

AUTOMATYCZNE PRZETWARZANIE INFORMACJI	N O R M A   B R A N Ż O W A	<b>BN-89</b>
	Przetwarzanie informacji i komputery <b>Techniki rozszerzania kodów 7- i 8-bitowego</b>	<b>3101-06</b>
		Zamiast BN-76/3101-06
		Grupa katalogowa 1960

BN-89/3101-06 (eqv CT CЭB 360-86)

**PRZEDMOWA**

Przedmiotem niniejszej normy są techniki rozszerzania kodów 7- i 8-bitowego z zastosowaniem znaków rozszerzania kodu wg PN-89/T-42108. Norma obejmuje także strukturę kodów podlegających rozszerzaniu, relacje między kodami 7- a 8-bitowym oraz klasyfikację technik rozszerzania wraz ze strukturą poszczególnych klas.

Norma ma zastosowanie podczas wymiany informacji i jej przetwarzania w komputerach, urządzeniach wejściowo-wyjściowych, urządzeniach przygotowania, zdalnego przetwarzania i transmisji danych, systemach terminali, oprogramowaniu i środkach systemów przetwarzania informacji.

Norma stanowi polską wersję normy CT CЭB 360-86.

Zauważone w tekście źródłowym błędy skorygowano w oparciu o ISO 2022 (1986).

W celu dostosowania normy do warunków krajowych wprowadzono następujące odstępstwa od oryginału:

a) część wstępną bez tytułu, w której omówiono przedmiot i zakres stosowania normy, umieszczono w pierwszych zdaniach niniejszej Przedmowy.

b) odsyłacze do norm CT CЭB zamieniono na odsyłacze do odpowiednich Polskich Norm.

c) ponumerowano załączniki wg kolejności ich wystąpienia nie wyodrębniając tzw. załączników informacyjnych.

d) rosyjskie symbole znaków funkcyjnych (w tym znaków rozszerzania kodu, znaku spacji i kasowania) zastąpiono symbolami ISO w języku angielskim; oryginalne symbole są zamieszczone w PN-89/T-42108, w której podano symbolikę ISO i RWPG.

e) ilustracje rysunkowe (rys. 1 ÷ 7) przyjęto wg angielskiej wersji normy (ISO 2022 (1986)).

f) ze względu na uzyskanie czytelności tablic pominięto w nich pełne nazwy znaków funkcyjnych wszędzie tam, gdzie one występują; nazwy te wraz z określeniami są zawarte w PN-89/T-42108.

g) w załączniku 3 p. 24 zamiast odsyłacza wprowadzono pełne określenie funkcji sterującej.

Przedmowa oraz Informacje dodatkowe stanowią krajowe uzupełnienie normy.

**NORMA MIĘDZYNARODOWA CT CЭB 360-86****1. POSTANOWIENIA OGÓLNE**

**1.1. Założenia podstawowe.** Struktura, nazwy, określenia znaków i pozycji tablicy kodu 7-bitowego, przeznaczonych do rozszerzania, powinny być zgodne z PN-88/T-42109/01<sup>1)</sup>, PN-84/T-42109/02 i PN-86/T-42109/03. Analogiczna zasada obowiązuje w stosunku do kodu 8-bitowego, który powinien być zgodny z PN-79/T-42112/01<sup>2)</sup>.

Klasyfikacja, nazwy, symbole i określenia znaków alfanumerycznych — wg PN-89/T-42108<sup>3)</sup>.

**1.2. Cel rozszerzania kodu.** Podane w normach wymienionych w 1.1 kody umożliwiają reprezentację zbiorów

zawierających odpowiednio do 128 znaków (kod 7-bitowy) i do 256 znaków (kod 8-bitowy). Zgodnie z tymi normami można reprezentować także inne znaki graficzne łącząc dwa lub większą liczbę znaków ze znakami funkcyjnymi COFNIĘCIE (BS) lub POWRÓT KARETKI (CR). Jeśli utworzone zgodnie z tymi normami zbiory znaków nie zawierają wystarczającej liczby funkcji sterujących lub znaków graficznych, to wymagania te mogą być spełnione dzięki zastosowaniu technik rozszerzania, określonych w niniejszej normie.

Podane w niniejszej normie zasady mogą być wykorzystane także do tworzenia dodatkowych środków rozszerzenia kodu, np. przy formułowaniu niektórych parametrów funkcji sterujących. Środki te jednak nie są przedmiotem tej normy.

**1.3. Rozszerzane struktury kodów.** Przyjmując za punkt wyjścia podstawowe tablice kodowe wg PN-88/

<sup>1)</sup> W oryginale: CT CЭB 356-86.

<sup>2)</sup> W oryginale: CT CЭB 358-76.

<sup>3)</sup> Odpowiednikiem PN-89/T-42108 jest norma CT CЭB 359-86.

Zgłoszona przez Instytut Maszyn Matematycznych  
Ustanowiona przez Dyrektora Instytutu Maszyn Matematycznych dnia 30 listopada 1989 r.  
jako norma obowiązująca od dnia 1 lipca 1990 r.  
(Dz. Norm. i Miar nr 2/1990, poz. 3)

T-42109/01<sup>1)</sup> (dla kodu 7-bitowego) oraz wg PN-79/T-42112/01<sup>2)</sup> (dla kodu 8-bitowego) w niniejszej normie opisano różne środki rozszerzania zbiorów funkcji sterujących i zbiorów znaków graficznych kodu oraz przedstawiono następujące struktury i metody tworzenia kodów rozszerzonych, związanych z kodami 7- i 8-bitowym:

- 1) rozszerzanie kodu 7-bitowego, wykonywane w jego środowisku,
- 2) strukturę rodziny kodów 8-bitowych kompatybilną ze strukturą kodu 7-bitowego,
- 3) rozszerzanie kodu 8-bitowego wykonywane w jego środowisku,
- 4) relacje między kodami 7- i 8-bitowym.

**1.4. Korzyści wynikające ze znormalizowania metod rozszerzania kodu.** Znormalizowane metody rozszerzania kodu powinny być stosowane w celu zapewnienia identycznego sposobu rozszerzania kodu we wszystkich przypadkach omówionych w 1.3, a także dla ułatwienia ich możliwych modyfikacji. Korzyści są następujące:

- 1) zmniejszenie prawdopodobieństwa sprzeczności informacji między systemami współpracującymi,
- 2) zapewnienie warunków dla rozszerzania kodu na etapie projektowania systemu,
- 3) możliwość opracowania znormalizowanych metod przywoływania uzgodnionych zbiorów znaków,
- 4) zapewnienie wymiany informacji między środowiskami kodów 7- i 8-bitowego itp.

**1.5. Wymagany rodzaj przetwarzania danych.** Metody rozszerzania kodu są przeznaczone do stosowania przy szeregowym przetwarzaniu danych w kierunku do przodu. Wykorzystywanie tych metod w stosunku do danych przetwarzanych w inny sposób lub do danych sformatowanych w celu przetwarzania rekordów o stałej długości może spowodować niepożądane rezultaty lub wymagać dodatkowego specjalnego przetwarzania w celu zapewnienia poprawnej interpretacji.

**1.6. Warunki spełnienia wymagań dotyczących metod rozszerzania kodu.** W niniejszej normie podano zbiór możliwych metod rozszerzania kodu. W stosunku do odrębnych urządzeń, środków, czy systemów można dokonać wyboru spośród nich. W dokumentacji urządzeń, środków i systemów powinien być określony podzbiór wybranych metod rozszerzania kodu. Wykorzystany podzbiór tych metod będzie zgodny z niniejszą normą, jeśli będą spełnione następujące wymagania:

- 1) metody rozszerzania kodu, opisane w niniejszej normie, powinny być realizowane przez określone w niej funkcje sterujące, którym przydzielono symbole i reprezentacje kodowe,
- 2) nie należy wykorzystywać reprezentacji kodowych, zarezerwowanych dla przyszłej normalizacji i rejestracji, których znaczenia nie zostały określone,
- 3) nie należy stosować żadnych sekwencji ESC o znaczeniach innych niż zarejestrowane,
- 4) w przypadku gdy dwa systemy stosujące różne podzbiory wybranych metod rozszerzania kodu mają

komunikować się ze sobą, powinny stosować dla celów tej komunikacji tylko te metody rozszerzania kodu, które są dla nich wspólne.

**1.7. Terminy i definicje** podano w załączniku 3.

**1.8. Przydzielanie kombinacji kodowych.** Kombinacje kodowe przeznaczone do przypisania określonym zbiorom znaków lub określonym funkcjom sterującym dla potrzeb ich przywoływania lub oznaczania powinny być wykorzystywane zgodnie z procedurami rejestracji. Informacje na ten temat podano w załączniku 4.

## 2. FORMY ZAPISU

W niniejszej normie zgodnie z PN-88/T-42109/01<sup>1)</sup> i PN-79/T-42112/01<sup>2)</sup> przyjęto oznaczenia pozycji tablic kodowych, cmówione w załączniku 5.

## 3. ROZSZERZANIE KODU 7-BITOWEGO W JEGO ŚRODOWISKU

### 3.1. Wprowadzenie

**3.1.1. Struktura kodu 7-bitowego.** Tablica kodu 7-bitowego zgodna z PN-88/T-42109/01<sup>1)</sup>, podana w załączniku 6, jest podstawą dla metod rozszerzania kodu, wykorzystywanych w stosunku do zbiorów znaków w kodzie 7-bitowym, określonych w PN-88/T-42109/01<sup>1)</sup>, PN-84/T-42109/02 i PN-86/T-42109/03.

**3.1.2. Rozszerzanie przez zamianę.** Jeśli zbiory znaków zawarte w normach wymienionych w 3.1.1 nie spełniają wymagań danego zastosowania, to wymagania te mogą być spełnione przez wykorzystanie kodu o podobnej strukturze, ale zawierającego niektóre znaki inne. Taka zamiana powinna być traktowana jako opracowanie nowego kodu, nie będącego przedmiotem żadnej z norm podanych w 3.1.1.

**3.1.3. Rozszerzanie przez zwiększanie zestawów znaków.** Postanowienia niniejszej normy umożliwiają zwiększanie zestawu znaków o znaki dodatkowe względem 128 znaków zawartych w tablicach kodu 7-bitowego, określonych w normach wg 3.1.1. Zestawy znaków mogą być zwiększane dzięki zastosowaniu:

- 1) pojedynczych funkcji sterujących,
- 2) zbiorów składających się z 32 funkcji sterujących,
- 3) zbiorów zawierających 94 znaki graficzne,
- 4) zbiorów zawierających 96 znaków graficznych,
- 5) zbiorów o więcej niż 94 lub 96 znakach graficznych, reprezentowanych przez więcej niż 1 bajt, tzw. zbiorów wielobajtowych.

Dodatkowe zbiory znaków wymienione w 3) ÷ 5) nie powinny zawierać znaków SPACJA (SP), KASOWANIE (DEL) i żadnych znaków funkcyjnych.

**3.1.4. Elementy rozszerzania kodu.** W wielu zastosowaniach wymagane są kombinacje podanych w 3.1.3 środków rozszerzania kodu. Na rys. 1 przedstawiono elementy rozszerzania kodu wraz z odpowiadającymi im oznaczeniami:

<sup>1)</sup> W oryginale: CT C9B 356-86.

<sup>2)</sup> W oryginale: CT C9B 358-86.

1) C0 — zbiór zawierający 32 znaki funkcyjne (kolumny 0 i 1 tablicy kodu).

2) C1 — zbiór dodatkowy zawierający 32 funkcje sterujące.

3) dodatkowe jednostkowe funkcje sterujące.

4) G0 — zbiór złożony z 94 znaków graficznych (kombinacje kodowe pozycji 2/1 ÷ 7/14); zbiór wielobajtowy może występować także jako G0,

5) G1, G2, G3 — zbiory dodatkowe zawierające 94 znaki graficzne (kombinacje kodowe pozycji 2/1 ÷ 7/14) lub 96 znaków graficznych (kombinacje kodowe pozycji 2/0 ÷ 7/15); zbiory wielobajtowe mogą być wykorzystywane także jako G1, G2, G3.

