

Arithmatika Komputer

Pertemuan 3

Oleh :

Riyanto Sigit, S.T, M.Kom

Nur Rosyid Mubtada'i S.Kom

Setiawardhana , S.T

Hero Yudo Martono, S.T

Politeknik Elektronika Negeri Surabaya - ITS

2005

2.4.Representasi Floating Point

- Untuk menuliskan bilangan floating point (bilangan pecahan) dilakukan dengan menuliskan dalam bentuk eksponensial. Sehingga bilangan tersebut memiliki bilangan dasar, bilangan pemangkat dan basis bilangan tersebut

? Formatnya

$$X = X * B_X^{EX}$$

X= bilangan floating point

B_X= basis dari bilangan X

Ex= Bilangan pemangkat Basis

Misalnya:

$$300 = 3 * 10^2$$

? Konversi

- Konversi bilangan floating point berbasis deka ke basis biner harus dilakukan terlebih dahulu sebelum merubah kedalam format representasi floating point. Yaitu dengan cara:
 - Dekan ke biner
 - Contoh:
 - $3,75 \rightarrow 11,11$
 - didapat dari
 - biner dari angka 3 adalah 11 didapat dari $1*2^1+1*2^0=3$.
 - Sedangkan untuk merubah 0,75 menjadi bilangan biner dapat dilakukan dengan cara
 - $0,75*2=1,5$ didapat bilangan didepan koma 1 sisanya dikalikan lagi dengan 2 yaitu
 - $0,5*2=1,0$ didapat bilangan didepan koma 1 dan sisanya 0

? Konversi

- Untuk merubah kembali dapat dilakukan dengan cara :

$$1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2}$$

$$2 + 1 + 0.5 + 0.25 = 3.75$$

Exponensial

- Penulisan bilangan floating point dengan cara exponensial dapat menyebabkan adanya kemungkinan sebuah bilangan ditulis dengan cara yang bermacam-macam

$$300 = 0,3 * 10^3 = 3 * 10^2 = 30 * 10^1 = 30^0 * 100 = 3000 * 10^{-1} \dots$$

demikian juga pada bilangan biner

$$11,11 = 11,11 * 2^0 = 1,111 * 2^1 = 0,1111 * 2^2$$

Standarisasi untuk penulisan bilangan

- Normalisasi bentuknya adalah :

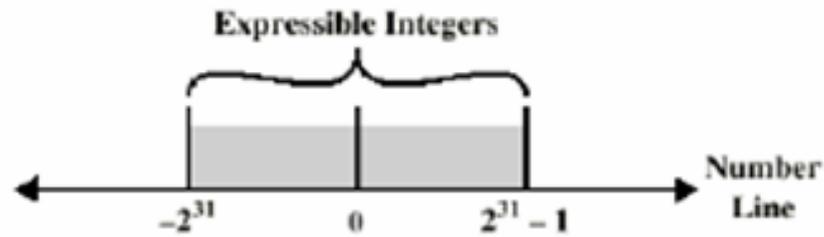
$$\pm 1,bbb\dots b \times 2^{\pm E}$$

- Bit pertama significand selalu 1 sehingga tidak perlu disimpan (implisit) dalam field significand. Dan b adalah bilangan biner (1 atau 0).
- Untuk keperluan yang luas, maka diadakan standar bagi representasi bilangan floating point ini, yaitu standard IEEE 754. Standar ini juga mendefinisikan operasi aritmetikanya

