fracționare, subunitare. Dacă virgula este aşezată după cifra c.m.p.s., se operează cu numere întregi. În continuare se presupune această poziționare, considerând că se lucrează cu numere întregi.

În *forma cu virgulă mobilă*, fiecare număr este caracterizat prin două valori:

- *Mantisa*, care indică mărimea exactă a numărului într-un anumit domeniu;
- *Exponentul*, care indică ordinul de mărime a numărului, fiind puterea la care se ridică baza mantisei. Exponentul indică deci implicit poziția virgulei binare.

2.5. Reprezentarea numerelor în virgulă fixă

2.5.1. Reprezentarea numerelor cu semn

În continuare se vor nota cu x, y numerele în reprezentarea binară obişnuită, la care se ataşează semnul. Un număr cu n cifre de mărime se va scrie sub forma:

$$\pm x = \pm x_{n-1}\, x_{n-2} \ldots x_1\, x_0$$

De exemplu:

`±6 = ±0110`

Se vor nota cu X, Y numerele în reprezentarea din calculator, care conțin şi cifrele de semn:

$$X = x_n\, x_{n-1}\, x_{n-2} \ldots x_1\, x_0$$

După modul de exprimare a numerelor negative, există trei forme uzuale de reprezentare a numerelor cu semn în virgulă fixă:

- În mărime şi semn (`MS`);
- În complement față de 1 (`C1`);
- În complement față de 2 (`C2`).

Pentru toate formele, un număr pozitiv se exprimă în acelaşi fel:

$$X = 0 \cdot 2^n + \sum_{i=0}^{n-1} x_i 2^i = 0 \cdot 2^n + x \tag{2.21}$$

Numărul *pozitiv cu semn* se reprezintă deci adăugând cifra 0 de semn în fața numărului fără semn:

$$X = 0\, x_{n-1}\, x_{n-2} \ldots x_1\, x_0$$

Considerăm pentru simplitate numere cu 4 biți de mărime şi un bit de semn. De exemplu, numărul fără semn `5` se reprezintă prin:

```
 5              0101
```

iar numărul cu semn `+5` prin:

```
+5          0 0101
```

Cea mai simplă formă de reprezentare este cea *în mărime şi semn*. Un număr negativ reprezentat în mărime şi semn are expresia:

$$X = 1 \cdot 2^n + \sum_{i=0}^{n-1} x_i 2^i = 1 \cdot 2^n + x \tag{2.22}$$

Deci, numărul negativ se reprezintă prin adăugarea cifrei 1 de semn în faţa numărului fără semn:

$$X = 1\ x_{n-1}\ x_{n-2}\ \ldots\ x_1\ x_0$$

De exemplu, numărul `-5` se va reprezenta prin:

```
-5(MS) 1 0101
```

Există mai multe dezavantaje ale acestei reprezentări. Prima este că adunarea şi scăderea necesită circuite mai complexe. A doua este că există două reprezentări pentru valoarea `0`:

```
+0          0 0000
-0          1 0000
```

De aceea este mai dificil să se testeze dacă o valoare este 0 (o operaţie frecventă), decât în cazul în care ar exista o singură reprezentare.

În cazul reprezentării *în complement faţă de 1*, un număr negativ se reprezintă prin complementul faţă de 1 al numărului pozitiv cu aceeaşi valoare absolută. Complementul faţă de 1 al unui număr binar se obţine prin înlocuirea biţilor de 1 cu 0, şi a celor de 0 cu 1. Un număr negativ reprezentat în complement faţă de 1 are expresia:

$$X = 1 \cdot 2^n + \sum_{i=0}^{n-1} \overline{x_i} 2^i = 2^{n+1} - x - 1 \tag{2.23}$$

unde $\overline{x_i} = 1 - x_i$ reprezintă complementul faţă de 1 al cifrei x_i.

