

Tugas Mandiri - 2

Pengantar Sistem Digital
Semester Ganjil 2022/2023

Revisi 1.1

Petunjuk pengerjaan:

- Kerjakan dengan tulisan tangan atau diketik.
- Tuliskan Nama, Kelas, dan NPM pada setiap lembar jawaban.
- Tuliskan penjelasan dari cara mendapatkan jawaban tersebut.
- Apabila ditulis tangan, hasil pekerjaan di scan / foto dan dimasukkan ke dalam satu file berformat .pdf.
- Format nama file (tanpa tanda kurung) :
[KodeAsdos]_TM2_[Nama]_[NPM].pdf.
- Tugas mandiri dikumpulkan **Jumat, 16 September 2022 pukul 17.00** pada slot yang sudah disediakan di SCELE.
- Jika **mengumpulkan telat sebelum pukul 23:59 pada hari yang sama**, akan dikenakan **penalti sebesar 50 poin**. Lebih dari waktu tersebut, tugas mandiri **tidak akan dinilai**

1. [24] Lengkapilah tabel berikut!

Desimal	BCD	Excess-3	2421	Excess-5
15	0000 0000 0001 0101	0000 0000 0100 1000	0000 0000 0001 1011	0000 0000 0110 1010
290	0000 0010 1001 0000	0000 0101 1100 0011	0000 0010 1111 0000	0000 0111 1110 0101
1050	0001 0000 0101 0000	0100 0011 1000 0011	0001 0000 1011 0000	0110 0101 1010 0101
2142	0010 0001 0100 0010	0101 0100 0111 0101	0010 0001 0100 0010	0111 0110 1001 0111

2. [16] Ubah bilangan desimal berikut menjadi heksadesimal *floating-point* IEEE-754 32-bit! **Jawab dengan cara pengerjaan!**

a. 1927 = **0x44F0E000**

$$\begin{aligned}
 (1927)_{10} &= 1024 + 512 + 256 + 128 + 4 + 2 + 1 \\
 &= 2^{10} + 2^9 + 2^8 + 2^7 + 2^2 + 2^1 + 2^0 \\
 &= (11110000111)_2 \\
 &= 1.1110000111 \times 2^{10}
 \end{aligned}$$

Sign Bit = 0

Exponential Bit = $127 + 10 = 137 = 10001001$

Fraction Bit = 111 0000 111 0 000

IEEE-754 32 Bit : 0100 0100 1111 0000 1110 0000 0000 0000

Heksadesimal : **0x44F0E000**

b. 20.5 = **0x41A40000**

$$\begin{aligned}
 (20.5)_{10} &= 16 + 4 + 0.5 \\
 &= 2^4 + 2^2 + 2^{-1} \\
 &= (10100.1)_2 \\
 &= 1.01001 \times 2^4
 \end{aligned}$$

Sign Bit = 0

Exponential Bit = $127 + 4 = 131 = 1000 0011$

Fraction Bit = 010 0100 0000 0000 0000 0000

IEEE-754 32 Bit : 0100 0001 1010 0100 0000 0000 0000 0000

Heksadesimal : **0x41A40000**

c. $-525 = \text{0xC4034000}$

$$\begin{aligned}
 (-525)_{10} &= -(512 + 8 + 4 + 1) \\
 &= -(2^9 + 2^3 + 2^2 + 2^0) \\
 &= -(1000001101)_2 \\
 &= -(1.000001101)_2 \times 2^9
 \end{aligned}$$

$$\text{Sign Bit} = 1$$

$$\text{Exponential Bit} = 127 + 9 = 136 = 1000\ 1000$$

$$\text{Fraction Bit} = 000\ 0011\ 0100\ 0000\ 0000\ 0000$$

$$\text{IEEE-754 32 Bit} : 1100\ 0100\ 0000\ 0011\ 0100\ 0000\ 0000\ 0000$$

$$\text{Heksadesimal} : \text{0xC4034000}$$

d. $-2004.25 = \text{0xC4FA8800}$

$$\begin{aligned}
 (-2004.25)_2 &= -(1024 + 512 + 256 + 128 + 64 + 16 + 4 + 0.25) \\
 &= -(2^{10} + 2^9 + 2^8 + 2^7 + 2^6 + 2^4 + 2^{-2}) \\
 &= -(11111010100.01) \\
 &= -(1.111101010001)_2 \times 2^{10}
 \end{aligned}$$

$$\text{Sign Bit} = 1$$

$$\text{Exponential Bit} = 127 + 10 = 137 = 1000\ 1001$$

$$\text{Fraction Bit} = 111\ 1010\ 1000\ 1000\ 0000\ 0000$$

$$\text{IEEE-754 32 Bit} : 1100\ 0100\ 1111\ 1010\ 1000\ 1000\ 0000\ 0000$$

$$\text{Heksadesimal} : \text{0xC4FA8800}$$

3. [18] Lakukanlah operasi aritmatika pada bilangan *floating-point* IEEE-754 32-bit berikut beserta hasilnya dalam heksadesimal IEEE-754 32-bit! **Jawab dengan cara pengerjaan!**

