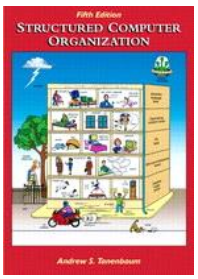
Class plan

Tuesday, January 22, 2013
5:27 PM



CSC260 Class Information

Professor David Sher

Office: B3068 Phone # 572-7383 ext. 26830 Email: davidbsher-Spring13@yahoo.com

webpage:: http://www.matcmp.ncc.edu/sherd/classdoc/csc260 Twitter:: Math4Sher - #shercsc260

Search twitter with **https://twitter.com/search/realtime?q=%23shercsc260**

Office Hours: Thursday 1-2:15, Wednesday 11:00-12:15

Book: Structured Computer Organization 5TH Edition by Andrew Tannenbaum

Lecture Plan		
TOPIC	**Periods**	**Tannenbaum SECTIONS**
1. Numerical Formats (2's complement and floating pt.)	1	Appendix A&B
2. Digital logic: gates (NOT,NAND,NOR, etc.) integrated circuits, combinatorial circuits	3	
3. Historical evolution of and contemporary multilevel machines, virtual machines.		3.1.1, 3.2, 3.3
4. Instruction execution & machine cycle: instruction and execution phases, hardware involved (MAR, MBR, IR, PC. buses, IAR)	1	1
5. Memory organization and properties: bits, metabits, addresses, secondary memory, ROM, RAM, EPROM, dynamic *vs*. static, volatility	2	2.1
6. I/O devices and processors/channels; modes of transmissions	1	2.2, 3.3
7. Codes: various types including error correcting & frequency dependent	1 3	2.3,5.1 2.2
8. **Example** microarchitecture: data path; design, timing and sequencing	3 3	4 5&handout
9. Another approach to assembly language programming - The **Java Virtual Machine**	3 1	6.1,6.2 6.3
10. Operating System - Virtual Memory		7.2
11. Operating System - Multiprocessing	1	
12. Macros: implementing a macro facility in an assembler		7.4

13. Linking and loading: object modules, linking, resolving external references, binding, dynamic relocation and linking		

Essays:

1. Give several examples of how using fewer bits in numbers and dates and other kinds of data can make a cell phone faster.
2. Describe several cell phone apps and explain for each app which part of the computer (like the cpu or the ram memory or the bus) is most important for running the app well.
3. For several apps (as in question 2) explain why they would run better on an IBM PC or a Java Virtual Machine. Explain how they could take advantage of the features of the machine.

Projects:	**Grading:**
1. Numerical Format Translations 2. Hardware Design 3. JVM Program.	**10%** Quizzes (~20) **24%** Projects (3) **21%** Essays (3) **45%** Exams (3)

DISABILITIES STATEMENT

If you have -- or suspect you have -- a physical, psychological, medical, or learning disability that may have an impact on your ability to carry out the assigned coursework, I urge you to contact the staff at the Center for Students With Disabilities, Bldg. U (behind the old College Union), 572-7241, TTY 572-7617. The counselors at CSD will review your concerns and determine with you what accommodations are necessary and appropriate. All information and documentation will be kept confidential.

Rules of the College Classroom

I. Classroom

- **A.** Professor decides where students sit.
- **B.** Professor has full control over when exams are given
- **C.** Many classes will start with Quizzes.
- **D.** If you come late, you will not receive any of this credit.
- **E.** You are responsible for making up any material that you missed.
- **F.** Class materials may not be available on days after they are given out.
- **G.** No profanity.

II. Expulsion (for the day) 3 Expulsions (not absenses) are an F

- **H.** *!*$**!*!!!%*

  Cursing
- **I.** Talking during exam or quiz (also 0 on exam or quiz)
- **J.** Talking during lecture
- **K.** Making noises, whistling, or playing music.
- **L.** Touching other people or their stuff
- **M.** Cell phone ringing. Cell Phones should be in silent mode during class.
- **N.** Talking on a cell phone

- **O.** Being removed 3 times from class or vandalizing any classroom will result in an F as your class grade (Failing the entire course). Expulsion means being thrown out of class for behavior (not being absent). Vandalism will also get an F.

III. Assignments

- **P.** Late Assignments will not be accepted.
- **Q.** Unless a doctor's note or police report is submitted.
- **R.** All take home assignments must be typed.

S. If you are not in school when the assignment is due mail the assignment.

T. Use a post office to mail the assignment to:
David Sher
Math Department
Nassau Community College
1 Education Dr.
Garden City NY 11530

24 hour post office in NYC: is located at 421 Eighth Avenue, between 31st Street and 33rd Street in the New York City borough of Manhattan, across the street from Pennsylvania Station and Madison Square Garden.

Hicksville Post Office ***Lobby Hours: Mo-Fr 7:30am-11:30pm, Sa 8am-3pm:*** 185 West John St., Hicksville (OysBy)

U. Assignments will be returned only to the student who did them. (You can not pick up your friend's assignment.)

IV. Exams & Withdrawel

V. Exams and quizzes are open notes, open book.

W. Students may not leave once they start an exam until they finish.

X. Cell phones and computers must be off during exams.

Y. Students must have their college ID to take the exam

Z. Students can withdraw till 1 week before the *final*. They must fill out a withdrawal form or they will get a letter grade.

Using Twitter

- Use https://twitter.com/search/realtime?q=%23shercsc260 to search for #shercsc260 before class without logging in to twitter
- You can also login
- Text "follow @math4Sher" To 40404
- Tweets well be texted to you
- "off @math4sher" stops twitter sms
- Focus on tweets containing #shercsc260

How to call your professor (checklist)

- ✔ Call 1-516-572-7383 ext 26830
- ✔ State your name
- ✔ Say the date and time
- ✔ State which section you are in
- ✔ State your phone number
- ✔ Explain what you need (for example: more time on an assignment).
- ✔ Explain why you need it (for example: you were sick and have a note from your doctor).
- ✔ State times to call back
- ✔ Restate your phone number
- ✔ Restate your name
- ✔ Restate the day and time

How to email your professor (checklist)

- ✔ Email to davidbsher-spring13@yahoo.com
- ✔ CC to: David.Sher@ncc.edu
- ✔ In the subject line should be your section
- ✔ and what you want (briefly)

- ✔ In the body: State your name
- ✔ explain what you need (for example: more time on an assignment).
- ✔ Explain why you need it (for example: you were sick and have a note from your doctor).
- ✔ Restate your email
- ✔ Give an alternate email
- ✔ If you want a phone reply give your phone #
- ✔ State times to call back

Grades and other course details

Grade	Average	Other issues
A	90 100	Getting thrown out of 3 classes for misbehavior will result in are F. Vandalizing school equipment will result in an F. Students who do not buy the textbook usually fail. Use the math center in B130 to avoid failing and to catch up. Come to the professor's office hours for extra help in B3068 See the website at http:// www.matcmp.ncc.edu/sherd/classdoc/csc260 .
B+	85-90	
B	80-85	
C+	75-80	
C	70-75	
D+	65-70	
D	60-65	
F	0-60	

Sample Essay 2
Why use comments to build a JAVA ISP

David Sher

Comments are an important tool for programming. They are especially

Comments are an important tool for programming. They are especially important when constructing large projects. Below we discuss why comments are particularly useful for building a program like a java ISP.

- A java ISP. is a valuable program. Comments allow you to copyright the source code of this program and thus prevent others from using your work to make money from a java ISP.
- Many revisions of a java ISP. have come out. Each time it is revised code that may have been written years ago needs to be changed. Comments help programmers understand old code and speed changing it.
- Comments can notify programmers that two pieces of code depend on each other and when one is changed the other must be too. Such comments avoid inconsistencies and bugs in revisions. Since a java ISP is frequently revised, this is an important use for comments when building A java ISP.
- A java ISP is a large programming project. Many programmers work together to build it. Comments help each programmer understand what the other programmers are doing.
- A java ISP contains many function calls in its code to perform the many operations accessed through the menus. Comments can clarify what each function actually does and what their parameters are for.
- Because a java ISP was built by multiple programmers, you may not know which programmer to ask about a feature or bug in the program. Comments can indicate which programmer built or revised which part of the program; then you know who to look up

- **Project 1:**
  Translating into Arrays of Bits and Implementing Operations
  Due:_____

  You will write a program in the language of your choice with a method that takes a number between -127 and 127 and translates it into 8 bytes that were either '1' or '0'. These 8 bytes represent the number in 2's complement. You should also have a function that can take an array of bytes and output it. You should write a method that takes an array of bytes and returns true if the array is a palindrome and false otherwise. You also need to write a method that returns true if the array of bytes represents the number of letters in your (the programmer's) last name and false otherwise without translating the array of bytes back into an integer. You should write a similar method for the number of letters in your (the programmer's) first name. You will write a program that accepts a number from the user and shows what each method returns.

Project 2

Designing Circuits

Due:_____

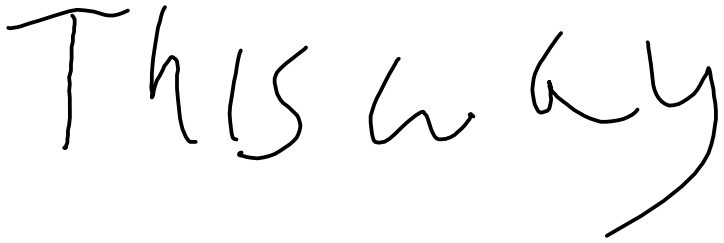
Design a circuit to take the bits from project 1 and do project 1. Your circuit should consist of logic gates and wires. The circuit will take 6 bits representing a 2's complement number in 6 input wires and have 3 output wires for the three methods.

Project 3

JVM Progamming

Due:_____

Write a JVM method that translates a string into an array of binary bits. Write a JVM method that implements the methods you should have implemented in Project 1. Read a number from the user and output the results of your methods using your algorithms.



1/28

Monday, January 28, 2013
1:53 PM

Computers manipulate numbers
Numbers hang inside wires
Digital means a small number of fixed distinct values.
Number are binary 0 or 1 voltage (charge)
Synchronized devices specify how long each operation takes
Asynchronous devices signal when results ready
Paradoxically asynchronous is faster
Synchronous hardware is simpler
Easier to debug
Choice between how many wires and how many cycles.
32 bit computers get 32 bit numbers in 1 cycle
64 bit computers get 64 bit numbers in 1 cycle
Integers are represented by 2's complement
To negate a number complement (0 to 1, 1 to 0) and add 1

23
128 64 32 16 8 4 2 1
0 0 0 1 0 1 1 1
-23

1 1 1 0 1 0 0 0

1 1 1 0 1 0 0 1

Only 1 representation of 0

Arithmetic worksthe same for positive and negative

High bit (128) tells if positive or negative

Hw

-75 38 -110 in 2's complement 8 bit

sp

1/29

Wednesday, January 30, 2013
1:48 PM

-88 8bit 2s complement
88 binary 01011000
Complement 10100111
Add 1 to get 10101000 is -88
Check
Complement: 01010111 add 1 to get 01011000 = 88

Floating pt.
Why
Need fractions
Why not fixed point
32 bits not enuf for fixed point.

Scientific notation is natural
1 <= number < 10
Power of 10
3e8 = 3*10^8 = 300,000,000

Floating point
0 <= significand < 1
Power of 2

Sign bit
Pg 695

18 = 10010*2^0 = .10010*2^5
5+127=132 = 10000100
0 10000100 00100000000000000000000
18 in single precision

22/7
11
2/7 0
4/7 0
8/7 1
2/7 0
4/7 0
8/7 1

11.001001001 …
.11001001001 … *2^2
0 10000001 00100100100100100100100

-22/7
1 10000001 00100100100100100100100

HW

Translate to single precision floating pt.

.1

3*10^8

-2/11

2/4

Monday, February 04, 2013
1:46 PM

-.3

1 01111110 00110011001100110011001

.3
.6 0
1.2 1
.4 0
.8 0
1.6 1
1.2 1
.01001

$2^{10} \approx 1000$

$2^{20} = (2^{10})^2 \approx 1000000$

Essay1 due 2/13

Hw: read 3.1

2/6

Wednesday, February 06, 2013
11:38 AM

The New York State Mathematics Association of Two-Year Colleges (NYSMATYC) is sponsoring a statewide math contest on 3/14. The contest is a one-hour long test consisting of 20 challenging questions. Students receive 4 points for a correct answer, no points for answers left blank and a negative point for incorrect answers. To get an idea of the type of questions on the test, old exams and their solutions are available at **http://www.nysmatyc.org/league/oldexams.php**.

