L1 (CHAPTER 2)

Data Representation Exercises ANS

Acknowledgement: some slides taken from Yifeng Zhu's courseware

Review

Decimal, Binary and Hex

Decimal	Binary	Hex
0	0000	0x0
1	0001	0x1
2	0010	0x2
3	0011	0x3
4	0100	0x4
5	0101	0x5
6	0110	0x6
7	0111	0x7
8	1000	0x8
9	1001	0x9
10	1010	0xA
11	1011	0xB
12	1100	0xC
13	1101	0xD
14	1110	0xE
15	1111	0xF

Prefix 0x denotes hex

Question: Number Conversion

- Q: Convert 0x3A56E2F8 into binary

- Q: Convert binary number 111010 into hex

Answer: Number Conversion

- Q: Convert 0x3A56E2F8 into binary
- A: 0011 1010 0101 0110 1110 0010 1111 1000 (simple table lookup for each hex symbol)
- Q: Convert binary number 111010 into hex
- A: 0x3A (group 111010 into two parts 0011 1010, followed by table lookup)

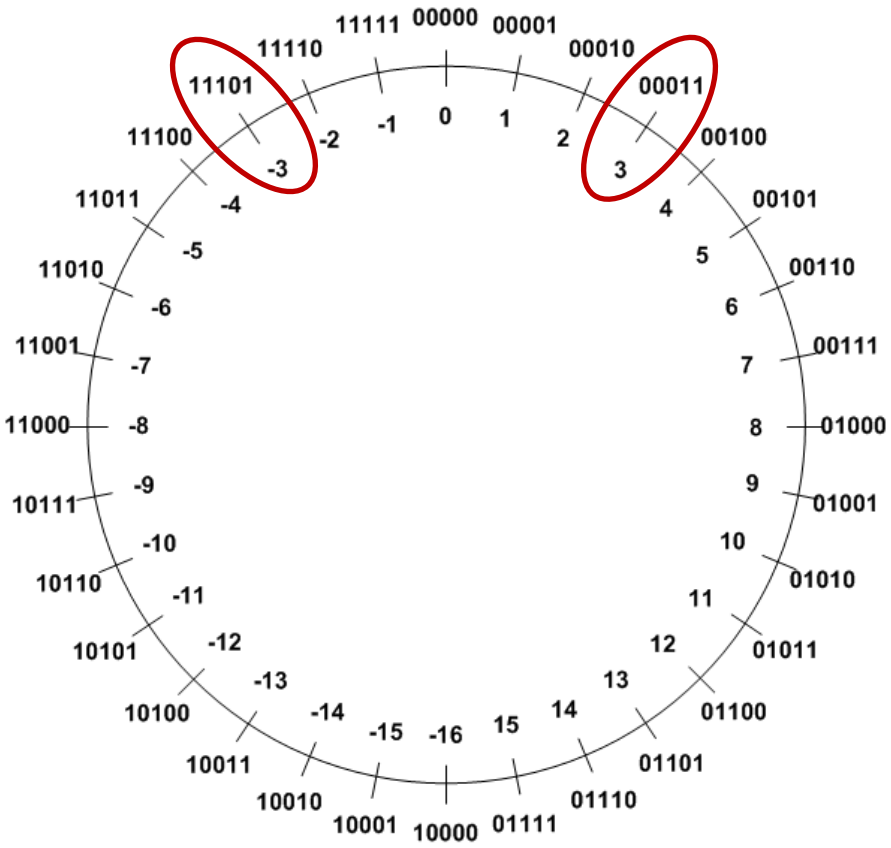
Review

Signed Integers

Method 3: Two's Complement

Two's Complement ($\overline{\alpha}$):

$$\alpha + \overline{\alpha} = 2^n$$



TC of a number can be obtained by its bitwise NOT plus one.

