Solutions - Homework 2

(Due date: September 29th)

Presentation and clarity are very important! Show your procedure!

PROBLEM 1 (36 PTS)

- Calculate the result of the following operations with 32-bit floating point numbers. Truncate the results when required. When doing fixed-point division, use 8 fractional bits. Show your procedure.

✓ 80123000 + FACE8000 ✓ 60A10000 + C2F97000 ✓ 7F90BEAD + DFEA0C98	✓ CA09E378 - 80000000 ✓ FAD90000 - 09DECADE ✓ FEE32B88 - FF800000	✓ 80000000 × 497424FE ✓ 7A09D300 × 7F800000 ✓ 8B092000 × 0FACE000	✓ 80000000 ÷ BEEFFACE ✓ FF800000 ÷ 48500000 ✓ 390D3800 ÷ C9600000

✓ $X =$ 80123000 + FACE8000:

80123000: 1000 0000 0001 0010 0011 0000 0000 0000

$e + bias = 00000000 = 0 \rightarrow \textit{Denormal number} \rightarrow e = -126$

80123000 = $-0.00100100011 \times 2^{-126}$

FACE8000: 1111 1010 1100 1110 1000 0000 0000 0000

$e + bias = 11110101 = 245 \rightarrow e = 245 - 127 = 118$

FACE8000 = $-1.10011101 \times 2^{118}$

$X = -0.00100100011 \times 2^{-126} - 1.10011101 \times 2^{118}$

$X = -\dfrac{0.00100100011}{2^{244}} \times 2^{118} - 1.10011101 \times 2^{118}$

The division by 2^{244} requires more than $p + 1 = 24$ bits for proper representation. Thus, we can approximate this first operand by 0.

$X = -1.10011101 \times 2^{118}$

X = 1111 1010 1100 1110 1000 0000 0000 0000 = FACE8000

✓ $X =$ 60A10000 + C2F97000:

60A10000: 0110 0000 1010 0001 0000 0000 0000 0000

$e + bias = 11000001 = 193 \rightarrow e = 193 - 127 = 66$ $\quad$ $Mantissa = 1.0100001$

60A10000 = 1.0100001×2^{66}

C2F97000: 1100 0010 1111 1001 0111 0000 0000 0000

$e + bias = 10000101 = 133 \rightarrow e = 133 - 127 = 6$ $\quad$ $Mantissa = 1.11110010111$

C2F97000 = $-1.11110010111 \times 2^{6}$

$X = 1.0100001 \times 2^{66} - 1.11110010111 \times 2^{6}$

$X = 1.0100001 \times 2^{66} - \dfrac{1.11110010111}{2^{60}} \times 2^{66}$

Representing the division by 2^{60} requires more than $p + 1 = 24$ bits. Thus, we can approximate the 2nd operand with 0.

$X = 1.0100001 \times 2^{66}$

X = 0110 0000 1010 0001 0000 0000 0000 0000 = 60A10000

✓ $X =$ 7F90BEAD + DFEA0C98:

7F90BEAD: 0111 1111 1001 0000 1011 1110 1010 1101

$e + bias = 11111111 = 255, f \neq 0$

7F90BEAD = NaN

$X = NaN + \# = NaN$

✓ $X =$ CA09E378 - 80000000:

CA98D378: 1100 1010 0000 1001 1110 0011 0111 1000

$e + bias = 10010100 = 148 \rightarrow e = 148 - 127 = 21$ $\quad$ $Mantissa = 1.00010011110001101111$

CA09E378 = $-1.00010011110001101111 \times 2^{21}$

80000000: 1000 0000 0000 0000 0000 0000 0000 0000
$e + bias = 00000000 = 0 \rightarrow Denormal\ number \rightarrow e = -126$
80000000 = 0

$X = -1.00010011110001101111 \times 2^{21} = 1.0773 \times 2^{21} = 2259166$

X = 1100 1010 0000 1001 1110 0011 0111 1000 = CA09E378

✓ X = FAD90000 - 09DECADE:
FAD90000: 1111 1010 1101 1001 0000 0000 0000 0000
$e + bias = 11110101 = 245 \rightarrow e = 245 - 127 = 118$ $Mantissa = 1.1011001$
FAD9000 = $-1.1011001 \times 2^{118}$