1/13

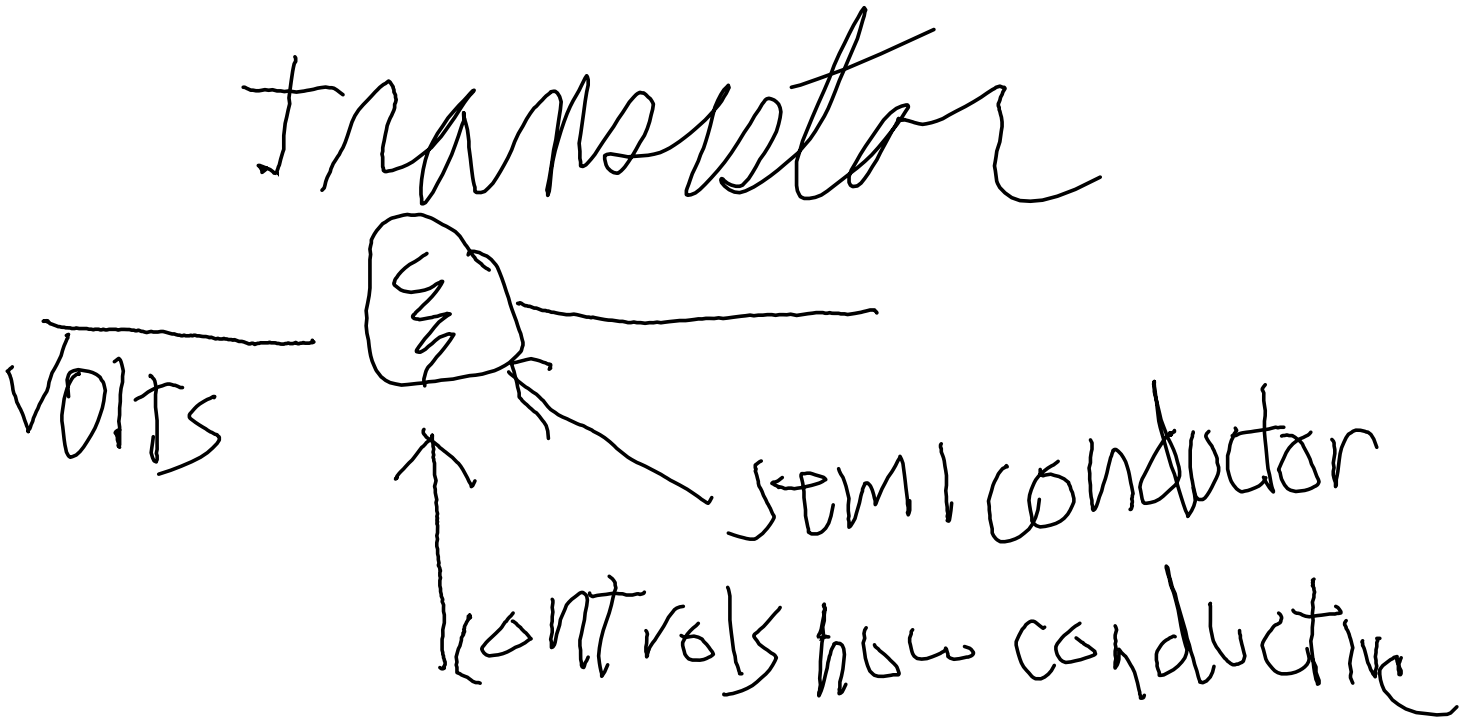
2/13	0
4/13	0
8/13	0
16/13=1 3/13	1
6/13	0
12/13	0
24/13=1 11/13	1
22/13=1 9/13	1
18/13=1 5/13	1
10/13	0

20/13=1 7/13	1
14/13=1 1/13	1

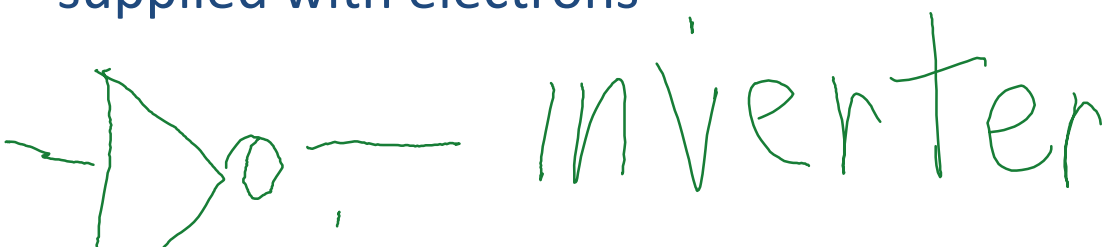
.000100111011

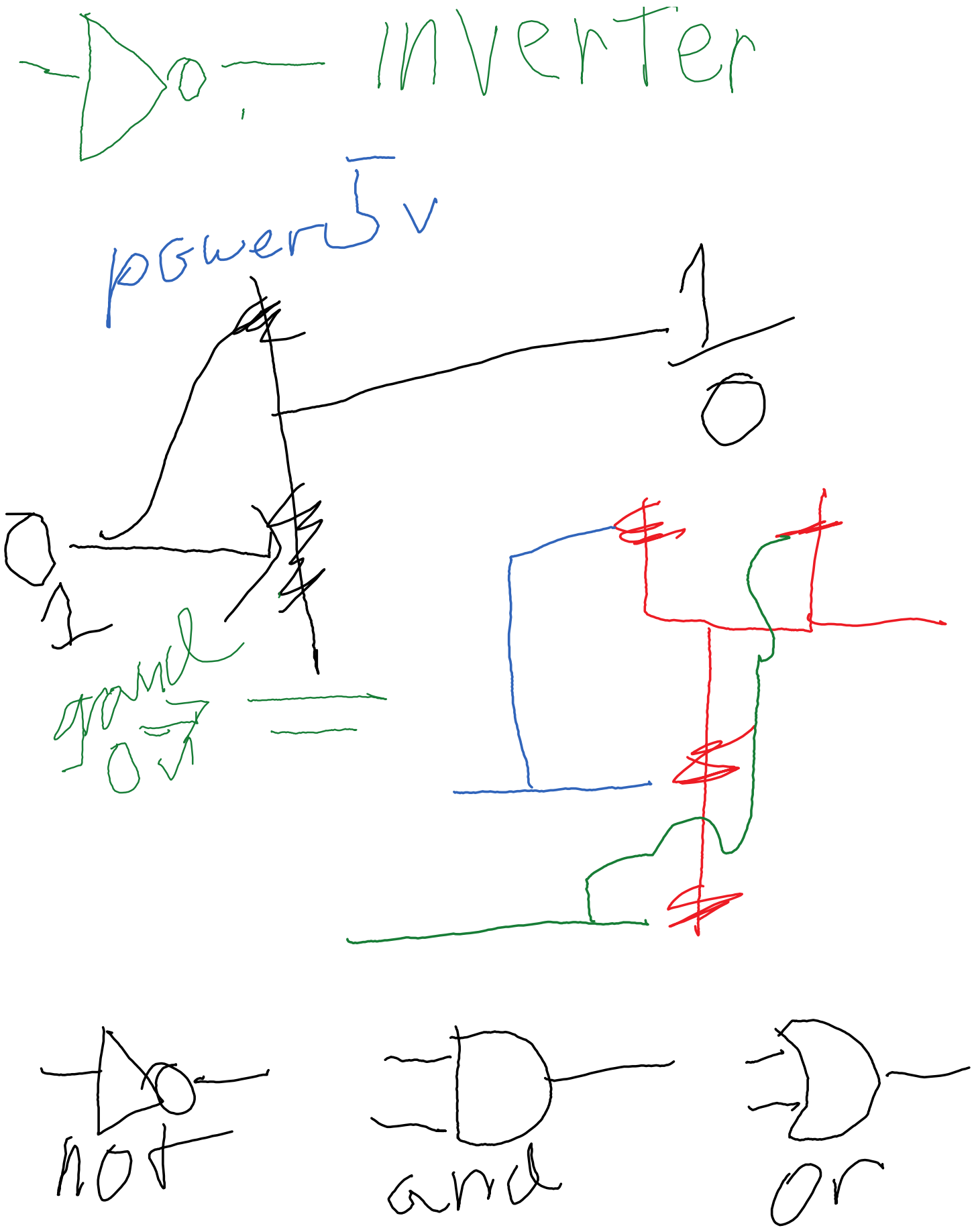
0 01111011 00111011000100111011000

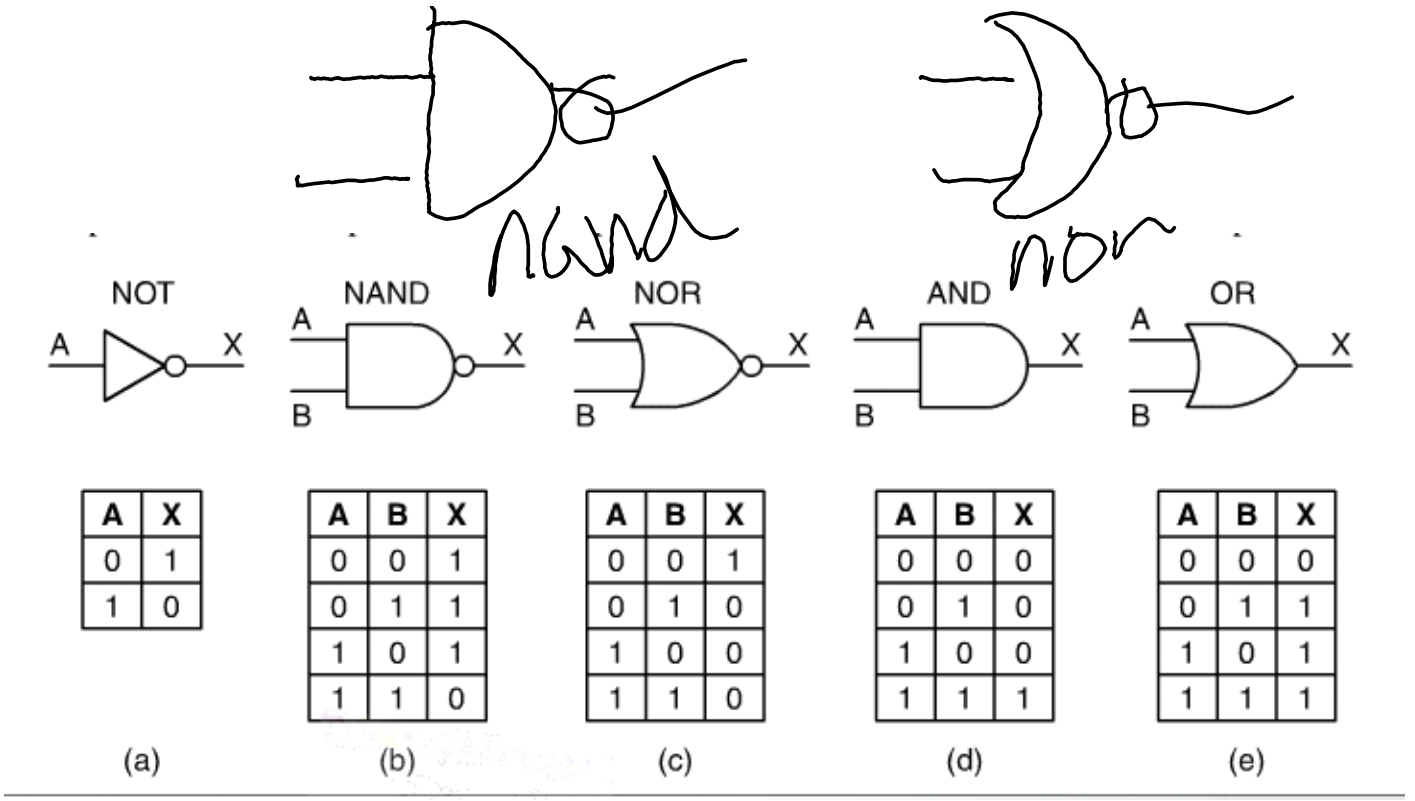
Essay 1 due 2/14 emailed



Semiconductor only conducts when it is supplied with electrons




inverter
power 5 v
ground 0 v
not
and
or



A	X
0	1
1	0

(a)

A	B	X
0	0	1
0	1	1
1	0	1
1	1	0

(b)

A	B	X
0	0	1
0	1	0
1	0	0
1	1	0

(c)

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

(d)

