CURSUL 6

4. REPREZENTAREA ÎN VIRGULĂ MOBILĂ (VM)

În general, un număr N se poate reprezenta în virgulă mobilă (VM) în forma următoare:

$$\boxed{N = \pm M \cdot B^{\pm E}}$$

Un număr reprezentat în VM are două componente. Prima componentă este **mantisa** (M), care indică valoarea exactă a numărului într-un anumit domeniu, fiind reprezentată de obicei ca un număr fracționar cu semn. A doua componentă este **exponentul** (E), care indică ordinul de mărime al numărului. B este **baza** exponentului.

Această reprezentare poate fi memorată într-un cuvânt binar cu trei câmpuri: semnul, exponentul şi mantisa.

De exemplu, presupunând un cuvânt de 32 de biţi, o asignare posibilă a biţilor la fiecare câmp poate fi următoarea:

31	30 ... 23	22 ... 0
S	EXPONENT	MANTISA

Aceasta este o reprezentare în mărime şi semn, deoarece semnul are un câmp separat față de restul numărului. Câmpul de semn constă dintr-un bit care indică semnul numărului, „0" pentru un număr pozitiv şi „1" pentru un număr negativ. Nu există un câmp rezervat pentru baza B, deoarece această bază este implicită şi ea nu trebuie memorată, fiind aceeaşi pentru toate numerele.

De obicei, câmpul rezervat exponentului nu conține exponentul real, ci o valoare numită **caracteristică**, care se obține prin adunarea unui deplasament la exponent, astfel încât să rezulte întotdeauna o valoare pozitivă. Astfel, nu este necesar să se rezerve un câmp separat pentru semnul exponentului. Caracteristica C este deci exponentul deplasat:

$$\boxed{C = E + \text{deplasament}}$$

Valoarea reală a exponentului se poate afla prin scăderea deplasamentului din caracteristica numărului.

Unul din *avantajele* utilizării exponentului deplasat constă în simplificarea operațiilor executate cu exponentul, datorită lipsei exponenților negativi. *Al doilea avantaj* se referă la modul de reprezentare al numărului zero. Mantisa numărului zero are cifre de 0 în toate pozițiile. Exponentul numărului zero poate avea, teoretic, orice valoare, rezultatul fiind tot zero. La unele calculatoare, dacă un rezultat are mantisa zero, exponentul rămâne la valoarea pe care o are în momentul respectiv, rezultând un "zero impur". La majoritatea calculatoarelor, se recomandă ca numărul zero să aibă cel mai mic exponent posibil, rezultând astfel un "zero pur". În cazul exponenților deplasați, exponentul cu cea mai mică valoare este 0. Deci, prin utilizarea caracteristicii, reprezentarea în VM a numărului 0 este aceeaşi cu reprezentarea în VF, adică toate pozițiile sunt 0. Aceasta înseamnă că se pot utiliza aceleaşi circuite pentru testarea valorii zero.

Un *alt avantaj* al utilizării exponenților deplasați este că numerele pozitive în VM sunt ordonate în acelaşi fel ca şi numerele întregi. Deci, mărimea numerelor în virgulă mobilă poate fi comparată utilizînd un comparator pentru numere întregi.

Un *dezavantaj* al utilizării exponenților deplasați este că adunarea lor este mai complicată, deoarece necesită scăderea deplasamentului din suma exponenților.

În reprezentarea de mai sus, mantisa constă din 23 de biți. Deși virgula binară nu este reprezentată, se presupune că ea este așezată înaintea bitului c.m.s. al mantisei.

De exemplu: dacă B = 2, numărul 1.75 poate fi reprezentat sub mai multe forme:

$$+0.111 \cdot 2^1$$

S	EXPONENT	MANTISA
0	0000 0001	1110 0000 0000 0000 0000 000

$$+0.00111 \cdot 2^3$$

S	EXPONENT	MANTISA
0	0000 0011	0011 1000 0000 0000 0000 000

Pentru simplificarea operațiilor cu numere în VM și pentru creșterea preciziei acestora, se utilizează reprezentarea sub formă normalizată. Un număr în VM este *normalizat* dacă bitul c.m.s. al mantisei este 1.

Din cele două reprezentări ale numărului 1.75, ilustrat mai sus, prima este cea normalizată.

Deoarece bitul c.m.s. al unui număr normalizat în VM este întotdeauna 1, acest bit nu este de obicei memorat, fiind un *bit ascuns* la dreapta virgulei binare. Aceasta permite ca mantisa să aibă un bit semnificativ în plus. Astfel, câmpul de 23 de biți este utilizat pentru memorarea unei mantise de 24 de biți cu valori cuprinse între 0.5 și 1.0.

