

- indeksu, który przechowywany jest w rejestrze o numerze określonym w części adresowej rozkazu; ten drugi modyfikator używany jest tylko przy pewnych typach rozkazów (np. przy wybieraniu wektora z tablicy dwuwymiarowej) i niektórych komputerach.

W komputerach JS RIAD oraz IBM 360/370 występują następujące składniki adresu rzeczywistego (rys.7.1):

- przesunięcie, które jest 12-bitową liczbą zawartą w polu D rozkazu; umożliwia ono adresowanie obszaru pamięci o wielkości 4096 bajtów, rozpoczynającego się od adresu bazowego,

- adres bazowy, który jest 24-bitową liczbą zawartą w rejestrze stałoprzecinkowym o numerze określonym przez pole B rozkazu,

- indeks, który jest także 24-bitową liczbą zawartą w rejestrze, określonym przez pole X (występujące tylko w rozkazach typu RX).

W komputerach serii ODRA 1300 istnieje 5 różnych rodzajów modyfikacji: indeksowa, pośrednia, względna, znakowa i bazowa. Przy modyfikacji indeksowej adres rzeczywisty tworzymy jako sumę:

- adresu pierwotnego N (12-bitowego),
- zawartości jednego z trzech rejestrów indeksowych, oznaczonych przez M1, M2, M3 (lub X1, X2, X3) i wybranego wg zawartości 2-bitowego pola M; jeśli zawartością pola M jest 0, adres rzeczywisty jest równy pierwotnemu.

## 7.2. Reprezentacja danych w maszynach cyfrowych

Ze względu na stosowane rozwiązania techniczne maszyn cyfrowych, liczby oraz znaki alfanumeryczne (czyli znaki alfabetu, cyfry i znaki specjalne) najwygodniej jest reprezentować w maszynie cyfrowej za pomocą łańcuchów (stałej lub zmiennej długości), zer i jedynek, czyli po prostu łańcuchów bitów.

### 7.2.1. Reprezentacja liczb. Pozycyjne systemy liczenia

Danej liczbie można przyporządkować w sposób jednoznaczny dowolną kombinację zer i jedynek <sup>\*)</sup>, ale w takim przypadku reguły wykonywania działań na tych liczbach byłyby bardzo złożone. Dlatego też w codziennej praktyce korzystamy z tzw. systemów liczenia, czyli zbioru reguł tworzenia i zapisu liczb. Interesować nas będą tylko tzw. pozycyjne systemy liczenia, to znaczy takie systemy, w których każdej cyfrze danej liczby przyporządkowana jest stała wartość liczbowa zależna tylko od położenia tej cyfry w liczbie. W szczególności rozważone będą takie systemy pozycyjne, w których liczbę  $L$  przedstawiamy jako sumę odpowiednich całkowitych wielokrotności (wskazywanych przez cyfry) całkowitych potęg liczby całkowitej  $N$  (różnej od 0 i  $\pm 1$ ), zwanej podstawą systemu liczenia (ang. radix, base of notation).

A więc zgodnie z powyższym rozwinięciem liczby  $L$ , reprezentowanej przez ciąg cyfr  $C_n C_{n-1} \dots C_1 C_0 C_{-1} C_{-2} \dots C_m$  w systemie z podstawą  $N$ , jest następująca suma:

$$L = \sum_{i=-m}^n C_i N^i.$$

Systemami najczęściej spotykanymi w praktyce komputerowej są systemy: dwójkowy ( $N=2$ ), ósemkowy (oktalny) ( $N=8$ ), dziesiętny ( $N=10$ ) oraz szesnastkowy (heksagonalny, heksadecymalny) ( $N=16$ ). W tablicy 7.1 przedstawiono cyfry  $C_i$  używane w wymienionych powyżej systemach.

Tablica 7.1

Podstawa systemu liczenia $N$	Stosowane cyfry $C_i$
2	0,1
8	0,1,2,3,4,5,6,7
10	0,1,2,3,4,5,6,7,8,9
16	0,1,2,3,4,5,6,7,8,9, A,B,C,D,E,F

<sup>\*)</sup>Należy pamiętać o tym, że mając do dyspozycji łańcuch  $n$  bitów można z jego pomocą przedstawić  $2^n$  różnych liczb lub znaków.