09DECADE: 0000 1001 1101 1110 1100 1010 1101 1110
$e + bias = 00010011 =\rightarrow e = 19 - 127 = -108$ $Mantissa = 1.10111101100101011011111$
09DECADE = $1.10111101100101011011111 \times 2^{-108}$

$X = -1.1011001 \times 2^{118} - 1.10111101100101011011111 \times 2^{-108}$
$X = -1.1011001 \times 2^{118} - \dfrac{1.10111101100101011011111}{2^{226}} \times 2^{118}$
Representing the division by 2^{226} requires more than $p + 1 = 24$ bits. Thus, we approximate the 2nd operand with 0.

$X = -1.1011001 \times 2^{118}$
X = 1111 1010 1101 1001 0000 0000 0000 0000= FAD90000

✓ X = FEE32B88 - FF800000:
FF800000: 1111 1111 1000 0000 0000 0000 0000 0000
$e + bias = 11111111 = 255, f = 0$
FF800000 = $-\infty$
$X = \# - (-\infty) = +\infty$
X = 7F800000

✓ X = 80000000 × 497424FE:
80000000: 1000 0000 0000 0000 0000 0000 0000 0000
$e + bias = 00000000 = 0 \rightarrow Denormal\ number \rightarrow e = -126$
80000000 = 0
$X = 0 \times \# = 0$
X = 80000000

✓ X = 7A09D300 × 7F800000:
7F800000: 0111 1111 1000 0000 0000 0000 0000 0000
$e + bias = 11111111 = 255, f = 0$
7F800000 = $+\infty$
$X = (\#) \times \infty = \infty$
X = 0111 1111 1000 0000 0000 0000 0000 0000 = 7F800000

✓ X = 8B092000 × 0FACE000:
8B092000: 1000 1011 0000 1001 0010 0000 0000 0000
$e + bias = 00010110 = 22 \rightarrow e = 22 - 127 = -105$ $Mantissa = 1.0001001001$
8B092000 = $-1.0001001001 \times 2^{-105}$

0FACE000: 0000 1111 1010 1100 1110 0000 0000 0000
$e + bias = 00011111 = 31 \rightarrow e = 31 - 127 = -96$ $Mantissa = 1.0101100111$
0FACE000 = $1.0101100111 \times 2^{-96}$

$X = -1.0001001001 \times 2^{-105} \times 1.0101100111 \times 2^{-96}$
$X = -1.01110010011001011111 \times 2^{-201} = -0 \times 2^{-126}$
$e + bias = -201 + 127 = -74 < 0$

Here, there is underflow (not even denormalized numbers different than zero cannot represent it). X is assigned the value -0.
X = 1000 0000 0000 0000 0000 0000 0000 0000 = 80000000

✓ X = 80000000 ÷ BEEFFACE:
80000000: 1000 0000 0000 0000 0000 0000 0000 0000
$e + bias = 00000000 = 0 \rightarrow Denormal\ number \rightarrow e = -126$
80000000 = 0

$X = -0 \div -\# = 0$
X = 00000000

✓ X = FF800000 ÷ 48500000:
FF800000: 1111 1111 1000 0000 0000 0000 0000 0000
$e + bias = 11111111 = 255, f = 0$
FF800000 = $-\infty$

$X = -\infty \div \# = -\infty$
X = FF800000

✓ X = 390D3800 ÷ C9600000:
390D3800: 0011 1001 0000 1101 0011 1000 0000 0000
$e + bias = 001110010 = 114 \rightarrow e = 114 - 127 = -13$ $Mantissa = 1.000110100111$
390D3800 = $1.000110100111 \times 2^{-13}$

C9600000: 1100 1001 0110 0000 0000 0000 0000 0000
$e + bias = 10010010 = 146 \rightarrow e = 146 - 127 = 19$ $Mantissa = 1.11$
C9600000 = -1.11×2^{19}

$$X = -\frac{1.000110100111 \times 2^{-13}}{1.11 \times 2^{19}}$$

```
                    0000000000000010100001
1110000000000  ) 1000110100111100000000
                  1110000000000
                    1110100111000
                    1110000000000
                        10011100000000
                         1110000000000
                           101100000000
```