În unele cazuri, bitul ascuns se presupune poziționat la stânga virgulei binare. Astfel, mantisa memorată M va reprezenta de fapt valoarea 1.M. În acest caz, numărul normalizat 1.75 va avea următoarea formă:

$$+1.11 \cdot 2^0$$

S	EXPONENT	MANTISA
0	0000 0000	1100 0000 0000 0000 0000 000

Presupunând că bitul ascuns este poziționat la stânga virgulei binare în formatul prezentat, un număr normalizat diferit de zero reprezintă următoarea valoare:

$$\boxed{N = (-1)^S \cdot (1.M) \cdot 2^{E-128}}$$

unde: S indică bitul de semn.

Pentru alegerea unui format în VM trebuie realizat un compromis între dimensiunea mantisei și cea a exponentului. Creșterea dimensiunii mantisei va conduce la creșterea preciziei numerelor, iar creșterea dimensiunii exponentului va conduce la creșterea domeniului numerelor care pot fi reprezentate. Singura cale de a crește atât precizia, cât și domeniul numerelor, este de a utiliza un număr mai mare de biți pentru reprezentare.

Cele mai multe calculatoare utilizează cel puțin două formate, în *simplă precizie* (se exemplu, pe 32 de biți), și *dublă precizie* (de exemplu, pe 64 de biți).

4.1. Reprezentarea numerelor în formatul IEEE 754

În trecut au existat diferențe considerabile în modul de execuție a operațiilor în VM la diferite familii de calculatoare. Aceste diferențe se refereau la numărul de biți alocați pentru exponent și pentru mantisă, la gama exponenților, la modurile de rotunjire și la operațiile executate la apariția unor condiții de excepție, ca depășirea superioară sau cea inferioară.

Astfel, IEEE (*Institute of Electrical and Electronics Engineers*) a elaborat un standard pentru reprezentarea numerelor în virgula mobilă și pentru operațiile aritmetice în această reprezentare. Standardul IEEE 754 a fost publicat în anul 1985.

Standardul IEEE 754 definește următoarele formate sau precizii:

- Formatul scurt (precizie simplă): 4 octeți;
- Formatul lung (precizie dublă): 8 octeți;
- Formatul temporar (precizie extinsă): 10 octeți.

Parametrii principali ai acestor formate sunt prezentați în *Tabelul 1*. Standardul nu precizează ca obligatorie implementarea tuturor formatelor, dar recomandă implementarea combinației cu precizie simplă și precizie simplă extinsă, sau a formatelor cu precizie simplă, precizie dublă și precizie dublă extinsă.

Tabelul 1 – Parametrii formatelor definite de standardul IEEE 754

	Precizie simplă	Precizie simplă extinsă	Precizie dublă	Precizie dublă extinsă
Biți ai mantisei	24	≥ 32	53	≥ 64
Exponent real maxim	127	≥ 1023	1023	≥ 16383
Exponent real minim	-126	≤ −1022	-1022	≤ −16382
Deplasament exponent	127	Nespecificat	1023	Nespecificat

Pentru toate formatele, baza implicită este 2. Formatele cu precizie simplă, precizie dublă și precizie dublă extinsă sunt prezentate în figura 1. Coprocesoarele matematice și unitățile de calcul în virgulă mobilă ale procesoarelor implementează de obicei aceste formate.

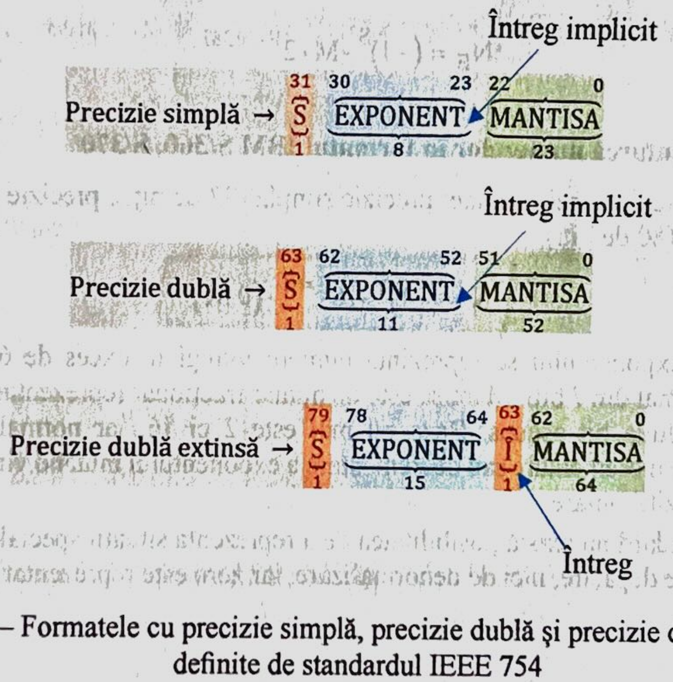

Figura 1 – Formatele cu precizie simplă, precizie dublă și precizie dublă extinsă definite de standardul IEEE 754

