CSE 321b: Computer Organization (II)
Third Year, Computer & Systems Engineering

Solution to Assignment #3

1. **Apply Booth's algorithm to multiply -8 (multiplicand) by +5 (multiplier).**

A	Q	Q_{-1}	M	
0000	0101	0	1000	Initial values
1000	0101	0		A ← A – M
1100	0010	1		Shift
0100	0010	1		A ← A + M
0010	0001	0		Shift
1010	0001	0		A ← A – M
1101	0000	1		Shift
0101	0000	1		A ← A + M
0010	1000	0		Shift

2. **Show steps to divide +27 (dividend) by -4 (divisor) using non-restoring division algorithm.**

A	Q	M	
000000	011011	111100	Initial values
000000	011011	000100	Take absolute of Q & M
000000	11011?		Shift
111100			Subtract
111100	11011**0**		Q_0 ← 0
111001	10110?		Shift
111101			Add
111101	10110**0**		Q_0 ← 0
111011	01100?		Shift
111111			Add
111111	01100**0**		Q_0 ← 0
111110	11000?		Shift
000010			Add
000010	11000**1**		Q_0 ← 1
000101	10001?		Shift
000001			Subtract
000001	10001**1**		Q_0 ← 1
000011	00011?		Shift
111111			Subtract
111111	00011**0**		Q_0 ← 0
000011	000110		Add
000011	111010		Adjust signs of A & Q

3. Suppose the IEEE 754 Standard has a binary14 format …

(a) Convert the following numbers to their binary14 counterparts:

i. -11.375

$-11.375 = -2^{3.508} = -1.421875 * 2^3 \rightarrow$ 1 1000010 011011

ii. $-3.3882 * 10^{-21}$

$-3.3882 * 10^{-21} = -2^{-67.99997} = 0.0156253 * 2^{-62} \rightarrow$ 1 0000000 000001

(b) Perform the following calculations …

i. 1 0000011 101101 + 0 0000000 101111

(1) Check for special cases
- No special cases

(2) Transform subtraction to addition and negate second number
- Not needed

(3) Align
- First number (normal): unbiased exponent = 3-*b*, significand = 1.10110100
- Second number (subnormal): unbiased exponent = 1-*b*, significand = 0.10111100
- Second number has a smaller exponent
- Add 2 to its exponent and shift its fraction to the right 2 positions!!
- Exponent of second number = 0000011
- Significand of second number = 0.00101111

(4) Add significands (taking signs into consideration)
- Significand of result = -1.10110100 + 0.00101111 = -1.10000101 (no overflow!)
- Fraction of result = 1000010**1**
- Sign bit of result = 1

(5) Normalize
- Not needed
- Exponent of result = 0000011

(6) Round
- Candidate fractions are 100001 and 100010
- Guard bits = 01
- Since result is negative and we use rounding down ==> pick larger candidate
- Fraction of result = 100010

Result = 1 0000011 100010

ii. 0 0001111 011011 – 1 1100001 101001

(1) Check for special cases
- No special cases

(2) Transform subtraction to addition and negate second number
- Change "–" ==> "+"
- Sign of second number = 0

(3) Align

- First number has a smaller exponent
- Add 82 to its exponent and shift its fraction to the right 82 positions!!
- Exponent of first number = 1100001
- Significand of first number = 0.00000000
- Result equals second number!

Result = 0 1100001 101001