Alexander Saaby Einshøj

INF-1100 OBLIGATORISK INNLEVERING 1

1)

1. 0b1010
2. 0b1001000
3. 0b10000110

2)

1. 10
2. -102
3. -1

3)

1. 0x2A
2. 0x5E
3. 0xCF

4)

1. 0b1010 1011 1100 1101
2. 0b0000000111101111
3. 0b1111 1111 1111 1111 1111 1111 1111 1111

5)

a. 5
b. 9
c. 10

6)

1. 0b00001010 + 0b00100001 = 0b00101011
2. 0b00010000 + 0b11111111 = 0b00001111
3. 0b11111111 + 0b1010 = 0b11111111 + 0b11111010 = 0b11111001 (carry is ignored)
4. 0b01111000 + 0b01100100 = 0b11011100 (OVERFLOW!)

7)

1. 0b01001010 & 0b00100001 = 0b00000000
2. 0b01001011 & 0b11111111 = 0b01001011
3. 0b11110000 | 0b10101010 = 0b11111010
4. 0b11110111 | 0b11101110 = 0b11111111
5. 0b11011111 ^ 0b00100010 = 0b00000010
6. ~0b11011111 = 0b00100000

8)

1) Du har et desimaltall A.

$\mathbf{A = a_n * 2^n + a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + \ldots + a_0 * 2^0}$

2) La sign bit være 0

3) Hvis A er et partall, er RMB (rightmost bit) 0; hvis A er oddetall, er RMB 1

4) Del tallet på 2. Hvis tallet er et oddetall, trekk først fra 1.

5) Gjenta steg 3 til venstresiden av likningen er 0, og fyll ut en verdi for a_i hver gang du gjør det.

6) Hvis A originalt er negativt, utfør $\mathbf{\sim (a_n * 2^n + a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + \ldots + a_0 * 2^0)+1}$

9)

1) Hvis du har n bits, må du se på bit a_n. Hvis den er 0, betyr det at tallet er positivt, er den 1 må du gjøre det samme som steg 6 i forrige oppgave; ~ $(\mathbf{a_n * 2^n + a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + \ldots + a_0 * 2^0}) + \mathbf{1}$.

2) Nå er tallet positivt og vi kan regne ut verdien av det ved å se på bit-strengen vi nå har foran oss.

3) Hvis tallet i utgangspunktet var negativt, setter vi et minustegn foran det.

10)

Hvis man setter >> = logical right shift operation, vil man ved å sette $\mathbf{a_n * 2^n + a_{n-1} * 2^{n-1} + a_{n-2} * 2^{n-2} + \ldots + a_0 * 2^0}$ >> 1, dele det tilsvarende desimaltallet på 2.