S reprezintă semnul numărului. Pentru exponentul deplasat se rezervă 8 biţi în formatul scurt, 11 biţi în formatul lung şi 15 biţi în formatul temporar. Deplasamentul exponentului pentru cele trei formate este de 127 ($7F_{(16)}$), 1023 ($3FF_{(16)}$), respectiv 16383 ($3FFF_{(16)}$). Valorile minime (0) şi cele maxime (255, 2047, respectiv 32767) ale exponentului nu sunt utilizate pentru numerele normalizate, ele fiind utilizate pentru reprezentarea unor valori speciale.

Bitul ascuns este utilizat şi la standardul IEEE 754, dar mantisa este reprezentată într/un mod diferit. Reprezentarea mantisei este denumită *significand* în standardul IEEE. În cazul formatelor cu precizie simplă şi precizie dublă, mantisa constă dintr-un bit implicit cu valoarea 1 (partea întreagă), virgula binară implicită şi biţii fracţiei F:

$$\boxed{M = 1.F}$$

Dacă toţi biţii fracţiei sunt 0, mantisa este 1.0; dacă toţi biţii fracţiei sunt 1, mantisa este cu puţin mai mică decât 2.0. Deci:

$$1.0 \le M \le 2.0$$

Formatul cu precizie dublă extinsă este utilizat pentru reprezentarea numerelor în cadrul unităţilor de calcul în VM şi a coprocesoarelor matematice, în scopul reducerii erorilor datorate rotunjirilor. În acest format, bitul 63 reprezintă partea întreagă a mantisei, care nu este implicită. Numerele în formatul temporar nu sunt întotdeauna normalizate, de aceea nu încep în mod obligatoriu cu un bit de 1. Din acest motiv, acest bit este reprezentat în mod explicit, fiind notat cu Î în cadrul formatului. Valoarea mantisei este în acest caz:

$$M = \hat{I}.F$$

Valoarea unui număr în precizie simplă (N_S), în precizie dublă (N_D) şi în precizie dublă extinsă (N_E) este:

$$\boxed{N_S = (-1)^S \cdot M \cdot 2^{E-127}}$$

$$\boxed{N_D = (-1)^S \cdot M \cdot 2^{E-1023}}$$

$$\boxed{N_E = (-1)^S \cdot M \cdot 2^{E-16383}}$$

4.2. Reprezentarea numerelor în formatul IBM S/360, S/370

Standardul prezintă trei formate: precizie simplă (32 de biţi), precizie dublă (64 de biţi) şi precizie extinsă (80 de biţi).

$$\boxed{N = (-1)^S \cdot 16^{E-64} \cdot (0.M)}$$

În câmpul exponentului se reprezintă numere întregi în exces de 64 şi pentru toate versiunile este format din 7 biţi. Mantisa este un număr fracţionar reprezentat în semn-mărime, nefolosindu-se soluţia bit ascuns. Baza nu mai este 2 ci 16, iar normalizarea se face în consecinţă, fiecare incrementare sau decrementare a exponentului mutând virgula la stânga sau la dreapta cu 4 poziţii binare.

În acest standard nu există posibilitatea de a reprezenta situaţii speciale de tip NaN (*Not a Number*), nici de depăşire, nici de denormalizare, iar zero este reprezentat cu zero peste tot.

Exemple:

1. Care este reprezentarea binară a numărului -0.75 în simplă precizie?

Numărul -0.75 poate fi scris în binar: -0.11.

$$0.75 \cdot 2 = 1.50 \quad \Leftarrow \text{MSB}$$

$$0.50 \cdot 2 = 1.00 \quad \Leftarrow \text{LSB}$$

$$-0.75_{(10)} = -0.11_{(2)}$$

Notația științifică a numărului este: $-0.11 \cdot 2^0$, iar forma normalizată a acestei notații este: $-1.1 \cdot 2^{-1}$. Caracteristica va fi: $-1 + 127 = 126$.

$$126_{(10)} = 0111\,1110_{(2)}$$

Reprezentarea numărului în simplă precizie este deci:

31	30 … 23	22 … 0
1	0111 1110	1000 0000 0000 0000 0000 000

2. Care este numărul zecimal reprezentat de următorul cuvânt:

31	30 … 23	22 … 0
1	1000 0001	0100 0000 0000 0000 0000 000

Bitul de semn este 1, câmpul rezervat caracteristicii conține: $81_{(16)}$.

$$81_{(16)} = 8 \cdot 16^1 + 1 \cdot 16^0 = 128 + 1 = 129_{(10)}$$

iar câmpul mantisei conține: $0.01_{(2)}$, având notația științifică: $1 \cdot 2^{-2} = 0.25$, iar forma normalizată: 1.25.