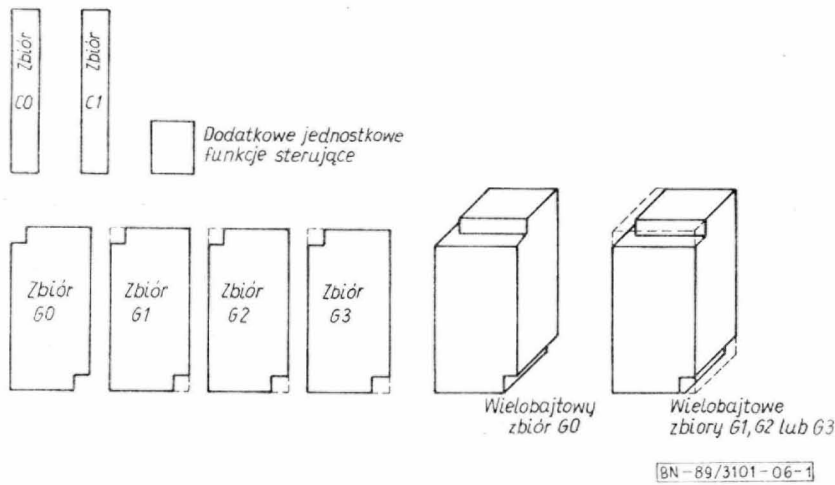
Zaleca się wykorzystywanie znormalizowanych zbiorów znaków, podanych w normach wg 3.1.1 jako zbiorów odpowiednio C0 i G0.

niż łaciński, zawierających także znaki alfabetu łacińskiego).

3) inne kody 7-bitowe o strukturze wg 3.1.1, mogące zawierać 94 znaki graficzne, 96 znaków graficznych i (lub) zbiory wielobajtowe w kolumnach 2 ÷ 7. W celu zapewnienia możliwości rozszerzania kodu, przedstawionych w niniejszej normie, należy pozostawić bez zmian znaczenia funkcje i pozycje w tablicy kodu znaków ESC, SI oraz SO.

**3.1.6. Znaki rozszerzania kodu wg PN-88/T-42109/01<sup>1)</sup>**. W powyższej normie występują następujące znaki podstawowego zbioru C0, przeznaczone do rozszerzania kodu; ESC, SO, SI, DLE, określone wg PN-89/T-42108<sup>2)</sup>.

W niniejszej normie nie opisano zastosowania znaku DLE zarezerwowanego dla zapewnienia dodatkowych



Rys. 1. Elementy rozszerzania kodu

**3.1.5. Poziomy zgodności.** Dla potrzeb wymiany informacji są ustalane różne poziomy zgodności, które mogą być przestrzegane podczas stosowania środków rozszerzania kodu. Wyodrębnia się trzy poziomy:

1) wersja kodu zgodna z zestawem podstawowym wg PN-88/T-42109/01<sup>1)</sup>,

2) wariant kodu 7-bitowego zgodny z zestawem podstawowym wg PN-88/T-42109/01<sup>1)</sup>, spełniający następujące wymagania:

a) w kolumnach 0 i 1 występują tylko znaki funkcyjne.

b) dziesięć znaków sterowania transmisją oraz znaki NUL, SO, SI, CAN, SUB, ESC, SP, DEL pozostają na tych samych miejscach w tablicy i nie zmieniają znaczeń.

c) znaki SPACJA (SP), KASOWANIE (DEL) i jeden lub dwa zbiory o 94 znakach graficznych są umieszczone w kolumnach 2 ÷ 7; nie są wykorzystywane zbiory o 96 znakach graficznych, zbiory wielobajtowe i funkcje przesunięcia, z wyjątkiem funkcji SI oraz SO.

d) znaki graficzne wg zestawu podstawowego zawartego w PN-88/T-42109/01<sup>1)</sup> nie są przemieszczane na inne pozycje (ta zasada nie dotyczy alfabetów innych

funkcji sterujących, mających zastosowanie w transmisji danych).

**3.1.7. Pozostałe znaki rozszerzania kodu.** W PN-89/T-42108<sup>2)</sup>, oprócz znaków podanych w 3.1.6, określono i przedstawiono do wykorzystania w środowisku kodu 7-bitowego następujące dodatkowe sterujące funkcje przesunięć: LS2, LS3, SS2 i SS3. Sposoby ich kodowania opisano w załączniku 1.

Dodatkowe funkcje sterujące przeznaczone do wykorzystania w środowisku kodu 8-bitowego, a mianowicie funkcje LS1R, LS2R oraz LS3R wg PN-89/T-42108<sup>2)</sup> (zobacz 5.2.1) są używane w środowisku kodu 7-bitowego tylko do konwersji danych ze środowiska 7-bitowego na 8-bitowe i na odwrót (zobacz 7.2 i 7.4). Rezultaty działania funkcji sterujących LS1R, LS2R i LS3R w kodzie 7-bitowym są analogiczne, jak rezultaty działania funkcji odpowiednio SO, LS2 i LS3.

**3.1.8. Znaki graficzne złożone.** Dla zbiorów znaków zawierających dodatkowe znaki graficzne reprezentowane przez połączenie dwóch lub więcej znaków graficznych na tej samej pozycji, np. litery ze znakami diakrytycznymi, przewiduje się, że:

<sup>1)</sup> W oryginale: CT CЭB 356-86.

<sup>2)</sup> Odpowiednikiem PN-89/T-42108 jest norma CT CЭB 359-86.

1) znaki graficzne powodujące przemieszczenie czynnej pozycji o jedną pozycję znakową do przodu (tzw. znaki spacji) są używane w połączeniu ze znakiem BS lub CR.

2) znaki graficzne nie powodujące takiego przemieszczenia (tzw. znaki niespacji) są używane w połączeniu ze spacującymi znakami graficznymi.

Zgodnie z PN-88/T-42109/01<sup>1)</sup> zezwala się na wykorzystywanie pierwszej z tych metod do reprezentowania liter ze znakami diakrytycznymi (pierwszy lub drugi poziom zgodności wg 3.1.5). Druga metoda jest przeznaczona dla trzeciego poziomu zgodności (zobacz 3.1.5). Twórcy zbiorów graficznych przeznaczonych do zarejestrowania (zobacz załącznik 4) powinni jednoznacznie wskazać w każdym zbiorze znaki niespacji.

Nowa norma określająca zbiór znaków powinna zawierać postanowienie dotyczące ograniczeń w ich łączeniu, jeśli nie zostało to ustalone podczas rejestracji tego zbioru.

**3.2. Rozszerzanie zbioru znaków graficznych za pomocą funkcji przesunięć.** W środowisku kodu 7-bitowego są wykorzystywane następujące funkcje przesunięć: SO, SI, LS2, LS3, SS2, SS3, określone w PN-89/T-42108<sup>2)</sup>.

**3.2.1. Wykorzystywanie funkcji przesunięć blokujących.** W środowisku kodu 7-bitowego funkcje przesunięć blokujących SO, SI, LS2 oraz LS3 powinny być używane wyłącznie do rozszerzania zbioru znaków graficznych.

Każda z funkcji przesunięć SO, LS2 lub LS3 przywołuje dodatkowy zbiór 94- lub 96-znakowy, odpowiednio: G1, G2, G3. Zbiór 94-znakowy jest przywoływany na pozycje  $2/1 \div 7/14$ , znak SPACJA (SP) — na pozycję 2/0, a znak KASOWANIE (DEL) — na pozycję 7/15. Zbiór 96-znakowy jest przywoływany na pozycje  $2/0 \div 7/15$  tablicy kodu. Dozwolone jest pozostawienie na niektórych pozycjach miejsc pustych, tzn. bez znaków graficznych. Nie jest wymagane, z wyjątkiem przypadków omówionych niżej, aby wszystkie znaki graficzne zbioru dodatkowego różniły się od znaków zbioru wcześniej przywołanego.

Funkcja przesunięcia SI powinna przywoływać 94 znaki graficzne zbioru G0 (na pozycje  $2/1 \div 7/14$ ) i sprawiać, aby pozycje 2/0 i 7/15 przyjmowały swe zwykle znaczenie odpowiednio SPACJI (SP) i KASOWANIA (DEL).

Jeśli zbiór został już przywołany, powtórne użycie odpowiadającej mu funkcji przesunięcia nie daje żadnego rezultatu.

Wystąpienie funkcji przesunięcia blokującego nie powinno mieć wpływu na znaczenia następujących kombinacji kodowych:

- 1) reprezentujących znaki funkcyjne w kolumnach 0 i 1;
- 2) występujących w dowolnej sekwencji ESC;
- 3) kombinacji kodowej występującej bezpośrednio po SS2 lub SS3.

Znaki SPACJA (SP) i KASOWANIE (DEL) powinny być umieszczone tylko na pozycjach odpowiednio 2/0 i 7/15 wtedy i tylko wtedy, gdy przywołany zbiór jest 94-znakowy; znaki te nie powinny być umieszczane na innych pozycjach w dowolnym zbiorze.

Jednak znaki inne niż SPACJA, lecz reprezentujące odstępy innych rozmiarów lub przeznaczeń, mogą występować na dowolnych pozycjach w dowolnym zbiorze znaków graficznych lub funkcyjnych.

Na początku wymiany informacji powinien być określony stan przesunięcia przez wykorzystanie jednej z funkcji przesunięć blokujących, jak to opisano w 3.4 (zobacz także rozdz. 6).

**3.2.2. Wykorzystywanie funkcji przesunięć jednostkowych.** Funkcje przesunięć jednostkowych SS2 i SS3 są używane wyłącznie do rozszerzania zbioru znaków graficznych. Funkcja SS2 powinna przywoływać jeden znak z ostatnio oznaczonego zbioru G2, funkcja SS3 — jeden znak z ostatnio oznaczonego zbioru G3.

Te przywołania zmieniają znaczenie jednej kombinacji kodowej występującej bezpośrednio po nich (zobacz 3.3.9) i przydzielają jej znaczenie odpowiedniej kombinacji kodowej ze zbioru G2 lub G3.

Dopuszcza się, aby po funkcji SS2 lub SS3 występowała tylko jedna z kombinacji kodowej z pozycji  $2/1 \div 7/14$  przy 94-znakowych zbiorach G2 lub G3 i z pozycji  $2/0 \div 7/15$  przy 96-znakowych zbiorach G2 lub G3 (zobacz 7.3). Użycie funkcji przesunięcia jednostkowego nie powinno wpływać na bieżący stan przesunięcia, ustalony przez funkcję przesunięcia blokującego.

**3.2.3. Wykorzystywanie nie więcej niż trzech dodatkowych zbiorów znaków graficznych.** W pewnych zastosowaniach nie wymaga się więcej niż trzech dodatkowych zbiorów 94- lub 96-znakowych, które mogą być jednoznacznie identyfikowane jako G1, G2 i G3. Zbiory te są oznaczane za pomocą odpowiednich sekwencji ESC, opisanych w 3.3.7 ÷ 3.3.10. Zgodnie z postanowieniami wg 3.4 dopuszcza się pominięcie tych sekwencji po uzgodnieniu między stronami uczestniczącymi w wymianie informacji. Dowolny z tych dodatkowych zbiorów może być wówczas przywołany za pomocą odpowiednich funkcji przesunięć.

**3.2.4. Wykorzystywanie więcej niż trzech dodatkowych zbiorów znaków graficznych.** Jeśli wymagane jest używanie więcej niż trzech dodatkowych zbiorów znaków graficznych lub więcej niż jednego takiego zbioru oznaczonego jako G0 lub G1, lub G2, lub G3, to zbiory  $G0 \div G3$  powinny być oznaczane za pomocą odpowiednich sekwencji ESC, opisanych w 3.3.7 ÷ 3.3.10. Każde kolejne użycie funkcji przesunięcia powinno przywołać odpowiedni, wcześniej oznaczony zbiór.

Aby oznaczyć zbiór jako G1, G2, G3 za pomocą sekwencji ESC, nie należy powracać do zbioru G0 za pomocą funkcji SI.

Użycie funkcji przesunięcia powinno przywoływać znaki graficzne zbioru oznaczonego jako ostatni dla wykorzystania na mocy tej funkcji, ale nie dopuszczalny jest wpływ tego na identyfikację dowolnych, wcześniej oznaczonych zbiorów. Oznaczony zbiór może być przywoływany dowolną liczbą razy przez ponowne użycia

<sup>1)</sup> W oryginale: CT CЭB 356-86.

