computer… remember?



Review: Combinational Circuits

Combinational Circuit

- Always gives same output for given set of inputs
  - ex: adder always generates sum and carry, regardless of previous inputs
- Cycles are not allowed
  - Cannot have feedback from output back to input



- Useful for many, but not all, aspects of computation
  - Arithmetic Logic Unit (ALU)

Today's Challenge

How can we remember information with just

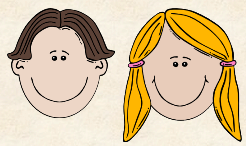
- AND
- OR
- NOT

?????????

Quick Review: Boolean Logic

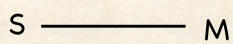
How to express:

Matt will go to party if and only if Sue goes to party?



Two boolean variables: M and S

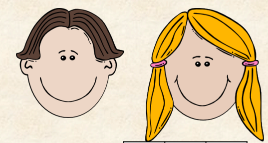
M = S

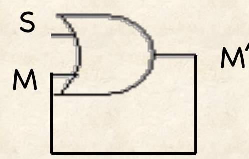


More Complicated Expression

Matt doesn't like changing his mind…

Matt will go to the party if Sue goes or if he already wanted to go





Enumerate ALL Scenarios in Truth Table

S	M	M′
0	0	0
0	1	1
1	0	1
1	1	1

Sequential Circuits

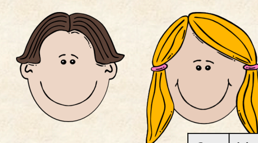
Sequential Circuit (vs. Combinational)

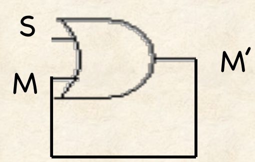
- Stores information: state
- Output depends on state + input
  - Given same input might produce different output, depending on stored information
  - *Example:* ticket counter
    - Advances when push button, output depends on previous state
- Cycles are allowed
  - Can have feedback from output to input
- Useful for building memory!

More Complicated Expression

Matt doesn't like changing his mind…

Matt will go to the party if and only if Sue goes or he already wanted to go





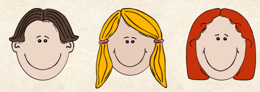
S	M	M′
0	0	0
0	1	1
1	0	1
1	1	1

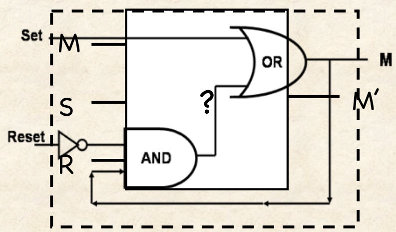
Problem with this circuit (or Matt)?

Once going, can't change mind! Once M=1, always M=1

How can Matt change his Mind?

Matt will go to the party if and only if Sue goes OR (he already wanted to go AND Rita does not go)





How would you express?

M′ = S OR (M AND NOT R)

R, S: "**control**" inputs

What is S doing?

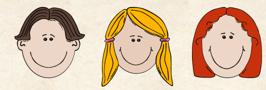
Setting state (to 1)

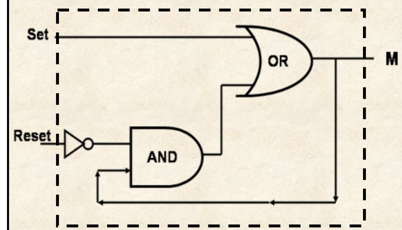
What is R doing?

Resetting state (to 0)

How can Matt change his Mind?

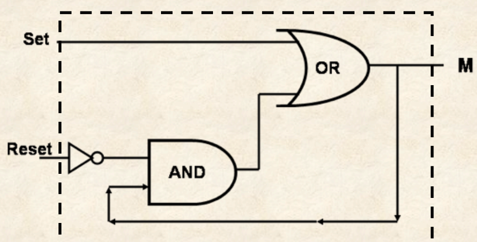
Matt will go to the party if Sue goes OR if the following holds: (he already wanted to go AND Rita does not go)





S	R	M	M′
0	0	0	
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	

R-S Flip-Flop (Caution: Simplified !!)



