CSCI 160

CLASS 4

HW from class 3: Why does this algorithm work?

Justification

We started with:

We know: $N = \underbrace{\overbrace{\quad\quad}^{\text{integer part}}}_{n} \cdot \underbrace{\overbrace{\quad\quad}^{\text{fraction part}}}_{m}$

$$\boxed{N_{(r-1)c} = r^n - r^{-m} - N}$$

1) $M + N_{(r-1)c} = M + r^n - r^{-m} - N$

2) Whether we follow a) or b) depends on the presence of an e. a. c.

What is the magnitude of the e.a.c.? It's an 'overflow' for our representation above:

$\boxed{\text{e.a.c}}$: $\underbrace{10 \ldots\ldots 0}_{n}$ $\boxed{= r^n}$

On **branch a)** there is an e. a. c., which means we have:

$$\underbrace{M + r^n - r^{-m} - N}_{\text{from 1)}} \geq \underbrace{r^n}_{\text{e.a.c.}} \quad \text{<-->} \quad M - N \geq r^{-m} \quad \text{<-->}$$

=smallest positive number in our representation

<--> $M - N > 0$ or $M > N$

It also means, that the case when

$M - N = 0$ takes branch b), which means that 0 will be expressed as -0 by this Algorithm.

We continue by justifying the computations in the branch a) and then branch b).

Remember: $\boxed{N_{(r-1)c} = r^n - r^{-m} - N}$ $N = \underbrace{\qquad}_{n}^{\text{integer part}} \cdot \underbrace{\qquad}_{m}^{\text{fraction part}}$

<u>**Alg (M-N):**</u> 1) $M + N_{(r-1)c}$ 2) a) e.a.c. -> add it to l.s.d.
b) no e.a.c. -> compute -(r-1)'s compl. of 1)

<u>**branch a)**</u> - continued

Branch a) says: "add the e.a.c. to the l.s.d.". This means (s. ex) we have to subtract the value of the e.a.c., which is r^n and add a 1 to the l.s.d., which has the value r^{-m} to the value from 1):

Here it is: $M + r^n - r^{-m} - N - r^n + r^{-m} = \boxed{M - N > 0}$, which is what we wanted.

<u>**branch b)**</u> is taken when there is no e.a.c.. From the Hint (class 3) we know that this is the case when $\boxed{M - N \leq 0}$

Branch b) says: compute the (r-1)'s complement from the result at 1) and give it a negative sign:

$$- (M + N_{(r-1)c})_{(r-1)c} = - (M + r^n - r^{-m} - N)_{(r-1)c} = - (r^n - r^{-m} - (M + r^n - r^{-m} - N)) =$$

$$= - (r^n - r^{-m} - M - r^n + r^{-m} + N) = - (-M + N) = - |M - N| \leq 0 \text{ for } -M + N \geq 0.$$

This is what we wanted. <u>Note:</u> 0 will be expressed as -0 by this Alg., as mentioned before.

Example in decimal

Remember: $\boxed{N_{(r-1)c} = r^n - r^{-m} - N}$

Alg (M-N): 1) $M + N_{(r-1)c}$ 2) a) e.a.c. -> add it to l.s.d.
b) no e.a.c. -> compute -(r-1)'s compl. of 1)

i) $r = 10$, $n = 4$, $m = 2$

$r - 1 = 9$

M: 32.1
N: .64

1 1 1 1

1) M: 0032.10 +
N_{9C}: 9999.35

e.a.c.= 1 0031.45 +

->case2-a) 1

$\boxed{0031.46}$

ii) Swap M <–> N, to get to case 2-b)

M: .64
N: 32.1

1 1

1) M: 0000.64 +
N_{9C}: 9967.89

9968.53 no e.a.c. –>case2-b)

–> –9's compl.:

$\boxed{-0031.46}$

NOTE: Use this table describing various binary codes, for the last 3 exercises on the next page.