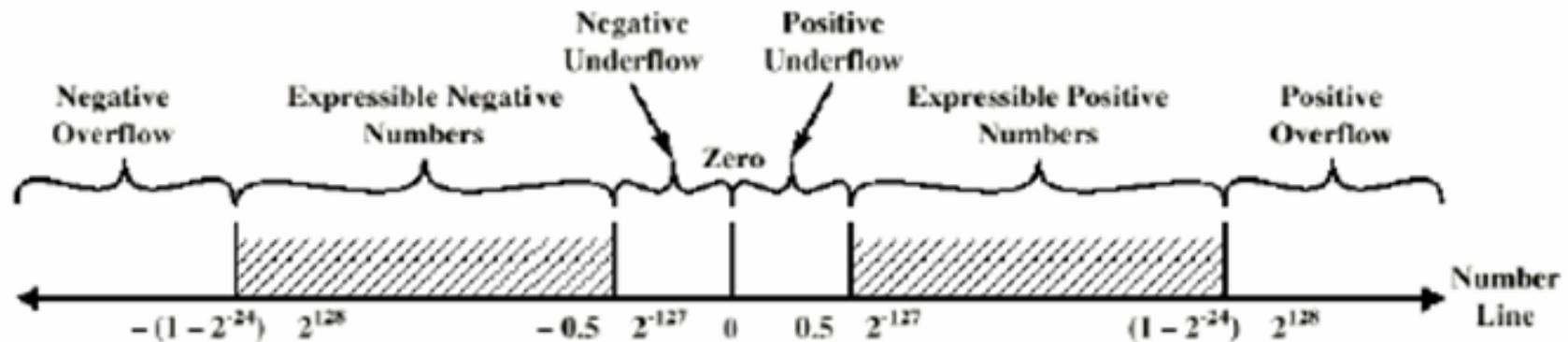
Format penulisan menurut standar IEEE 754

1 bit	8 bit	23 bit
tanda	biased exponent	Fraction / significand
(a) Format tunggal (single format) 32 bit		
1 bit	11 bit	52 bit
	biased exponent	Fraction / significand
(b) Format ganda (double format) 64 bit		

- Pada format tunggal, bit paling kiri digunakan untuk representasi tanda 0 jika positif dan 1 jika negatif, sedangkan 8 bit berikutnya adalah pangkat (exponen) yang direpresentasikan dalam bentuk bias.
- Bagian 23 bit terakhir digunakan untuk menunjukkan bit dari bilangan fraction nya



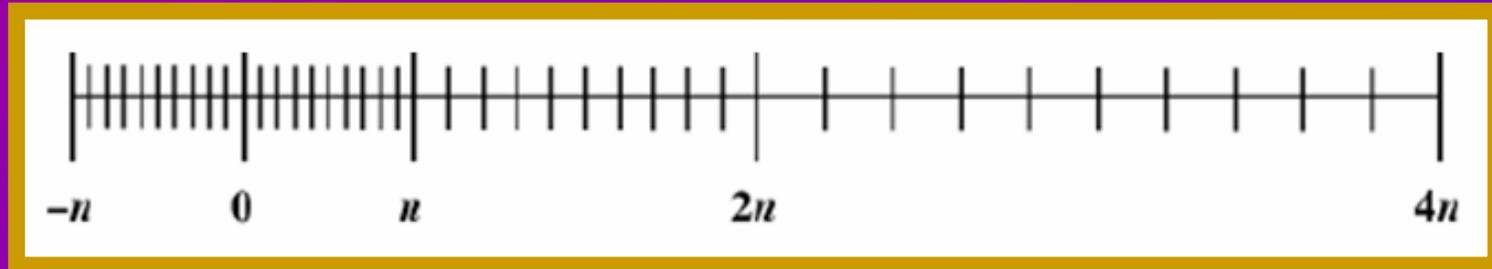
(a) Twos Complement Integers



(b) Floating-Point Numbers

- Untuk word 32 bit, range bilangan yang dapat dicakup diperlihatkan pada gambar berikut.
- Terdapat beberapa segmen tidak termasuk dalam range yang diperlihatkan pada gambar tersebut, yaitu underflow dan overflow