### Przykład 7.1

Poniżej podamy przykłady liczb w wymienionych systemach liczenia, ich rozwinięcia oraz algorytm przechodzenia z systemu dwójkowego w ósemkowy i szesnastkowy.

$$11010_2 = 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 26_{10},$$

$$32_8 = 3 \cdot 8^1 + 2 \cdot 8^0 = 26_{10},$$

$$1A_{16} = 1 \cdot 16^1 + 10 \cdot 16^0 = 26_{10}.$$

Chcąc zamienić liczbę dwójkową  $L_2$  w ósemkową liczbę  $L_8$  (liczbę szesnastkową  $L_{16}$ ), stosujemy następujący prosty algorytm:

1) dzielimy liczbę dwójkową  $L_2$  od końca na grupy 3-bitowe, tzw. triady (grupy 4-bitowe, tzw. tetrazy), przy czym brakujące na początku bity uzupełniamy zerami,

2) każdą triadę (tetradę) zastępujemy odpowiednią cyfrą ósemkową (szesnastkową) zgodnie z rozwinięciem liczby dwójkowej tworzącej triadę (tetradę).

### Przykład 7.2

Zamienimy poniższą liczbę dwójkową na ósemkową i szesnastkową:

$$\underbrace{1\ 1\ 0}_6 \underbrace{1\ 0\ 1}_5 \underbrace{0\ 1\ 1}_3_2 = 653_8 = 6 \cdot 8^2 + 5 \cdot 8^1 + 3 \cdot 8^0 = 427_{10}.$$

$$\underbrace{0\ 0\ 0\ 1}_1 \underbrace{1\ 0\ 1\ 0}_A \underbrace{1\ 0\ 1\ 1}_B_2 = 1AB_{16} =$$

$$= 1 \cdot 16^2 + 10 \cdot 16^1 + 11 \cdot 16^0 = 427_{10}.$$

## **7.2.2. Liczby stałoprzecinkowe i zmiennoprzecinkowe**

Liczbami stałoprzecinkowymi nazywamy liczby, w których położenie przecinka jest umowne, ale nie ulega zmianie.

W reprezentacji binarnej liczby wyróżniamy tzw. bit znaku. Jest to najczęściej najbardziej znaczący (skrajny lewy) bit liczby dwójkowej, przy czym na ogół "0" oznacza liczbę dodatnią, zaś "1" - liczbę ujemną. Jeśli umieścimy przecinek bezpośrednio za bitem znaku, mamy do czynienia z ułamkiem właściwym; jeśli natomiast umieścimy przecinek po ostatnim, najmniej znaczącym bicie, mamy liczbę całkowitą.

Jeśli mamy do dyspozycji  $n$  pozycji bitowych, to liczby całkowite, które można przedstawić  $n$ -pozycyjnym łańcuchem bitów, należą do przedziału  $\langle -2^{n-1}, 2^{n-1} - 1 \rangle$ . Istnieje wiele sposobów przedstawiania liczb ujemnych w maszynie cyfrowej. Najprostszym sposobem reprezentacji jest metoda "znak-moduł", polegająca na tym, że łańcuch bitów reprezentujący liczbę ujemną różni się tylko bitem znaku od łańcucha reprezentującego liczbę dodatnią o tej samej wartości bezwzględnej. Inne metody reprezentacji liczb ujemnych są omówione w wielu podręcznikach m.in. [2], [12], [16], zaś metoda "uzupełnienie do dwójki" będzie przedstawiona poniżej przy omawianiu reprezentacji liczb w systemie IBM 360/370 oraz JS RIAD.

Liczbę zmiennoprzecinkową  $L$  wyrażamy w następującej postaci:

$$L = m \cdot N^w,$$

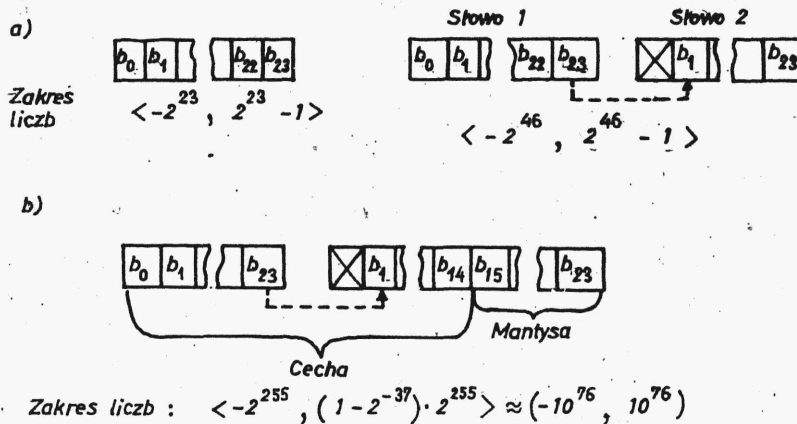
gdzie:

- $w$  - liczba całkowita, zwana cechą lub wykładnikiem,
- $m$  - liczba ułamkowa, zwana częścią ułamkową lub mantysą,
- $N$  - podstawa stosowanego systemu liczenia.