Example **1**: TC(3)

	Binary	Decimal
Original number	00011	3
Step 1: Invert every bit	11100	
Step 2: Add 1	+ 00001	
Two's complement	11101	-3

Review

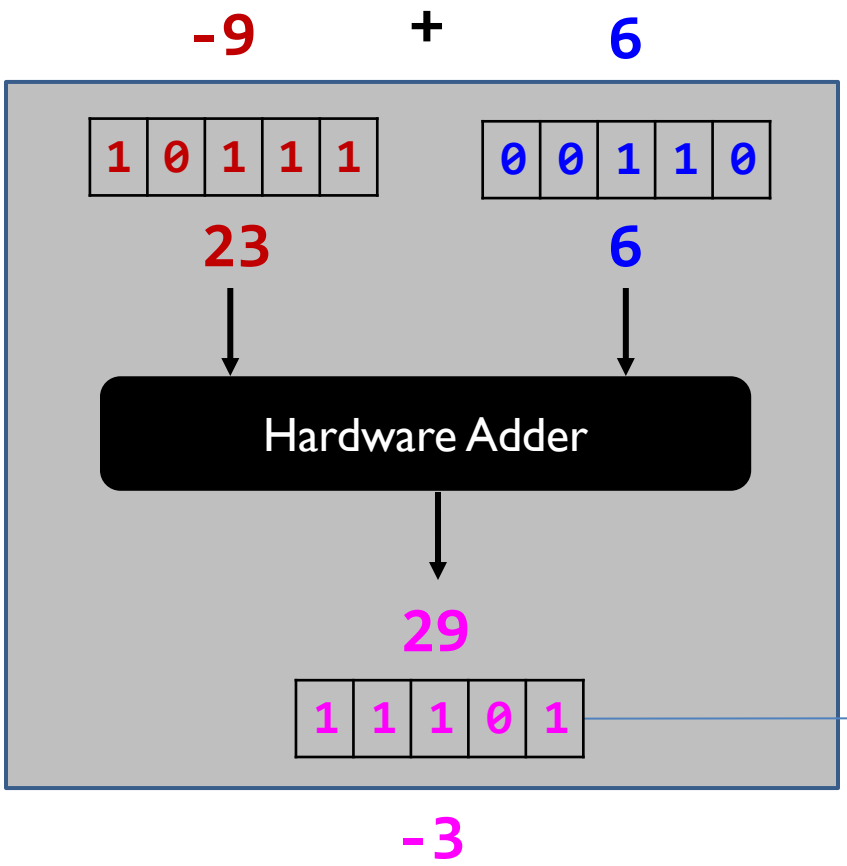
Signed Integer Representation Overview

- Three ways to represent signed binary integers:
  - Signed magnitude
    - $\boldsymbol{value} = (-\mathbf{1})^{\boldsymbol{sign}} \times \boldsymbol{Magnitude}$
  - One's complement ($\tilde{\alpha}$)
    - $\alpha + \tilde{\alpha} = \mathbf{2}^{\boldsymbol{n}} - \mathbf{1}$
  - Two's complement ($\overline{\alpha}$)
    - $\alpha + \overline{\alpha} = \mathbf{2}^{\boldsymbol{n}}$

	Sign-and-Magnitude	One's Complement	Two's Complement
Range	$[-2^{n-1}+1, 2^{n-1}-1]$	$[-2^{n-1}+1, 2^{n-1}-1]$	$[-2^{n-1}, 2^{n-1}-1]$
Zero	Two zeroes (± 0)	Two zeroes (± 0)	One zero
Unique Numbers	$2^n - 1$	$2^n - 1$	2^n

Review

Adding two integers



9
01001 flip 10110 +1 **-9** 10111

Two's Complement

flip 00010 +1 **3** 00011

Two's Complement Counterpart

- Same bit patterns, different interpretation.
  - Unsigned addition: 23+6=29
  - Signed addition: -9+6=-3
- This example shows that the hardware adder for adding unsigned numbers, also works correctly for adding signed numbers.