a. $0x40D0CCCD + 0x40A40000 = 0x413A6667$

$$0x40D0CCCD = 0100\ 0000\ 1101\ 0000\ 1100\ 1100\ 1100\ 1101$$

0	1000 0001	101 0000 1100 1100 1100 1101
	Sign Exponential	Fraction

$$\text{Exponent} = (128 + 1) - 127 = 2$$

$$\text{Mantisa} = 1.101\ 0000\ 1100\ 1100\ 1100\ 1101 \times 2^2$$

$$0x40A40000 = 0100\ 0000\ 1010\ 0100\ 0000\ 0000\ 0000\ 0000$$

0	1000 0001	010 0100 0000 0000 0000 0000
	Sign Exponential	Fraction

$$\text{Exponent} = (128 + 1) - 127 = 2$$

$$\text{Mantisa} = 1.01001 \times 2^2$$

Addition

$$\begin{aligned} & 1.101\ 0000\ 1100\ 1100\ 1100\ 1101 \times 2^2 \\ & 1.010\ 01 \times 2^2 \\ & = 10.111\ 0100\ 1100\ 1100\ 1100\ 1101 \times 2^2 + \\ & = 1.0111\ 0100\ 1100\ 1100\ 1100\ 1101 \times 2^3 \\ & \text{Exponent} = 127 + 3 = 130 \end{aligned}$$

0	1000 0010	011 1010 0110 0110 0110 0111
	Sign Exponential	Fraction

$$\text{IEEE-754 32 Bit} = 0x413A6667$$

b. $0x45098014 - 0x425CD1EC = 0x45060CCD$

$$0x45098014 = 0100\ 0101\ 0000\ 1001\ 1000\ 0000\ 0001\ 0100$$

0	1000 1010	000 1001 1000 0000 0001 0100
	Sign Exponential	Fraction

$$\text{Exponent} = (128 + 8 + 2) - 127 = 11$$

$$\text{Mantisa} = 1.0001001100000000000101 \times 2^{11}$$

$$0x425CD1EC = 0100\ 0010\ 0101\ 1100\ 1101\ 0001\ 1110\ 1100$$

0	1000 0100	101 1100 1101 0001 1110 1100
	Sign Exponential	Fraction

$$\text{Exponent} = (128 + 4) - 127 = 5$$

$$\begin{aligned} \text{Mantisa} &= 1.101110011010001111011 \times 2^5 \\ &= 0.0000011011001101000111 \times 2^{11} \end{aligned}$$

Substraction

$$\begin{aligned} & 1.0001001100000000000101 \times 2^{11} \\ & 0.00000110111001101000111 \times 2^{11} \\ & = 1.000100000010011001101 \times 2^{11} \\ & \text{Exponent} = 127 + 11 = 138 = (128 + 8 + 2) \end{aligned}$$

0	100 0101 0 000 0110 0000 1100 1100 1101
	Sign Exponential Fraction

$$\text{IEEE-754 32 Bit} = 0x45060CCD$$

4. [22] Saat kita melakukan operasi aritmatika antara dua bilangan *floating-point* IEEE-754, salah satu titik desimal *mantissa* harus digeser jika eksponen kedua *mantissa* tidak sama.
- Apakah arah pergeseran titik desimal (ke kanan atau kiri) jika *mantissa* yang di-shift memiliki eksponen yang lebih kecil?
Titik desimal akan digeser atau di-shift **ke arah kiri**.
 - Mengapa titik decimal harus digeser dalam kasus ini?
Tujuan digesernya titik desimal yaitu agar **kedua *mantissa* memiliki nilai eksponen yang sama** dan **mempertahankan nilai leading bit**.
Lalu, kenapa harus digeser ke arah kiri?
Karena *mantissa* terbatas hanya menyimpan 24 bit (dengan 23 bit *fraction*) saja, maka lebih baik kita **menghilangkan bit paling kanan (LSB)** dibandingkan menghilangkan leading bit. Ketika digeser ke arah kiri pun maka nilai eksponen juga akan membesar hal ini menyebabkan nilai eksponen yang lebih kecil akan membesar sampai sama dengan nilai eksponen yang satunya.
Contoh:
N1: $1.000110000000000000000000 \times 2^4$
N2: $1.000000000000000000000011 \times 2^{24}$
Jika N1 dan N2 dilakukan operasi aritmatika maka perlu disetarakan dulu eksponen nya dengan menggeser titik desimal N1 sebanyak 20 kali ke kiri. Sehingga menjadi
N1: $0.000000000000000000000001 \times 2^{24}$
Terlihat bahwa nilai LSB pada N1 menghilang ketika dilakukan penyetaraan eksponen.