Łatwo zauważyć, że danej liczbie  $L$  mogą odpowiadać różne pary  $(m;w)$ , np.  $(0,23 ; -1)$ ,  $(0,023 ; 0)$  dla liczby  $0,023$ . Na ogół liczby zmiennoprzecinkowe przedstawiane są w postaci znormalizowanej, czyli takiej postaci, aby zawsze liczba  $L$  była reprezentowana przez dokładnie jedną parę  $(m;w)$ . W tym celu wartość mantysy sprowadzamy do odpowiedniego zakresu, na przykład  $\langle 1/N, 1 \rangle$ . Normalizacja liczb zmiennoprzecinkowych ułatwia ich porównywanie, ponieważ mantysy porównujemy dopiero wtedy, gdy wykładniki są równe.

### 7.2.3. Reprezentacja liczb w komputerach serii ODRA 1300

Liczby stałoprzecinkowe mogą być pojedynczej lub podwójnej długości i zajmują odpowiednio jedno lub dwa słowa maszynowe 24-bitowe, przy czym w obu przypadkach bitem znaku jest bit zerowy  $b_0$  pierwszego słowa, zaś w liczbach podwójnej długości wykorzystane są dodatkowo bity  $b_1 - b_{23}$  drugiego słowa (rys.7.2a).



Rys.7.2. Reprezentacja liczb w komputerach serii ODRA 1300:  
a) liczby stałoprzecinkowe pojedyncze i podwójne, b) liczby zmiennoprzecinkowe

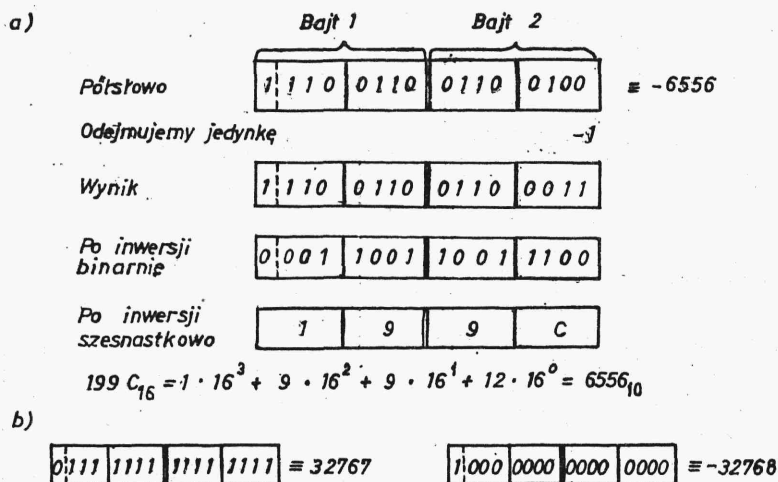
Liczby zmiennoprzecinkowe są postaci  $L = m \cdot 2^w$  i zajmują po dwie komórki pamięci, przy czym na mantysę  $m$  przeznaczone jest 38 bitów (wraz z bitem znaku), zaś na wykładnik  $w$  - 9 bitów (wraz ze znakiem). Ze względu na normalizację wartość mantysy  $m$  należy do jednego z przedziałów  $\langle -1, -\frac{1}{2} \rangle$  lub  $\langle \frac{1}{2}, 1 \rangle$ , zaś wartości wykładnika  $w$  do przedziału  $\langle -256, 255 \rangle$  (rys.7.2b).