- Gambar berikut menyajikan kerapatan bilangan – bilangan dalam floating point



- Terdapat hubungan antara range dan ketelitian.
- Bila kita menambah jumlah bit eksponen maka kita akan mengembangkan range bilangan yang dapat diekspresikan.
- Konsekuensinya adalah mengurangi kerapatan bilangan dan juga kepresisiannya.
- Satu – satunya cara untuk meningkatkan range dan kepresisian adalah menggunakan jumlah bit yang lebih banyak, misalnya dengan format 64 bit.
- Pada arsitektur IBM S/370 menggunakan basis 16, hal yang didapatkan adalah jumlah range yang lebih besar dengan jumlah bit eksponen yang tetap, tetapi resikonya adalah berkurang ketelitiannya

Contoh konversi bilangan ke format floating point IEEE

Bilangan= -113.3125

$$113_{10} = 1110001_2$$

sedangkan 0.3125 didapat:

$$0.3125 \times 2 = 0.625 \quad 0$$

$$0.625 \times 2 = 1.25 \quad 1$$

$$0.25 \times 2 = 0.5 \quad 0$$

$$0.5 \times 2 = 1.0 \quad 1$$

jadi $113.3125_{10} = 1110001.0101_2$.

di normalisasi menjadi: $1110001.0101 \times 2^0 = 1.1100010101 \times 2^6$

Contoh konversi bilangan ke format floating point IEEE

karena mantisa terdiri dari 23 bit maka akan menjadi 110001010100000000000000.

Bilangan pemangkat adalah 6 maka dirubah ke bias dengan cara $6 + 127 = 133 = 10000101_2$.

Karena negatif maka bilangan tanda = 1

1	10000101	110001010100000000000000
---	----------	--------------------------

Sama dengan

$C2E2A000_{16}$.

2.5. Aritmatika Floating Point

- Operasi aritmatika pada bilangan floating point dapat mengoperasikan bilangan mantisa dan bilangan pemangkatnya **secara terpisah**

Contoh :

Jika :

$$X = X_S * B^{XE}$$

$$Y = Y_S * B^{YE}$$

Maka

$$X+Y = (X_S * B^{XE-YE} + Y_S) * B^{YE}$$

$$X-Y = (X_S * B^{XE-YE} - Y_S) * B^{YE}$$

dengan syarat

$$XE \leq YE$$

$$X*Y = (X_S * Y_S) * B^{XE+YE}$$

$$X/Y = (X_S / Y_S) * B^{XE-YE}$$

Contoh kasus :

$$X = 30 = 0.3 * 10^2$$

$$Y = 200 = 0.2 * 10^3$$

$$X+Y = (0.3*10^{2-3}+0.2)* 10^3 = 0.23* 10^3 = 230$$

$$X-Y = (0.3*10^{2-3}-0.2)* 10^3 = -0.17* 10^3 = -170$$

$$X*Y = (0.3*0.2)* 10^{2+3} = 0.06*10^5 = 6000$$

$$X/Y = (0.3/0.2)* 10^{2-3} = 1.5*10^{-1} = 0.15$$

Permasalahan

- Masalah yang sering muncul pada operasi aritmatika bilangan floating point :
 - Exponent Overflow, sebuah eksponen positif melampaui nilai eksponen maksimum. Dalam sebagian sistem biasa ditandai dengan $+\infty$ atau $-\infty$.
 - Exponent Underflow, sebuah eksponen negatif melampaui nilai eksponen minimum. Hal ini berarti bilangan terlalu kecil untuk dapat direpresentasikan. Biasanya didekatkan dengan nilai 0
 - Significand Overflow, dalam penambahan dua significand yang bertanda sama dapat menghasilkan carry out bit yang paling berarti
 - Significand Underflow, dalam proses penggabungan significand, digit dapat mengalir ke ujung kanan significand. Hal ini biasanya diperlukan pembulatan nilai

- Untuk operasi ini, kedua operand harus dipindahkan ke register yang digunakan oleh ALU.
- Bila format floating point-nya memiliki bit significand implisit maka harus dibuat eksplisit.
- Umumnya, eksponen dan significand akan disimpan di register lain, yang akan digabungkan kembali bila hasil telah diperoleh

4 langkah dalam algoritma penambahan dan pengurangan

1. Periksa bilangan – bilangan nol

Apabila salah satu operand bernilai nol maka hasilnya adalah operand lainnya. ($A + 0 = A$, $0 - A = -A$)

2. Perataan significand

Untuk dapat dilakukan operasi penjumlahan maupun pengurangan maka nilai eksponen haruslah sama. Untuk itu harus dilakukan penyamaan eskponen sebelum significand ditambahkan atau dikurangkan.