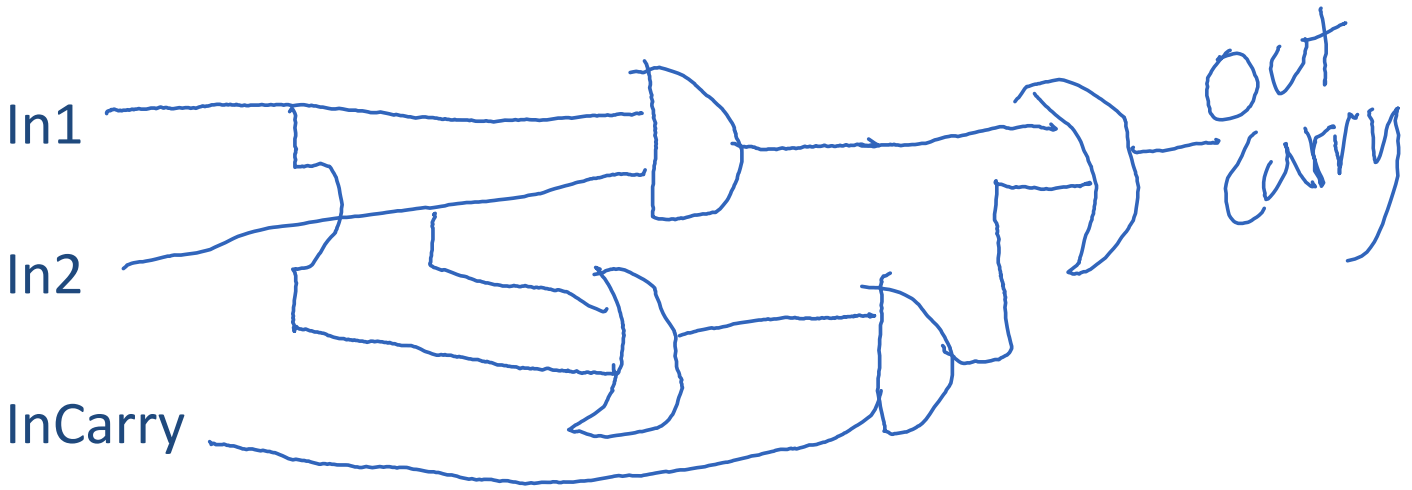
A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

(e)

Screen clipping taken: 2/6/2013 2:54 PM

ADDING

In1	In2	InCarry	Out	OutCarry
0	0	0	0	0
1	0	0	1	0
0	1	0	1	0
1	1	0	0	1
0	0	1	1	0
1	0	1	0	1
0	1	1	0	1
1	1	1	1	1

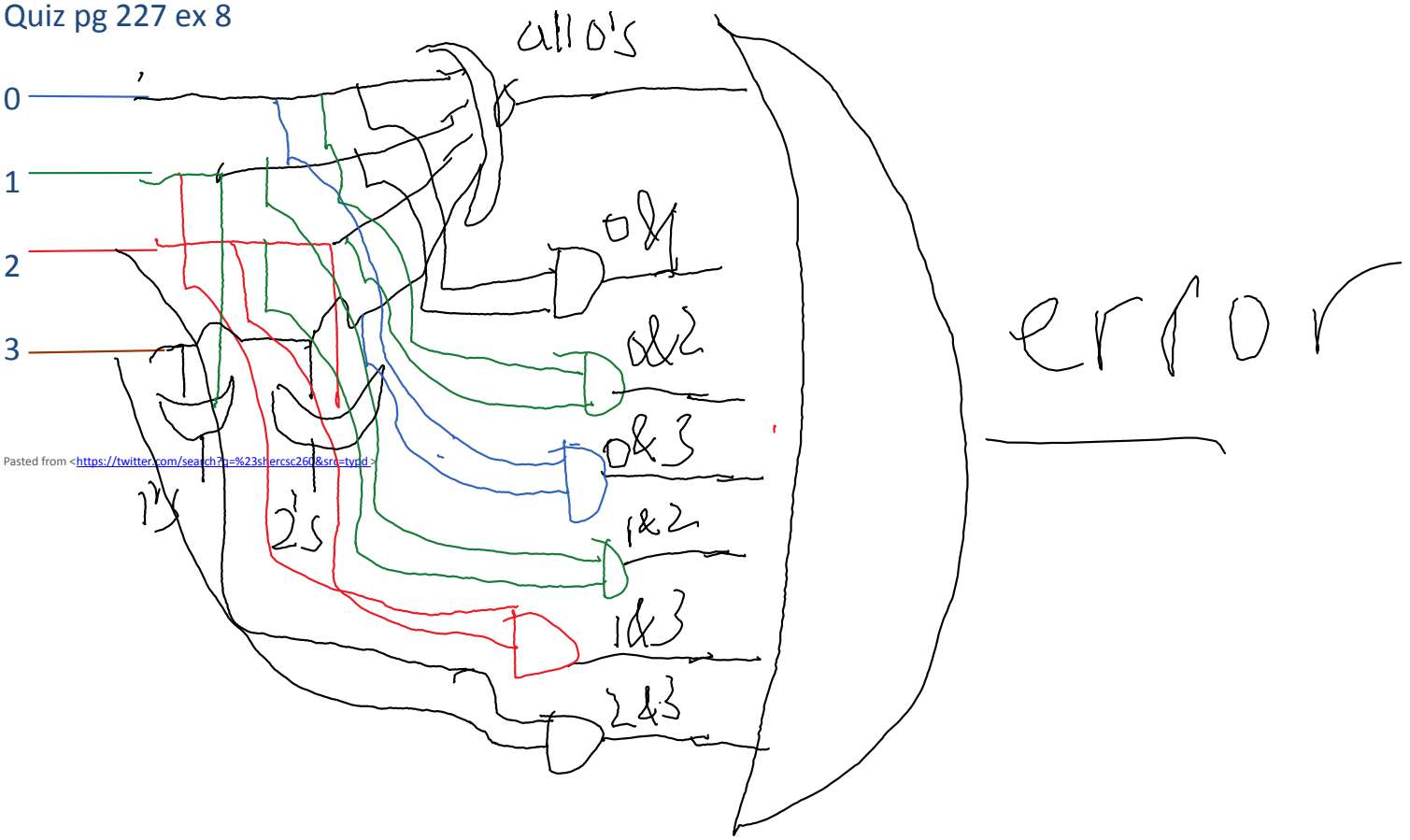


Proj 1 due 2/23

2/11

Monday, February 11, 2013
1:46 PM

Quiz pg 227 ex 8

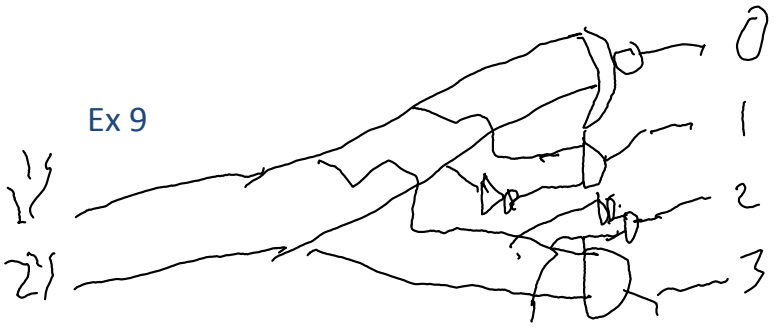


$$\sim(A \wedge B) = \sim A \vee \sim B$$

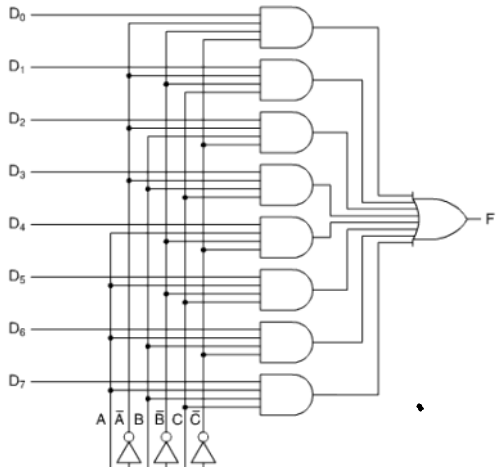
$$\sim(A \vee B) = \sim A \wedge \sim B$$

$$A \vee B = \sim(\sim A \wedge \sim B)$$

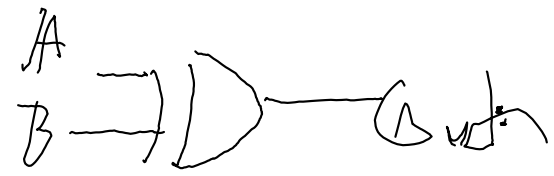
Ex 9



Pg 149



Screen clipping taken: 2/11/2013 3:06 PM



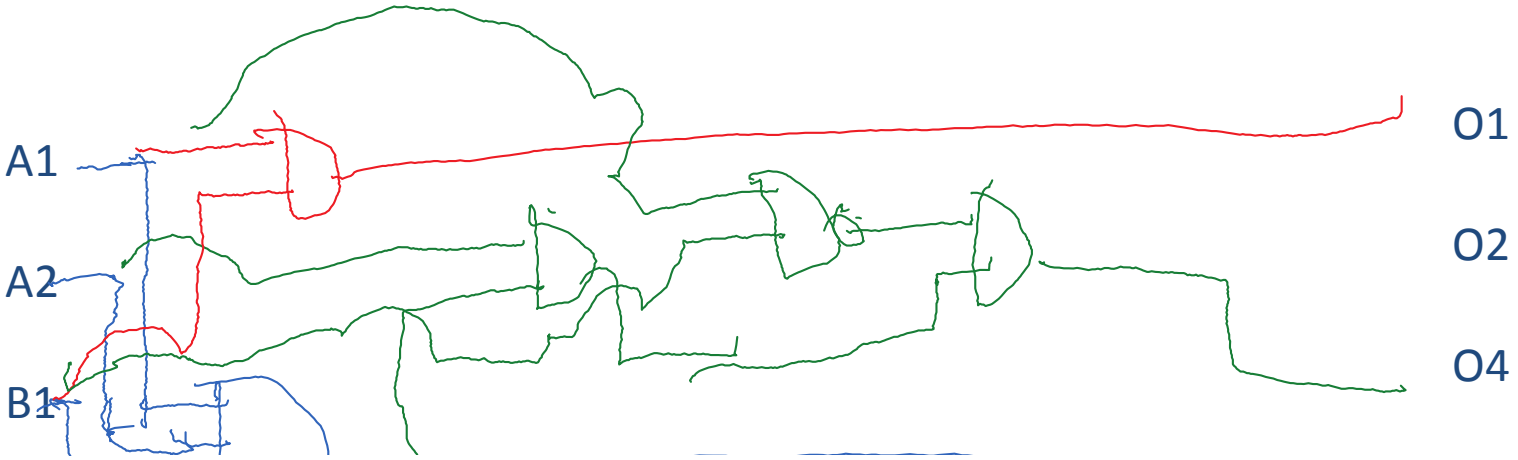
A	B	Out
1	1	0
0	1	1
1	0	1
0	0	0

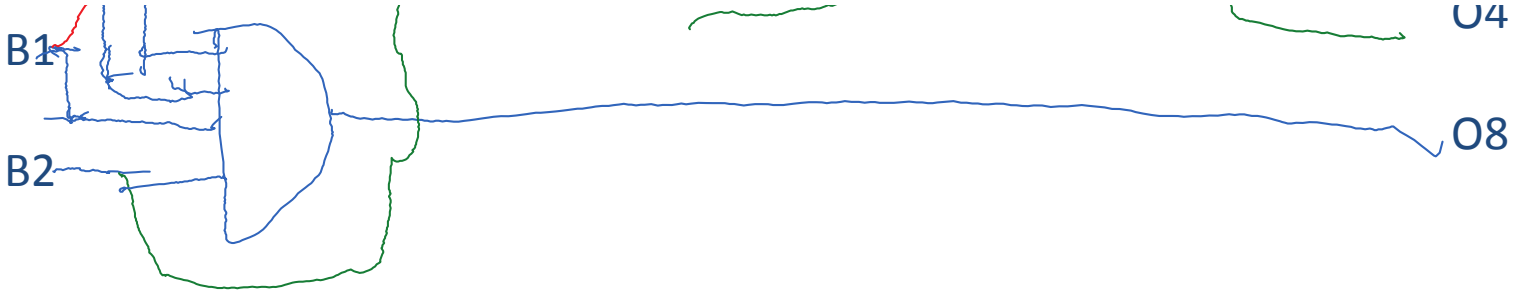
HW
read
3.2
3.3

2/13

Wednesday, February 13, 2013
1:56 PM

A1	A2	B1	B2	Out1	Out2	Out4	Out 8
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0
1	1	0	0	0	0	0	0
0	0	1	0	0	0	0	0
1	0	1	0	1	0	0	0
0	1	1	0	0	1	0	0
1	1	1	0	1	1	0	0
0	0	0	1	0	0	0	0
1	0	0	1	0	0	0	0
0	1	0	1	0	0	1	0
1	1	0	1	0	1	1	0
0	0	1	1	0	0	0	0
1	0	1	1	1	1	0	0
0	1	1	1	0	1	1	0
1	1	1	1	1	0	0	1