### 7.2.4. Reprezentacja liczb w komputerach IBM System 360/370 oraz JS RIAD

Liczby stałoprzecinkowe mogą zajmować półsłowa (16 bitów = 2 bajty), słowa (32 bity = 4 bajty) oraz podwójne słowa

(64 bity = 8 bajtów). Lewy skrajny bit jest bitem znaku, przy czym zgodnie ze wspomnianą powyżej konwencją "0" oznacza liczbę dodatnią, zaś "1" - ujemną. Do reprezentacji liczb ujemnych stosowana jest metoda "uzupełnienia do dwójki". Przy zamianie ujemnej liczby dwójkowej (tzn. skrajny lewy bit liczby dwójkowej zawiera "1") na ujemną dziesiętną wykonywany jest następujący algorytm:

- 1) odejmujemy od skrajnego prawego bitu łańcucha jedynkę,
- 2) wykonujemy inwersję wszystkich bitów łańcucha ("0" → "1", "1" → "0"),
- 3) rozwinięcie otrzymanej w kroku 2 liczby binarnej da nam wartość bezwzględną szukanej liczby ujemnej (rys.7.3).



Rys.7.3. Reprezentacja liczb stałoprzecinkowych w komputerach JS RIAD i IBM 360/370: a) konwersja liczby binarnej w ujemną liczbę dziesiętną, b) maksymalne (co do wartości bezwzględnej) liczby o długości półsłowa

Liczby zmiennoprzecinkowe są reprezentowane w szesnastkowym systemie liczenia, czyli są wyrażane w następującej postaci:

$$L = (m)_{16} \cdot 16^w,$$

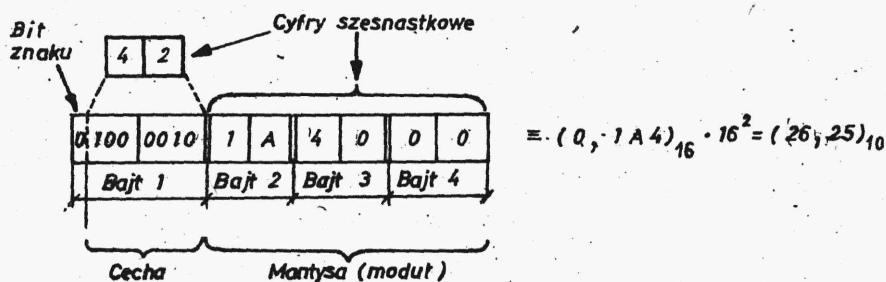
gdzie m jest znormalizowaną mantysą.

### Przykład 7.3

$$(26,25)_{10} = (1A,4)_{16} \quad \text{przed normalizacją,}$$

$$(26,25)_{10} = (0,1A4)_{16} \cdot 16^2 \quad \text{po normalizacji.}$$

Mantysa liczb zmiennoprzecinkowych jest ułamkiem składającym się z 6 cyfr szesnastkowych (24 bity)\* w postaci "znak-moduł". Wykładnik w zapisany jest przy użyciu 7 bitów, więc jego wartość zawiera się w przedziale  $\langle 0, 127 \rangle$ . Aby umożliwić reprezentację bardzo małych i bardzo dużych wartości przyjęto, że wykładnik w interpretacji programowej zawiera się w przedziale  $\langle -64, +63 \rangle$ . A więc chcąc zapisać w pamięci maszyny liczbę znormalizowaną szesnastkową musimy do wykładnika dodać  $64_{10} = 40_{16}$ . Na rys.7.4 przedstawiona jest wewnętrzna reprezentacja liczby  $(26,25)_{10} = (0,1A4)_{16} \cdot 16^2$  o znormalizowanej części ułamkowej. Wartości liczb przedstawionych w powyższy sposób należą do przedziału  $(5,4 \cdot 10^{-79}; 7,2 \cdot 10^{75})$ .



Rys.7.4. Reprezentacja liczby dodatniej zmiennoprzecinkowej w JS RIAD oraz IBM 360/370.

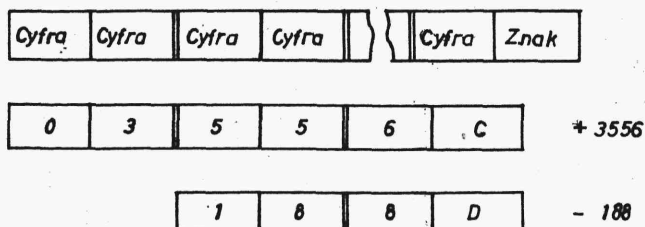
W systemie IBM 360/370 stosowana jest także tzw. arytmetyka dziesiętna oraz liczby dziesiętne kodowane dwójkowo (ang. Binary Coded Decimal = BCD). Arytmetyka ta stosowana jest

\*) Dotyczy to liczb zmiennoprzecinkowych pojedynczej precyzji, które różnią się od tzw. liczb podwójnej precyzji dłuższą mantysą. Identyczna sytuacja ma miejsce w komputerach ODRA 1300.