Alignment:
$$\frac{1.000110100111}{1.11} = \frac{1.000110100111}{1.110000000000} = \frac{1000110100111}{1110000000000}$$

Append x = 8 zeros: $\frac{1000110100111\mathbf{00000000}}{1110000000000}$

Integer division
$Q = 10100001, R = 101100000000$
$\rightarrow Qf = 0.10100001$

Thus:
$$X = \frac{-1.0000011 \times 2^{-13}}{1.11 \times 2^{19}} = -0.10100001 \times 2^{-32} = -1.0100001 \times 2^{-33} = -1.2578 \times 2^{-33} = -1.4643 \times 10^{-10}$$
$e + bias = -33 + 127 = 94 = 01011110$

X = 1010 1111 0010 0001 0000 0000 0000 0000 = AF210000

PROBLEM 2 (14 PTS)

- Complete the table for the following DFX formats:

$B = 2^{n-p_0-2}$
$num0 \in [-2^{n-p_0-2}, 2^{n-p_0-2} - 2^{-p_0}]$
$num1 \in [-2^{n-p_1-2}, -2^{n-p_0-2} - 2^{-p_1}] \cup [2^{n-p_0-2}, 2^{n-p_1-2} - 2^{-p_1}]$
$Dynamic\ Range = 20\log_{10}(2^{n-2-p_1+p_0})$

DFX format	p_0	p_1	Number of bits of significand	Boundary value	$num0$ range	$num1$ range	Dynamic Range (dB)
8_4_2	4	2	7	4	$[-4,3.9375]$	$[16,-4.25]\ U$ $[4,15.75]$	48.1648
12_6_4	6	4	11	16	$[-16,15.984375]$	$[-64,-16.0625]\ U$ $[16,63.9375]$	72.2472
16_8_6	8	6	15	64	$[-64,63.99609375]$	$[-256,-64.015625] U$ $[64,255.984375]$	96.3296
24_16_8	16	8	23	64	$[-64,63.99998474]$	$[-16384,-64.00390625]\ U$ $[64,16383.99609375]$	180.618

PROBLEM 3 (20 PTS)

- Convert the following signed fixed point numbers in format [16 8] to the dual fixed point format 16_8_3. If more bits are required, you are allowed to use the format 17_8_3.

FX	AB.CE	0C.4F	8B.EE	8F.27	81.BE	81.E4	0A.BB	FA.09
DFX								

✓ AB.CE:
1010 1011.1100 1110 ⇒ To DFX 16_8_3 ($num0$): 0010101111001110 = not a $num0$!
⇒ To DFX 16_8_3 ($num1$): 1111110101011110 = FD5E

✓ 0C.4F:
0000 1100.0100 1111 ⇒ To DFX 16_8_3 ($num0$): 0000110001001111 = 0C4F

✓ 8B.EE:
1000 1011.1110 1110 ⇒ To DFX 16_8_3 ($num0$): 0000101111101110 = not a $num0$!
⇒ To DFX 16_8_3 ($num1$): 1111110001011111 = FC5F

✓ 8F.27:
1000 1111.0010 0111 ⇒ To DFX 16_8_3 ($num0$): 0000111100100111 = not a $num0$!
⇒ To DFX 16_8_3 ($num1$): 1111110001111001 = FC79

✓ 81.BE:
1000 0001.1011 1110 ⇒ To DFX 16_8_3 ($num0$): 0000000110111110 = not a $num0$!
⇒ To DFX 16_8_3 ($num1$): 1111110000001101 = FC0D

✓ 81.E4:
1000 0001.1110 0100 ⇒ To DFX 16_8_3 ($num0$): 0000000111100100 = not a $num0$!
⇒ To DFX 16_8_3 ($num1$): 1111110000001111 = FC0F

✓ 0A.BB:
0000 1010.1011 1011 ⇒ To DFX 16_8_3 ($num0$): 0000101010111011 = 0ABB

✓ FA.09:
1111 1010.0000 1001 ⇒ To DFX 16_8_3 ($num0$): 0111101000001001 = 7A09

PROBLEM 4 (30 PTS)

- Calculate the result of the following operations where the numbers are represented in dual fixed-point arithmetic. Note that the results must be in the same format. Include an overflow bit when necessary.