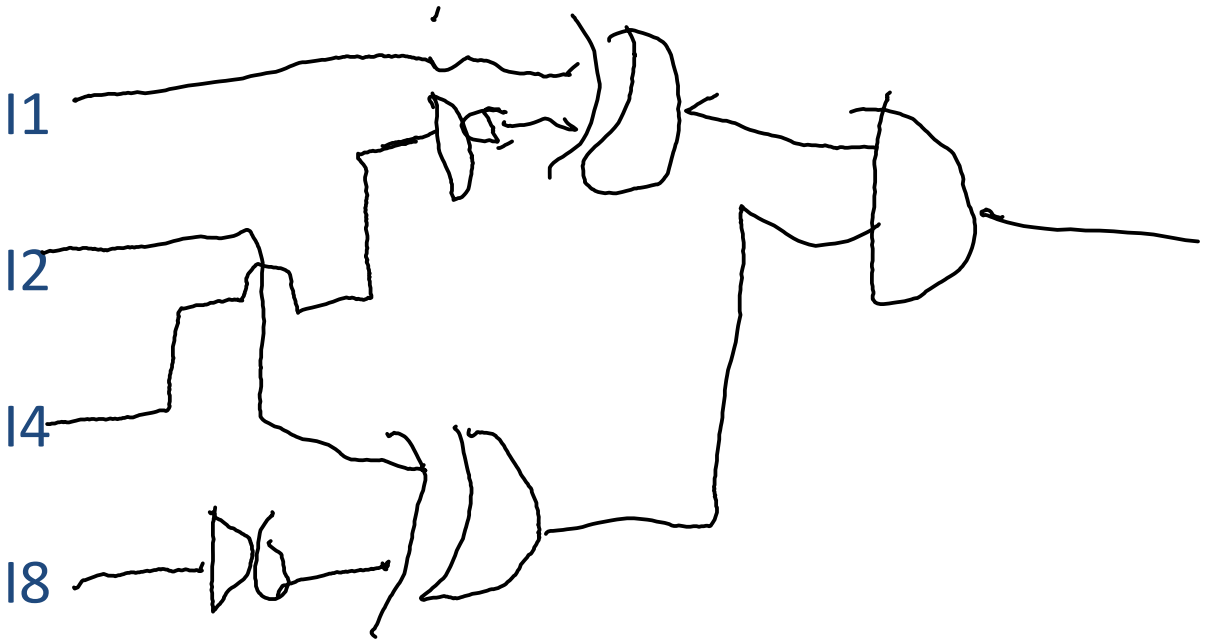


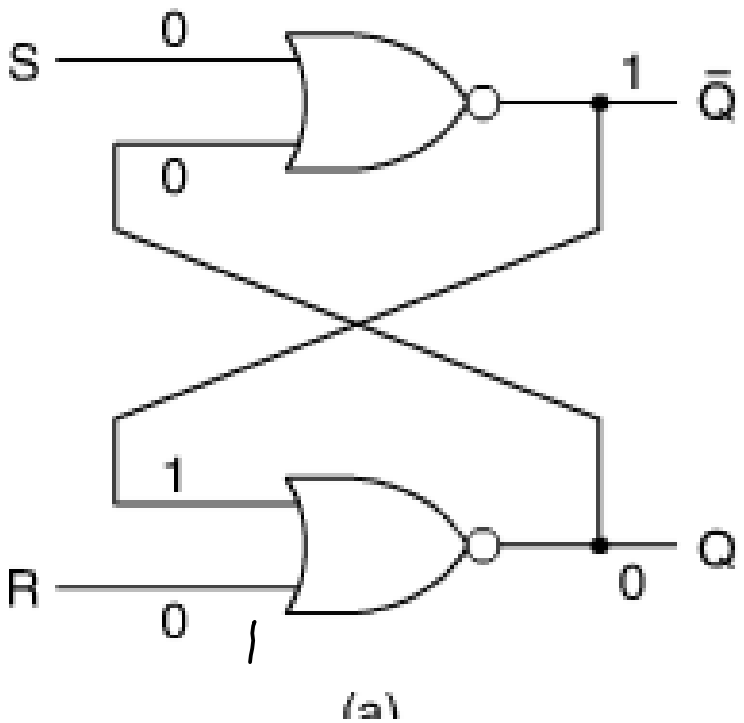

B1
B2
O4
O8

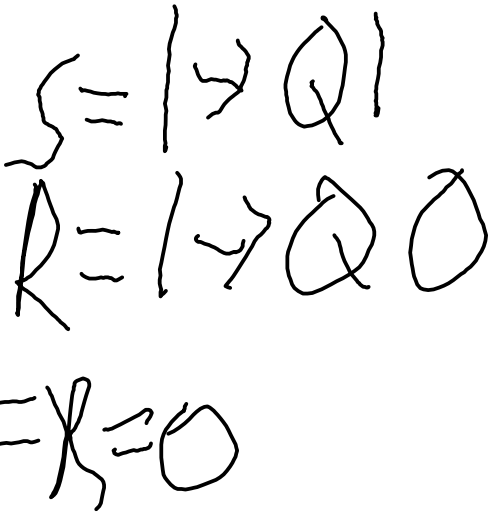
2/25

Monday, February 25, 2013
1:50 PM

Exam 3/6
Proj 2 3/16







Screen clipping taken: 2/25/2013 2:34 PM

Cpu - central processing unit
Alu - arithmetic logic unit

Cpu
Alu +registers+command parsing

Alu
Add , negate, invert, and or, shift (x=x<<2)

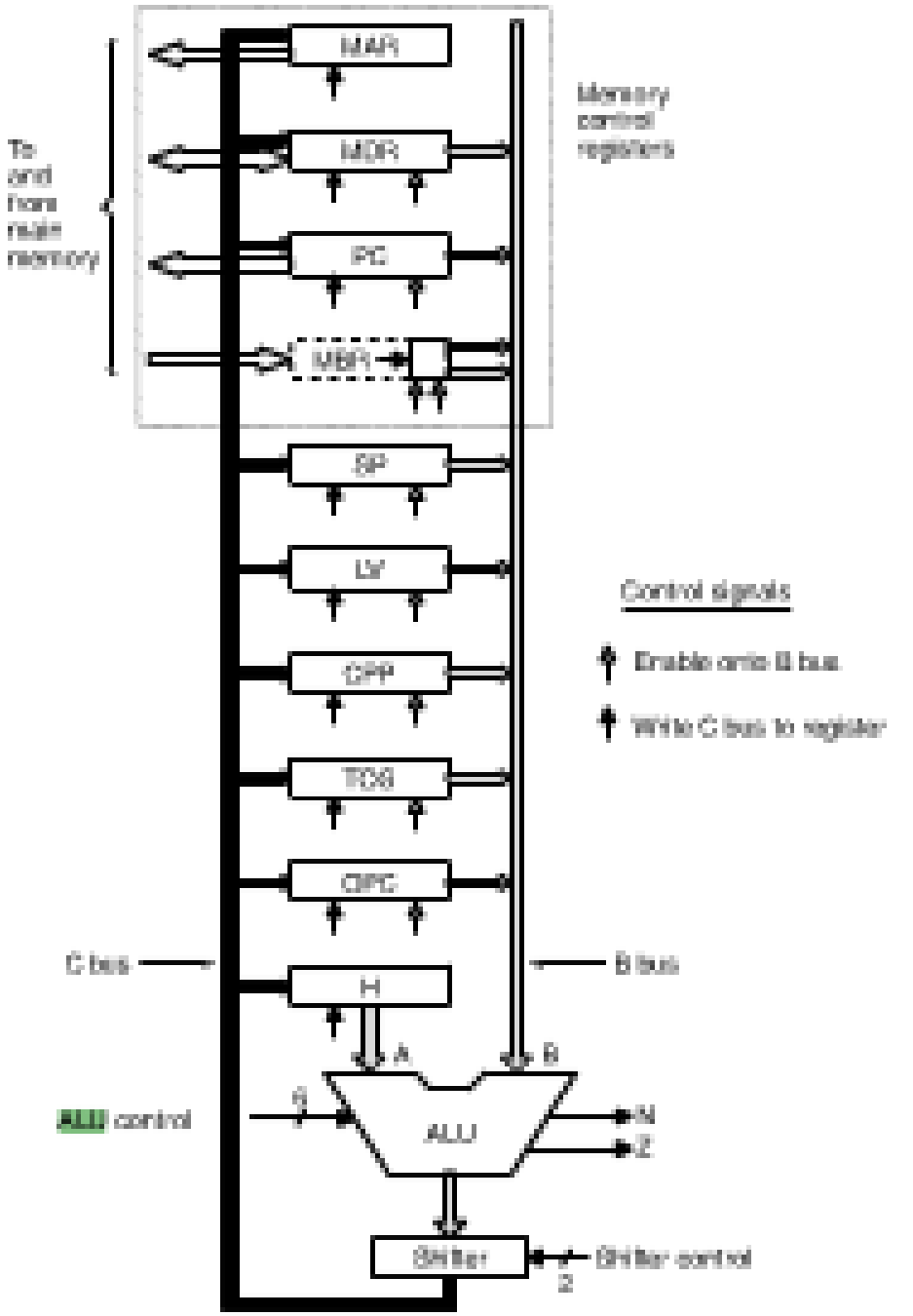


Figure 4-6. The data path of the example microarchitecture used in this chapter.

Screen clipping taken: 2/25/2013 2:57 PM

2/27

Tuesday, February 26, 2013
12:58 PM

the American Mathematical Association of Two-Year Colleges (AMATYC) Student Mathematics League contest will be held on Tuesday, March 5, 2013, in room B-213 during Club Hour.
The level of the test is precalculus mathematics. The test contains 20 questions drawn from a standard syllabus in College Algebra and Trigonometry and may involve precalculus algebra, trigonometry, synthetic and analytic geometry, and probability; questions that are completely self-contained may be included as well. All questions are short-answer or multiple choice.
Calculators are allowed, including graphing calculators, provided they do not have a typewriter-like keyboard (for example, the TI-92) or a disk drive.
Past contest questions and answers are available at:
http://www.amatyc.org/SML/old-competition-questions.htm
For interested students, a sign-up sheet is posted outside my office, room D-3123.

Pasted from <https://bl2prd0310.outlook.com/owa/>

10pts

1. Translate -100 into 8 bit two's complement.

20 pts

2. Translate -2/9 into the class' floating pt format.

35pts

3. Design a circuit that determines if a 4 bit number is prime.

35pts

4. Assume a 8ns bus cycle. How many cycles would it take to do a write given the timings from page 181?

1. Translate 100 and negate
   64+32+4=100
   01100100
   Complement: 10011011
   +1 : 10011100
   Check
   Complement 01100011
   +1: 01100100
2. -2/9 to binary - means 1st bit is 1