If Set = 1 (and Reset = 0), M = 1
If Reset = 1 (and Set = 0), M = 0
If Set = 0, Reset = 0, M keeps old value!

Not best if both Set and Reset = 1 (who wins?)

Random Access Memory (RAM)

Memory: Remembers lots of bits, not just 1 bit

- Logical $k \times m$ array of stored bits

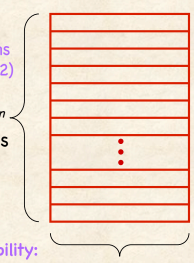
Two operations:

Write: Specify n-bit address and m-bits of data; remember that data at that address

Read: Specify n-bit address and see m-bits of data stored there

Address Space: number of locations (usually power of 2)

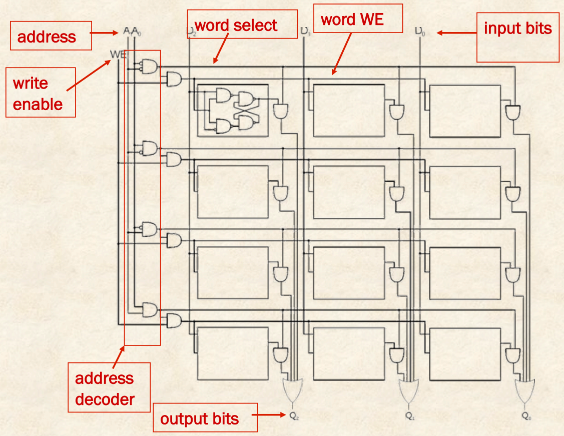
$k = 2^n$ locations



m bits

Addressability: number of bits per location (e.g., 32 bits)

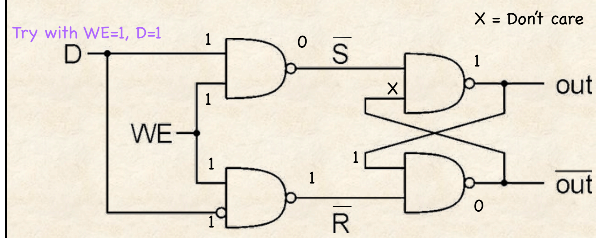
2^2 x 3 bit Random Access Memory



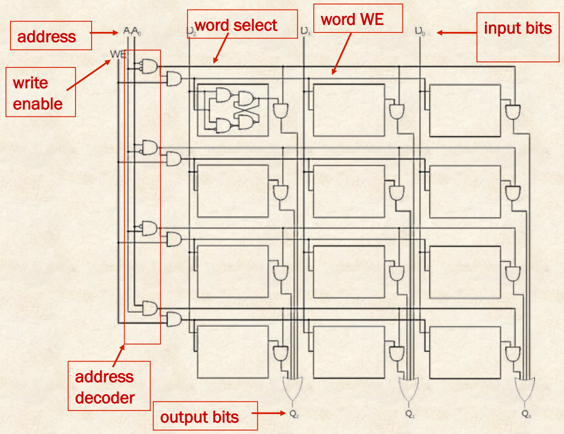
Convenient 1-Bit Memory

Two inputs: D (data) and WE (write enable)

- when WE = 1, latch is set to value of D
- when WE = 0, latch holds previous value (ignores D)

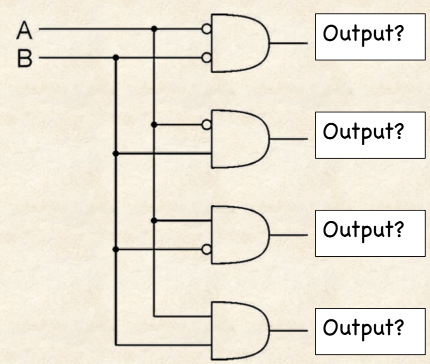


2^2 x 3 bit Random Access Memory



Mystery Circuit

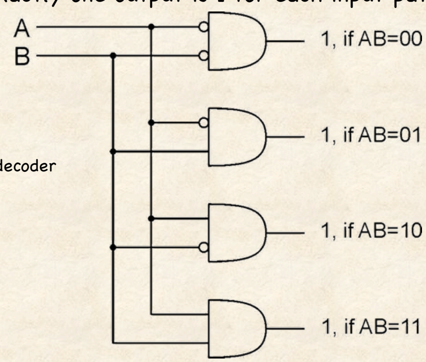
What does this combinational circuit do?



