CS241 Lecture 1 - Foundations of Sequential Programs

Graham Cooper

August 6, 2015

Intro

Prof: Brad Lushman
Room: DC 3110
ISA: Akshaya Senthi cs241@uwaterloo.ca
Web: https://www.student.cs.uwaterloo.ca/c̃s241

What is a "Sequential Program"

- "Ordinary" (Not concurrent or parallel)
- "Single-threaded" (Only one thing going on at a time)

Foundations

- understand how seq.programs "work"
- from the groud up

Starting Point:

- bare hardware (sort of)
- for CS241 (simulated MIPS machine)
- (only interprets 1s & 0s)

At the end we will get programs written in a C-like language to run on MIPS

Binary and Hexidecimal Numbers

Bits

- 0 or 1, high or low voltages
- binary digit
- configurations of magnetic media (CD etc)

- all the computer understands
- group bits together

Byte: 8 bits eg 11001001, $2^8 = 256$ possible bytes

Word:

- machine specific grouping of bytes
- we will assume a 32-bit computer architecture
- 1 word = 32 bits = 4 bytes

4 bits (1/2 a btye is called a nibble)

Q: Given a byte, (or a word) in the computer's memory, what does it mean? ie what does 11001001 mean?

A: It could mean many things.

1) A number, What number?

- Binary number system
- $1 * 2^7 + 1 * 2^6 + 0 * 2^5 + 0 * 2^4 + 1 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0 = 201$

How can we represent negative numbers?

Simple Way

- Reserve the first bit to represent the sign
- 0 for positive 1 for negative
- sign magnitude representation
- now 11001001 = -(64 + 8 + 1) = -73
- two representations for 0: 00000000 10000000
- waste (we don't need two zeros)
- arithmetic is tricky (eg. add a positive and negative number)

Better Way:

2's Compliment Representation

1. Interpret the n-bit number as an unsigned integer
2. If the first bit is 0, done!
3. else, subtract 2^n

eg: n = 3

	000	001	010	011	100	101	110	111
1)	0	1	2	3	4	5	6	7
2)					-8	-8	-8	-8
3)					-4	-3	-2	-1

As a number line:

000	001	010	011	100	101	110	111

- Therefore: n bits represent $-2^(n-1)$ ... $2^(n-1)-1$
- only 1 zero
- left bit gives sign
- arithmetic is cleaner
- eg. same addition circuitry works for both unsigned and 2's compliment
- Int 2's comp $11001001 = 201 - 2^8 = 201 - 256 = -55$

Convenience: Hexadecimal Notation: 1100 1001 = C9 .. CLOUD9 BOIZZZ

- base 16, 0..9,A..F
- more compact than binary
- each hex digit is 4 bits
- 0xC9 0x is hex

Q: Given a byte, 11001001, how can we tell if it is unsigned, sign-magnitude, or 2's compliment?

A: We can't! Need to remember what our intent was when we stored the byte!

2) What else could the bits represent?

- A character, which character?
- Need a mapping from chars to numbers, a convention
  - ASCII
    - uses 7 bits
    - 8th bit is supposed to be 0 (companies used it for other characters)
    - compatibility issues
    - 11001001 is not 7-bit ASCII
    - 01001001 is ASCII for I
  - Other Encodings:
    - EBCDIC
    - UNICODE

3) An instruction

Our instructions will be 32 bits long

4) Garbage (unused memory)

Assignment 0
Download MIPS reference sheet on the course webpage and bring to every class