Question: 2's Complement

- For each of the following binary numbers, give the corresponding binary number of the negative of its value, for 2's-complement system
- (a) x=01010101
- (b) x=10101010
- (c) x=10000000

Answer: 2's Complement

- For each of the following binary numbers x, give the corresponding binary number of –x in 2's-complement representation?
- (a) x=01010101
  - -x = 10101011
- (b) x=10101010
  - -x = 01010110
- (c) x=10000000
  - -x=10000000

Question: Number Conversion

- Q: What is the decimal value of binary number x=10100111 as either unsigned int, or signed int in 2's complement representation?
- What about x=11100001?
- What about x=10000000?

Answer: Number Conversion

- Q: What is the decimal value of binary number x=10100111 as either unsigned int, or signed int in 2's complement representation?
- A: if unsigned int, then x=2^7+2^5+2^2+2^1+2^0=167
- If signed int, then it is a negative number, since leftmost sign bit is 1. First convert it into its positive counterpart of bitwise NOT plus one to get 01011001, which is equal to decimal 2^6+2^4+2^3+2^0=89. Hence x=-89
- Similarly, for x=11100001 (unsigned int 225), first convert it into its positive counterpart of bitwise NOT plus one to get 00011111, which is equal to decimal 31. Hence x=-31
- Similarly, for x=10000000 (unsigned int 128), first convert it into its positive counterpart of bitwise NOT plus one to get 10000000, which is equal to decimal 2^7. Hence x=-2^7=-128

	uint	Int
10100111	167	-89
11100001	225	-31
10000000	128	-128

Question: Number Conversion

- Q: Which number is larger: 1001 or 0011 in binary?
- Q: Which number is larger: 0xFFFFFFFF or 0x00000001 in hex?

Answer: Number Conversion

- Q: Which number is larger: 1001 or 0011 in binary?
- A: depends on the number system.
  - If unsigned int, then 1001 is 9, and 0011 is 3 in decimal, and 9 > 3
  - If signed int, then 1001 is -7 (negative of 0111), and 0011 is 3 in decimal, and -3 < 3
- Q: Which number is larger: 0xFFFFFFFF or 0x00000001 in hex?
- Q: depends on the number system.
  - If unsigned int, then 0xFFFFFFFF is 2^32-1, and 0x00000001 is 1 in decimal, and 2^32-1 > 1
  - If signed int, then 0xFFFFFFFF is -1 (negative of 0x00000001), and 0x00000001 is 1 in decimal, and -1 < 1

Question: Number Range

- Which range of decimals can be expressed with a 6-bit number (assuming Two's complement representation)?

Answer	Range
A	-32 … 32
B	-64 … 63
C	-31 … 32
D	-16 … 15
E	-32 … 31

Answer: Number Range

- Which range of decimals can be expressed with a 6-bit number (assuming Two's complement representation)?

Answer	Range
A	-32 … 32
B	-64 … 63
C	-31 … 32
D	-16 … 15
E	-32 … 31

$$[-2^{n-1}, 2^{n-1}-1] = [-2^5, 2^5-1] = [-32,31]$$

Question: Number Range

- Which range of decimals can be expressed with a 6-bit unsigned integer?

Answer	Range
A	-32 … 32
B	-64 … 63
C	-31 … 32
D	-16 … 15
E	-32 … 31

Answer: Number Range

- Which range of decimals can be expressed with a 6-bit unsigned integer?

Answer	Range
A	-32 … 32
B	-64 … 63
C	-31 … 32
D	-16 … 15
E	0 … 63

$$[0, 2^n - 1] = [0, 2^6 - 1] = [0,63]$$

Question: Integer arithmetic

- Q: What is the result of 1001 + 0011?