<sup>2)</sup> Odpowiednikiem PN-89/T-42108 jest norma CT CЭB 359-86.

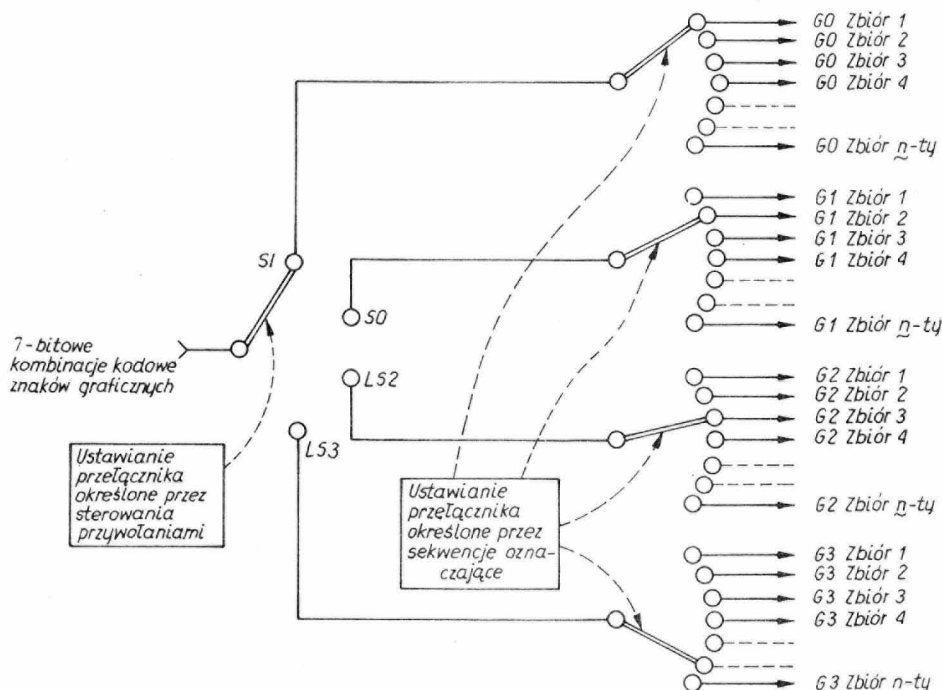


odpowiedniej funkcji przesunięcia aż do chwili zamiany zbioru przez inną oznaczającą sekwencję ESC.

Oznaczenie następnego zbioru znaków graficznych za pomocą sekwencji ESC nie powinno zmieniać bieżącego stanu przesunięcia.

Jeżeli zbiór znaków graficznych oznaczany za pomocą sekwencji ESC należy do klasy, która jest w danej chwili przywołana, nowy zbiór też powinien być przywołany.

Na rys. 2 przedstawiono schemat wyżej opisanych procesów oznaczania i przywoływania zbiorów.



BN-89/3101-06-2

Rys. 2. Wykorzystywanie więcej niż 3 zbiorów znaków graficznych za pomocą funkcji przesunięć blokujących

### 3.3. Rozszerzanie kodu za pomocą sekwencji ESC

**3.3.1. Przeznaczenie sekwencji ESC.** Sekwencje ESC umożliwiają wykorzystywanie jednostkowych funkcji sterujących lub ich zbiorów, nie będących w obu przypadkach funkcjami sterowania transmisją. Sekwencje te są ponadto używane do oznaczania zbiorów znaków graficznych, oznaczania innych zastosowań niektórych lub wszystkich kombinacji kodowych kodu 7-bitowego oraz do oznaczania zbiorów znaków w kodzie o liczbie bitów innej niż 7.

Przykładowo sekwencje ESC umożliwiają używanie nie występujących w kodzie:

- 1) jednostkowej funkcji sterującej,
- 2) zbioru funkcji sterujących,
- 3) zbioru znaków graficznych.

Umożliwiają także wykorzystywanie kodu o innej strukturze.

**3.3.2. Budowa sekwencji ESC.** Sekwencja ESC składa się z dwóch lub większej liczby kombinacji kodowych 7-bitowych. Pierwszą zawsze powinna być kombinacja reprezentująca znak ESC, a ostatnią — kombinacja re-

prezentująca znak finalny. Sekwencja ESC może także zawierać dowolną liczbę kombinacji kodowych reprezentujących znaki wewnętrzne.

Znaczenie sekwencji ESC powinno być określone przez kombinację kodową reprezentującą jej znak wewnętrzny lub znaki wewnętrzne, jeśli takie występują i przez kombinację reprezentującą jej znak finalny.

Jako wewnętrzne mogą być użyte znaki z kolumny 2 tablicy kodu (w liczbie 16). Mają one ogólny symbol litery I.

Jako finalne mogą występować znaki z kolumn 3 ÷ 7, z wyjątkiem pozycji 7/15. Odpowiada im ogólny symbol w postaci litery F.

**UWAGA.** Mimo że w niniejszej normie sekwencje ESC są opisywane za pomocą symboli literowych lub numerów pozycji w tablicy kodu, znaczenie sekwencji jest określone za pomocą wchodzących w jej skład kombinacji kodowych i nie wpływa na nie jakiegokolwiek dowolne znaczenie, przypisane każdej z tych kombinacji z osobna.

Znaki funkcyjne z kolumn 0 i 1 oraz znak na pozycji 7/15 nie mogą być używane w charakterze znaków wewnętrznych ani też znaków finalnych sekwencji ESC.

**UWAGA.** Znaki objęte powyższym zakazem mogą na skutek błędu wystąpić w sekwencji i w niektórych zastosowaniach może okazać się konieczne użycie metod wykrywania i korygowania tych błędów. Zagadnienia te jednak nie są przedmiotem niniejszej normy.

**3.3.3. Kategorie sekwencji ESC.** W niniejszej normie przedstawiono zastosowanie sekwencji ESC. Jednak sekwencje o niektórych znakach finalnych z kolumny 3, oznaczonych symbolicznie  $F_p$  (rys. 3 i 4), są zarezerwowane do prywatnego użytku. Prywatne sekwencje ESC nie są rejestrowane zgodnie z załącznikiem 4.

**UWAGA.** Użytkownicy prywatnych sekwencji ESC powinni mieć na względzie fakt, że inni użytkownicy mogą przypisać danej sekwencji inne znaczenie lub może okazać się, że to samo znaczenie jest wyrażane przez dwie różne sekwencje. Co więcej, takie znaczenia mogą w przyszłości być przypisane sekwencjom zarejestrowanym. Ostrzeżenie strony uczestniczące w wymianie informacji, że stosowanie prywatnych sekwencji ESC może w konsekwencji obu stronom zmniejszyć możliwości tej wymiany.

**3.3.3.1. Sekwencje dwuznakowe** powinny mieć ogólną postać ESC F.

Są one wykorzystywane do reprezentowania jednostkowych dodatkowych funkcji sterujących. 79 dwuznakowych sekwencji ESC tworzy trzy grupy w zależności od znaku finalnego, jak zilustrowano to na rys. 3.

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								
8				$F_p$	$F_e$			
9								
10								
11								
12								
13								
14								
15								

BN-89/3101-06-3

Rys. 3. Znaki finalne dwuznakowych sekwencji ESC

Sekwencja ESC  $F_s$ <sup>1)</sup> reprezentuje jedną dodatkową funkcję sterującą o przypisanym na stałe znaczeniu, niezależnym od wykorzystanego znaku finalnego.

Do tego celu przewidziano 31 znaków finalnych z kolumn 6 i 7. W załączniku 4 p. 3 omówiono procedury rejestracji tych sekwencji.

Sekwencja ESC  $F_e$  reprezentuje odrębną funkcję sterującą oznaczonego na bieżąco (w danej chwili) zbioru CI, zawierającego 32 funkcje sterujące (zobacz p. 3.3.6). Jej znaczenie zależy od wykorzystanego znaku finalnego. Do tego celu służą 32 znaki finalne z kolumn 4 i 5. W niektórych zastosowaniach wystarcza użycie tylko jednego takiego zbioru dodatkowego. Wówczas zbiór identyfikuje się albo za pomocą sekwencji ESC, jak to opisano w 3.3.6, albo zgodnie z umową między nadawcą i odbiorcą informacji. Jeśli istnieje potrzeba wykorzystywania w systemie kilku dodatkowych zbiorów funkcji sterujących, to kolejny używany zbiór oznacza się i przywołuje za pomocą odpowiedniej sekwencji ESC.

<sup>1)</sup> Występujące w dalszej części normy sformułowania „sekwencja ESC  $F_s$ ”, „sekwencja ESC  $F_e$ ” oraz „sekwencja ESC  $F_p$ ” należy rozumieć odpowiednio jako: „sekwencja typu ESC  $F_s$ ”, „sekwencja typu ESC  $F_e$ ” i „sekwencja typu ESC  $F_p$ ”.

Sekwencja ESC  $F_p$  reprezentuje w zależności od znaku finalnego jednostkową dodatkową funkcję sterującą o nieznormalizowanym znaczeniu, przeznaczoną do prywatnego użytku. Jej znaczenie, w zależności od wykorzystanego znaku finalnego, powinno być uzgodnione między nadawcą i odbiorcą informacji. Do tego celu przewidziano 16 znaków finalnych z kolumny 3.

**3.3.3.2. Sekwencje trójznakowe** powinny mieć ogólną postać ESC I F.

Wszystkie typy trójznakowych sekwencji ESC w zależności od przeznaczenia połączono w klasy odpowiednio do ich znaków wewnętrznych, co przedstawiono w 3.3.4 ÷ 3.3.13 i w tabl. I.

Sekwencje te dzielą się na trzy typy w zależności od znaków finalnych, jak to ilustruje rys. 4.

	0	1	2	3	4	5	6	7
0								
1								
2								
3								
4								
5								
6								
7								
8				$F_p$				
9								
10								
11								
12								
13								
14								
15								

BN-89/3101-06-4

Rys. 4. Wewnętrzne i finalne znaki trójznakowych sekwencji ESC

Sekwencje ESC I  $F_i$  są przeznaczone do znormalizowanego użycia przez ISO. Do tego celu przewidziano 63 znaki  $F_i$  z kolumn 4 ÷ 7. Znak finalny 7/14, jeśli występuje z wewnętrznymi znakami 2/1, 2/2, 2/4, 2/8 ÷ 2/11 lub 2/13 ÷ 2/15 wskazuje, że zbiór oznaczony jest pusty, tzn. nie zawiera żadnego znaku i że kombinacje kodowe, reprezentujące znaki pustego zbioru, nie powinny być wykorzystywane.

Sekwencje ESC I  $F_c$  są przeznaczone do znormalizowanych zastosowań przez RWPG (nie przez ISO). Do tego celu przewidziano 8 znaków  $F_c$  na pozycjach 3/8 ÷ 3/15.

Do prywatnego użytku zarezerwowano sekwencje ESC I  $F_p$ . Do tego celu przewidziano 8 znaków  $F_p$  na pozycjach 3/0 ÷ 3/7 tablicy kodu.

**3.3.3.3. Sekwencje czteroznakowe i dłuższe** powinny mieć ogólną postać ESC L...I F, przy czym symbole L...I reprezentują dwa lub większą liczbę znaków wewnętrznych.