Contoh dalam desimal :

$$(1234 \times 10^0) + (345 \times 10^{-2}) = (1234 \times 10^0) + (3,45 \times 10^{-0}) = 1237,45 \times 10^{-0}$$

3. Tambahkan atau kurangkan significand

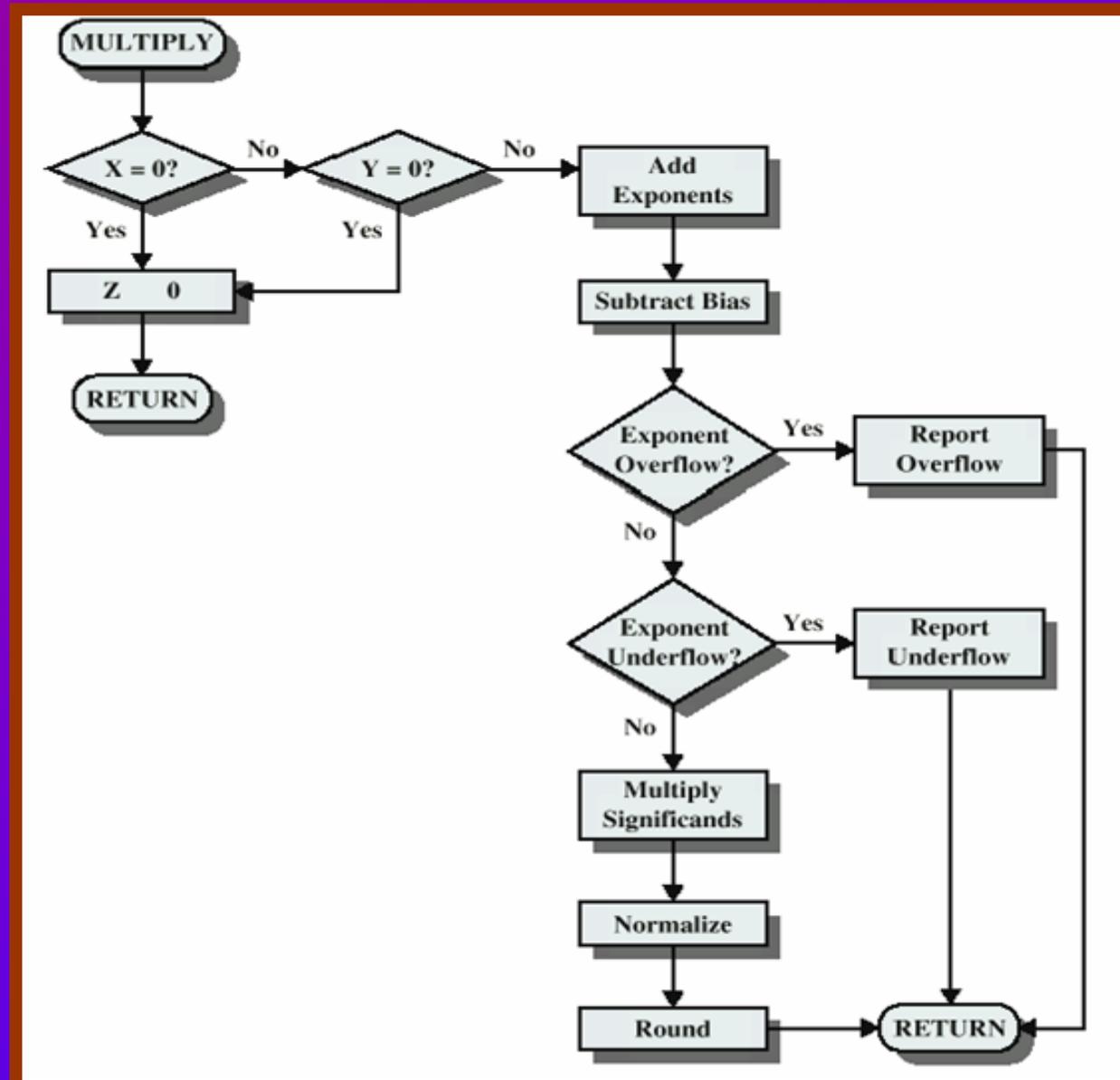
Setelah terjadi perataan maka dilakukan penambahan atau pengurangan dengan memperhatikan bit tandanya