Table

Four Different Binary Codes for the Decimal Digits

Decimal Digit	BCD 8421	2421	Excess-3	8, 4, −2, −1
0	0000	0000	0011	0000
1	0001	0001	0100	0111
2	0010	0010	0101	0110
3	0011	0011	0110	0101
4	0100	0100	0111	0100
5	0101	1011	1000	1011
6	0110	1100	1001	1010
7	0111	1101	1010	1001
8	1000	1110	1011	1000
9	1001	1111	1100	1111
	1010	0101	0000	0001
Unused	1011	0110	0001	0010
bit	1100	0111	0010	0011
combi-	1101	1000	1101	1100
nations	1110	1001	1110	1101
	1111	1010	1111	1110

1.14 Obtain the 1's and 2's complements of the following binary numbers:

(a) 10000000 (b) 00000000

(c) 11011010 (d) 01110110

(e) 10000101 (f) 11111111.

1.15 Find the 9's and the 10's complement of the following decimal numbers:

(a) 52,784,630 (b) 63,325,600

(c) 25,000,000 (d) 00,000,000.

1.24 Formulate a weighted binary code for the decimal digits, using weights

(a)*6, 3, 1, 1

(b) 6, 4, 2, 1

1.25 Represent the decimal number 5,137 in (a) BCD, (b) excess-3 code, (c) 2421 code, and (d) a 6311 code.

1.33* The state of a 12-bit register is 100010010111. What is its content if it represents

(a) three decimal digits in BCD?

(b) three decimal digits in the excess-3 code?

(c) three decimal digits in the 84-2-1 code?

(d) a binary number?

r's Complement-Representation

r = base

Remember: $\boxed{N_{(r-1)c} = r^n - r^{-m} - N}$

$N = \underbrace{\overbrace{\qquad\qquad}^{\text{integer part}}}_{n} \bullet \underbrace{\overbrace{\qquad\qquad}^{\text{fraction part}}}_{m}$

n, m = number of locations

Definition

r = 2

How did you compute N_{2c} = 2's complement of N? By swapping 0 <-> 1, and adding 1 to l.s.d., which is equivalent to computing the 1's complement and adding a 1 to l.s.d., which equals 2^{-m}

$$N_{2c} = N_{1c} + 2^{-m} = 2^n - 2^{-m} - N + 2^{-m} = 2^n - N$$

Base r:

$$\boxed{N_{rc} = r^n - N}$$

In our base r representation, where we allocate n digits to the integer part and m digits to the fraction part, we define the r's complement of a number N as follows:

$$N_{rc} = r^n - N$$

Show that in the above representation, given two non-negative numbers M and N as inputs, the algorithm below computes the value of $M - N$, in the same representation. Compute all steps of the algorithm below in the boxes, as indicated. **Justify each step in its box. Compare $M - N$ to 0 in each of the branches of step 2) below, and show the exact inequality covered by each branch in the places indicated.**

ALGORITHM to compute $M - N$:

1) **Compute** $M + N_{rc}$

2) (a) **If there is an e.a.c. then ignore it. Stop.**

   (b) **If there is no e.a.c. then compute:**

   $-(M + N_{rc})_{rc}$

   where $(M + N_{rc})$ **is already computed at step 1) above. Stop.**

Justification

Step 1): [ ]

Step 2) (a): **For this step you got:** $\boxed{M - N \quad 0}$

[ ]

Step 2) (b): **For this step you got:** $\boxed{M - N \quad 0}$

[ ]

Addendum to Q1.2

Answer this **Addendum to Q1.2** before you answer the other Quiz 1.2 questions.

Suppose M = N.

After we apply the two subtraction algorithms we studied:
1. (r – 1)’s complement representation, and
2. r’s complement representation,

which sign (+, or –) should the exact result have in each of the above two algorithms?
Encircle the appropriate sign in spaces below:

1. (r – 1)’s complement

   M – N = + – 0

2. r’s complement

   M – N = + – 0