Sekwencje te powinny być interpretowane następująco:

1) pierwszy znak wewnętrzny określa klasę zastosowania, analogiczną względem klasy, której odpowiada taki sam znak wewnętrzny w sekwencji trójznakowej;

2) z wyjątkiem przypadków, w których pierwszy znak wewnętrzny jest zarezerwowany do przyszłego znormalizowania lub w których jest on znakiem z pozycji 2/0, 2/4, 2/5 lub 2/6 (zobacz odpowiednio p. 3.3.12, 3.3.9, 3.3.11 i 3.3.13 lub rozdz. 6), drugim znakiem wewnętrznym należy posługiwać się zgodnie z następującą regułą:

a) pozycja 2/0 jest zarezerwowana do oznaczania zbiorów określanych dynamicznie (zwanym dalej skróto zbiorami DRCS) (zobacz 3.3.10),

b) pozycje 2/1, 2/2, 2/3 są przeznaczone do rejestracji funkcji sterujących i zbiorów znaków graficznych,

c) pozycje 2/4 ÷ 2/15 są zarezerwowane do przyszłego znormalizowania;

3) trzeci i wszystkie pozostałe znaki wewnętrzne są przeznaczone do rejestracji funkcji sterujących i zbiorów znaków graficznych;

4) wszystkie sekwencje ESC o finalnym znaku symbolizowanym przez  $F_p$  są zarezerwowane do prywatnego użytku i nie określone w niniejszej normie;

5) wykorzystanie pozycji 7/14 w charakterze znaku finalnego określającego zbiór pusty, jak to opisano w 3.3.3.2, pozostaje w mocy także dla sekwencji czteroznakowych i dłuższych.

**3.3.4. Jednostkowe dodatkowe funkcje sterujące.** Sekwencja ESC 2/3 F reprezentuje jednostkową dodatkową funkcję sterującą, określoną przez finalny znak tej sekwencji.

**3.3.5. Zbiory 32 znaków funkcyjnych z kolumn 0 i 1.** Sekwencja ESC 2/1 F oznacza i przywołuje zbiór C0 32 znaków funkcyjnych, reprezentowanych przez kombinacje kodowe z kolumn 0 i 1.

Dziesięć znaków sterowania transmisją, występujących w zbiorze C0 powinno zachować swoje znaczenia i pozycje w tablicy kodu. Zbiór C0 nie powinien zawierać żadnych innych znaków sterowania transmisją.

W celu zmniejszenia prawdopodobieństwa nieporozumień podczas wymiany informacji zbiór C0 powinien zawierać:

- 1) 10 znaków sterowania transmisją,
- 2) znaki funkcyjne NUL, SO, SI, CAN, SUB i ESC o niezmiennym znaczeniu i niezmienionej pozycji w tablicy kodu.

Należy uwzględnić fakt, że zmiany znaczeń znaków funkcyjnych podczas wymiany informacji mogą wpływać także na pracę urządzeń. Na przykład kombinacja kodowa reprezentująca znak HT (TABULACJA POZIOMA) działa jako „tabulacja pozioma” na system przeznaczony do reagowania na ten znak funkcyjny.

**3.3.6. Zbiory 32 funkcji sterujących reprezentowanych przez sekwencje ESC  $F_c$ .** Sekwencja ESC 2/2 F oznacza i przywołuje zbiór 32 funkcji sterujących C1, nie wpływając przy tym na zbiór C0.

Każda funkcja sterująca zbioru C1 jest reprezentowana przez sekwencję ESC  $F_c$ , a nie przez jedną kombinację kodową. Zbiór C1 nie powinien zawierać funkcji sterowania transmisją (zobacz uwaga do 3.3.8).

**3.3.7. Zbiory 94 znaków graficznych.** Sekwencja ESC 2/8 F oznacza zbiór 94 znaków graficznych, wykorzystywany jako G0. Przywoływanie oznaczonego zbioru jest realizowane przez znak SI.

Sekwencja ESC 2/9 F oznacza zbiór 94 znaków graficznych, wykorzystywany jako G1. Przywoływanie oznaczonego zbioru jest realizowane za pomocą znaku SO.

Sekwencja ESC 2/10 F oznacza zbiór 94 znaków graficznych, wykorzystywany jako G2. Oznaczony zbiór jest przywoływany za pomocą funkcji LS2, a poszczególne znaki z oznaczonego zbioru są przywoływane za pomocą funkcji SS2.

Sekwencja ESC 2/11 F oznacza zbiór 94 znaków graficznych, wykorzystywany jako G3. Oznaczony zbiór jest przywoływany przez funkcję LS3, a poszczególne znaki z oznaczonego zbioru — przez funkcję SS3.

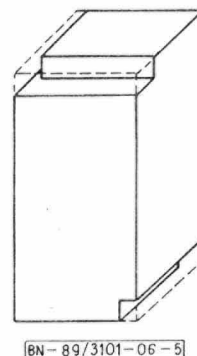
**3.3.8. Zbiory 96 znaków graficznych.** Sekwencja ESC 2/13 F oznacza zbiór 96 znaków graficznych, wykorzystywany jako G1. Przywoływanie oznaczonego zbioru jest realizowane za pomocą znaku SO.

Sekwencja ESC 2/14 F oznacza zbiór 96 znaków graficznych, wykorzystywany jako G2. Oznaczony zbiór jest przywoływany przez funkcję LS2, a poszczególne znaki zbioru oznaczonego — przez funkcję SS2.

Sekwencja ESC 2/15 F oznacza zbiór 96 znaków graficznych, używany jako G3. Oznaczony zbiór jest przywoływany przez funkcję LS3, a poszczególne znaki oznaczonego zbioru — przez funkcję SS3.

**UWAGA.** Podczas rejestrowania zbiorów znaków każdemu ze zbiorów zostaje przydzielony jeden znak finalny. Dla zbiorów znaków funkcyjnych C0 i C1 przydziela się odrębne grupy znaków finalnych, tzn. zbiór zostaje zarejestrowany albo jako C0, albo jako C1. Inaczej jest w przypadku rejestrowania zbiorów graficznych. Zbiory te nie są rejestrowane jako G0 lub G1, lub G2, lub G3, lecz jako wszystkie cztery rodzaje jednocześnie. Każdy zbiór może być wykorzystywany jako dowolny z czterech zbiorów, dzięki zastosowaniu wewnętrznego znaku sekwencji ESC, jak to określono w 3.3.7 ÷ 3.3.9, z wyjątkiem zbiorów 96 znaków graficznych, które nie powinny być wykorzystywane jako G0.

**3.3.9. Zbiory znaków graficznych o reprezentacjach wielobajtowych.** Sekwencja ESC 2/4 I F oznacza zbiór znaków graficznych, z których każdy jest reprezentowany przez dwa lub większą liczbę bajtów, a każdy bajt odpowiada kombinacji kodowej z kolumn 2 ÷ 7. Zbiór ten przedstawiono na rys. 5.



Rys. 5. Zbiór znaków graficznych o reprezentacjach wielobajtowych

Sekwencja ESC 2/4 2/8 F oznacza zbiór graficzny wielobajtowy, który będzie wykorzystywany jako G0. Oznaczony zbiór jest przywoływany za pomocą SI.

Sekwencja ESC 2/4 2/9 F lub ESC 2/4 2/13 F oznacza wielobajtowy zbiór graficzny, który będzie wykorzystywany jako G1. Oznaczony zbiór jest przywoływany za pomocą SO.

Sekwencja ESC 2/4 2/10 F lub ESC 2/4 2/14 F oznacza wielobajtowy zbiór graficzny, który będzie wykorzystywany jako G2. LS2 przywołuje oznaczony zbiór, a poszczególne znaki zbioru oznaczonego są przywoływane przez SS2.

Sekwencja ESC 2/4 2/11 F lub ESC 2/4 2/15 F oznacza wielobajtowy zbiór graficzny, który ma być wykorzystywany jako G3. Oznaczony zbiór jest przywoływany przez LS3, a poszczególne znaki zbioru oznaczonego — przez SS3.

Wyjątkiem od tych reguł są sekwencje ESC 2/4 4/0, ESC 2/4 4/1 oraz ESC 2/4 4/2, które zostały przydzielone zarejestrowanym w przeszłości wielobajtowym zbiorom G0.

Zbiór wielobajtowy oznaczony sekwencją ESC 2/4 2/8 F, ESC 2/4 2/9 F, ESC 2/4 2/10 F lub ESC 2/4 2/11 F zawiera co najwyżej  $94^n$  znaków. Każdy znak jest reprezentowany przez ciąg złożony z  $n$  bajtów, tzn. z kombinacji kodowych z przedziału  $2/1 \div 7/14$ . Przy tym  $n = 2, 3, 4 \dots$

Zbiór wielobajtowy oznaczony przez sekwencję ESC 2/4 2/13 F, ESC 2/4 2/14 F lub ESC 2/4 2/15 F zawiera co najwyżej  $96^n$  znaków. Każdy znak jest reprezentowany przez ciąg złożony z  $n$  bajtów, tzn. kombinacji kodowych z przedziału  $2/0 \div 7/15$ . Przy tym  $n = 2, 3, 4 \dots$

W obrębie jednego zbioru wielobajtowego każdy znak graficzny jest reprezentowany przez tę samą liczbę bajtów  $n$ .

Kiedy funkcja przesunięcia jednostkowego jest wykorzystywana do przywoływania znaku ze zbioru wielobajtowego, to wbrew zasadom jej zwykłego używania obejmuje ona dwie lub większą liczbę kolejnych kombinacji kodowych do reprezentowania jednego znaku zbioru wielobajtowego.

Sekwencje o pierwszym znaku wewnętrznym 2/4 i drugim znaku wewnętrznym albo 2/0  $\div$  2/7, albo 2/12 są zarezerwowane do przyszłego znormalizowania.

Znaczenia znaku finalnego sekwencji podano w tabl. 1.

Tablica 1. Znaki finalne sekwencji oznaczających zbiory wielobajtowe

Numer kolumny w tablicy kodu	Liczba bajtów reprezentujących jeden znak w zbiorze wielobajtowym
3	2 lub więcej, do prywatnego użytku
4 i 5	2
6	3
7	4 i więcej

Jeśli należy zarejestrować więcej niż 63 zbiory, w sekwencji ESC oznaczającej zbiór wielobajtowy po drugim znaku wewnętrznym mogą wystąpić kolejne znaki wewnętrzne, przy tym trzeci powinien być z przedziału  $2/1 \div 2/3$ .

**3.3.10. Zbiory znaków określone dynamicznie (DRCS).** Zbiór DRCS jest zbiorem znaków graficznych, których

zobrazowanie powinno być określone i przesłane (tzn. załadowane) do chwili ich wykorzystania. Określenie można uczynić w sposób jawny lub za pomocą odniesienia. Określonymi znakami mogą być litery, znaki specjalne lub elementarne znaki rysunków. Po jednokrotnym załadowaniu zbiór DRCS jest traktowany jako element repertuaru zbiorów graficznych, który może być oznaczony przez odpowiednie sekwencje ESC jako G0, G1, G2 lub G3.

Sekwencja ESC I 2/0 F oznacza zbiór DRCS, przy czym I zawarte w przedziale  $2/8 \div 2/11$  wskazuje zbiór 94-znakowy, wykorzystywany odpowiednio jako G0, G1, G2 lub G3, a I zawarte w przedziale  $2/13 \div 2/15$  wskazuje zbiór 96-znakowy wykorzystywany odpowiednio jako G1, G2 lub G3 w taki sam sposób, jak to określono w 3.3.7 i 3.3.8. Znak finalny F powinien mieć się w przedziale  $4/0 \div 7/14$ .

Za pomocą takich czteroznakowych sekwencji można zidentyfikować 126 zbiorów, tzn.  $2 \times 63$  zbiory. Powinno to wystarczać dla maksymalnych potrzeb, ale jeśli okaże się konieczne przyjęcie większej liczby zbiorów, to dla ich oznaczania można rozbudować sekwencje ESC, umieszczając między drugim znakiem wewnętrznym a znakiem finalnym jeden lub większą liczbę dodatkowych znaków wewnętrznych. Sekwencje, w których jako drugi znak wewnętrzny występuje 2/0 oraz sekwencje, w których jako pierwszy znak wewnętrzny występuje znak z przedziału  $2/0 \div 2/7$  lub znak 2/12, są zarezerwowane do przyszłego znormalizowania.

Wielobajtowe zbiory graficzne mogą także być zbiorem określanymi dynamicznie (DRCS). Taki zbiór jest oznaczany przez sekwencję ESC 2/4 I 2/0 F. Znak wewnętrzny sekwencji ma przydzielone te same kombinacje kodowe z przedziału  $2/8 \div 2/11$  lub  $2/13 \div 2/15$  i takie same znaczenia, jak określone w 3.3.10. W obrębie sekwencji można wstawiać dodatkowe znaki wewnętrzne, jeśli istnieje potrzeba identyfikowania więcej niż 63 zbiorów.