4. Normalisasi hasilnya

Dilakukan pergeseran untuk memenuhi standar yang ditentukan

Perkalian dan Pembagian Floating Point

Perkalian



Langkah Perkalian

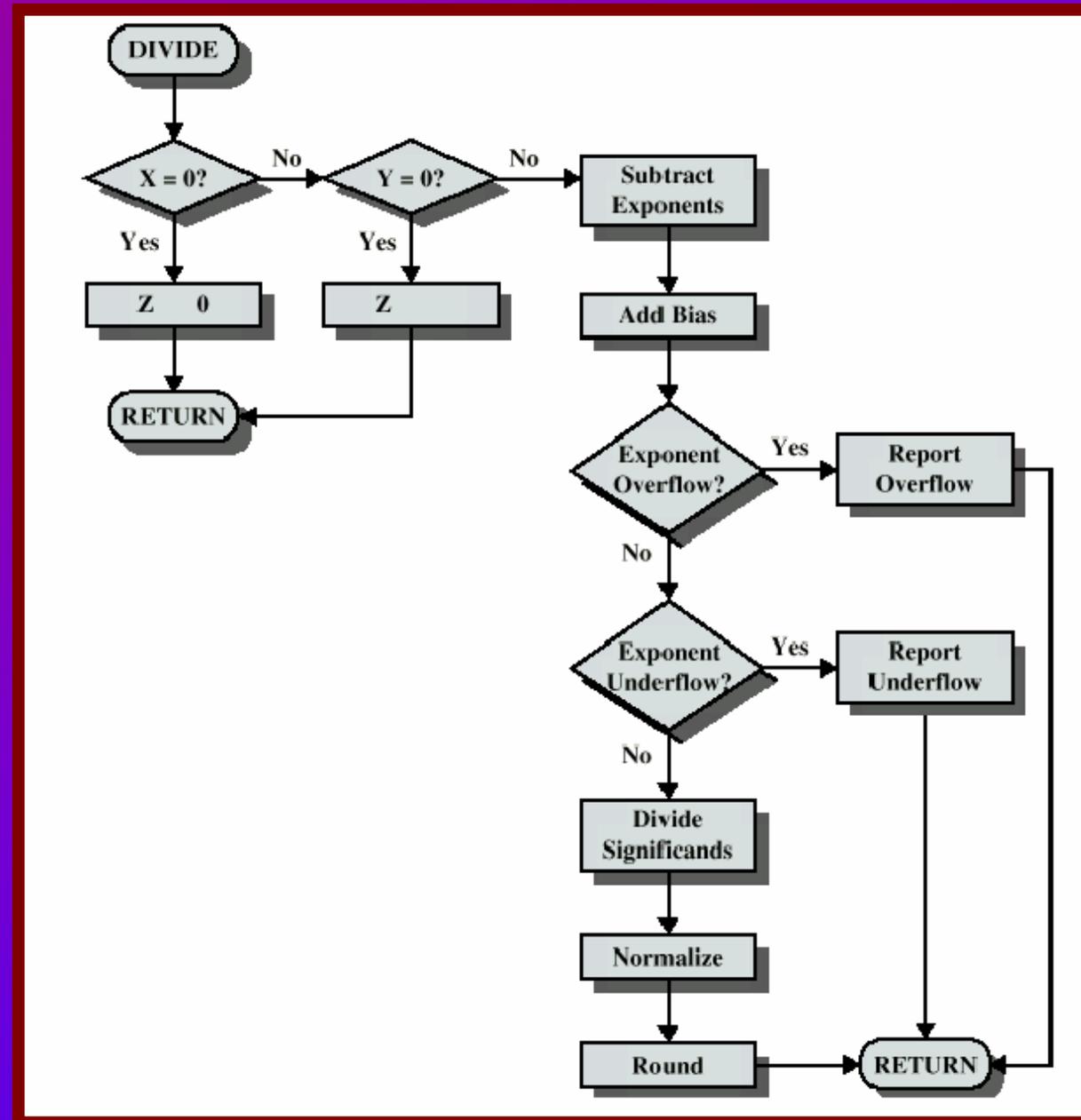
- Bila salah satu operand sama dengan 0, maka 0 dilaporkan sebagai hasilnya. Langkah berikutnya adalah menambah eksponennya.
- Bila eksponen yang disimpan dalam bentuk yang terbias, maka jumlah eksponen akan menggandakan biasnya. Jadi, nilai bias harus dikurangi dari jumlahnya.
- Hasil harus overflow atau underflow untuk dilaporkan sebagai akhir algoritma.
- Apabila eksponen berada pada range yang valid maka langkah selanjutnya adalah mengalikan significand-nya, perlu diperhatikan tandanya.
- Perkalian dilakukan seperti perkalian integer.
- Hasil akan menggandakan bit operand-nya, bit ekstra akan hilang saat pembulatan.
- Setelah hasil didapatkan, diadakan normalisasi dan pembulatan