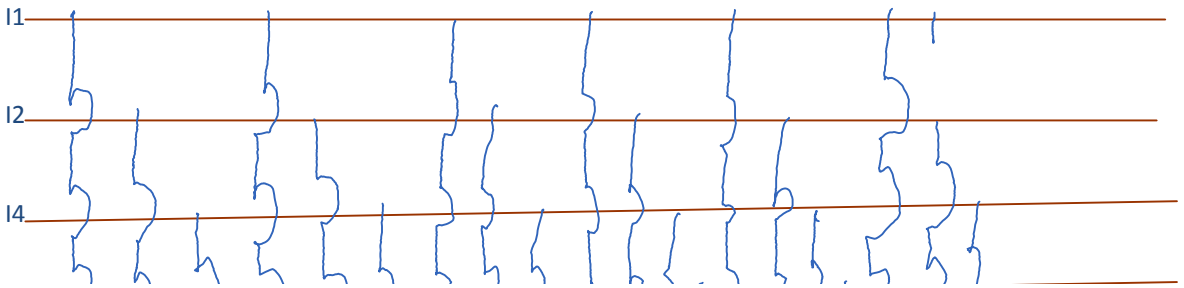
4	0
8	0
16->7	1
14->5	1
10->1	1
2	0

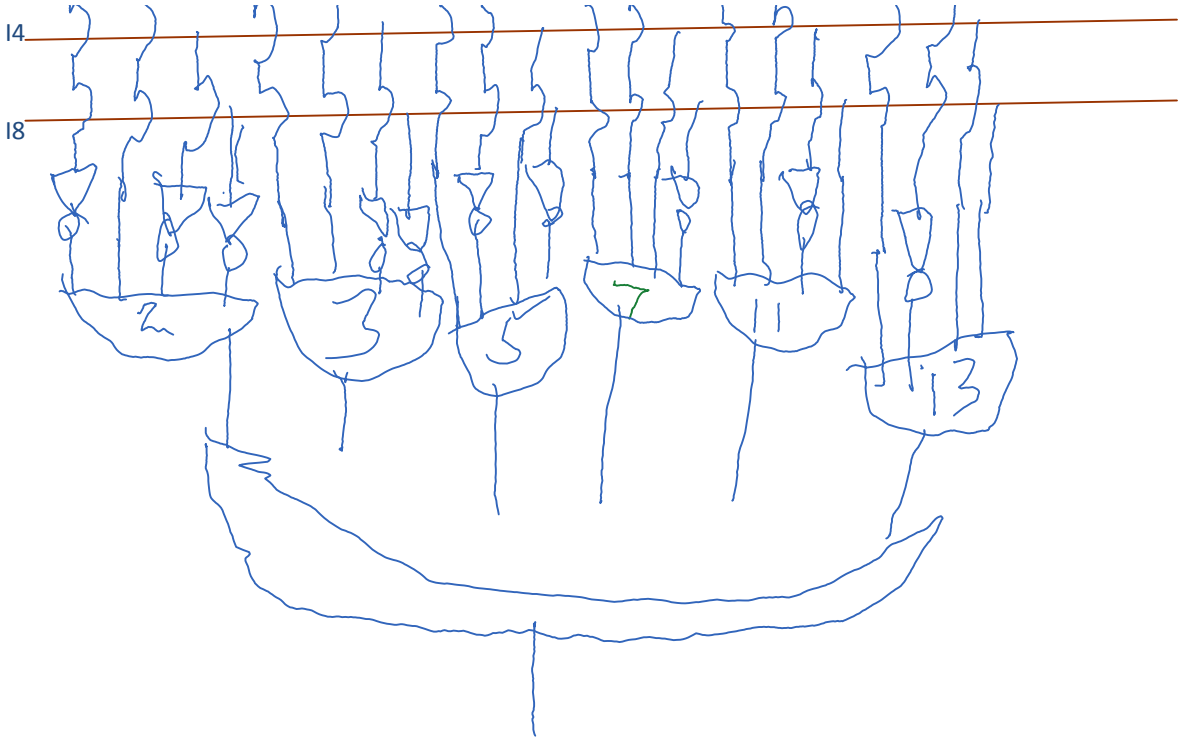
ignore

.001110001110000111000111 …
1.11000111000111000111 … *2^-3

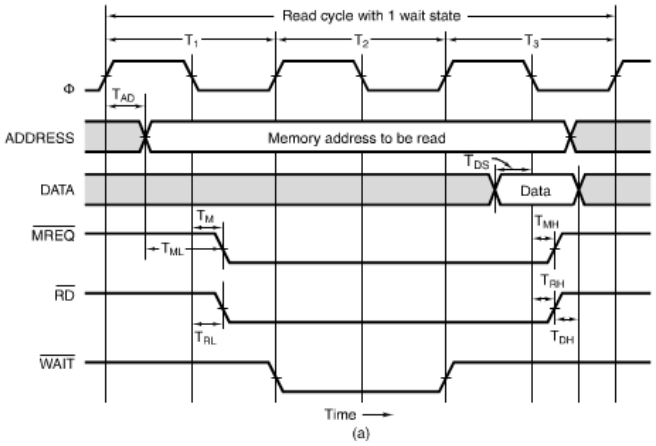
Exponent 127-3=124 =01111100
1 01111100 11000111000111000111000

3. 2 or 3 or 5 or 7 or 11 or 13
   0010
   0011
   0101
   0111
   1011
   1101





Screen clipping taken: 3/4/2013 2:49 PM



Screen clipping taken: 3/4/2013 2:51 PM

Symbol	Parameter	Min	Max	Unit
T_{AD}	Address output delay		4	nsec
T_{ML}	Address stable prior to $\overline{MREQ}$	2		nsec
T_M	$\overline{MREQ}$ delay from falling edge of Φ in T_1		3	nsec
T_{RL}	$\overline{RD}$ delay from falling edge of Φ in T_1		3	nsec
T_{DS}	Data setup time prior to falling edge of Φ	2		nsec
T_{MH}	$\overline{MREQ}$ delay from falling edge of Φ in T_3		3	nsec
T_{RH}	$\overline{RD}$ delay from falling edge of Φ in T_3		3	nsec
T_{DH}	Data hold time from negation of $\overline{RD}$	0		nsec

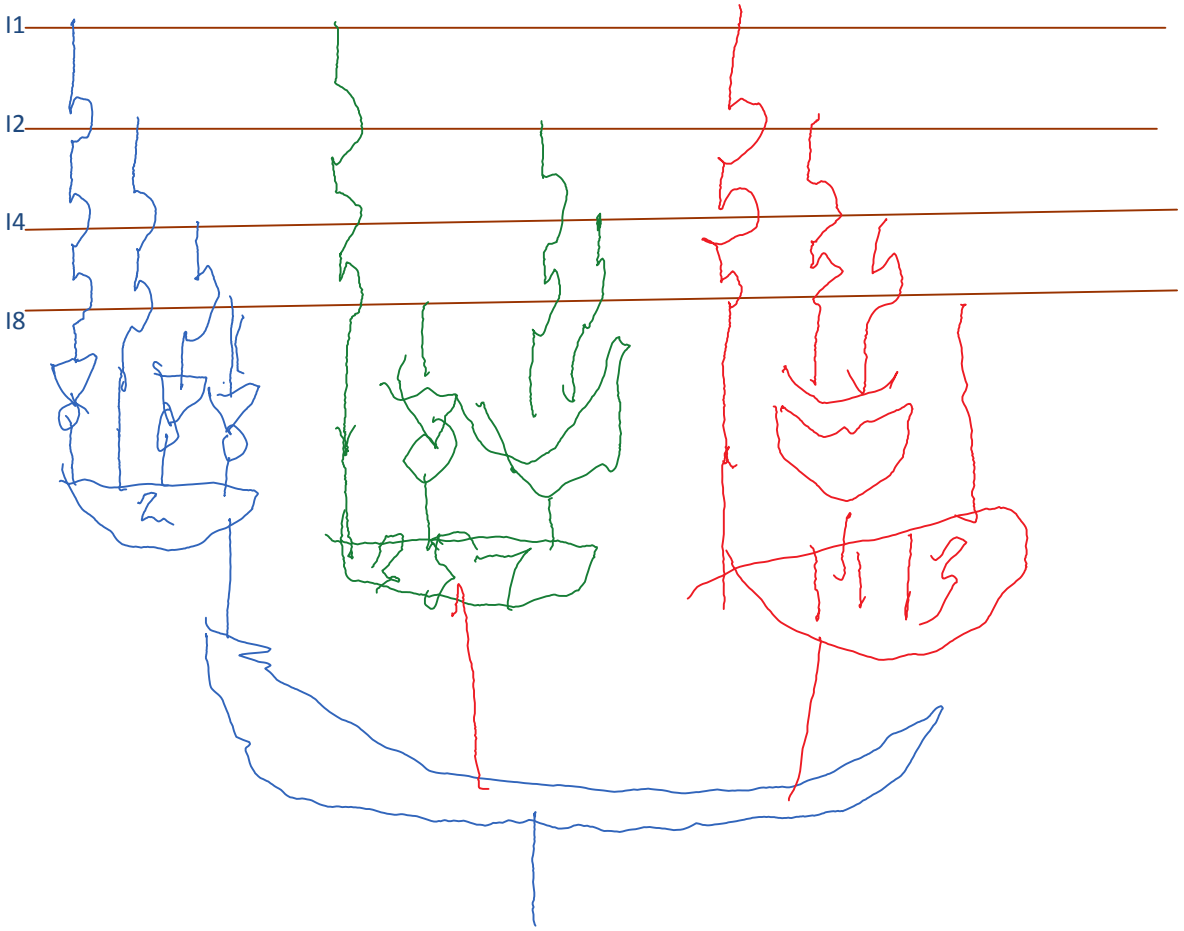
(b)

Tad+tm
And tad +tml
And tad+trl <8 fits into 1 cycle
Tds + tmh+tdh and tds+trh+tdh <8 fit into 1 cycle

Answer 3 cycles

If tdh =4 then tds+tmh+tdh takes 2 cycles and we need 4 cycles

3. 2 or 3 or 5 or 7 or 11 or 13
   0010
   0011
   0101
   0111
   1011
   1101



3/11

Monday, March 11, 2013
1:48 PM

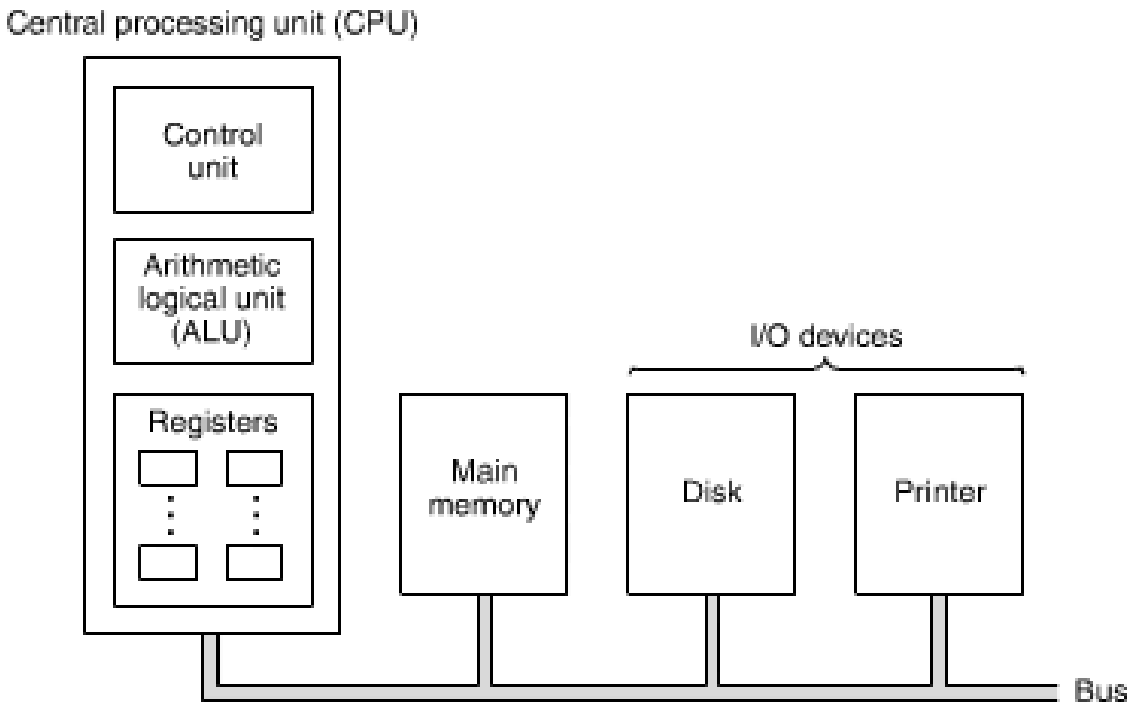


Figure 2-1. The organization of a simple computer with one CPU and two I/O devices.