**UWAGI.** 1. Ta klasa sekwencji ESC należy do wyjątkowych, ponieważ przypisanie znaków finalnych (a możliwe, że także i wewnętrznych) jest wykonywane nie przez organizacje rejestrujące (zobacz procedury rejestracji opisane w załączniku 4), lecz przez użytkownika. Zaleca się przypisywanie kolejnych znaków finalnych począwszy od pozycji 4/0.  
2. Potrzeba tej szczególnej sekwencji ESC, tak różniącej się od zwykłej sekwencji trójznakowej używanej do reprezentowania zbiorów zarejestrowanych, wynika z faktu, że obejmuje ona ścisły opis kształtu lub kroju pisma znaków.

**3.3.11. Inne systemy kodowania.** Sekwencje ESC 2/5 F oraz ESC 2/5 I F (z wyjątkiem ESC 2/5 4/0) oznaczają i przywołują system kodowania inny niż opisany w niniejszej normie, przy czym ten system nie musi być kodem znakowym.

Sekwencja ESC 2/5 4/0 została wybrana i jest zalecana do wykorzystania w innych systemach kodowania w celu powrotu do systemu kodowania zgodnego z niniejszą normą. Sekwencja ESC 2/5 4/0 przywraca stan systemu kodowania (tzn. stan sekwencji powiadamiających oraz zbiorów oznaczonych i przywołanych) do stanu, który istniał w czasie przywołania innego systemu



kodowania. W niniejszej normie nie określono innych stanów systemu, np. pozycji czynnej.

Sekwencje oznaczające inne systemy kodowania podzielono na następujące kategorie, w zależności od wykorzystania lub nie sekwencji powrotu ESC 2/5 4/0:

- |  |   |  |
|--|---|--|
| 1) ESC 2/5 F<br>ESC 2/5 I F<br>ESC 2/5 L...I F,<br>gdzie I = 2/1 ÷ 2/3 | } | inny system kodowania wykorzystuje sekwencję powrotu ESC 2/5 4/0   |
| 2) ESC 2/5 2/15 F<br>ESC 2/5 2/15 L...I F                              | } | inny system kodowania nie używa sekwencji powrotu ESC 2/5 4/0 (może mieć inne środki powrotu lub nie mieć ich wcale) |

W kategorii 1) mogą być używane drugi znak wewnętrzny z przedziału 2/1 ÷ 2/3 i dalsze kolejne znaki wewnętrzne, o ile trzeba oznaczyć (zidentyfikować, zarejestrować) więcej niż 63 systemy kodowania.

W kategorii 2) mogą być używane dalsze znaki wewnętrzne, jeśli istnieje potrzeba oznaczenia (zidentyfikowania, zarejestrowania) więcej niż 63 systemów kodowania.

Sekwencje, których pierwszy znak wewnętrzny jest znakiem z pozycji 2/5, a znak 2/0 lub 2/4 ÷ 2/14 występuje jako drugi znak wewnętrzny, są zarezerwowane dla przyszłego znormalizowania.

Przedstawione powyżej środki stwarzają możliwości przełączania systemu kodowania zgodnego z niniejszą normą na inne systemy kodowania, kiedy nie jest to wykonywane na mocy Protokołu wyższego poziomu (zobacz rozdz. 8).

**3.3.12. Powiadomienia o środkach rozszerzania kodu.** Sekwencje ESC 2/0 F powiadomają o środkach rozszerzania kodu, wykorzystywanych w kolejnym strumieniu danych. Użycie tych sekwencji opisano w rozdz. 6.

**3.3.13. Modyfikowanie zbiorów zarejestrowanych.** W załączniku 4 podano informację o międzynarodowych organizacjach zajmujących się rejestrowaniem zbiorów znaków. Jeśli jest wykorzystywana sekwencja ESC 2/6 F, to powinna ona bezpośrednio poprzedzać sekwencję oznaczającą zbiór i informować o modyfikacji zarejestrowanego zbioru. Znak finalny F, będąc znakiem z przedziału 4/0 ÷ 7/14, określa numer modyfikacji zawarty odpowiednio w przedziale 1 ÷ 63. Wykonując modyfikację można tylko do zbioru dołączyć jeden lub więcej znaków i o tym należy poinformować organizację rejestrującą zbiory (zobacz załącznik 4) z adnotacją, że przedłożony zbiór jest modyfikacją zbioru zarejestrowanego. Jeśli zbiór zmodyfikowany nie jest poza tą zmianą zgodny z poprzednią wersją, to powinien być na nowo oznaczony przez inną sekwencję ESC.

**UWAGA.** Połączenie sekwencji oznaczającej numer modyfikacji zbioru z sekwencją oznaczającą jego poprzednią wersję ułatwia rozpoznawanie przez wcześniej wyprodukowane urządzenia lub systemy nowej wersji zbiorów znaków.

Sekwencje o pierwszym znaku wewnętrznym 2/6 i dalszych kolejnych znakach wewnętrznych są zarezerwowane do przyszłego znormalizowania.

**3.3.14. Sekwencje trójznakowe bez przydzielonych znaczeń.** Sekwencjom ESC 2/7 F i ESC 2/12 F nie przydzielono znaczeń. Są one zarezerwowane do przyszłego znormalizowania.

**3.3.15. Wykaz znaczeń znaków wewnętrznych sekwencji ESC** podano w tabl. 2. Lista sekwencji ESC — wg załącznika 2.

Tablica 2. Znaczenia znaków wewnętrznych sekwencji ESC

Pierwszy znak wewnętrzny	Drugi znak wewnętrzny															
	2/0	2/1	2/2	2/3	2/4	2/5	2/6	2/7	2/8	2/9	2/10	2/11	2/12	2/13	2/14	2/15
<i>Pozycje tablicy kodu</i>	Klasa sekwencji															
2/0	Sekwencje powiadamiające															
2/1	Funkcje sterujące		Zbiór C0		<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">60</div> <div style="border: 1px solid black; padding: 2px;">61</div> <div style="border: 1px solid black; padding: 2px;">62</div> <div style="border: 1px solid black; padding: 2px;">63</div> </div>											
2/2			Zbiór C1													
2/3			Jednostkowe funkcje sterujące													
2/4	Znaki graficzne		Zbiory wielobajtowe		<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">61</div> <div style="border: 1px solid black; padding: 2px;">62</div> <div style="border: 1px solid black; padding: 2px;">63</div> </div>											
2/5	Inne systemy kodowania															
2/6	Modyfikacja zbiorów zarejestrowanych															
2/7																
2/8	94 znaki graficzne		Zbiór 60		<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">60</div> <div style="border: 1px solid black; padding: 2px;">61</div> <div style="border: 1px solid black; padding: 2px;">62</div> <div style="border: 1px solid black; padding: 2px;">63</div> </div>											
2/9			Zbiór 61													
2/10			Zbiór 62													
2/11			Zbiór 63													
2/12																
2/13	96 znaków graficznych		Zbiór 61		<div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 2px;">61</div> <div style="border: 1px solid black; padding: 2px;">62</div> <div style="border: 1px solid black; padding: 2px;">63</div> </div>											
2/14			Zbiór 62													
2/15			Zbiór 63													
<p><b>*) DRCS</b> – zbiór określony dynamicznie (zobacz 3.3.10).</p> <p>Obszary zakreślowane oznaczają kombinacje kodowe zarezerwowane do przyszłego znormalizowania.</p>																

**3.4. Oznaczanie i przywoływanie na początku wymiany informacji.** Na początku wymiany informacji wszystkie oznaczenia powinny być określone przez wykorzystanie odpowiednich sekwencji ESC, a stan przesunięcia — przez użycie odpowiednich funkcji przesunięć blokujących. Ostrzega się strony uczestniczące w wymianie informacji, decydując się na nieużywanie takich oznaczeń, że w konsekwencji możliwości każdej ze stron mogą podczas wymiany zostać ograniczone.

**3.5. Schemat rozszerzania kodu 7-bitowego w jego środowisku.** Na rys. 6 przedstawiono w schematycznej formie znormalizowane środki rozszerzania kodu 7-bitowego.

#### 4. STRUKTURA RODZINY KODÓW 8-BITOWYCH

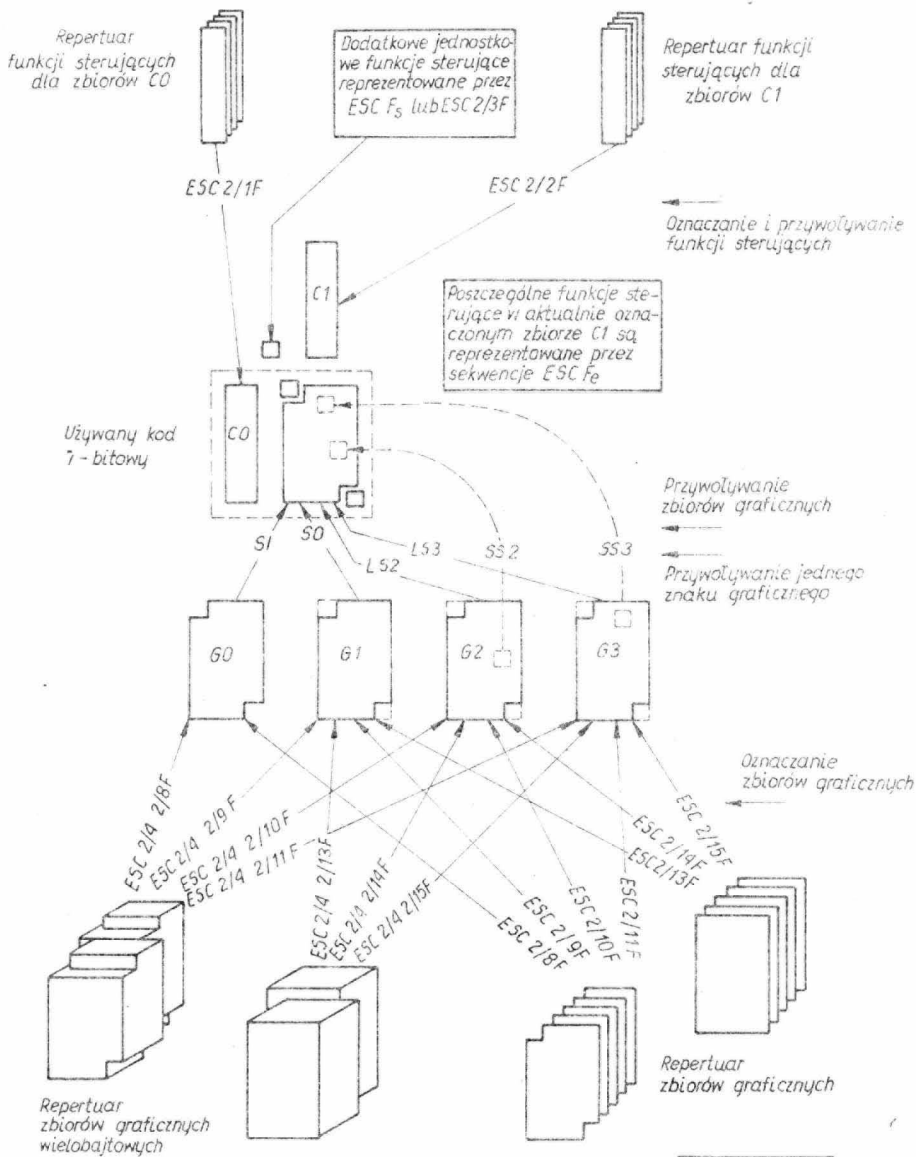
**4.1. Tworzenie rodziny kodów 8-bitowych na podstawie kodu 7-bitowego.** Rodzinę kodów 8-bitowych można utworzyć przez dostawienie jednego bitu o największej

wadze do każdej kombinacji kodowej kodu 7-bitowego, uzyskując w ten sposób zbiór 256 kombinacji 8-bitowych. Znaki występujące w kodzie 7-bitowym są przypisywane 128 kombinacjom kodowym, których ósmym bitem jest ZERO. W ten sposób zbiór określony w 3.1 staje się określoną i integralną częścią kodu 8-bitowego, w zakresie struktury, spełniającego wymagania niniejszej normy. Dodatkowe 128 kombinacji kodowych, w których ósmym bitem jest JEDEN, jest przeznaczonych do dalszego rozszerzania kodu.