Complementul faţă de 1 al unui număr negativ se obţine prin complementarea tuturor cifrelor numărului fără semn şi adăugarea cifrei de semn 1:

$$X = 1\,\overline{x}_{n-1}\,\overline{x}_{n-2} \ldots \overline{x}_1\,\overline{x}_0$$

De exemplu:

```
+5          0 0101
-5(C1)      1 1010
```

Există şi în acest caz două reprezentări pentru 0:

```
+0          0 0000
-0          1 1111
```

Un număr negativ reprezentat în *complement faţă de 2* are expresia:

$$X = 1 \cdot 2^n + \sum_{i=0}^{n-1} \overline{x_i} 2^i + 1 \cdot 2^0 = 2^{n+1} - x \qquad (2.24)$$

Complementul față de 2 al unui număr se poate obține în mai multe moduri. O posibilitate o reprezintă utilizarea relației de definiție (2.24). De exemplu, considerând $n = 4$, complementul față de 2 al numărului `-5` va fi:

$2^{4+1} - 5 = 32 - 5 = 27 = 11011$

O altă posibilitate este obținerea complementului față de 2 în două etape:

1. Se obține complementul față de 1.
2. Se consideră rezultatul ca un întreg fără semn, la care se adună valoarea 1.

De exemplu:

```
+5          0 0101
-5(C1)      1 1010 +
                 1
            ______
-5(C2)      1 1011
```

Practic, complementul față de 2 al unui număr se poate determina pornind de la numărul pozitiv cu semn, astfel:

1. Se scriu cifrele numărului începând cu cifra c.m.p.s., neschimbate, până la primul 1 inclusiv.
2. Se complementează cifrele întâlnite în continuare.

Există o singură reprezentare pentru 0 în C2. În plus, operațiile de adunare şi scădere se efectuează cel mai simplu în această reprezentare. Se poate arăta că reprezentarea prin C2 conduce la aflarea valorii reale a numărului, dacă cifra de semn se consideră negativă.

În cazul reprezentării în MS şi C1, gama numerelor care pot fi exprimate prin n biți de mărime este:

$$-(2^n - 1) \le x \le 2^n - 1$$

Pentru reprezentarea în C2, această gamă este:

$$-2^n \le x \le 2^n - 1$$

De exemplu, pentru numere de un octet (7 cifre de mărime şi o cifră de semn), gama pentru reprezentarea în MS şi C1 este:

$$-(2^7 - 1) \le x \le 2^7 - 1$$

adică:

$$-127 \le x \le 127$$

iar în C2:

$$-128 \le x \le 127$$

În Tabelul 2.5 se prezintă reprezentarea unor numere cu 4 biți de mărime şi un bit de semn în MS, C1 şi C2.

Tabelul 2.5. Diferite moduri de reprezentare a unor numere cu semn.

X	MS	C1	C2
+15	0 1111	0 1111	0 1111
+14	0 1110	0 1110	0 1110
…			
+2	0 0010	0 0010	0 0010
+1	0 0001	0 0001	0 0001
0	0 0000 1 0000	0 0000 1 1111	0 0000
-1	1 0001	1 1110	1 1111
-2	1 0010	1 1101	1 1110
…			
-14	1 1110	1 0001	1 0010
-15	1 1111	1 0000	1 0001
-16	-	-	1 0000

2.5.2. Reguli de deplasare a numerelor cu semn

De multe ori sunt necesare operații de deplasare a numerelor cu semn. Aceste deplasări trebuie efectuate astfel încât să se modifice numai valoarea numerelor, nu şi semnul. Deplasarea la stânga cu o poziție este echivalentă înmulțirii cu 2, iar deplasarea la dreapta este echivalentă împărțirii cu 2 (înmulțirii cu 2^{-1}).

Pentru stabilirea regulilor de deplasare, se consideră exemplele din Tabelul 2.6.

Tabelul 2.6. Stabilirea regulilor de deplasare a numerelor cu semn.

		MS	C1	C2
X	+ 6	0 0110	0 0110	0 0110
$X \cdot 2$	+12	0 1100	0 1100	0 1100
$X \cdot 2^{-1}$	+ 3	0 0011	0 0011	0 0011
X	- 6	1 0110	1 1001	1 1010
$X \cdot 2$	-12	1 1100	1 0011	1 0100
$X \cdot 2^{-1}$	- 3	1 0011	1 1100	1 1101

La deplasare participă numai cifrele de mărime ale numerelor. Din analizarea tabelului rezultă următoarele:

- În cazul deplasării numerelor pozitive, în pozițiile rămase libere după deplasarea la stânga sau la dreapta se introduc cifre de 0.
- La numerele negative reprezentate în MS, în pozițiile rămase libere după o deplasare la stânga sau la dreapta se introduc cifre de 0.
- La numerele negative reprezentate în C1, în pozițiile rămase libere după o deplasare la stânga sau la dreapta se introduc cifre de 1.
- La numerele negative reprezentate în C2, în pozițiile rămase libere după o deplasare la stânga se introduc cifre de 0, iar după o deplasare la dreapta se introduc cifre de 1 (deci, se repetă semnul numărului).