Report 2 due 3/21
Exam 2 on 4/1
Review on 3/19

TAD =6 has no effect
TDH=6 adds a cycle
4 cycles

Screen clipping taken: 3/11/2013 1:49 PM

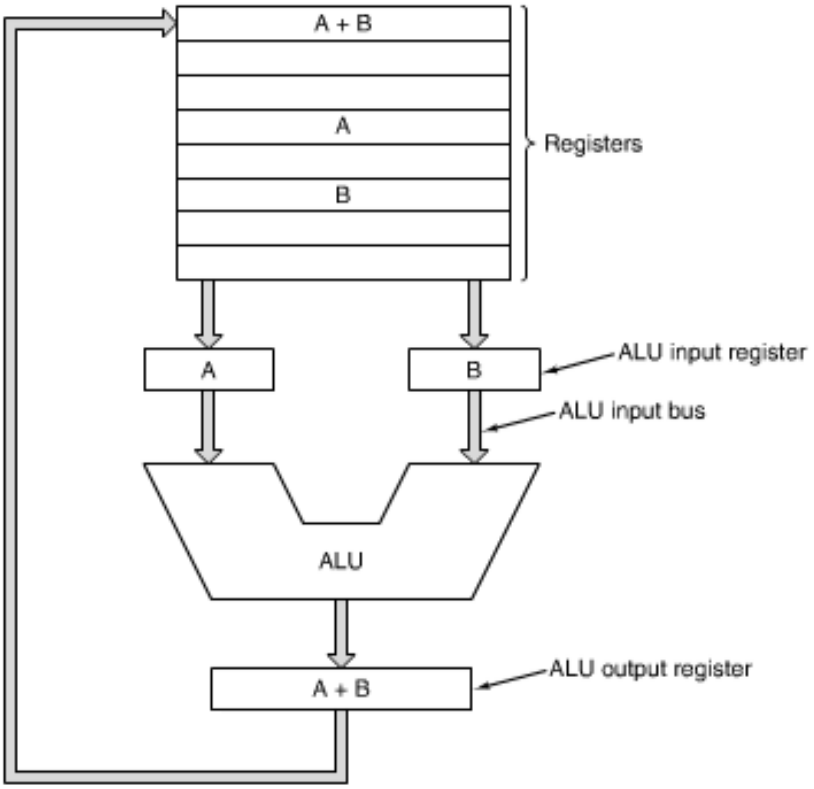


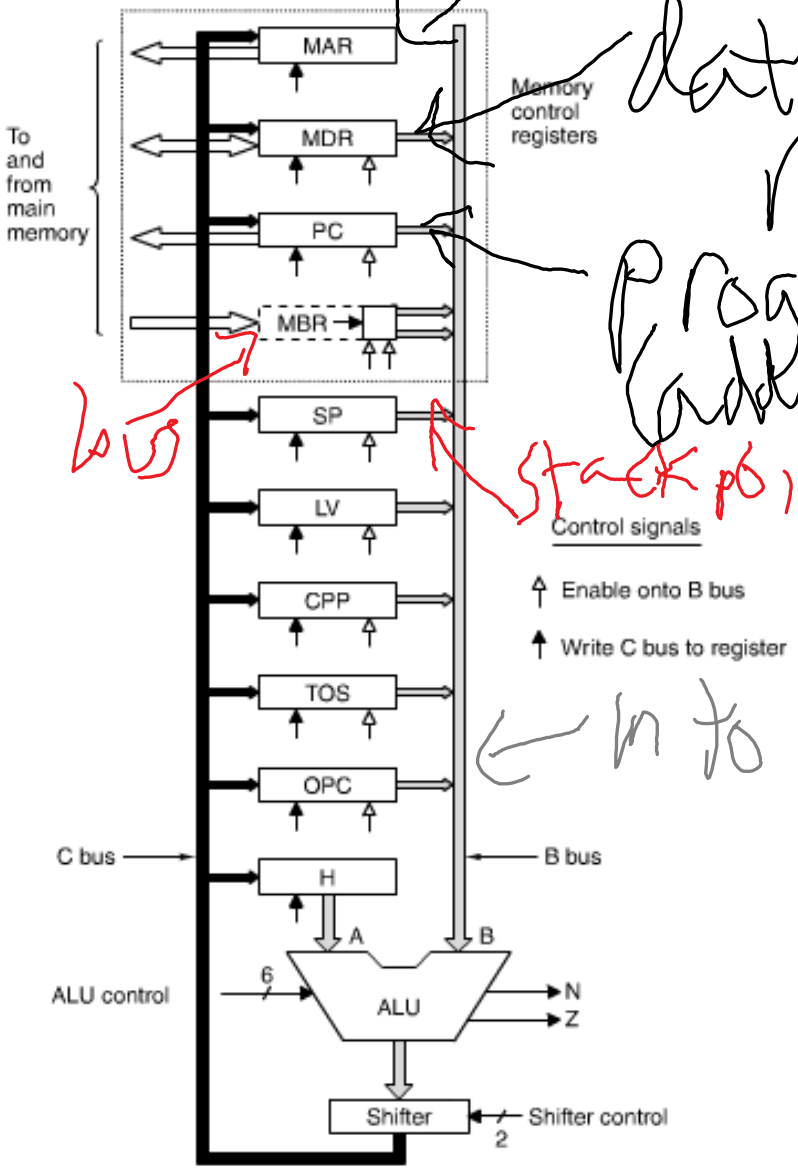
Figure 2-2. The data path of a typical von Neumann machine.

Screen clipping taken: 3/11/2013 1:51 PM

F_0	F_1	ENA	ENB	INVA	INC	Function
0	1	1	0	0	0	A
0	1	0	1	0	0	B
0	1	1	0	1	0	$\overline{A}$
1	0	1	1	0	0	$\overline{B}$
1	1	1	1	0	0	A + B
1	1	1	1	0	1	A + B + 1
1	1	1	0	0	1	A + 1
1	1	0	1	0	1	B + 1
1	1	1	1	1	1	B – A
1	1	0	1	1	0	B – 1
1	1	1	0	1	1	–A
0	0	1	1	0	0	A AND B
0	1	1	1	0	0	A OR B
0	1	0	0	0	0	0
1	1	0	0	0	1	1

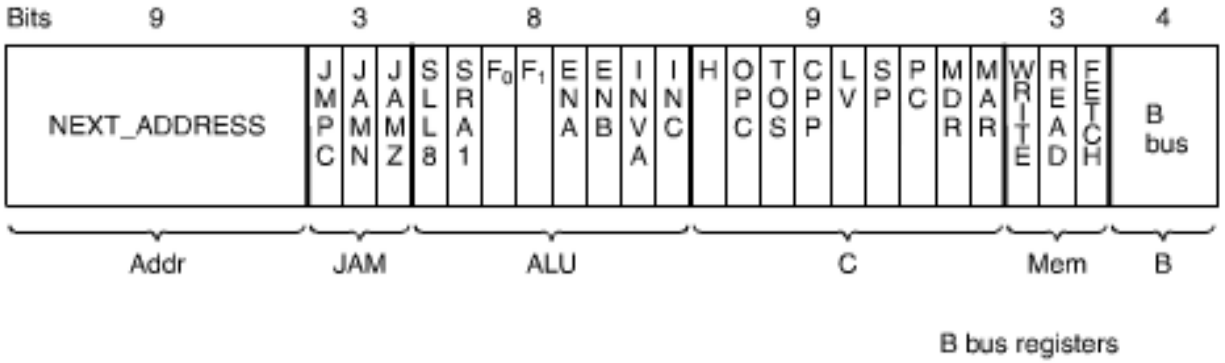
0	1	0	0	0	0	0
1	1	0	0	0	1	1
1	1	0	0	1	0	−1

addresses
data to & from memory
program counter (address of instr in memory)



log
stack pointer
into ALU

Pg 233



Screen clipping taken: 3/11/2013 2:01 PM

Screen clipping taken: 3/11/2013 1:59 PM

Screen clipping taken: 3/11/2013 1:55 PM

10.25

1010.01
1.01001x2^3
0 10000010 01001000000000000000000

HW Read 4.1, read 4.3

3/18

Monday, March 18, 2013
1:47 PM

quiz write the microinstruction that code an instruction that pops the stack as long as the top is negative.

Label	Operations
nop1	goto (MBR)
iadd1	MAR = SP = SP – 1; rd
iadd2	H = TOS
iadd3	MDR = TOS = MDR+H; wr; goto (MBR1)
isub1	MAR = SP = SP – 1; rd

Screen clipping taken: 3/18/2013 2:12 PM

Label	Operations	Comments
ifeq1	MAR = SP = SP – 1; rd	Read in next-to-top word of stack
ifeq2	OPC = TOS	Save TOS in OPC temporarily
ifeq3	TOS = MDR	Put new top of stack in TOS
ifeq4	Z = OPC; if (Z) goto T; else goto F	Branch on Z bit
if_icmpeq1	MAR = SP = SP – 1; rd	Read in next-to-top word of stack

Screen clipping taken: 3/18/2013 2:14 PM

Pasted from <https://twitter.com/search?q=%23shercsc260&src=typd>

popN1 N=TOS;if(N)goto popN4 else goto popN2
popN2 goto MBR instruction done
popN4 MAR=SP=SP-1;rd top of stack is negative
popN5
popN6 N=TOS=MDR; if(N) goto popN4 else goto MBR

3/20

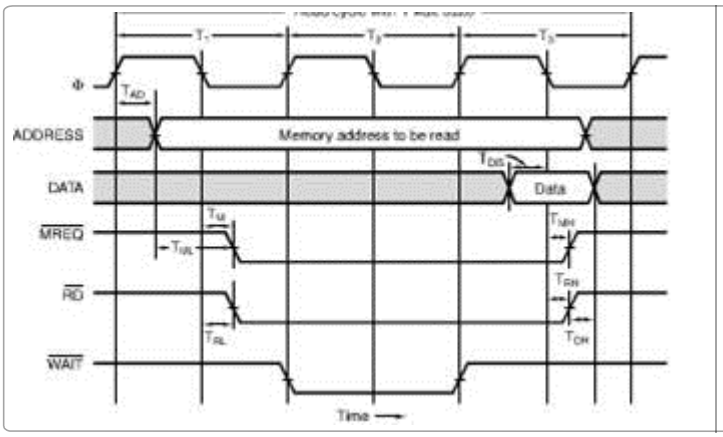
Wednesday, March 20, 2013
1:56 PM

40 pts

1. Write the microcode for a an instruction that jumps to the address if the top of the stack is the sum of the next two numbers on the stack.
   MAR=SP=SP-1; rd gets 2nd item on stack and pops
   H=TOS put original top of stack where we can work with it
   H=H-MDR subtract first number from original top
   MAR=SP-1;rd gets 3rd item on stack, does not pop
   TOS=MDR put popped value into top of stack
   Z=H-MDR;if(Z)goto T else goto F conditionally jump

20 pts

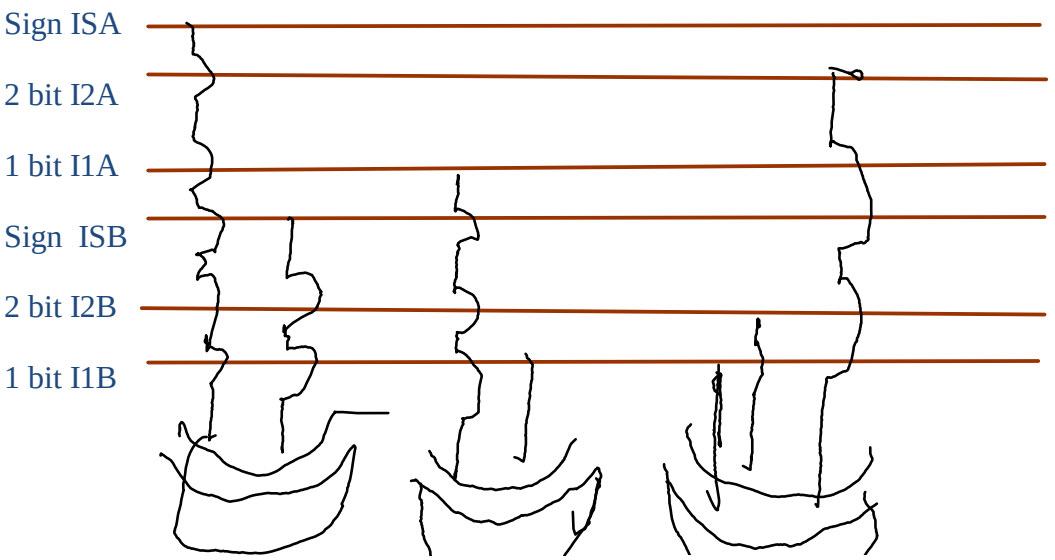
2. T_{DS}=6ns and T_{ML}=4ns. How many cycles would it take to do a read given the timings from page 181? Show why.
   $T_{AD}+T_{ML}$=8ns which fits in 1 cycle
   T_{DS} does not fit in the first 1/2 cycle so we need an extra cycle
   So 4 cycles for memory

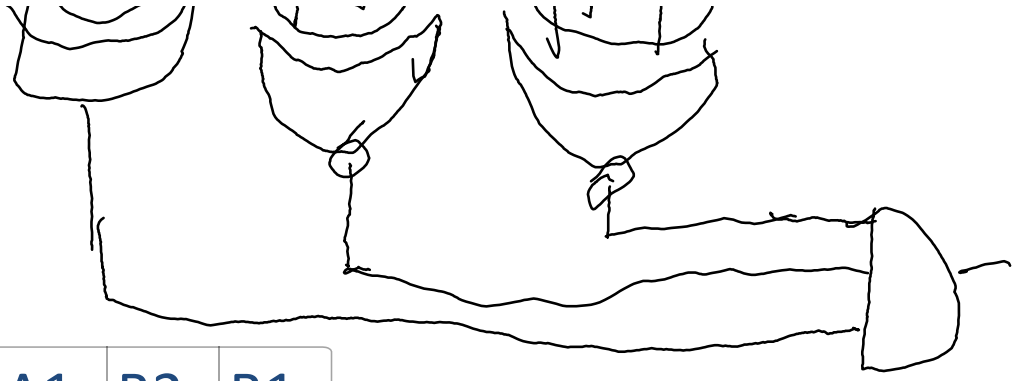


Symbol	Parameter	Min	Max	Unit
T_{AD}	Address output delay		4	nsec
T_{ML}	Address stable prior to MREQ	2		nsec
T_{MR}	MREQ delay from falling edge of Φ in T_1		3	nsec
T_{RL}	RD delay from falling edge of Φ in T_1		3	nsec
T_{DS}	Data setup time prior to falling edge of Φ	2		nsec
T_{MH}	MREQ delay from falling edge of Φ in T_3		3	nsec
T_{RH}	RD delay from falling edge of Φ in T_3		3	nsec
T_{DH}	Data hold time from negation of RD	0		nsec

40 pts

3. Design a circuit that outputs 1 if one of the two three bit inputs is the negation (in two's complement) of the other 0 otherwise.