**4.2. Określenie pojęcia rodziny kodów 8-bitowych.** W celu zaspokojenia potrzeb różnorodnych gałęzi przemysłu, dziedzin zastosowania lub systemów w niniejszej normie określa się pojęcie rodziny kodów 8-bitowych w następujący sposób:

1) zbiór 32 dodatkowych znaków funkcyjnych może być wybrany z kolumn 08 i 09;

2) zbiór 94 lub 96 dodatkowych znaków graficznych może być wybrany z kolumn 10 ÷ 15. Jeśli zbiór 94



Rys. 6. Rozszerzanie kodu 7-bitowego w jego środowisku (pokazano wszystkie funkcje przesunięć)

znaków graficznych jest przywoływany do kolumn 10 ÷ 15, to pozycje 10/00 i 15/15 nie powinny być wykorzystywane.

## 5. ROZSZERZANIE KODU 8-BITOWEGO

Metody rozszerzania kodu 8-bitowego są zbliżone z metodami wykorzystywanymi do rozszerzania kodu 7-bitowego.

Znak ESC powinien być używany w kodzie 8-bitowym w taki sam sposób, jak w kodzie 7-bitowym do tworzenia sekwencji ESC. Znaczenia tych sekwencji nie ulegają zmianie w kodzie 8-bitowym. Znaki z kolumn 08 ÷ 15 nie powinny występować w sekwencjach ESC; pojawienie się któregośkolwiek z tych znaków oznacza stan błędu. W niniejszej normie nie podano znormalizowanych procedur wykrywania i korygowania takich błędów.

**5.1. Elementy rozszerzania kodu 8-bitowego w jego środowisku,** przedstawione na rys. 1, scharakteryzowano bliżej w tabl. 3.

Tablica 3. Elementy rozszerzania kodu 8-bitowego w jego środowisku

Zbiór	Opis zbioru	Zajmowane kolumny w tablicy kodu
C0	32 znaki funkcyjne	00 ÷ 01
C1	32 znaki funkcyjne	08 ÷ 09
G0	94 znaki graficzne	02 ÷ 07
G1	94 lub 96 znaków graficznych	02 ÷ 07 lub 10 ÷ 15
G2	94 lub 96 znaków graficznych	02 ÷ 07 lub 10 ÷ 15
G3	94 lub 96 znaków graficznych	02 ÷ 07 lub 10 ÷ 15

Zbiory C0 i C1 powinny być oznaczane i przywoływane za pomocą tych samych sekwencji ESC, jak w środowisku kodu 7-bitowego (zobacz 3.3.5 i 3.3.6). Zbiory G0, G1, G2 i G3 powinny być oznaczane przez te same sekwencje ESC, jak w środowisku kodu 7-bitowego (zobacz 3.3.7 ÷ 3.3.10).

**5.2. Rozszerzanie zbioru znaków graficznych za pomocą funkcji przesunięć.** W środowisku kodu 8-bitowego wykorzystuje się następujące funkcje przesunięć, określone w PN-89/T-42108<sup>1)</sup>: LS0, LS1, LS1R, LS2, LS2R, LS3, LS3R, SS2, SS3.

Informacje dotyczące kodowania tych funkcji przedstawiono w tabl. 5 i załączniku 1 tabl. Z1-1.

**5.2.1. Stosowanie funkcji przesunięć blokujących.** W środowisku kodu 8-bitowego występuje siedem funkcji przesunięć blokujących. Są one używane wyłącznie do rozszerzania zbioru znaków graficznych. Z wyjątkiem funkcji LS0, która przywołuje tylko zbiory 94-znakowe, każda z pozostałych sześciu funkcji przywołuje jeden dodatkowy zbiór 94- lub 96-znakowy do kolumn 02 ÷ 07 lub 10 ÷ 15. W tabl. 4 przedstawiono siedem funkcji przesunięć blokujących.

Tablica 4. Funkcje przesunięć blokujących używane w środowisku kodu 8-bitowego

Funkcja	Zbiór przywoływany	Kolumny tablicy kodu podlegające działaniu funkcji
LS0 <sup>1)</sup>	G0	02 ÷ 07
LS1	G1	02 ÷ 07

cd. tabl. 4

Funkcja	Zbiór przywoływany	Kolumny tablicy kodu podlegające działaniu funkcji
LS1R	G1	10 ÷ 15
LS2	G2	02 ÷ 07
LS2R	G2	10 ÷ 15
LS3	G3	02 ÷ 07
LS3R	G3	10 ÷ 15

Jeśli dowolny zbiór został już przywołany, to powtórne użycie odpowiadającej mu funkcji przesunięcia nie daje żadnego rezultatu.

Stosowanie funkcji przesunięcia blokującego nie powinno wpływać na znaczenie następujących kombinacji kodowych:

- 1) reprezentujących znaki funkcyjne w kolumnach 00 i 01, 08 i 09,
- 2) występujących w dowolnej sekwencji ESC,
- 3) występujących po funkcjach SS2 lub SS3.

Znaki SPACJA (SP) i KASOWANIE (DEL) powinny występować tylko na pozycjach odpowiednio 02/00 i 07/15, ale tylko w tym przypadku, gdy zbiór przywołany składa się z 94 znaków graficznych. Znaki te nie powinny występować na dowolnych innych pozycjach w dowolnym zbiorze. Jednak znaki inne niż SPACJA, ale reprezentujące odstępów o innych rozmiarach lub przeznaczeniu, mogą występować na dowolnych pozycjach w dowolnym zbiorze znaków graficznych lub funkcji sterujących.

Na początku wymiany informacji stan przesunięć powinien być określony przez wykorzystanie funkcji przesunięć blokujących, jak to określono w 3.4, w rozdz. 6 i w tabl. 5.

**5.2.2. Stosowanie funkcji przesunięć jednostkowych.** W kodzie 8-bitowym zasady używania tych funkcji są takie same, jakie obowiązują dla kodu 7-bitowego (zobacz 3.2.2). Dla kombinacji kodowej występującej po funkcji SS2 lub SS3 jest dopuszczalny tylko znak z kolumn 02 ÷ 07. Z wyjątkiem sytuacji opisanej w 7.3, wszystkie kombinacje kodowe kolumn 10 ÷ 15 nie powinny występować po funkcji SS2 ani po SS3. Użycie funkcji przesunięcia jednostkowego nie powinno mieć wpływu na bieżący stan, ustawiony przez jedną lub więcej funkcji przesunięć blokujących.

**5.3. Rozszerzanie kodu za pomocą sekwencji ESC.** Jeśli kod 8-bitowy jest określony zgodnie z 5.1, to rozszerzanie kodu powinno być realizowane za pomocą sekwencji ESC, jak to przedstawiono poniżej w 5.3.1 ÷ 5.3.3.

**5.3.1. Dwuznakowe sekwencje ESC** powinny charakteryzować się taką samą strukturą, jak w środowisku kodu 7-bitowego (zobacz 3.3.3.1).

Sekwencje ESC  $F_n$  reprezentują jednostkowe dodatkowe funkcje sterujące o tym samym znaczeniu, które one mają w środowisku kodu 7-bitowego (zobacz 3.3.3.1).

Użycie sekwencji ESC  $F_n$  w środowisku kodu 8-bitowego nie jest przedmiotem niniejszej normy, jednak jeśli sekwencje te są wykorzystywane w specjalnych warunkach (zobacz tabl. 5), ich znaczenie powinno być takie samo, jak dla środowiska kodu 7-bitowego.

<sup>1)</sup> Odpowiednikiem PN-89/T-42108 jest norma CT CEB 359-86.

**5.3.2. Trójznakowe sekwencje ESC** powinny cechować się taką samą strukturą i znaczeniem, jak dla środowiska kodu 7-bitowego (zobacz 3.3.3.2).

**5.3.3. Czteroznakowe i dłuższe sekwencje ESC** powinny mieć taką samą strukturę i znaczenia, jak dla środowiska kodu 7-bitowego (zobacz 3.3.3.3).

**UWAGA.** Tę samą strukturę i znaczenia, jak dla środowiska kodu 7-bitowego mają sekwencje ESC oznaczające i przywołujące systemy kodowania inne niż określone w niniejszej normie (zobacz 3.3.11). Sekwencje ESC, które oznaczają zbiory graficzne wielobajtowe i zbiory określone dynamicznie (DRCS), mają strukturę i znaczenie zgodne odpowiednio z 3.3.9 i 3.3.10.

**5.4. Wielobajtowe zbiory znaków graficznych.** W środowisku kodu 8-bitowego, tak jak w środowisku kodu 7-bitowego, wielobajtowe zbiory znaków graficznych mogą być oznaczane i przywoływane jako zbiory G0, G1, G2 lub G3 (zobacz 3.3.9). Znak graficzny takiego zbioru wielobajtowego jest reprezentowany przez dwa lub więcej bajtów, z których wszystkie składają się z kombinacji kodowych albo z kolumn 02 ÷ 07, albo z kolumn 10 ÷ 15 w zależności od tego, gdzie zbiór wielobajtowy został przywołany. W ten sposób ósmy

bit ( $b_8$ ) każdego bajtu w danym zbiorze wielobajtowym powinien albo zawsze wynosić ZERO albo zawsze JEDEN.

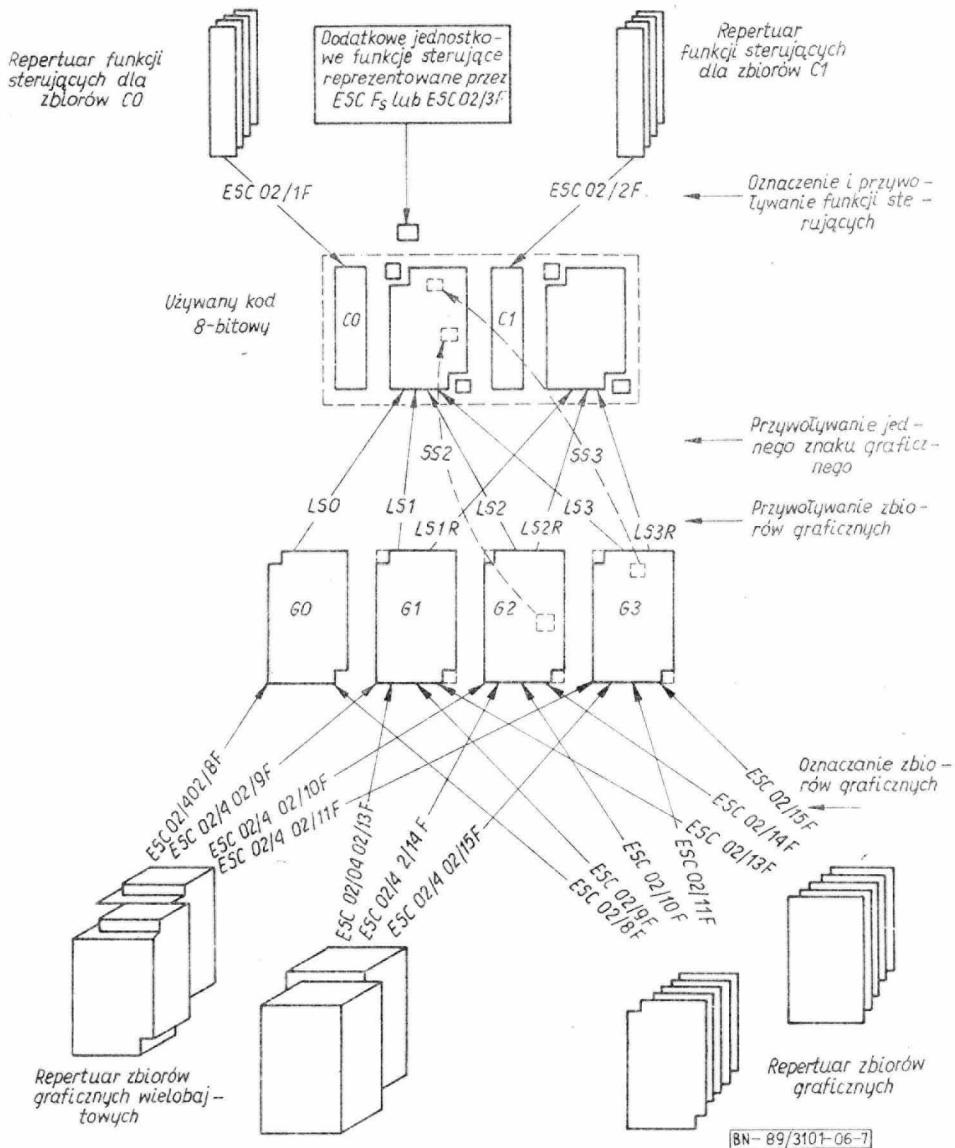
**UWAGI.** 1. Jeśli ósme bity ( $b_8$ ) wszystkich bajtów w danej reprezentacji wielobajtowej nie są sobie równe, to można wykrywać i korygować ten błąd, jednak nie jest to przedmiotem niniejszej normy.