2.5.3. Operații cu numere reprezentate în virgulă fixă

2.5.3.1. Adunarea numerelor reprezentate în C2

Metodele de adunare şi scădere a numerelor reprezentate în C2 demonstrează avantajele acestei reprezentări. Aceste operații se pot efectua ca şi în cazul numerelor fără semn.

Considerăm câteva exemple de adunare. În primele exemple, numerele sunt de acelaşi semn.

```
a)      + 9             0 1001
        + 5             0 0101
        ---             ------
        +14             0 1110
```

Rezultatul este corect.

```
b)      + 9             0 1001
        +11             0 1011
        ---             ------
        +20             1 0100
```

Se obține un număr negativ, deci rezultatul este incorect. Deoarece $+20 > 2^4 = 16$, se depăşeşte capacitatea de reprezentare a rezultatului.

```
c)      - 9              1 0111
        - 5              1 1011
        ---              ------
        -14             ①1 0010
```

Rezultatul este corect, deoarece $-14 > -2^4 = -16$. Se obține un număr negativ reprezentat în C2. Apare un transport de la cifra de semn, care se neglijează.

```
d)      - 9              1 0111
        -11              1 0101
        ---              ------
        -20             ①0 1100
```

Rezultatul este eronat, deoarece se obține un număr pozitiv.

În exemplele următoare, numerele sunt de semne contrare.

```
e)    + 7        0 0111
      - 4        1 1100
      + 3      ①0 0011

f)    - 7        1 1001
      + 4        0 0100
      - 3        1 1101
```

În cazul numerelor de semne contrare, rezultatul este întotdeauna corect, deoarece nu poate apare depăşire de capacitate.

Din exemplele prezentate, se poate formula *regula generală de adunare* a două numere reprezentate în C2:

> Se adună numerele bit cu bit, inclusiv biții de semn, care sunt tratați la fel cu biții de mărime, şi se ignoră eventualul transport de la bitul de semn. Dacă rezultatul este negativ, apare ca un număr reprezentat în C2.

Dacă rezultatul este mai mare în valoare absolută decât valoarea maximă care poate fi reprezentată în registru, apare o *depăşire* (exemplele *b* şi *d*). La apariția depăşirii, UAL trebuie să semnaleze acest fapt, astfel încât rezultatul să nu fie utilizat.

Se observă că depăşirea poate apare indiferent dacă există sau nu transport de la cifra de semn. Pentru *detectarea apariției depăşirii*, se poate aplica următoarea regulă simplă:

> La adunarea a două numere de acelaşi semn, apare depăşire dacă şi numai dacă rezultatul are semn contrar semnului numerelor.

2.5.3.2. Scăderea numerelor reprezentate în C2

Scăderea a două numere reprezentate în C2 se poate efectua fie prin metoda scăderii directe, dacă se dispune de scăzătoare elementare, fie prin metoda adunării complementului față de 2, dacă se dispune numai de sumatoare elementare. Se poate enunța următoarea regulă de scădere:

> Pentru scăderea unui număr (scăzător) dintr-un altul (descăzut), se calculează complementul față de 2 al scăzătorului şi se efectuează adunarea acestuia la descăzut.

Se consideră următorul exemplu, în care s-a notat cu D descăzutul, cu S scăzătorul, iar cu S' complementul față de 2 al scăzătorului.

```
D     + 7        0 0111
S     - 4        1 1100
S′    -(-4)      0 0100

D     + 7        0 0111
S′    + 4        0 0100
      +11        0 1011
```

Regula pentru detectarea depăşirii poate fi enunțată astfel:

> La scăderea a două numere de semne contrare apare depăşire dacă şi numai dacă rezultatul are acelaşi semn cu scăzătorul.