Valoarea numărului este:

$$(-1)^1 \cdot 1.25 \cdot 2^{129-127} = -1.25 \cdot 2^2 = -5.0$$

3. Care este reprezentarea binară a numărului -10.375, în virgulă mobilă, simplă precizie?

- Numărul pozitiv se transformă în binar și se obține:

$10,375_{(10)} = 1010,011_{(2)}$

Binar
$10 : 2 = 5 + 0$
$5 : 2 = 2 + 1$
$2 : 2 = 1 + 0$
$1 : 2 = 0 + 1$
$0.375 \cdot 2 = 0.75$
$0.75 \cdot 2 = 1.5$
$0.5 \cdot 2 = 1.0$
$1010.011_{(2)}$

10 : 2 = 5, rest 0; 5 : 2 = 2, rest 1; 2 : 2 = 1, rest 0; 1

$10_{(10)} = 1010_{(2)}$

$0,375 \cdot 2 = 0,750$

$0,75 \cdot 2 = 1,50$

$0,50 \cdot 2 = 1,00$

- Se scrie numărul obținut în binar, sub formă normalizată:

$$1010.011_{(2)} = 1.010011 \cdot 2^3$$

- Se determină valoarea exponentului:

$$E = 3 + 127 = 130$$

- Se transformă noul exponent în binar:

Binar
$130:2 = 65 + 0$
$65:2 = 32 + 1$
$32:2 = 16 + 0$
$16:2 = 8 + 0$
$8:2 = 4 + 0$
$4:2 = 2 + 0$
$2:2 = 1 + 0$
$1:2 = 0 + 1$
$1000\ 0010_{(2)}$

$130_{(10)} = 10000010_{(2)}$

- Se determină bitul de semn al mantisei: 1;
- Se scrie numărul:

S	EXPONENT	MANTISA
1	1000 0010	0100 1100 0000 0000 0000 000

4. Care este reprezentarea hexazecimală a numărului $-228.15_{(10)}$, în virgulă mobilă, simplă precizie?

- Numărul pozitiv se transformă în binar și se obține:

Binar
$228:2 = 114 + 0$
$114:2 = 57 + 0$
$57:2 = 28 + 1$
$28:2 = 14 + 0$
$14:2 = 7 + 0$
$7:2 = 3 + 1$
$3:2 = 1 + 1$
$1:2 = 0 + 1$
$0.15 \cdot 2 = 0.30$
$0.30 \cdot 2 = 0.60$
$0.60 \cdot 2 = 1.20$
$0.20 \cdot 2 = 0.40$
$0.40 \cdot 2 = 0.80$
$0.80 \cdot 2 = 1.60$
$0.60 \cdot 2 = 1.20$
$0.20 \cdot 2 = 0.40$
$0.40 \cdot 2 = 0.80$
$0.80 \cdot 2 = 1.60$
$1110\ 0100.0010\ 0110\ 0110\ 01_{(2)}$

$228_{(10)} = 11100100_{(2)}$

- Se scrie numărul obținut în binar, sub formă normalizată:

$$1110\ 0100{,}0010\ 0110\ 0110\ 01_{(2)} = 1{,}1100\ 1000\ 0100\ 1100\ 1100\ 100 \cdot 2^7$$

- Se determină valoarea exponentului: $E = 7 + 127 = 134$;
- Se transformă noul exponent în binar:

Binar
$134 : 2 = 67 + \mathbf{0}$
$67 : 2 = 33 + \mathbf{1}$
$33 : 2 = 16 + \mathbf{1}$
$16 : 2 = 8 + \mathbf{0}$
$8 : 2 = 4 + \mathbf{0}$
$4 : 2 = 2 + \mathbf{0}$
$2 : 2 = 1 + \mathbf{0}$
$1 : 2 = 0 + \mathbf{1}$
$1000\ 0110_{(2)}$

- Se determină bitul de semn al mantisei: 1;
- Se scrie numărul, în binar:

$$\overset{S}{1}\ \overset{\text{EXPONENT}}{1000\ 0110}\ \overset{\text{MANTISA}}{1100\ 1000\ 0100\ 1100\ 1100\ 100}$$

(rotunjirea se face prin adăugarea unei unități, după cei 23 de biți ai mantisei)

- Se scrie numărul, în hexazecimal:

$1100\ 0011\ 0110\ 0100\ 0010\ 0110\ 0110\ 0101_{(2)} =$

$$= \underbrace{1100}_{C}\ \underbrace{0011}_{3}\ \underbrace{0110}_{6}\ \underbrace{0100}_{4}\ \underbrace{0010}_{2}\ \underbrace{0110}_{6}\ \underbrace{0110}_{6}\ \underbrace{0101}_{5}{}_{(2)} = C3642665_{(16)}$$

5. Care este reprezentarea hexazecimală a numărului $-27.25_{(10)}$, în virgulă mobilă, simplă precizie?

- Numărul pozitiv se transformă în binar și se obține:

$27,25_{(10)} = 11011,01_{(2)}$

Binar
$27 : 2 = 13 + \mathbf{1}$
$13 : 2 = 6 + \mathbf{1}$
$6 : 2 = 3 + \mathbf{0}$
$3 : 2 = 1 + \mathbf{1}$
$1 : 2 = 0 + \mathbf{1}$
$0.25 \cdot 2 = \mathbf{0}.5$
$0.50 \cdot 2 = \mathbf{1}.0$
$1\,1011.01_{(2)}$

$27_{(10)} = 11011_{(2)}$

$0.25 \cdot 2 = 0.50$

$0.50 \cdot 2 = 1.00$

- Se scrie numărul obținut în binar, sub formă normalizată:

$$11011.01_{(2)} = 1.101101 \cdot 2^4$$

- Se determină valoarea exponentului: $E = 4 + 127 = 131$;
- Se transformă noul exponent în binar:

$131_{(10)} = 10000011_{(2)}$

Binar
$131 : 2 = 65 + 1$
$65 : 2 = 32 + 1$
$32 : 2 = 16 + 0$
$16 : 2 = 8 + 0$
$8 : 2 = 4 + 0$
$4 : 2 = 2 + 0$
$2 : 2 = 1 + 0$
$1 : 2 = 0 + 1$
$1000\ 0011_{(2)}$

131 | 2
130 65 | 2
1 64 32 | 2
1 32 16 | 2
0 16 8 | 2
0 8 4 | 2
0 4 2 | 2
0 2 1
0

- Se determină bitul de semn al mantisei: 1;
- Se scrie numărul, în binar:

S	EXPONENT	MANTISA
1	1000 0011	1011 0100 0000 0000 0000 000

- Se scrie numărul, în hexazecimal:

$1100\ 0001 1101 1010\ 0000\ 0000\ 0000\ 0000_{(2)} =$

$= \underbrace{1100}_{C}\ \underbrace{0001}_{1}\ \underbrace{1101}_{D}\ \underbrace{1010}_{A}\ \underbrace{0000}_{0}\ \underbrace{0000}_{0}\ \underbrace{0000}_{0}\ \underbrace{0000}_{0}{}_{(2)} = C1DA0000_{(16)}$

6. Care este reprezentarea hexazecimală a numărului $0.1_{(10)}$, în virgulă mobilă, simplă precizie?

- Numărul se transformă în binar și se obține:

Binar
$0.1 \cdot 2 = 0.2$
$0.2 \cdot 2 = 0.4$
$0.4 \cdot 2 = 0.8$
$0.8 \cdot 2 = 1.6$
$0.6 \cdot 2 = 1.2$
$0.2 \cdot 2 = 0.4$
$0.4 \cdot 2 = 0.8$
$0.8 \cdot 2 = 1.6$
$0.6 \cdot 2 = 1.2$
$0.2 \cdot 2 = 0.4$
$0.4 \cdot 2 = 0.8$
$0.8 \cdot 2 = 1.6$
$0.000110011001_{(2)}$

- Se scrie numărul obținut în binar, sub formă normalizată:

$$0.000110011001_{(2)} = 1.10011001 \cdot 2^{-4}$$

- Se determină valoarea exponentului: $E = -4 + 127 = 123$;
- Se transformă noul exponent în binar:

Binar
$123:2 = 61 + 1$
$61:2 = 30 + 1$
$30:2 = 15 + 0$
$15:2 = 7 + 1$
$7:2 = 3 + 1$
$3:2 = 1 + 1$
$1:2 = 0 + 1$
$0111\ 1011_{(2)}$

123 | 2
122 61 | 2
(1) 60 30 | 2
(1) 30 15 | 2
(0) 14 7 | 2
(1) 6 3 | 2
(1) 2 (1)
(1)

$123_{(10)} = 01111011_{(2)}$

- Se determină bitul de semn al mantisei: 1;
- Se scrie numărul, în binar:

S	EXPONENT	MANTISA
0	0111 1011	1001 1001 1001 1001 1001 101

(rotunjirea se face prin adăugarea unui unități, după cei 23 de biți ai mantisei)