2. Występowanie wielobajtowych zbiorów znaków graficznych nie ma wpływu na konwersję kodu 7-bitowego na 8-bitowy i na odwrót (zobacz rozdz. 7).

**5.5. Zgodność z wymaganiami niniejszej normy.** Kod 8-bitowy można uznać za zgodny z wymaganiami niniejszej normy, jeśli kolumny 00 ÷ 07 spełniają wymagania wg 3.1.5 1) lub 2), kolumny 08 i 09 zawierają tylko znaki funkcyjne, a kolumny 10 ÷ 15 są wykorzystywane tylko dla znaków graficznych.

Aby zapewnić możliwość stosowania środków rozszerzenia kodu danej normy, należy pozostawić dla znaku ESC i wykorzystywanych znaków przesunięć odpowiadające im znaczenia i pozycje tablicy kodu bez zmian (zobacz tabl. 7).

**5.6. Schemat rozszerzania kodu 8-bitowego w jego środowisku** przedstawiono na rys. 7.



BN-89/3101-06-7

Rys. 7. Rozszerzanie kodu 8-bitowego w jego środowisku (pokazano wszystkie funkcje przesunięć)



## 6. POWIADOMIENIA O STOSOWANYCH ŚRODKACH ROZSZERZANIA KODU

**6.1. Postanowienia ogólne.** Na początku wymiany informacji strony w niej uczestniczące mogą żądać od siebie powiadomienia o środkach rozszerzania kodu, wykorzystywanych w kolejnym strumieniu danych. Jeśli podobne powiadomienie powinno być włączone do strumienia kodowanych znaków, to powinny być użyte jedna lub kilka trójznakowych sekwencji ESC klasy ESC 2/0 F. Na mocy uzgodnienia między nadawcą i odbiorcą taka sekwencja powiadamiająca może nie występować. Znak finalny sekwencji powiadamiającej informuje o środkach wykorzystywanych do reprezentowania zbiorów graficznych i niektórych zbiorów funkcji sterujących w środowiskach kodów 7- i 8-bitowego. W tabl. 5 przedstawiono używane w tym celu znaki finalne włącznie z opisem środków rozszerzania kodu oraz ilustrującymi go schematami tablic kodowych.

Sekwencje o pierwszym znaku wewnętrznym 2/0 zawierające następne kolejne znaki wewnętrzne są zarezerwowane do przyszłego znormalizowania.

**6.2. Ograniczenia.** Powiadomienia 4/1, 4/3 i 4/4 nie powinny być wykorzystywane w połączeniu z powiadomieniami 5/0, 5/2, 5/3, 5/4, 5/5, 5/6 i 5/7.

Powiadomienia 4/12, 4/13 i 4/14 nie powinny być używane włącznie z innym dowolnym powiadomieniem.




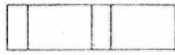

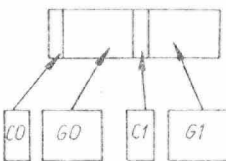

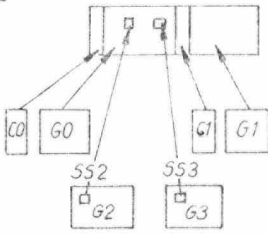

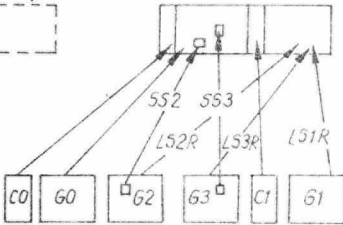
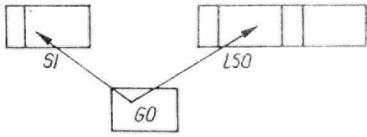
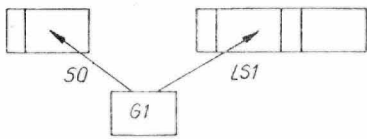
**UWAGI.** 1. W środowisku kodu 7-bitowego dane, o których powiadamia sekwencja ESC 2/0 4/4 mają tę samą postać co dane, o których powiadamia sekwencja ESC 2/0 4/2. Obie sekwencje są przeznaczone dla tych sytuacji wymiany, w których w środowisku kodu 7-bitowego należy wprowadzić rozróżnienie między danymi pochodzącymi z dwóch typów środowisk kodów 8-bitowych, tzn. mających zbiór G1 w kolumnach  $02 \div 07$  lub w kolumnach  $10 \div 15$ .

2. Jako przykłady sekwencji, które mogłyby być wykorzystywane w środowisku kodu 8-bitowego do powiadamiania o używaniu zbiorów G0, G1 i G3 za pomocą przesunięć blokujących i zbioru G2 za pomocą przesunięcia jednostkowego, mogą służyć następujące sekwencje:  
ESC 02/00 05/00 ESC 02/00 05/02 ESC 02/00 05/07  
ESC 02/00 05/10.

Tablica 5. Powiadomienia o środkach rozszerzenia kodu

Znak finalny	Środki rozszerzania kodu	Środowisko kodu 7-bitowego	Środowisko kodu 8-bitowego
1	2	3	
4/1	Będzie wykorzystywany zbiór G0. Sekwencja ESC, która oznacza ten zbiór, przywołuje go także do kolumn $2 \div 7$ . Nie powinny być wykorzystywane funkcje przesunięć blokujących. W środowisku kodu 7-bitowego kolumny $10 \div 15$ nie są wykorzystywane. Zobacz 6.2		
4/2	Będą wykorzystywane zbiory G0 i G1. W środowisku kodu 7-bitowego SI przywołuje G0 do kolumn $2 \div 7$ , a SO przywołuje G1 do kolumn $2 \div 7$ . W środowisku kodu 8-bitowego do kolumn $02 \div 07$ LS0 przywołuje G0, a LS1 przywołuje G1; w tym czasie kolumny $10 \div 15$ nie są wykorzystywane		
4/3	Zbiory G0 i G1 będą wykorzystywane tylko w środowisku kodu 8-bitowego. Oznaczające sekwencje ESC także przywołują zbiory G0 i G1 odpowiednio do kolumn $02 \div 07$ i $10 \div 15$ . Nie powinny być używane funkcje przesunięć blokujących. Zobacz 6.2	<i>Nie używane</i>	
4/4	Będą wykorzystywane zbiory G0 i G1. W środowisku kodu 7-bitowego do kolumn $2 \div 7$ SI przywołuje G0, a SO przywołuje G1. W środowisku kodu 8-bitowego oznaczające sekwencje ESC także przywołują zbiory G0 i G1 odpowiednio do kolumn $02 \div 07$ i $10 \div 15$ . Nie powinny być używane funkcje przesunięć blokujących. Zobacz 6.2		
4/5	Funkcje przesunięć są w pełni zachowane podczas konwersji danych wykonywanej między środowiskami kodu 7-bitowego i 8-bitowego	Zobacz 7.4	
4/6	Będzie wykorzystywany zbiór C1. Zarówno w środowisku kodu 7-bitowego, jak i 8-bitowego każda funkcja sterująca ze zbioru C1 będzie reprezentowana przez sekwencję ESC F <sub>c</sub> .	Zobacz 3.3.3.1 i 5.3.1	
4/7	Będzie wykorzystywany zbiór C1. W środowisku kodu 7-bitowego każda funkcja sterująca ze zbioru C1 będzie reprezentowana przez sekwencję ESC F <sub>c</sub> . W środowisku kodu 8-bitowego każda funkcja sterująca ze zbioru C1 będzie reprezentowana przez jedną i tylko jedną kombinację kodową z kolumn 08 i 09.	Zobacz 3.3.3.1	

cd. tabl. 5

Znak finalny	Środki rozszerzania kodu	Środowisko kodu 7-bitowego	Środowisko kodu 8-bitowego
I	2	3	
4/8	Wszystkie zbiory znaków graficznych są 94-znakowe	Zobacz 3.3.7	
4/9	Zbiory znaków graficznych mogą być 94- lub 96-znakowe	Zobacz 3.3.7 i 3.3.8	
4/10	W środowiskach kodu 7- i 8-bitowego jest używany kod 7-bitowy		
4/11	W środowisku kodu 8-bitowego jest używany kod 8-bitowy	<i>Nie używane</i> 	
4/12	Wersja kodu 8-bitowego 1 poziomu. Będą wykorzystywane zbiory C0, G0, C1, G1. Funkcje przesunięć nie powinny być używane	<i>Nie używane</i> 	
4/13	Wersja kodu 8-bitowego 2 poziomu. Będą wykorzystywane środki 1 poziomu oraz zbiory G2 i G3. Z funkcji przesunięć są używane tylko przesunięcia jednostkowe SS2 i SS3	<i>Nie używane</i> 	
4/14	Wersja kodu 8-bitowego 3 poziomu. Będą wykorzystywane środki 2 poziomu i funkcje przesunięć blokujących LS1R, LS2R i LS3R	<i>Nie używane</i> 	
5/0	Jako dodatek do dowolnej innej kategorii zbiorów graficznych, które mogą być używane, będzie wykorzystywany zbiór G0. Powinien on być przywoływany w środowisku kodu 7-bitowego za pomocą SI, a w środowisku kodu 8-bitowego za pomocą LS0. Zobacz 6.2		
5/2	Jako dodatek do dowolnej innej kategorii zbiorów graficznych, które mogą być używane, będzie wykorzystywany zbiór G1. Powinien on być przywoływany w środowisku kodu 7-bitowego za pomocą SO, a w środowisku kodu 8-bitowego za pomocą LS1. Zobacz 6.2		

cd. tabl. 5

Znak finalny	Środki rozszerzania kodu	Środowisko kodu 7-bitowego	Środowisko kodu 8-bitowego
1	2	3	
5/3	Jako dodatek do dowolnej innej kategorii zbiorów graficznych, które mogą być używane, będzie wykorzystywany zbiór G1. Powinien on być przywoływany w środowisku kodu 7-bitowego za pomocą SO, a w środowisku kodu 8-bitowego za pomocą LS1R. Zobacz 6.2		
5/4	Jako dodatek do dowolnej innej kategorii zbiorów graficznych, które mogą być używane, będzie wykorzystywany zbiór G2. Powinien on być przywoływany w obu środowiskach kodów 7- i 8-bitowego za pomocą LS2. Zobacz 6.2		
5/5	Jako dodatek do dowolnej innej kategorii zbiorów graficznych, które mogą być używane, będzie wykorzystywany zbiór G2. Powinien on być przywoływany w środowisku kodu 7-bitowego za pomocą LS2, a w środowisku kodu 8-bitowego — za pomocą LS2R. Zobacz 6.2		
5/6	Jako dodatek do dowolnej innej kategorii zbiorów graficznych, które mogą być używane, będzie wykorzystywany zbiór G3. Powinien on być przywoływany zarówno w środowisku kodu 7-bitowego, jak i 8-bitowego za pomocą LS3. Zobacz 6.2		
5/7	Jako dodatek do dowolnej innej kategorii zbiorów graficznych, które mogą być używane, będzie wykorzystywany zbiór G3. Powinien on być przywoływany w środowisku kodu 7-bitowego za pomocą LS3, a w środowisku kodu 8-bitowego za pomocą LS3R. Zobacz 6.2		
5/10	Jako dodatek do dowolnej innej kategorii zbiorów graficznych, które mogą być używane, będzie wykorzystywany zbiór G2. Zarówno w środowisku kodu 7-bitowego, jak i w środowisku kodu 8-bitowego SS2 powinna przywoływać jeden znak z tego zbioru		
5/11	Jako dodatek do dowolnej innej kategorii zbiorów graficznych, które mogą być używane, będzie wykorzystywany zbiór G3. Zarówno w środowisku kodu 7-bitowego, jak i 8-bitowego SS3 powinna przywoływać jeden znak z tego zbioru		

## 7. RELACJE MIĘDZY KODAMI 7- I 8-BITOWYM

**7.1. Konwersja kodu 7-bitowego na 8-bitowy i na odwrot.** Konwersja ta zależy od tego, jakie środki rozszerzania kodu występują w danym zastosowaniu. Środki te są identyfikowane za pomocą sekwencji powiadamiających, opisanych w rozdz. 6.