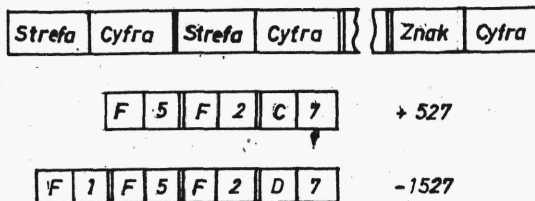
do tych procedur przetwarzania danych, w których występuje niewiele obliczeń, ale dużo jest danych wejściowych oraz danych wyjściowych. W takich przypadkach czasochłonny jest proces konwersji danych z postaci dziesiętnej w binarną i odwrotnie. Liczby dziesiętne reprezentowane są w postaci liczb całkowitych ze znakiem. Liczby ujemne kodowane są w sposób naturalny z użyciem znaku minus, czyli w postaci "znak-moduł". Cyfry dziesiętne 0-9 kodowane są binarnie przy użyciu 4 bitów (0000-1001), zaś pozostałe kody (1010-1111) reprezentują znaki liczb: plus i minus (np. cyfry szesnastkowe C lub F reprezentują "plus", zaś D - "minus").

W argumentach i wynikach dziesiętnych przedstawionych w postaci liczb dziesiętnych zakodowanych dwójkowo, cyfry są upakowane po dwie w jednym bajcie (rys.7.5a). Tak upakowane cyfry tworzą w PAO pola o zmiennej długości. W prawej połowie ostatniego bajtu pola znajduje się znak zakodowanej w tym polu liczby dziesiętnej. Pola mogą mieć długość od 1 do 16 bajtów, czyli liczby dziesiętne mogą zawierać minimalnie liczbę jednocyfrową ze znakiem, maksymalnie liczbę 31-cyfrową ze znakiem.

a)



b)



Rys.7.5. Reprezentacja liczb dziesiętnych kodowanych dwójkowo w postaci: a) upakowanej, b) rozpakowanej

Liczby dziesiętne przesyłane pomiędzy PAO i urządzeniami we/wy (np. czytnik kart, drukarka) zakodowane są w postaci roz-



pakowanej, czyli jedna cyfra w każdym bajcie. Liczba w postaci rozpakowanej ma w lewej połowie bajtu tzw. kod strefy, zaś w prawej - kod cyfry (rys.7.5b). Znak liczby zakodowany jest w lewej połowie ostatniego bajtu. Liczby dziesiętne w postaci rozpakowanej nie są używane w działaniach arytmetycznych.

### **7.2.5. Reprezentacja danych alfanumerycznych w komputerach IBM System 360/370 i JS RIAD**

Do danych alfanumerycznych, zwanych dalej znakami alfanumerycznymi, zaliczamy litery danego alfabetu, cyfry oraz znaki specjalne jak np. + ? / : itd. Znaki, podobnie jak omawiane powyżej liczby, są reprezentowane przez łańcuchy bitów. Łańcuchy znaków są przechowywane w PAO w kolejnych komórkach pamięci.

W wielu obecnie stosowanych systemach komputerowych stosuje się kombinacje 6-bitowe do przedstawienia znaków alfanumerycznych, czyli przy ich użyciu można zakodować 64 różne symbole, a więc wszystkie litery, cyfry oraz najczęściej stosowane znaki specjalne. W komputerach ODRA 1300 w jednym słowie maszynowym można upakować 4 znaki 6-bitowe. Przy transmisji do urządzeń dopuszczających więcej znaków niż 64 (np. czytnik i perforator taśmy 8-ścieżkowej) używamy tzw. kodów przesunięć, na które przeznaczamy trzy znaki z podstawowych 64. W ten sposób można przedstawić  $3 \cdot (64 - 3) = 183$  znaki.

W systemie IBM 360/370 oraz JS RIAD używamy jednego bajtu, czyli 8 bitów, do reprezentacji jednego znaku. A więc można w ten sposób reprezentować  $2^8 = 256$  różnych znaków. Zestawy znaków w większości urządzeń drukujących zawierają od 39 do 88 symboli, mianowicie 10 cyfr dziesiętnych, 26 liter oraz najważniejsze znaki specjalne. Kombinacje bitów nie używane do kodowania znaków alfanumerycznych stosowane są do sterowania transmisją z/do urządzeń zewnętrznych.