EE486—Computer Arithmetic
Homework #1 Solutions (35 pts)

(3p) Problem 1.2

$\frac{N}{D} = Q + \frac{R}{D}$ where Q is the quotient and R is the remainder. Assume that D is not zero.

(1) case 1: $(N \geq 0) \Rightarrow Q_s = Q_m$ and $R_s = R_m$

(0.5) case 2a: $(N < 0,\ D > 0$ and $Rm = 0) \Rightarrow Q_s = Q_m$ and $R_s = R_m$

(0.5) case 2b: $(N < 0,\ D > 0$ and $Rm > 0) \Rightarrow Q_s = Q_m + 1$ and $R_s = R_m - D$

(0.5) case 3a: $(N < 0,\ D < 0$ and $Rm = 0) \Rightarrow Q_s = Q_m$ and $R_s = R_m$

(0.5) case 3b: $(N < 0,\ D < 0$ and $Rm > 0) \Rightarrow Q_s = Q_m - 1$ and $R_s = R_m + D$

Moral of the story: always look for the limiting cases.

(6p) Problem 1.9

In the following, the numbers (a, b, c) themselves are rounded first according to the statement of the problem. Then, the operations are rounded to give us the maximum and minimum value of the result.

(3) Case 1: If b and c are of the same sign their product will subtract from s. For s_{max} we will want to reduce their magnitude, and for s_{min} we want to increase their magnitude.

$$\begin{aligned} s_{max} &= \triangle(\triangle a - \bigtriangledown(T(b) \times T(c))\,) \\ s_{min} &= \bigtriangledown(\bigtriangledown a - \triangle(A(b) \times A(c))\,) \end{aligned}$$

Note that for s_{max} we used $(T(b) \times T(c))$. This helps us to combine the case for both numbers being positive or both being negative. If we insist on using $\triangle$ and $\bigtriangledown$ only then we must split this formula into $(\triangle b \times \triangle c)$ for b and c negative and $(\bigtriangledown b \times \bigtriangledown c)$ for b and c positive. Similarly for s_{min} and for the second case below.

(3) Case 2: If b and c are of opposite signs their product will add to s. For s_{max} we will want to increase their magnitude, and for s_{min} we want to decrease their magnitude. Notice that to *increase* the magnitude of a *negative* number we should use $\bigtriangledown$.

$$\begin{aligned} s_{max} &= \triangle(\triangle a - \bigtriangledown(A(b) \times A(c))\,) \\ s_{min} &= \bigtriangledown(\bigtriangledown a - \triangle(T(b) \times T(c))\,) \end{aligned}$$

(5p) Problem 1.10 - IEEE Operations

(1) a) Intermediate result (A * B) is below Min but will be carried as a de-normalized number so that:

$$(A * B) * C = (1).0100 \times 2^{-124}$$

(1) b) No intermediate results below Min:

$$A * (B * C) = (1).0100 \times 2^{-124}$$

Notice that if we did not have denormalized numbers and a flush to zero was used then the result in b) the same but that in a) would be zero.
(1) c) Even though the value of A is shifted out of the mantissa and guard bits, it will still be retained in the sticky bit and used to round up the last bit of the final result.

$$A + B + C = (1).\underbrace{0000000}_{7}1\underbrace{00000000000000}_{14}1 \times 2^5$$

(1) d) Result is still below Max.

$$C * D = (1).0100 \times 2^{127}$$

(1) e) Result is above Max and due to the rounding it is set to the maximum value.

$$(2 * C) * D = (1).\underbrace{11111 \cdots 11111}_{23} \times 2^{127}$$

(6p) Problem 1.11

(3) a) Assume that zero is handled using a reserved exponent value as in the IEEE standard. The implied binary point will be to the right of the MSB. Examples: 0.110 = 3/4, 1.010 = -3/4

(3) b) There are two possible answers to this question depending on how we interpret the range of the mantissa.

If we use the range exactly as it is written, $1/2 < m < 1$, then for a 4 bit mantissa we would allow the following normalized values:

Mantissa	Value
0.101	5/8
0.110	3/4
0.111	7/8
1.001	-7/8
1.010	-3/4
1.011	-5/8

We note that all the positives are of the form 0.1XX, and all the negatives are of the form 1.0XX. Therefore, we can use a hidden bit technique where the MSB bit is assumed to be the opposite of the first bit of the fraction.

However, this range means that there is no way to represent the numbers 1/2 or -1/2. The mantissa range should probably be $1/2 \leq m < 1$. If we use this range then the following normalized values are allowed:

Mantissa	Value
0.100	1/2
0.101	5/8
0.110	3/4
0.111	7/8
1.001	-7/8
1.010	-3/4
1.011	-5/8
1.100	-1/2

Because the value -1/2 does not fit the form 1.0XX, we can not use the system described above, and can not use a hidden bit.

(6p) Problem 1.12

More than one guard bit would only be required if in a single subtraction there was more than one right shift in the align phase and more than one left shift during post-normalization.

Assume the A and B are the same sign (otherwise we are performing addition), and let $|A| \geq |B|$. There are 4 cases that need to be checked:

- If no exponent aligning is required, then $|A| \geq |B| > 1/2\ |A|$. The range of results after the mantissa subtraction is then $0 \leq |A - B| < 1/2\ |A|$. This implies that at least one normalizing left shift will be required, with the possibility of a full mantissa left shift. However, because there was no initial aligning right shift, no guard digits are required.

- If a one-bit aligning right shift is required, then the range of the operands is $|A| > |B| > 1/4\, |A|$. The range of results after the mantissa subtraction is then $0 < |A - B| < 3/4\ |A|$. Here, too, there is the possibility that due to massive cancellation, a full-mantissa left shift may be required. But because the initial aligning right shift was only one bit, one guard digit is sufficient to maintain precision.

- If two aligning right shifts are required, then the range of the operands is $1/2\,|A| > |B| > 1/4\,|A|$. The range of results after the mantissa subtraction is then $1/2\,|A| < |A - B| < 3/4\,|A|$. Because $|A - B|$ is always greater than $1/2\,|A|$, at most a one bit normalizing left shift is required, and one guard digit is sufficient to maintain precision.
- In general, if n right shifts are required with $n > 1$, then $1/2^{n-1}\,|A| > |B|$, and after the mantissa subtraction $|A - B| > (1 - 1/2^{n-1})\,|A| \geq 1/2\,|A|$. Therefore, whenever more than a one bit aligning right shift is required, at most a one bit normalizing left shift will be required, and one guard digit will always be sufficient to maintain precision.

(4p) Problem 1.13

Part	S	significand	change to exp
A	1	1.000	-4
B	0	1.011	+1
C	0	1.000	+2
D	1	1.100	+1

(5p) Problem 1.14 - RP Rounding Table

For this problem treat A as being added directly to the LSB (L). not to the guard bit (G). It makes no difference whether the result is even or odd, so we do not need the value of L.

Sign	G	S	Action	A
X	0	0	Exact result. No rounding	0
0	0	1	Positive, round up mantissa	1
0	1	X	Positive, round up mantissa	1
1	X	X	Negative, truncate G and S	0