Decoder Circuit

n inputs, 2^n outputs

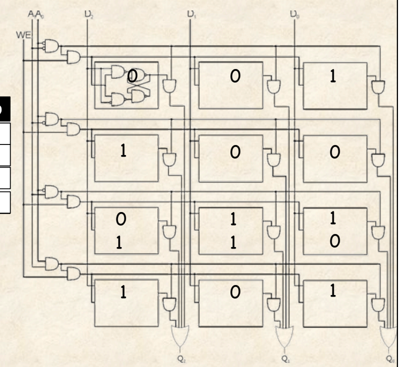
- exactly one output is 1 for each input pattern



Example RAM Operation

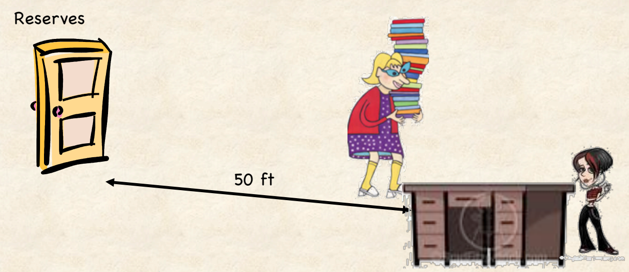
What happens when:

WE	A1	A0	D2	D1	D0	Q2	Q1	Q0
0	0	0	1	1	1			
0	0	1	1	1	1			
1	1	0	1	1	0			
0	1	0	1	1	1			



Other Components that Remember?

Cache Analogy: Inefficient Library



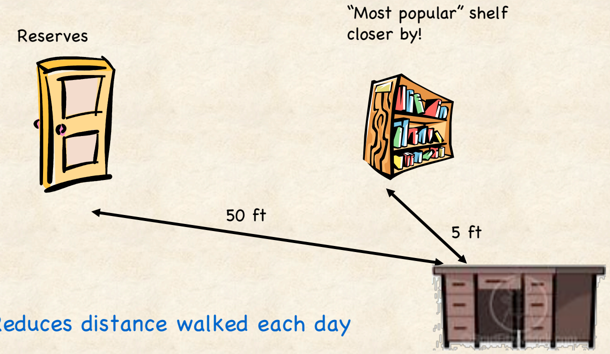
Distance covered?

- 1000 checkouts/returns per day
- 50ft x 2 x 1000 = 100,000 feet = ~ 20 miles

Please help, she's worn down and customers are waiting too long!!!

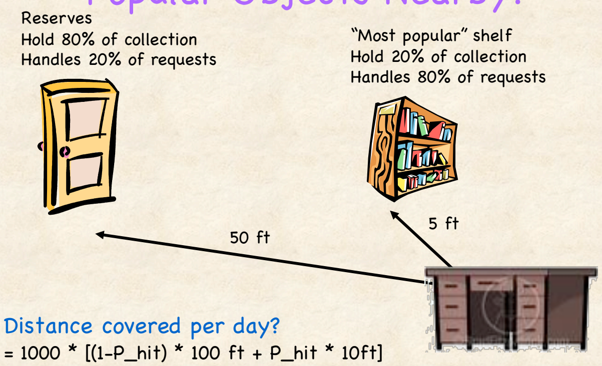
Ideas?

Better Arrangement: Popular Books Nearby



Reduces distance walked each day

Distance with Popular Objects Nearby?