- Se scrie numărul, în hexazecimal:

S → $\underbrace{0011\ 1101\ 1}_{E}100\ 1100\ 1100\ 1100\ 1100\ 1101_{(2)}$ =

$= \underbrace{0011}_{3}\ \underbrace{1101}_{D}\ \underbrace{1100}_{C}\ \underbrace{1100}_{C}\ \underbrace{1100}_{C}\ \underbrace{1100}_{C}\ \underbrace{1100}_{C}\ \underbrace{1101}_{D} = 3DCCCCCD_{(16)}$

7. Care este reprezentarea hexazecimală a numărului $1.2_{(10)}$, în virgulă mobilă, simplă precizie?

- Numărul se transformă în binar şi se obţine:

Binar
$0.2 \cdot 2 = 0.4$
$0.4 \cdot 2 = 0.8$
$0.8 \cdot 2 = 1.6$
$0.6 \cdot 2 = 1.2$
$0.2 \cdot 2 = 0.4$
$0.4 \cdot 2 = 0.8$
$0.8 \cdot 2 = 1.6$
$0.6 \cdot 2 = 1.2$
$0.2 \cdot 2 = 0.4$
$0.4 \cdot 2 = 0.8$
$0.8 \cdot 2 = 1.6$
$0.6 \cdot 2 = 1.2$
$1.001100110011_{(2)}$

- Numărul obţinut, sub formă normalizată, este:

$1.001100110011_{(2)}$

- Valoarea exponentului este: E = 127;
- Se transformă exponentul în binar:

Binar
$127:2 = 63 + 1$
$63:2 = 31 + 1$
$31:2 = 15 + 1$
$15:2 = 7 + 1$
$7:2 = 3 + 1$
$3:2 = 1 + 1$
$1:2 = 0 + 1$
$1111111_{(2)}$

$127 : 2$, 126, 1; $63 : 2$, 62, 1; $31 : 2$, 30, 1; $15 : 2$, 14, 1; $7 : 2$, 6, 1; $3 : 2$, 2, 1; 1

$127_{(10)} = 01111111_{(2)}$

- Se determină bitul de semn al mantisei: 1;
- Se scrie numărul, în binar:

S	EXPONENT	MANTISA
0	0111 1111	0011 0011 0011 0011 0011 001

(rotunjirea se face prin adăugarea unui unități, după cei 23 de biți ai mantisei)

- Se scrie numărul, în hexazecimal:

$$0011\,1111\,1001\,1001\,1001\,1001\,1001\,1010_{(2)} =$$

$$\underbrace{0011}_{3}\,\underbrace{1111}_{F}\,\underbrace{1001}_{9}\,\underbrace{1001}_{9}\,\underbrace{1001}_{9}\,\underbrace{1001}_{9}\,\underbrace{1001}_{9}\,\underbrace{1010}_{A}{}_{(2)} = 3F99999A_{(16)}$$

8. Următorul cuvânt pe 32 de biți este interpretat ca reprezentând, conform standardului IEEE 754, un număr în virgulă mobilă:

$$0100\ 0110\ 0001\ 1001\ 0000\ 0000\ 0000\ 0000$$

Să se afle valoarea numărului astfel reprezentat.

Putem afla valoarea numărului astfel reprezentat parcurgând următoarele etape:

- Separăm cele trei câmpuri:

S	EXPONENT	MANTISA
0	1000 1100	0011 0010 0000 0000 0000 000

- Identificăm semnul ca fiind pozitiv.
- Scădem din cel de-al doilea câmp

$$1000\ 1100_{(2)} = 1\cdot 2^7 + 1\cdot 2^3 + 1\cdot 2^2 = 140_{(10)}$$

valoarea $127_{(10)}$ pentru a afla valoarea exponentului.

Rezultă:

$$E = 140 - 127 = 13$$

- Adăugăm 1 în fața celui de al treilea câmp al mantisei:

$$1.M = 1.0011\ 0010\ 0000\ 0000\ 0000\ 000$$

☞ Obținem numărul reprezentat, mutând virgula cu 13 poziții la dreapta în numărul 1.M și obținem:

$$N = 2^{13} \cdot 1001\,1001\,0000\,00 \cdot 2^{-13} = 1 \cdot 2^{13} + 1 \cdot 2^{10} + 1 \cdot 2^{9} + 1 \cdot 2^{6} = 9792_{(10)}$$

Deci exponentul ia valori în intervalul $[-127,+128]$, iar precizia cu care poate fi reprezentat un număr este de 24 de biți.

9. Se dă numărul 1915.40625.

Se cere reprezentarea sa în standardul IEEE 754.

