2 Representing Numbers

Bits, Bytes and Words

- Bits refer to single digit, 0 or 1
- Bytes refer to combination of 8 bits
- Words refer to any fixed-width bits (multiples of 8 – 16, 32, 64…)

Base Conversion

Base ⟷ Base ⟷ Base

Shortcut to convert base
→ Always convert to base 2

Example: Convert 243_{10} to Base 16

→ From Base 10 to Base 2:
Step 1: Convert according to sum of binary

- 243_{10} → $1111\ 0011_2$

→From Base 2 to Base 16:
Step 1: Split the base 2 numbers into blocks of 4 (starts from right)

- 1111 | 0011
- "|" represents splitting

Step 2: Refer relationship between hex and decimal

- 1111 is "F" in HEX
- 0011 is "3" in HEX
- $1111\ 0011_2$ → $F3_{16}$

Comparison of One's and Two's

One's	Two's
Hard to detect overflow	Easy to detect overflow
Two zeros (+0 and -0)	Only one zero (+0)
Need to calculate carry bit and overflow	Ignore carry bit and overflow

One's Complement

- A method to represent binary without +ve & -ve
- Convert binary to One's by swapping the numbers (for -ve decimals)
- Easier to do subtraction by just adding One's (with carry bit and overflow)

+ve	One's	-ve	One's
0	000	-0	111
1	001	-1	110
2	010	-2	101
3	011	-3	100

for 3-bit numbers, maximum is +3, 1st digit in One's is to signify positive or negative

$$3-1 = 3+(-1) = 011+110 = 010$$

$$\begin{array}{r} 011 \\ +110 \\ \hline \textcircled{1}001 \\ +1 \\ \hline 010 \end{array}$$

carry bit

* 010 is also equivalent to 2 in base 10.

Two's Complement

- Does not have -0
- Convert binary to Two's by swapping numbers and adding 1 (disregard overflows carry bit)

+ve	Two's	-ve	Two's
0	000	-	
1	001	-1	111
2	010	-2	110
3	011	-3	101
-		-4	100

Without carry bit / overflow:

$$3-4 = 3+(-4) = 011+100 = 111$$

$$\begin{array}{r} 011 \\ +100 \\ \hline 111 \end{array}$$

* 111 ≈ −1 (base 10)

With carry bit / overflow:

$$-4-3 = (-4)+(-3) = 100+101 = 001$$

(ignore carry bit)

$$\begin{array}{r} 100 \\ +101 \\ \hline 1001 \end{array}$$

* 001 ≈ 1 (base 10) so this is false.
−4−3 = −7

Overflow Detection

Scenario	Numbers	Result In
Case 1	(+ve) + (+ve) 3 + 2 = 5 By Two's: 011 + 010= 101	-ve 101 is "-3" But actual answer is 5
Case 2	(-ve) + (-ve) -4 + (-3) = -7 By Two's: 100 + 101 = 001	+ve 001 is "1" But actual answer is -7

Error Detection Methods

	Parity	Checksums	CRC
Errors	Detect single bit	Detect multiple bits	
Condition	-	Numbers don't cancel each other	-
Method	Set parity bit to even/odd	Agree on a number, X	
Process	1. Set extra bit to 0/1 so that number of "1s" is even/odd	1. Sum up the numbers 2. Find the remainder when divided by X 3. Use the remainder to check if the sums are the same again or not	1. Concatenate the numbers 2. Find the remainder when divided by X 3. **same as checksums**