DFX Format: 8_4_2	Result	overflow		Result	overflow
FA+19			2B+A9		
E2+BB			C0+C2		
FB-90			88+1A		

DFX Format 16_8_4	Result	overflow		Result	overflow
72CD+0A98			8939-09A2		
C200+B8C3			E323-7AA9		
F990-0A32			D001+F170		

✓ FA+19:

1 1 1 1 1 0 1 0 +
0 0 0 1 1 0 0 1

➡

$c_7=1$ $c_6=1$ $c_5=1$ $c_4=1$ $c_3=1$ $c_2=1$ $c_1=0$ $c_0=0$

1 1 1 1 0.1 0 +
0 0 0 0 1.1 0 0 1 — optional

0 0 0 0 0.0 0 0 1

00000.0001 ⇒ To DFX 8_4_2 ($num0$): **0**000.0001 = 01 Overflow = 0

✓ E2+BB:

1 1 1 0 0 0 1 0 +
1 0 1 1 1 0 1 1

➡

$c_7=1$ $c_6=1$ $c_5=0$ $c_4=0$ $c_3=0$ $c_2=1$ $c_1=0$ $c_0=0$

1 1 0 0 0.1 0 +
0 1 1 1 0.1 1

0 0 1 1 1.0 1

00111.01 ⇒ To DFX 8_4_2 ($num0$): **0**1110100 = not a $num0$!
⇒ To DFX 8_4_2 ($num1$): **1**0011101 = 9D Overflow = 0

✓ FB-90:

1 1 1 1 1 0 1 1 -
1 0 0 1 0 0 0 0

➡

1 1 1 1 1 0.1 1 -
0 0 0 1 0 0.0 0

➡

$c_8=1$ $c_7=1$ $c_6=1$ $c_5=1$ $c_4=0$ $c_3=0$ $c_2=1$ $c_1=0$ $c_0=0$

1 1 1 1 1 0.1 1 +
1 1 1 1 0 0.0 0

1 1 1 0 1 0.1 1

111010.11 ⇒ To DFX 8_4_2 ($num0$): **0**0101100 = not a $num0$!
⇒ To DFX 8_4_2 ($num1$): **1**1101011 = EB Overflow = 0

✓ 2B+A9:

0 0 1 0 1 0 1 1 +
1 0 1 0 1 0 0 1

➡

$c_8=0$ $c_7=0$ $c_6=0$ $c_5=1$ $c_4=0$ $c_3=0$ $c_2=0$ $c_1=0$ $c_0=0$

0 0 0 1 0.1 0 1 1 +
0 1 0 1 0.0 1 — optional

0 1 1 0 0.1 1 ~~1 1~~

01100.1111 ⇒ To DFX 8_4_2 ($num0$): **0**1001111 = not a $num0$!
⇒ To DFX 8_4_2 ($num1$): **1**0110011 = B3 Overflow = 0

✓ C0+C2:

$c_8=1$ $c_7=1$ $c_6=0$ $c_5=0$ $c_4=0$ $c_3=0$ $c_2=0$ $c_1=0$ $c_0=0$

```
1 1 0 0 0 0 0 0 +        ➡   1 1 0 0 0 0.0 0 +
1 1 0 0 0 0 1 0              1 1 0 0 0 0.1 0
                             ---------------
                             1 0 0 0 0 0.1 0
```

100000.10 ⇒ To DFX 8_4_2 ($num0$): 00001000 = not a $num0$!
⇒ To DFX 8_4_2 ($num1$): 10000010 = not a $num1$! Overflow = 1

✓ 88+1A:

$c_7=0$ $c_6=0$ $c_5=0$ $c_4=0$ $c_3=0$ $c_2=0$ $c_1=0$ $c_0=0$

```
1 0 0 0 1 0 0 0 +        ➡   0 0 0 1 0.0 0 +
0 0 0 1 1 0 1 0              0 0 0 0 1.1 0 1 0
                             -----------------   optional
                             0 0 0 1 1.1 0 1 0
```

