Number representation: binary arithmetic; overflow

COSC 208, Introduction to Computer Systems, 2021-09-13

Announcements

- Project 1 Part 1 due Thursday (two days mercy Saturday night)
- Exam 1 this Friday

Warm-up

Express these decimal numbers using 8-bit two's complement:

Q1: -49

Q2: -11

🛑 **STOP HERE** after completing the warm-up; if you have extra time please **skip ahead** to the extra practice.

Binary arithmetic

Use 8-bit signed integers

Q3: 10 + 5

Q4: 7 + 15

Q5: -10 + 5

Q6: 10 - 5

Q7: 64 + 64

🛑 **STOP HERE** after completing the above questions; if you have extra time please **skip ahead** to the extra practice.

Overflow

For each of the following computations, determine whether the computation overflows, underflows, or neither. Assume we are using 8-bit signed integers.

Q8: `0b10000000` + `0b01111111`

Q9: `0b10000001` + `0b01111111`

Q10: `0b10000000` + `0b10000001`

Q11: `0b11000000` + `0b11000000`

Q12: `0b01111111` + `0b00000001`

Extra practice

Q13: *Convert 512 to unsigned binary.*

Q14: *Convert -42 to 8-bit signed binary.*

Q15: *Convert `0xFAB` to unsigned binary.*

Turn page

Q16: *Write a function called `valid_hex` that takes a string and returns 1 if it is a valid hexadecimal number; otherwise return 0. A valid hexadecimal number must start with `0x` and only contain the digits `0`-`9` and letters `A`-`F` (in upper or lower case).*

Worksheet created by Professor Aaron Gember-Jacobson