Pentru a reprezenta numărul în standardul IEEE 754 este necesar mai întâi conversia acestuia în binar:

Binar
$1915 : 2 = 957 + 1$
$957 : 2 = 478 + 1$
$478 : 2 = 239 + 0$
$239 : 2 = 119 + 1$
$119 : 2 = 59 + 1$
$59 : 2 = 29 + 1$
$29 : 2 = 14 + 1$
$14 : 2 = 7 + 0$
$7 : 2 = 3 + 1$
$3 : 2 = 1 + 1$
$1 : 2 = 0 + 1$
$0.40625 \cdot 2 = 0.81250$
$0.81250 \cdot 2 = 1.6250$
$0.6250 \cdot 2 = 1.250$
$0.250 \cdot 2 = 0.5$
$0.5 \cdot 2 = 1.0$
$111\,0111\,1011.0110\,1_{(2)}$

$$1915.40625_{(10)} = 111\,0111\,1011.0110\,1_{(2)} = 1.1101\,1110\,1101\,101 \cdot 2^{10}$$

☞ Ținând cont de forma unui număr în IEEE 754 (normalizată):

$$N = (-1)^{S} \cdot 2^{E-127} \cdot (1.M)$$

rezultă: $E = 127 + 10 = 137$.

☞ Se transformă exponentul în binar:

Binar
$137 : 2 = 68 + 1$
$68 : 2 = 34 + 0$
$34 : 2 = 17 + 0$
$17 : 2 = 8 + 1$
$8 : 2 = 4 + 0$
$4 : 2 = 2 + 0$
$2 : 2 = 1 + 0$
$1 : 2 = 0 + 1$
$1000\,1001_{(2)}$

☞ Reprezentarea conform IEEE 754 rezultă:

S	EXPONENT	MANTISA
0	1000 1001	1101 1110 1101 1010 0000 000

☞ Cei 32 de biţi ai reprezentării pot fi consideraţi într-o formă compactă ca şi un cod hex pe 8 simboluri:

$$0\ 1000\ 1001\ 1101\ 1110\ 1101\ 1010\ 0000\ 000_{(2)} =$$

$$= \underbrace{0100}_{4}\ \underbrace{0100}_{4}\ \underbrace{1110}_{E}\ \underbrace{1111}_{F}\ \underbrace{0110}_{6}\ \underbrace{1101}_{D}\ \underbrace{0000}_{0}\ \underbrace{0000}_{0}{}_{(2)} = 44EF6D00_{(16)}$$

10. Se dă numărul $1915.40625_{(10)}$.

Se cere reprezentarea sa în standardul IBM S360/370.

☞ Standardul IBM S360/370 codifică un număr sub formatul general:

$$N = (-1)^S \cdot 16^{E-64} \cdot (0.M)$$

☞ Deoarece baza reprezentării este $16 = 2^4$, mutarea virgulei pentru a obţine pentru N formatul IBM se va face în grupuri de câte 4 cifre binare.

☞ Conversia numărului din zecimal în binar este:

$$1915.40625_{(10)} = 111\ 0111\ 1011.0110\ 1_{(2)}$$

☞ Se mută virgula, spre stânga, cu 12 poziţii $\left(2^{12} = 2^{4\cdot 3} = 16^3\right)$:

$$1915.40625_{(10)} = 111\ 0111\ 1011.0110\ 1_{(2)} =$$

$$= 0.0111\ 0111\ 1011\ 0110\ 1 \cdot 2^{12} = 0.0111\ 0111\ 1011\ 0110\ 1 \cdot 16^3$$

☞ Rezultă că exponentul este:

$$E = 64 + 3 = 67$$

☞ Se transformă exponentul în binar:

Binar
$67:2 = 33 + 1$
$33:2 = 16 + 1$
$16:2 = 8 + 0$
$8:2 = 4 + 0$
$4:2 = 2 + 0$
$2:2 = 1 + 0$
$1:2 = 0 + 1$
$100\ 0011_{(2)}$

☞ Reprezentarea conform IBM S360/370 rezultă:

S	EXPONENT	MANTISA
0	100 0011	0111 0111 1011 0110 1000 0000

☞ Codul hex care rezultă pentru reprezentare este:

$$0\ 100\ 0011\ 0111\ 0111\ 1011\ 0110\ 1000\ 0000_{(2)} =$$

$$= \underbrace{0100}_{\Downarrow \atop 4}\ \underbrace{0011}_{\Downarrow \atop 3}\ \underbrace{0111}_{\Downarrow \atop 7}\ \underbrace{0111}_{\Downarrow \atop 7}\ \underbrace{1011}_{\Downarrow \atop B}\ \underbrace{0110}_{\Downarrow \atop 6}\ \underbrace{1000}_{\Downarrow \atop 8}\ \underbrace{0000}_{\Downarrow \atop 0}\,{}_{(2)} = 4377B680_{(16)}$$

11. Se consideră codul de 32 de biţi sub forma compactă 43B4D680, reprezentând un număr în standard IEEE 754. Care este valoarea numărului în zecimal?