00011.1010 ⇒ To DFX 8_4_2 ($num0$): 00111010 = 3A Overflow = 0

✓ 72CD+0A98:

```
                         0 0 0 0 0 1 0 1 0 0 1 1 0 0 0 0
0111001011001101    ➡    1 1 1 0 0 1 0.1 1 0 0 1 1 0 1 +
0000101010011000         0 0 0 1 0 1 0.1 0 0 1 1 0 0 0
                         -----------------------------
                         1 1 1 1 1 0 1.0 1 1 0 0 1 0 1
```

1111101.01100101 ⇒ To DFX 16_8_4 ($num0$): 0111110101100101 = 7D65 Overflow = 0

✓ C200+B8C3:

```
                       0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
1100001000000000 +  ➡  1 0 0 0 0 1 0 0 0 0 0.0 0 0 0 +
1011100011000011       0 1 1 1 0 0 0 1 1 0 0.0 0 1 1
                       -----------------------------
                       1 1 1 1 0 1 0 1 1 0 0.0 0 1 1
```

11110101100.0011 ⇒ To DFX 16_8_4 ($num0$): 0010110000110000 = not a $num0$!
⇒ To DFX 16_8_4 ($num1$): 1111101011000011 = FAC3 Overflow = 0

✓ F990-0A32:

```
                                                  1 1 1 1 1 1 1 0 0 0 0 0 0 0 0 0
1111100110010000  ➡  11110011001.0000 -      ➡    1 1 1 1 0 0 1 1 0 0 1.0 0 0 0 +
0000101000110011     00000001010.0011~~0010~~     1 1 1 1 1 1 1 0 1 0 1.1 1 0 1
                                                  -----------------------------
                                                  1 1 1 1 0 0 0 1 1 1 0.1 1 0 1
```

11110001110.1101 ⇒ To DFX 16_8_4 ($num0$): 0000111011010000 = not a $num0$!
⇒ To DFX 16_8_4 ($num1$): 1111100011101101 = F8ED Overflow = 0

✓ 8939-09A2:

```
                                                    1 1 1 1 1 1 1 0 1 1 0 0 0 0 0 0
1000100100111001 -  ➡  00010010011.1001 -     ➡     0 0 0 1 0 0 1 0 0 1 1.1 0 0 1 +
0000100110100010       00000001001.1010~~0010~~     1 1 1 1 1 1 1 0 1 1 0.0 1 1 0
                                                    -----------------------------
                                                    0 0 0 1 0 0 0 1 0 0 1.1 1 1 1
```

00010001001.1111 ⇒ To DFX 16_8_4 ($num0$): 0000100111110000 = not a $num0$!
⇒ To DFX 16_8_4 ($num1$): 1000100010011111 = 889F Overflow = 0

✓ E323-7AA9:

```
                                                  0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0
1110001100100011  ➡  11000110010.0011 -      ➡    1 1 0 0 0 1 1 0 0 1 0.0 0 1 1 +
0111101010101001     11111111010.1010~~1001~~     0 0 0 0 0 0 0 0 1 0 1.0 1 1 0
                                                  -----------------------------
                                                  1 1 0 0 0 1 1 0 1 1 1.1 0 0 1
```

11000110111.1001 ⇒ To DFX 16_8_4 ($num0$): 0011011110010000 = not a $num0$!
⇒ To DFX 16_8_4 ($num1$): 1110001101111001 = E379 Overflow = 0

✓ D001+F170:

```
                         1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
1101000000000001    ➡    1 0 1 0 0 0 0 0 0 0 0.0 0 0 1 +
1111000101110000         1 1 1 0 0 0 1 0 1 1 1.0 0 0 0
                         -----------------------------
                         1 0 0 0 0 0 1 0 1 1 1.0 0 0 1
```

10000010111.0001 ⇒ To DFX 16_8_4 ($num0$): 0010111000010000 = not a $num0$!
⇒ To DFX 16_8_4 ($num1$): 1100000101110001 = C171 Overflow = 0