

CS 2461

Lab- Week 6

# Announcements

## ■ Project 2

- due Monday!!!!
- don't wait... it's harder than it seems!!!!!!
- btw... this is a partner project but each person is expected to do their part and outline what you did in the report (so don't let your partner take over the whole thing)



## ■ Exam 1

- Tuesday
- Scary!!!!!!!!!!!!!!
- You'll be fine but....
- study!!!!!!!!!!!!!!
  - quizzes
  - homeworks
  - jeopardy



# Today....

- LC3 machine language programs
- Review & Project 2 questions

# LC3 Machine Code

Address	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x3000	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	1
x3001	0	0	0	1	0	1	1	0	1	0	1	1	1	0	0	0
x3002	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0
x3003	0	0	1	1	1	0	0	0	0	0	0	0	0	0	1	1
x3004	0	0	0	0	1	1	1	1	1	1	1	1	1	0	1	1
x3005	0	1	0	1	0	1	0	0	1	0	1	0	0	0	1	0
x3006	0	0	0	0	0	1	0	1	1	1	1	1	1	1	0	1
X3007	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1. Starting with first instruction at x3000: determine opcode
2. Determine operands – find address of operands
3. Execute the instruction to determine outcome
4. Go to next instruction

# Assembly Language Equivalent

Start LD R2, #3

ADD R3, R2, # -8

BRnz Goto

ST R4, #3

BRnzp Start

Goto AND R2, R2, #2

BRz # Start

# C bitwise operators: code from exercise sept27.c

- IF statement did not act as expected due to
  - casting of unsigned and signed integers
  - Logical AND: if integer operand is not zero then it treats it as True
- CallMeFirst: returns the XOR of operands x,y
- CallMeNext: returns  $2^x + 1$ 
  - $(1 \ll x)$  is  $1 * 2^x = 2^x$
- CallMeLast: returns  $(x-y)$ 
  - $\text{temp} = \sim y + 1$  is the 2's complement of y (i.e., it negates y)
- whoaml (x,n): returns the n-th byte of x
  - 0xFF is a 8-bit mask containing all 1's
  - $y=(n \ll 3)$  is  $n * 2^3 = 8n$ . So  $y=0$  (if  $n=0$ ) or  $y=8$  (if  $n=1$ ) or  $y=16$  ( $n=2$ ) or  $y=24$  ( $n=3$ )
  - 0xFF is shifted y times to the left...i.e., it is shifted to the n-th byte to get a mask that is all 0's except for all 1's in the n-th byte
  - This is then and-ed with x – therefore all bits except n-th byte are zero-ed out
  - Finally, this n-th byte is shifted to the rightmost byte (byte 0) and returned
- whataml: returns 1 if A is not zero else returns 0.
  - Observe the MSB of the XOR of A and  $-A$  is a 1 only when A is not zero.
  - This MSB is shifted right 31 (to LSB) and then Anded with 0x1