A2	A1	B2	B1
0	0	0	0
0	1	1	1
1	0	1	0
1	1	0	1

Micro instructions for jump if top is divisible by 8

D81 Z=TOS AND 1; if(Z)goto D82 else goto F

D82 OPC=TOS>>1

D83 Z=OPC aND 1; if(Z)goto D84 else goto F

D84 OPC=TOS>>1

D85 Z=OPC AND 1; if(Z)goto T else goto F

H=1

H=H+1

H=H+1

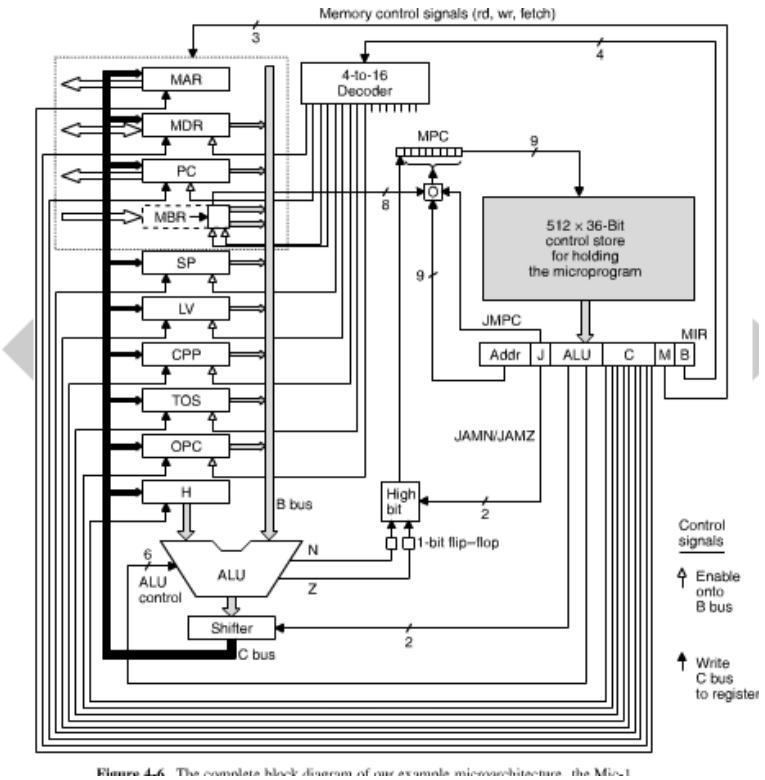
H=h+1

h=H+1
H=H+1
H=H+1
z=TOS AND H; if(Z)goto T else goto F

H=1
OPC=H=H+1
H=OPC+h+1
H=H+OPC
z=TOS AND H; if(Z)goto T else goto F

3/13

Wednesday, March 13, 2013
2:05 PM



Screen clipping taken: 3/13/2013 2:05 PM

Instruction ifse (if the top and next of the stack are equal) pops both from stack

Ifse1 mar=sp=sp-1;rd pop stack read new top

Ifse2 h=tos put old top into h

Ifse3 mar=sp=sp-1;rd pop stack read new top

Ifse4 opc=mdr while reading put old top in opc

Ifse4 opc-h;if(z) goto t else goto f

nop1	goto Main1	Do nothing
iadd1	MAR = SP = SP – 1; rd	Read in next-to-top word on stack
iadd2	H = TOS	H = top of stack
iadd3	MDR = TOS = MDR + H; wr; goto Main1	Add top two words; write to top of stack
isub1	MAR = SP = SP – 1; rd	Read in next-to-top word on stack

Screen clipping taken: 3/13/2013 2:49 PM

Iavg replaces the top two items on the stack with their average
Iavg1 mar=sp=sp-1;rd
Iavg2 h=tos
Iavg3 mdr=tos=(mdr+h)>>1;wr; goto Main1

Im3 multiplies top of stack by 3
h=tos
Tos=tos<<8 left shifted 8
Tos=tos>>1 left shifted 7
Tos=tos>>1 left shifted 6
Tos=tos>>1 left shifted 5
Tos=tos>>1 left shifted 4
Tos=tos>>1 left shifted 3
Tos=tos>>1 left shifted 2
Tos=tos>>1 left shifted 1
Mdr=tos=tos+h; wr; goto Main1

Ifodd jump if top of stack is odd
Mar=sp=sp-1;rd
h=tos
Tos=mdr popped
If(h AND 1)goto T; else goto F

Hw write the microcode for
Idiv4 jump if number is divisible by 4

4/3

Wednesday, April 03, 2013
1:40 PM

There are 10 kinds of people

Those who understand binary
Those who do not

Seen in episode of NCIS

Microcode question
If c exam
tos=tos xor -1

h=0 h will have answer
C1loop z=tos; if z then c1done else c1check
C1check z=tos and 1; if z then c1shift else c1count
C1count h=h+1
C1shift tos=tos>>1; goto c1loop
C1done mdr=tos=h
C1done2 mar=sp;wr; goto main1

Monday, April 08, 2013
1:50 PM

```
; --- Copyright Jonathan Meyer 1996. All rights reserved. ----------------
; File:      jasmin/examples/Count.j
; Author:    Jonathan Meyer, 10 July 1996
; Purpose:   Counts from 0 to 9, printing out the value
; ------------------------------------------------------------------------

.class public examples/Count
.super java/lang/Object

;
; standard initializer
.method public <init>()V
  aload_0
  invokenonvirtual java/lang/Object/<init>()V
  return
.end method

.method public static main([Ljava/lang/String;)V
      ; set limits used by this method
      .limit locals 4
      .limit stack 3

      ; setup local variables:

      ;    1 - the PrintStream object held in java.lang.System.out
      getstatic java/lang/System/out Ljava/io/PrintStream;
      astore_1

      ;    2 - the integer 10 - the counter used in the loop
```

```
    bipush 10
    istore_2

    ; now loop 10 times printing out a number

  Loop:

    ; compute 10 - <local variable 2> ...
    bipush 10
    iload_2
    isub
    invokestatic java/lang/String/valueOf(I)Ljava/lang/String;
    astore_3
    ; ... and print it
    aload_1    ; push the PrintStream object
    aload_3    ; push the string we just created - then ...
    invokevirtual java/io/PrintStream/println(Ljava/lang/String;)V

    ; decrement the counter and loop
    iload_2
     bipush 1
     isub
     istore_2
    iload_2
    ifne Loop

    ; done
    return

.end method
```

Count2.j:

```
; --- Copyright Jonathan Meyer 1996. All rights reserved. ----------------
; File:      jasmin/examples/Count.j
; Author:    Jonathan Meyer, 10 July 1996
; Purpose:   Counts from 0 to 9, printing out the value
; ------------------------------------------------------------------

.class public examples/Count
.super java/lang/Object

;
; standard initializer
.method public <init>()V
  aload_0
  invokenonvirtual java/lang/Object/<init>()V
  return
.end method

.method public static main([Ljava/lang/String;)V
      ; set limits used by this method
      .limit locals 10
      .limit stack 8

      ; setup local variables:

      ;    1 - the PrintStream object held in java.lang.System.out
      getstatic java/lang/System/out Ljava/io/PrintStream;
      astore 1

      ;    4 - the InputStream object held in java.lang.System.in
      getstatic java/lang/System/in Ljava/io/InputStream;
      astore 4
```

```
 aload 1   ; load up system.out
 ldc "Enter a number to count up to: " ; prompt for user
invokevirtual java/io/PrintStream/print(Ljava/lang/String;)V
 aload 4 ; load up system.in
 bipush 127
 newarray byte   ; array of 127 chars
 astore 6
 aload 6  ; array is stored in position 6 and also on stack
invokevirtual java/io/InputStream/read([B)I    ; reads into array
 istore 5  ; put the number of bytes in 5
 new java/lang/String  ; string to put the result in
 astore 7
 aload 7
 aload 6  ;  array of bytes to string
 iconst_0 ; offset 0
 iload 5 ; number of bytes
 iconst_2  ; subtract 2
 isub      ; remove line feed at end
 invokespecial java/lang/String/<init>([BII)V  ; create string from bytes
 aload_1   ; load up system.out
 aload 7
invokevirtual java/io/PrintStream/print(Ljava/lang/String;)V
 aload_1
 ldc  " is the number you entered"
invokevirtual java/io/PrintStream/println(Ljava/lang/String;)V

 aload 7
 invokestatic  java/lang/Integer/parseInt(Ljava/lang/String;)I
 istore_2
iload_2
istore 5
```

```
    ; now loop number from user times printing out a number

   Loop:

    ; compute (number from user - <local variable 2> ...
    iload 5
    iload_2
    isub
    invokestatic java/lang/String/valueOf(I)Ljava/lang/String;
    astore_3
    ; ... and print it
    aload_1    ; push the PrintStream object
    aload_3    ; push the string we just created - then ...
    invokevirtual java/io/PrintStream/println(Ljava/lang/String;)V

    ; decrement the counter and loop
    iload_2
     bipush 1
     isub
     istore_2
    iload_2
    ifne Loop



    ; done
    return

.end method
Console.j:
; Copyright David B. Sher 2013
; Input and output routines to console
; Don't have to repeat the laborious stuff.
```

```
.class public csc260/Console
.super java/lang/Object
;
; standard initializer
.method public <init>()V
  aload_0
  invokenonvirtual java/lang/Object/<init>()V
  return
.end method

; prints string to console
.method public static output(Ljava/lang/String;)V
.limit stack 3
.limit locals 2
        getstatic java/lang/System/out Ljava/io/PrintStream;
        aload 0 ; string to print
      invokevirtual java/io/PrintStream/print(Ljava/lang/String;)V ; print it
      ; done
      return

.end method
.method public static main([Ljava/lang/String;)V
      ; set limits used by this method
      .limit locals 10
      .limit stack 8
      ; done

       ldc "This should print"
       invokestatic csc260/Console/output(Ljava/lang/String;)V
      return
```

.end method

4/10

Wednesday, April 10, 2013
2:00 PM

Quiz if system.out is in variable 1 and x is in variable 2 write x-1,x,x+1

```
; make x-1
Iload 2
Bipush 1
iSub
    invokestatic java/lang/String/valueOf(I)Ljava/lang/String;
    astore_3
    ; ... and print it
    aload_1    ; push the PrintStream object
    aload_3    ; push the string we just created - then ...
    invokevirtual java/io/PrintStream/println(Ljava/lang/String;)V
; push x
Iload 2
    invokestatic java/lang/String/valueOf(I)Ljava/lang/String;
    astore_3
    ; ... and print it
    aload_1    ; push the PrintStream object
    aload_3    ; push the string we just created - then ...
    invokevirtual java/io/PrintStream/println(Ljava/lang/String;)V
; push x+1
Iload 2
Bipush 1
iAdd
    invokestatic java/lang/String/valueOf(I)Ljava/lang/String;
```

```
astore_3
; ... and print it
aload_1    ; push the PrintStream object
aload_3    ; push the string we just created - then ...
invokevirtual java/io/PrintStream/println(Ljava/lang/String;)V
```

4/15

Monday, April 15, 2013
2:09 PM

Cache

Cache keeps location,value pairs
When new added and full oldest lost

When reading
If location in cache quickly puts value on bus, without memory
If location is not in cache, goes to cpu and cache

When writing
Goes to memory and cache but quickly available from cache.

If reading and writing same locations a lot makes memory seem faster.
Cheaper than making all memory faster.

Essay 3 due 5/2

4/22

Monday, April 22, 2013
1:48 PM

quiz write a jasmin method that reads an array of floats from the user and returns it. It should take an array size parameter.

Pasted from <https://twitter.com/search?q=%23shercsc260&src=typd>

Average.j

```
; Copyright David B. Sher 2013
; Quiz contains method to average two numbers and tests method
.class public csc260/Average
.super java/lang/Object

;
; standard initializer
.method public <init>()V
   aload_0
   invokenonvirtual java/lang/Object/<init>()V
   return
.end method


; answer to quiz -  computes the average of its parameters
.method public static average(FF)F
.limit stack 2
.limit locals 2
    fload 0 ; load first parameter onto stack
    fload 1 ; load second parameter onto stack
    fadd    ; sum them
    ldc 2.0
    fdiv    ; divide by 2
```

```
    freturn
.end method ; average

; averages an array of floats
.method public static average([F)F
.limit stack 3
.limit locals 2
    bipush 0
    istore 1 ; put array iterator into location 1
    ldc 0.0    ; stack will hold sum
    ; sum the array
sumLoop:
   aload 0 ; get array length
    arraylength
    iload_1 ; get index
    isub  ;length-index
    ifle endSum  ; if length <= index leave loop
        aload_0
        iload_1  ; get array[index]
        faload
        fadd     ; add it to the sum
        iconst_1 ; increase index
        iload_1
        iadd
        istore_1
    goto sumLoop    ; loops
endSum:
    ; the sum of the array is on the stack
    aload_0  ;get array length
    arraylength
    i2f   ; translate to float
    fdiv  ; divide sum by length
```

```
    freturn
.end method


; tests the  average method
.method public static main([Ljava/lang/String;)V
.limit stack 3
.limit locals 3
    ; read first number
    ldc "Enter a real number: "
    invokestatic csc260/Console/output(Ljava/lang/String;)V
    invokestatic csc260/Console/getFloat()F ; first real is on stack
    ; read second number
    ldc "Enter another real number: "
    invokestatic csc260/Console/output(Ljava/lang/String;)V
    invokestatic csc260/Console/getFloat()F ; second real is on stack

    ; call average method
    invokestatic csc260/Average/average(FF)F


    ; output the average
    invokestatic csc260/Console/output(F)V
    ldc " is the average of your numbers"
    invokestatic csc260/Console/output(Ljava/lang/String;)V
    invokestatic csc260/Console/outputNewLine()V

    return
.end method ; main
```