☞ Desfăşurarea codului pe 32 de biţi oferă informaţii asupra semnului, mantisei şi exponentului numărului, ţinând cont de formula generală:

$$N = (-1)^S \cdot 2^{E-127} \cdot (1.M)$$

☞ Codul desfăşurat este:

$$\underbrace{4}_{\Downarrow \atop 0100}\ \underbrace{3}_{\Downarrow \atop 0011}\ \underbrace{B}_{\Downarrow \atop 1011}\ \underbrace{4}_{\Downarrow \atop 0100}\ \underbrace{D}_{\Downarrow \atop 1101}\ \underbrace{6}_{\Downarrow \atop 0110}\ \underbrace{8}_{\Downarrow \atop 1000}\ \underbrace{0}_{\Downarrow \atop 0000} = \underbrace{S}_{\Downarrow \atop 0}\ \underbrace{EXPONENT}_{\Downarrow \atop 100\ 0011\ 1}\ \underbrace{MANTISA}_{\Downarrow \atop 011\ 0100\ 1101\ 0110\ 1000\ 0000}$$

☞ Semnul este: $S = 0 \Rightarrow$ numărul este pozitiv;

☞ Exponentul este: $E = 1000\ 0111_{(2)} = 1 \cdot 2^7 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 135_{(10)}$;

☞ Mantisa este: $M = 0110\ 1001\ 1010\ 1101\ 0000\ 000$

$$N = 2^{135-127} \cdot (1.0110\ 1001\ 1010\ 1101\ 0000\ 000) =$$

$$= 2^8 \cdot (10110\ 1001.1010\ 1101\ 0000\ 000) \cdot 2^{-8} = 10110\ 1001.1010\ 1101$$

☞ În zecimal, numărul este:

$$N = 1 \cdot 2^8 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-3} + 1 \cdot 2^{-5} + 1 \cdot 2^{-6} + 1 \cdot 2^{-8} =$$

$$= 256 + 64 + 32 + 8 + 1 + 0.5 + 0.125 + 0.03125 + 0.015625 + 0.00390625 = 361.67578125$$

12. Se consideră codul de 32 de biţi sub forma compactă 43B4D680, reprezentând un număr în standard IBM S360/370. Care este valoarea numărului în zecimal?

☞ Desfăşurarea codului pe 32 de biţi oferă informaţii asupra semnului, mantisei şi exponentului numărului, ţinând cont de formula generală:

$$N = (-1)^S \cdot 16^{E-64} \cdot (0.M)$$

☞ Codul desfăşurat este:

$$\underbrace{4}_{\Downarrow \atop 0100}\ \underbrace{3}_{\Downarrow \atop 0011}\ \underbrace{B}_{\Downarrow \atop 1011}\ \underbrace{4}_{\Downarrow \atop 0100}\ \underbrace{D}_{\Downarrow \atop 1101}\ \underbrace{6}_{\Downarrow \atop 0110}\ \underbrace{8}_{\Downarrow \atop 1000}\ \underbrace{0}_{\Downarrow \atop 0000} = \underbrace{S}_{\Downarrow \atop 0}\ \underbrace{EXPONENT}_{\Downarrow \atop 100\ 0011}\ \underbrace{MANTISA}_{\Downarrow \atop 1011\ 0100\ 1101\ 0110\ 1000\ 0000}$$

☞ Semnul este: $S = 0 \Rightarrow$ numărul este pozitiv;

☞ Exponentul este: $E = 100\ 0011_{(2)} = 1 \cdot 2^6 + 1 \cdot 2^1 + 1 \cdot 2^0 = 67_{(10)}$;

☞ Mantisa este: $M = 1011\ 0100\ 1101\ 0110\ 1000\ 0000$;

$$N = 16^{67-64} \cdot (0.1011\ 0100\ 1101\ 0110\ 1000\ 0000) =$$

$$= 16^3 \cdot 1011\ 0100\ 1101.0110\ 1000\ 0000 \cdot 16^{-3} = 1011\ 0100\ 1101.01101$$

☞ **În zecimal, numărul este:**

$$N = 1 \cdot 2^{11} + 1 \cdot 2^9 + 1 \cdot 2^8 + 1 \cdot 2^6 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^0 + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-5} =$$

$$= 2048 + 512 + 256 + 64 + 8 + 4 + 1 + 0.25 + 0.125 + 0.03125 = 2893.40625$$