Answer: Integer arithmetic

- Q: Consider a 4-bit system. What is the result of 1001 + 0011?
- A: 1001 + 0011 = 1100
- Value of 1100 depends on the number system.
  - If unsigned int, then 1100 is 12, which is equal to 9 (1001) + 3 (0011)
  - If signed int, then 1100 is -4 (negative of 0100), which is equal to -7 (1001) + 3 (0011) in decimal

Review

Summary of Carry and Overflow Flags

Bit	Name	Meaning after add or sub
N	negative	result is negative
Z	zero	result is zero
V	overflow	signed overflow
C	carry	unsigned overflow

Carry flag C = 1 upon an **unsigned** addition if the answer is wrong (true result > 2^n-1)

Carry flag C = 0 (Borrow flag = 1) upon an unsigned subtraction if the answer is wrong (true result < 0)

Overflow flag V =1 upon a **signed** addition if the answer is wrong (true result > 2^{n-1}-1 or true result < -2^{n-1})

CPSR (Current Program Status Register)

20

20

Review

Signed or unsigned

- Whether the carry flag or the overflow flag should be used depends on the programmer's intention.

```
uint a;
uint b;
…
c = a + b
…
```

C Program

Check the carry flag
for unsigned addition

```
int a;
int b;
…
c = a + b
…
```

C Program

Check the overflow flag
for signed addition

21

Question: Addition

- Q: Consider a 4-bit system. What is the result of addition 1011+0110, assuming either unsigned integers, or signed integers in 2's-complement representation?

22

Answer: Addition

- Q: Consider a 4-bit system. What is the result of addition 1011+0110, assuming either unsigned integers, or signed integers in 2's-complement representation?

- A: A 4-bit unsigned int has the range $[0,2^4-1]=[0, 15]$; a 4-bit signed int has the range $[-2^3, 2^3-1]=[-8, 7]$
- 1011 is 11 in decimal as unsigned int; -5 in decimal as signed int; 0110 is 6 as either unsigned or signed int.
- 1011+0110 = 10001; the extra leftmost bit is discarded, so the result is 0001 (1 in decimal) for both cases.
- For unsigned addition, true result should be 11+6=17 in decimal. Since 17>15, the result is wrong, and Carry flag is set to 1.
- For signed addition, true result should be -5+6=1 in decimal. So the result is correct.

23

Question: Subtraction

- Q: Q: Consider a 4-bit system. What is the result of subtraction 1011-0110, assuming either unsigned integers, or signed integers in 2's-complement representation?

24

Answer: Subtraction

- Q: Consider a 4-bit system. What is the result of subtraction 1011-0110, assuming either unsigned integers, or signed integers in 2's-complement representation?

- A: 1011-0110 = 0101 (carry bit discarded), so the computed result is 0101 (5 in decimal) for both cases.
- For unsigned subtraction, true result should be 11-6=5 in decimal. So the result is correct
- For signed subtraction, true result should be -5-6=-11. Since -11 < -8, the result of 5 is wrong, and Overflow flag is set to 1.

25

Question: Subtraction

- Q: Consider a 4-bit system. What is the result of subtraction 0110-1011, assuming either unsigned integers, or signed integers in 2's-complement representation?

26

Answer: Subtraction

- Q: Consider a 4-bit system. What is the result of subtraction 0110-1011, assuming either unsigned integers, or signed integers in 2's-complement representation?

- A: 0110-1011 = 1011 (borrow bit discarded), so the computed result is 11 in decimal for unsigned, or -5 in decimal for signed.
- For unsigned subtraction, true result should be 6-11=-5 in decimal. Since -5 < 0, the result is wrong, and Carry flag is 0 (Borrow flag is 1).
- For signed subtraction, true result should be 6-(-5) =11. Since 11 > 7, the result is wrong, and Overflow flag is set to 1.

27

Question: True or False

- 1. Overflow is impossible when subtracting one unsigned number from another.
- 2. Overflow is impossible when subtracting two signed operands of the same sign.
- 3. There are two representations of zero in 2's complement representation.
- 4. In 2's complement, the absolute values of full-scale negative and full-scale positive are identical

28

Answer: True or False

- 1. Borrow=1 is impossible when subtracting one unsigned number from another. False
- 2. Overflow=1 is impossible when subtracting two signed operands of the same sign. True
- 3. There are two representations of zero in 2's complement representation. False
- 4. In 2's complement, the absolute values of smallest negative and largest positive numbers are identical. False

29