AverageArray.java

```
/**
```

```
    Program that gets an array of floats and outputs average
    Copyright David B. Sher 2013
*/
package csc260;

public class AverageArray
{
    /** holds the floats we are passing */
    private static float[] array = new float[30];
    private static int maxArraySize = 30;
    private static int arraySize =0;

    public static void main(String[] args)
    {
        // get numbers from user
        boolean done=false;
        do
        {
            Console.output("Enter a float number: ");
            array[arraySize++] = Console.getFloat();
            if(arraySize==maxArraySize)
            { // double storage for array
                maxArraySize <<=1;
                float[] tempArray= new float[maxArraySize];
                for(int index=0;index<arraySize;index++)
                    tempArray[index]=array[index];
                array=tempArray;
            }// end if
            Console.output("Another number? ");
            String answer = Console.getString();
            switch(answer.charAt(0))
            {
```

```
                case 'n':
                case 'N':
                    done = true; // number entry is done
                    // copy array to array of the right size
                    float[] tempArray= new float[arraySize];
                    for(int index=0;index<arraySize;index++)
                        tempArray[index]=array[index];
                    array=tempArray;
                    break; // if string starts with n stop reading numbers
                }
            } while(!done);
            Console.output("Averaging...");
            Console.outputNewLine();
            Console.output("The average of your numbers is " +
            Average.average(array));
            Console.outputNewLine();
        } // end main
}
ReadFloats.j
; contains a static method to read an array of floats from the user
; Copyright David B. Sher 2013
.class public csc260/ReadFloats
.super java/lang/Object

;
; standard initializer
.method public <init>()V
  aload_0
  invokenonvirtual java/lang/Object/<init>()V
  return
.end method
```

```
; answer to quiz goes here
.method public static readFarray(I)[F
.limit stack 4
.limit locals 3
    iload_0 ; get the size of the array
    newarray float
    astore 1 ; array lives in location 1
    iconst_0  ; put 0 into index in location 2
    istore_2
    ; loop reading floats into array
readLoop:
    iload_2
    iload_0  ; break if index == array size
    isub
    ifeq endLoop
    aload_1  ; put the array on the stack
    ldc "Enter array["
    invokestatic csc260/Console/output(Ljava/lang/String;)V
    iload_2
    dup    ; 2 copies of the index on the stack
    invokestatic csc260/Console/output(I)V
    ldc "] float: "
    invokestatic csc260/Console/output(Ljava/lang/String;)V
    invokestatic csc260/Console/getFloat()F
    fastore ; put number from user into array at index
    iload_2 ; increment index
    iconst_1
    iadd
    istore_2
    goto readLoop
endLoop:
```

```
    aload_1  ; return array
    areturn
.end method ; readFarray

; tests the  readFloatArray method
.method public static main([Ljava/lang/String;)V
.limit stack 3
.limit locals 3
    ; get the size of the array
    ldc "How many floats are in the array? "
    invokestatic csc260/Console/output(Ljava/lang/String;)V
    invokestatic csc260/Console/getInteger()I
    ; get the array
    invokestatic csc260/ReadFloats/readFarray(I)[F
    ; get the average of the numbers
    invokestatic csc260/Average/average([F)F
    invokestatic csc260/Console/output(F)V
    ldc " is the average of the numbers in the array."
    invokestatic csc260/Console/output(Ljava/lang/String;)V
    invokestatic csc260/Console/outputNewLine()V
    return
.end method ; main
```

4/24

Wednesday, April 24, 2013
1:59 PM

Proj 3 is due 5/10

quiz write a method that takes two integer array parameters and determines if they have the same contents (1 if same 0 ow)

Pasted from <https://twitter.com/search?q=%23shercsc260&src=typd>

Method does not read from or write to user

```
; does the quiz
.method public static arrayEqual([I[I)I
.limit stack 4
.limit locals 3    ; 0 first array  1 second array 2 index
    aload 0
    arraylength
    aload 1
    Arraylength
    sub
    Ifne  falseReturn
    Aconst_0 ; initialize index
    Iload 2
compareLoop:
    Aload_0
    Dup ; 2 copies of array 1 for length and 1 for accessing
    Arraylength
    Iload 2
    Sub
    Ifle done  ; if array completely searched then done
    Iload 2
```

```
    Iaload   ; access first array[index]
    Aload_1 ; 2nd array
    Iload 2
    Iaload ; access 2nd array[index]
    Isub
    Ifne falseReturn  ;  found difference
    Iconst_1 ; increment index
    Iload_2
    Iadd
    Istore_2
    Goto compareLoop ; loop


done:
    Iconst_1 ; return true
    Ireturn
falseReturn:
    Iconst_0
    Ireturn  ; returns false
.end method
```

Storage device vary according to a lot of dimensions
Most important

Device	Size	Permanence	Read speed	Write speed
Stone tablets	Tiny <100bytes	Huge >100000 years	Slow	Very slow
Clay	Tiny <100bytes	Huge >100000	Slow	Slow

tablets		years if baked		
Cd or dvd	Medium .5-5gb	Medium about 5 years if used	Medium	Slow to medium
Floppy disks	Medium small <2mb	Medium < 1 year	Medium	Medium
Hard disk	Large >1 tb	Medium < 10 years	High	High
Memory	Medium large < 100gb	Low < 1 week	Very high	Very high
ROM	Medium small	High years	Very High	Can't write
Registers	Tiny<100 bytes	Very low < 1 sec	Extremely high	Extremely high

HW read chapter 6

4/29

Monday, April 29, 2013
2:53 PM

Operating systems
Move data around memory devices
Control processes
Process is a running program
I/O

5/1

Wednesday, May 01, 2013
2:06 PM

Cache combine a small fast memory with a large slow memory to get the best of both

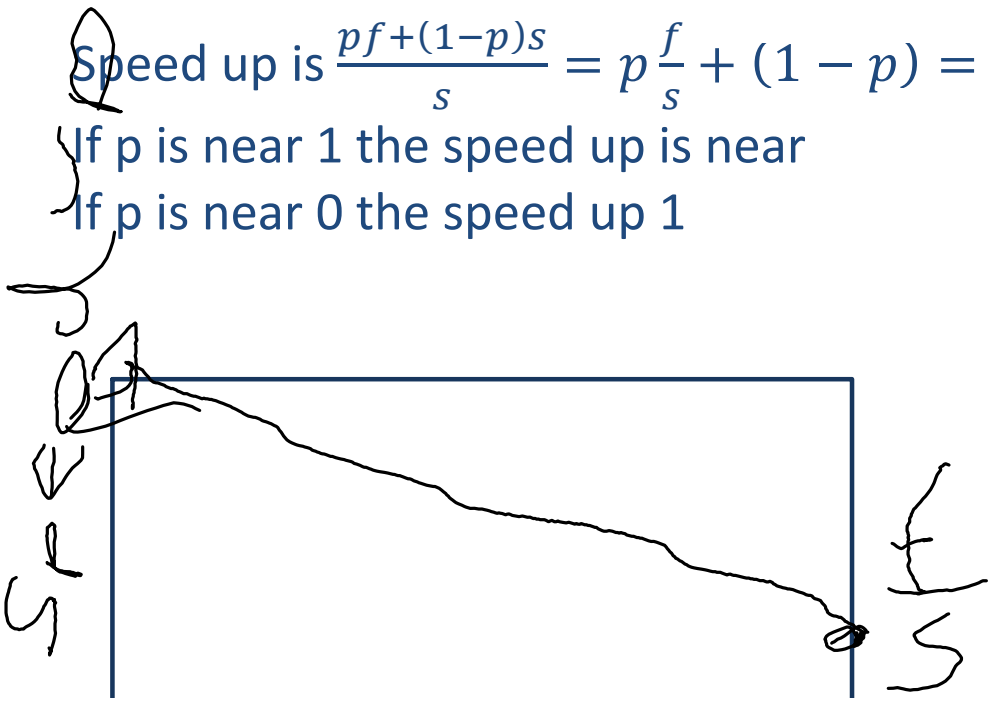
keep

Lets say cache is of size c memory is size m.
Cache access speed is f
Memory access speed is s

The probability that a memory access will be in the cache is p >= c/m.

Speed without a cache is s
Speed with the cache is pf+(1-p)s

Speed up is $\frac{pf+(1-p)s}{s} = p\frac{f}{s} + (1-p) = 1 - p\left(1 - \frac{f}{s}\right)$
If p is near 1 the speed up is near
If p is near 0 the speed up 1



A p of 1 means you never use the memory outside of cache
The limit on cache size is cash
Another limit is heat

Most of the complexity of caches and virtual memory involves making p as large as possible.

Take advantage of time and space locality.
In a short period of time most of the memory accesses will be the same or close.

The program counter ussually increases by 1 each time.
Exception is method call.

Predictive caching involves putting nearby elements in the cache.

Virtual memory
Use the memory as a cache for the disk.
Don't have to fit all the program into the memory.
Or all the data.
Programmers don't have to write different versions for different sized memories.

File on disk is broken into pages.
Disk is read 1 page at a time

5/6

Monday, May 06, 2013
2:02 PM

quiz If the memory is 1gb and the cache is 10mb the memory takes 100ns cache takes 20ns and the probability is 95% speedup? work

Pasted from <https://twitter.com/search?q=%23shercsc260&src=typd>

$$p\frac{f}{s} + (1-p)$$

.95*20/100+(1-.95)=0.24

$$\frac{pf + (1-p)s}{s}$$

(.95*20+(1-.95)*100)/100 = 0.24

Paging
Instead of reading individual bytes copy a page at a time

Pages generally range from .25kb to 8kb
When a position in disk is accessed the whole page is placed in memory (memory mapped)
Uneven access speed (is data all in one page (fast) or split between 2 (slower))

Virtual memory - most of the program is on the disk, pages read into memory as needed.

Only so much room in memory (otherwise no disk aka solid state disk).

If room in memory put new pages in.

If no room need to remove a page for each new one.

2 popular schemes: FIFO (queue) LRU (priority queue)
FIFO means First In First Out. LRU - Least recently used - page that has gone longest without access.

Operating system keeps track of pairs (virtual page, physical page)
Physical is where memory actually lives
Virtual is where its address is.

Lets say there are room for 3 pages in memory

Access	FIFO	LRU
1	1	1
2	12	12
1	12	21
3	123	213
4	234	134
2	234	342
5	345	425
4	345	254
2	452	542

Page faults

Pages in physical memory are called working set. A page fault occurs when we access a page not in the working set.
Lots of page faults is thrashing.
Working set size 4

Access	FIFO	LRU
1	1	1

2	12	12
1	12	21
3	123	213
4	1234	2134
2	1234	2134
5	2345	1345
4	2345	1345
2	2345	3452

Review Csc260 Exam 3

Monday, May 06, 2013
3:02 PM

Exam 3/13
Review 3/8
35 pts

1. Write the microcode for an instruction that copies everything in the stack to an array starting at the location on top of the stack. The bottom of the stack will be -1.

```
Cstack1  h=tos
Cstack2  MAR=SP=SP-1; rd        //pop stack
         ;                      // wait for new stack element
         Z=MDR+1; if(Z) goto cstackdone   // if at bottom done
         MAR=H; wr              // write next stack element to memory
         H=H+1                  // next item in array
         Goto cstack2           // loop
Cstackdone MAR=SP=SP-1;rd       // pop -1 off stack
         ;
         TOS=MDR; goto main     // put new top of stack
```

30 pts

2. Given a 3 page memory show how many pages had to be read from memory with this sequence of page references under FIFO and LRU, also what pages are in memory at the end: 1,2,3,1,4,1,3,2

Request	FIFO	LRU
1	1	1
2	12	12
3	123	123
1	123	231
4	234	314
1	341	341
3	341	413

Fifo has 6 page faults and ends with 412
LRU has 5 page faults and ends with 132

2	412	132

35 pts

3. Write a JVM method, sequentialArray, that creates and returns an array of ints that start with its first parameter and ends with its last parameter. For example sequentialArray(4,7) returns an array with 4,5,6,7 in it.

```
; defining int[] sequentialArray(int first,int last)
.method public static sequentialArray(II)[I
.limit locals 4
.limit stack 3
; find the size of the array
Iload 1 ; load last
Iload 0 ; load
Isub
Iconst_1
Iadd     ; last-first+1
Newarray int   ; array to return
Astore 2  ; local 2 will be the array to return
Iconst_0
Istore 3     ; local 3 will be the array index
InsertLoop:  aload 2
Iload 3
Iload 0
Iastore  ;put first into array
Iload 0 ;
Iload 1
Isub
Ifeq done ; if first=last then done with array
Iload 0;
Iconst_1
Iadd
Istore 0    ; increment first
Iload 3
Iconst_1
```

```
Iadd
Istore 3   ; increment index
Goto insertLoop
Done: aload 2
Areturn  ; return array
.end method
```