Langkah Pembagian

- Pada operasi pembagian hampir sama.
- Awalnya lakukan pemeriksaan terhadap nol.
- Bila divisor sama dengan nol maka terjadi error dan hasil diset tak hingga.
- Bila dividend bernilai nol maka hasilnya adalah 0.
- Langkah selanjutnya adalah eksponen dividend dikurang eksponen divisor-nya.
- Hal ini akan menghilangkan bias yang harus dikembalikan lagi.
- Selanjutnya diadakan pemeriksaan overflow dan underflow.
- Langkah akhirnya adalah membagi significand yang diikuti normalisasi dan pembulatan

Perkalian dan pembagian Floating Point

Pembagian



Kesimpulan

1. ALU mempunyai input Register dan Control Unit, sedangkan outputnya adalah Register dan Flag
2. Operasi bilangan integer pada ALU kebanyakan menggunakan representasi komplement dua
3. Operasi pengurangan adalah menggunakan operasi penambahan dengan bilangan negatif operator yang kedua
4. Representasi floating point pada format 32 bit terdiri terdapat 3 bagian yaitu 1 bit menyatakan tanda, 8 bit menyatakan eksponensial, 23 menyatakan bilangan di belakang koma
5. Operasi pada representasi floating point dioperasikan secara terpisah antara bilangan dan pangkatnya

Soal / Tugas

- Buatlah latihan berikut dalam representasi komplement 2
 - $5+7$
 - $6-3$
 - $3+4$
 - $7-2$
- Dengan Algoritma booth operasikan perkalian dibawah ini
 - $5*7$
 - $6*3$
 - $3*4$
 - $7*2$
- Dengan Algoritma booth pembagian komplement dua
 - $5/7$
 - $6/3$
 - $3/4$
 - $7/2$

Soal / Tugas

- Tuliskan representasi floating point untuk bilangan di bilangan dibawah ini.
 - 7,625
 - 0,5
 - -8,75
 - -1000,125
- Berapakah nilai floating point dari bilangan berikut
 - DA1F7000
 - FA51DA00
 - 4A4A7000
- Kalikanlah kedua bilangan tersebut diatas (a dan b)
 - DA1F7000
 - dikali dengan :
 - FA51DA00