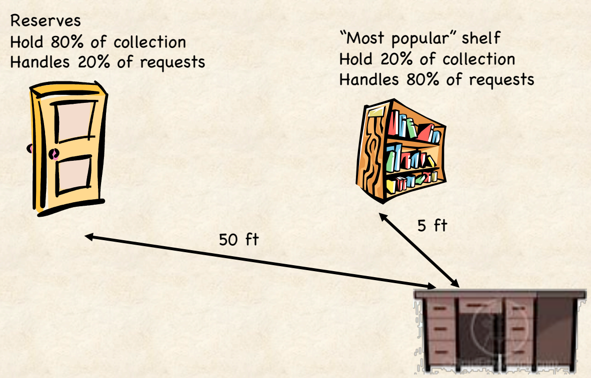


Distance covered per day?

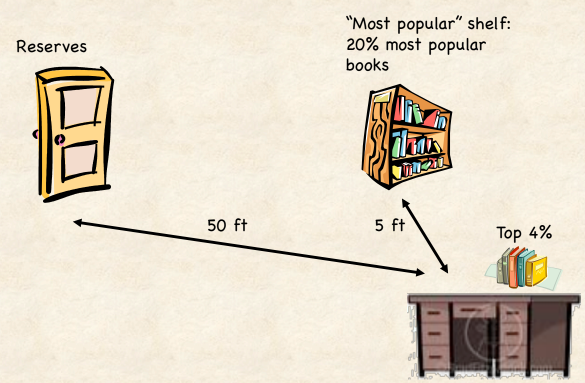
= 1000 * [(1-P_hit) * 100 ft + P_hit * 10ft]

= 1000 * [.20 * 100 + .80 * 10] = 28,000 ft instead of 100,000

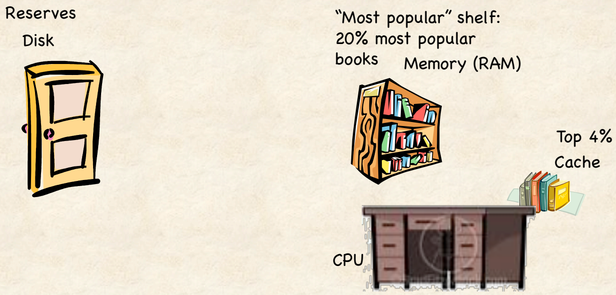
How can we improve?



Even better arrangement?



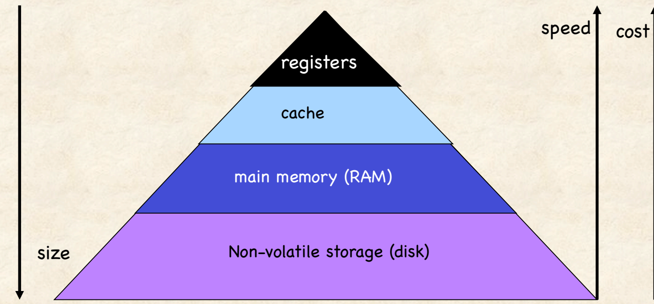
Computer ~~Librarian~~ arrangement



Often, today's computers have even more levels of caching
Level 1 : Closest to CPU
Level 2: Next closest…

Memory Hierarchy

Leverage memory hierarchy of machine architecture



Examples of Non-Volatile Storage

Hard disk drive

Tapes

RAIDs

Flash drives

NVRAM

Disk in Action

Modern Libraries

Today's Summary

Sequential circuits (vs. combinational)
- Can remember values using feedback loops in circuits!
- Implement Random Access Memory (RAM)

Homework 10 (LAST!!) due TODAY by 5 pm

Final Project : Card Game
- Due December 12 – In-class Demos

Intermediate Deadlines
- Wed (11/30): Find project partner -- TODAY
  - Google Doc to find others (email to cs202-tas@cs.wisc.edu)
- Fri (12/2): Project proposal
  - At least 1 sentence email to cs202-tas@cs.wisc.edu (cc partner)
  - Writing more is useful
- Wed (12/7): Project draft to Learn@UW dropbox
  - Whatever you have completed

EXTRA Instructor Office Hours

Tuesday and Thursday -- 1:30 – 4:30