- CPSC 275: Introduction to Computer Systems

CPSC 275: Introduction to Computer Systems

Fall 2025

- Syllabus
- Schedule
- Resources
- Upload
- Solution

Homework 6

NOTE: You are not required to hand in the following exercises, but you are strongly encouraged to complete them to strengthen your understanding of the concepts covered in class.

1. Given the two binary numbers `11111100` and `01110000` in an 8-bit arithmetic representation, which of these two numbers is larger if they are:
   A. Unsigned integers
   B. Signed integers

2. Given a tiny computer with a word size of 6 bits, what are the smallest and largest numbers in each of the following cases?
   A. Unsigned
   B. Signed

3. Assuming $w = 4$, we can assign a numeric value to each possible hexadecimal digit, assuming either an unsigned or a two's-complement interpretation. Fill in the following table according to these interpretations by writing out the nonzero powers of two in the summations shown in Slide 3, Lecture 6:

Hex	Bin	BinToUnsigned	BinToSigned
0xE	1110	$2^3 + 2^2 + 2^1 = 14$	$-2^3 + 2^2 + 2^1 = -2$
0x0			
0x5			
0x8			
0xD			
0xF			

4. Assuming $w = 4$ and a two's-complement interpretation and using the equation in Slide 10, Lecture 6, fill in the following table:

Dec	Bin	Hex	SignedToUnsigned
-8			
-3			
-2			
-1			
0			
5			

Explain how the equation applies to the entries in the table you generated.

5. Later in the course, we will look at listings generated by a *disassembler*, a program that converts an executable program file back to a more readable ASCII form. These files contain many hexadecimal numbers, typically representing values in two's complement form. Being able to recognize these numbers and understand their significance (for example, whether they are negative or positive) is an important skill.

   For the lines labeled A–J (on the right) in the following listing, convert the hexadecimal values (in 32-bit two's-complement form) shown to the right of the instruction names (`sub`, `mov`, and `add`) into their decimal equivalents:

```
8048337: 81 ec b8 01 00 00   sub $0x1b8,%esp             A.
804833d: 8b 55 08            mov 0x8(%ebp),%edx
8048340: 83 c2 14            add $0x14,%edx              B.
8048343: 8b 85 58 fe ff ff   mov 0xfffffe58(%ebp),%eax   C.
8048349: 03 02               add (%edx),%eax
804834b: 89 85 74 fe ff ff   mov %eax,0xfffffe74(%ebp)   D.
8048351: 8b 55 08            mov 0x8(%ebp),%edx
8048354: 83 c2 44            add $0x44,%edx              E.
8048357: 8b 85 c8 fe ff ff   mov 0xfffffec8(%ebp),%eax   F.
804835d: 89 02               mov %eax,(%edx)
804835f: 8b 45 10            mov 0x10(%ebp),%eax         G.
8048362: 03 45 0c            add 0xc(%ebp),%eax          H.
8048365: 89 85 ec fe ff ff   mov %eax,0xfffffeec(%ebp)   I.
804836b: 8b 45 08            mov 0x8(%ebp),%eax
804836e: 83 c0 20            add $0x20,%eax              J.
8048371: 8b 00               mov (%eax),%eax
```

- **Welcome: Sean**
  - LogOut