**7.2. Reprezentacja kodu 7-bitowego w środowisku kodu 8-bitowego.** W niektórych zastosowaniach, jak np. w trybie pracy „zapamiętaj i wyślij” jest wymagane pozostawienie informacji w reprezentacji 7-bitowej w środowisku kodu 8-bitowego. W tym przypadku w każdym ze znaków bit  $b_8$  jest ustawiany na ZERO.

Pewne funkcje przesunięć blokujących (tj. LS1R, LS2R i LS3R) dają inne rezultaty w kodzie 7-bitowym, a inne w kodzie 8-bitowym. Kiedy funkcje te są używane w środowisku kodu 8-bitowego i jeśli nie jest oczywiste, który kod jest używany (7-, czy 8-bitowy), to w celu zapewnienia jednoznacznej interpretacji danych należy wykorzystać sekwencję powiadamiającą odpowiednio ESC 2/0 4/10 lub ESC 2/0 4/11.

**7.3. Współdziałanie funkcji przesunięć.** Jeśli dane zakodowane w kodzie 7-bitowym i wykorzystujące środki przesunięć jednostkowych oraz przesunięć blokujących są konwertowane do postaci 8-bitowej, to zwykle reguły konwersji mogą spowodować, że bit o najstarszej wadze w kombinacji kodowej występującej bezpośrednio po SS2 lub SS3 zmieni się z ZERA na JEDEN. W tym przypadku należy przypisać znaczenie tylko siedmiu bitom o mniejszej wadze (zgodnie z 3.2.2 i 5.2.2).

Podobnie konwersja danych zakodowanych w kodzie 8-bitowym do postaci 7-bitowej i wykorzystujących środki przesunięć jednostkowych może spowodować, że funkcja przesunięcia blokującego wystąpi bezpośrednio po znaku przesunięcia jednostkowego. Znaki reprezentujące funkcję przesunięcia blokującego powinny być pominięte przy interpretacji funkcji przesunięcia jednostkowego, a występująca bezpośrednio po tym kombinacja kodowa powinna być interpretowana jako reprezentująca znak ze zbioru G2 lub G3.

**7.4. Ochrona informacji przy konwersji powrotnej.** Podczas konwersji informacji ze środowiska kodu 7-bitowego na środowisko kodu 8-bitowego nie występują trudności w zapewnieniu wielokrotnego wykorzystywania różnorodnych funkcji przywoływających. Możliwe, że w takiej sytuacji mogłyby być wykorzystane środki występujące w środowisku kodu 8-bitowego do zminimalizowania użycia funkcji przesunięć w tym środowisku. Jednak w niniejszej normie nie określono takich środków ani sposobów ich opracowania.

Podczas konwersji informacji, której źródłem jest środowisko kodu 8-bitowego, w którym wykorzystano różnorodne funkcje przesunięć określone w niniejszej normie, także nie występują trudności w reprezentowaniu informacji w środowisku kodu 7-bitowego. Jednak, jeśli istnieje potrzeba wykonania konwersji powrotnej, tzn. do środowiska kodu 8-bitowego, z zachowaniem

tych samych funkcji przesunięć, które wystąpiły poprzednio, to należy ten fakt uwzględniać w czasie pierwszej konwersji danych, tzn. ze środowiska kodu 8-bitowego na 7-bitowe. Do tego celu powinna być użyta sekwencja ESC 2/0 4/5 powiadamiająca o tym, że jest wymagana podobna ochrona lub że jest ona przewidziana.

## 8. STOSOWANIE OGRANICZNIKA METODY KODOWANIA

W zastosowaniu, w którym występuje Protokół Prezentacji lub inny ogólny poziom sterowania, początek napisu znakowego informacji zakodowanej jest wskazywany zgodnie z konwencją tego protokołu. Koniec tego napisu jest niekiedy wskazywany przez ogranicznik. Ogranicznik ten działa jak rozkaz powrotu od metody kodowania określonej w niniejszej normie do metody kodowania używanej w Protokole Prezentacji.

Ogranicznik jest określony w następujący sposób: **OGRANICZNIK METODY KODOWANIA:** funkcja sterująca, która ogranicza napis zakodowany zgodnie z niniejszą normą i która przełącza na ogólny poziom sterowania.

Reprezentacja kodowa tej funkcji to ESC 6/4.

**UWAGA.** Dla systemów kodowania innych niż określono w niniejszej normie może być dogodna funkcja ESC F<sub>c</sub>.

Wykorzystywanie ogranicznika nie jest obligatoryjne, jeśli protokół wyższego poziomu określa środki ograniczające napis, np. przez podanie długości tego napisu.

## 9. KONKRETNE ZNACZENIA SEKWENCJI ESC

W niniejszej normie nie określono konkretnych znaczeń sekwencji ESC. Są one ustalone zgodnie z procedurą obowiązującą przy rejestrowaniu zbiorów znaków (zobacz załącznik 4). Ta procedura powinna być uwzględniana w czasie przygotowywania i wdrażania sekwencji ESC i ich znaczeń.

**UWAGA.** Przypisywanie finalnych i w razie potrzeby wewnętrznych znaków wszystkim kategoriom sekwencji ESC (z wyjątkiem zarezerwowanych do przyszłego znormalizowania oraz powiadomień, DRCS, ESC F<sub>c</sub> i sekwencji do prywatnego użytku) jest wykonywane przez organizację rejestrującą zbiory znaków (zobacz załącznik 4).

## K O N I E C



## FUNKCJE PRZESUNIĘĆ

Tablica Z1-1. Reprezentacje kodowe funkcji przesunięć

Funkcja przesunięcia	Środowisko	
	kodu 7-bitowego	kodu 8-bitowego
SO	0/14	—
SI	0/15	—
LS0	—	00/15
LS1	—	00/14
LS1R	—	ESC 07/14
LS2	ESC 6/14	ESC 06/14
LS2R	—	ESC 07/13
LS3	ESC 6/15	ESC 06/15
LS3R	—	ESC 07/12
SS2	ESC 4/14	08/14
SS3	ESC 4/15	08/15

## UWAGI.

1. Jeśli reprezentacja funkcji SS2 jest wymagana w środowisku kodu 7-bitowego w postaci jednego bajtu, to powinna być ona kodowana jako kombinacja kodowa 1/9.
2. Jeśli jest wymagane reprezentowanie funkcji LS1R, LS2R i LS3R w środowisku kodu 8-bitowego, należy stosować odpowiednio sekwencje ESC 7/14, ESC 7/13 i ESC 7/12.
3. Znak DLE wg PN-89/T-42108<sup>1)</sup> jest reprezentowany przez kombinację kodową 1/0, a znak ESC — przez kombinację kodową 1/11.

Tablica Z1-2. Działanie funkcji przesunięć

Funkcja przesunięcia	Zbiór przywoływany	Strona tablicy kodu 8-bitowego, do której zbiór jest przywoływany
SO	G1	(pracuje tylko w środowisku kodu 7-bitowego)
SI	G0	(pracuje tylko w środowisku kodu 7-bitowego)
LS0	G0	lewa
LS1	G1	lewa
LS1R	G1	prawa
LS2	G2	lewa
LS2R	G2	prawa
LS3	G3	lewa
LS3R	G3	prawa
SS2	G2	lewa (jeden znak)
SS3	G3	lewa (jeden znak)

<sup>1)</sup> Występujące w załącznikach 1 ÷ 7 normy krajowe są odpowiednikami następujących norm RWPG:

PN-89/T-42108 eqv CT C9B 359-86.

PN-88/T-42109/01 eqv CT C9B 356-86.

PN-79/T-42112/01 eqv CT C9B 358-76.

## WYKAZ SEKWENCJI ESC OKREŚLONYCH W NINIEJSZEJ NORMIE

## 1. Forma zapisu sekwencji

W wykazie przedstawiono sekwencje ESC określone w niniejszej normie. Znaki w nawiasach wskazują, w jaki sposób można zwiększyć liczbę znaków wewnętrznych dla każdego typu sekwencji ESC, jeśli pierwsze serie znaków finalnych zostały wyczerpane:

*ln* ... reprezentuje jedną z kombinacji kodowych 2/1, 2/2 i 2/3,

*lm* ... reprezentuje jedną lub więcej kombinacji kodowych z pozycji 2/0 ÷ 2/15 (*lm* może nie występować).

## 2. Wykaz sekwencji ESC określonych w niniejszej normie

ESC 2/0 F .....	Powiadomienie
ESC 2/1 ( <i>ln lm</i> ) F .....	C0
ESC 2/2 ( <i>ln lm</i> ) F .....	C1
ESC 2/3 ( <i>ln lm</i> ) F .....	Jednostkowa dodatkowa funkcja sterująca
ESC 2/4 2/8 2/0 ( <i>lm</i> ) F ..	G0 DRCS wielobajtowy 94-znakowy
ESC 2/4 2/8 ( <i>ln lm</i> ) F ...	G0 wielobajtowy 94-znakowy <sup>2)</sup>
ESC 2/4 2/9 2/0 ( <i>lm</i> ) F ..	G1 DRCS wielobajtowy 94-znakowy
ESC 2/4 2/9 ( <i>ln lm</i> ) F ...	G1 wielobajtowy 94-znakowy

ESC 2/4 2/10 2/0 ( <i>lm</i> ) F	G2 DRCS wielobajtowy 94-znakowy
ESC 2/4 2/10 ( <i>ln lm</i> ) F ..	G2 wielobajtowy 94-znakowy
ESC 2/4 2/11 2/0 ( <i>lm</i> ) F	G3 DRCS wielobajtowy 94-znakowy
ESC 2/4 2/11 ( <i>ln lm</i> ) F ..	G3 wielobajtowy 94-znakowy
ESC 2/4 2/13 2/0 ( <i>lm</i> ) F	G1 DRCS wielobajtowy 96-znakowy
ESC 2/4 2/13 ( <i>ln lm</i> ) F ..	G1 wielobajtowy 96-znakowy
ESC 2/4 2/14 2/0 ( <i>lm</i> ) F	G2 DRCS wielobajtowy 96-znakowy
ESC 2/4 2/14 ( <i>ln lm</i> ) F ..	G2 wielobajtowy 96-znakowy
ESC 2/4 2/15 2/0 ( <i>lm</i> ) F	G3 DRCS wielobajtowy 96-znakowy
ESC 2/4 2/15 ( <i>ln lm</i> ) F ..	G3 wielobajtowy 96-znakowy
ESC 2/5 4/0 .....	Znormalizowane środki powrotu

<sup>2)</sup> Trzy sekwencje ESC, a mianowicie: ESC 2/4 4/0, ESC 2/4 4/1 oraz ESC 2/4 4/2 także oznaczają 94-znakowe zbiory G0 (zobacz 3.3.9).

ESC 2/5 (1n 1m) F .....	Inne systemy kodowania ze znormalizowanym powrotem <sup>1)</sup>	ESC 2/9 (1n 1m) F .....	G1 94-znakowy
ESC 2/5 2/15 (1m) F .....	Inne systemy kodowania bez znormalizowanego powrotu	ESC 2/10 2/0 (1m) F .....	G2 DRCS 94-znakowy
ESC 2/6 F .....	Modyfikacje zbiorów	ESC 2/10 (1n 1m) F .....	G2 94-znakowy
ESC 2/8 2/0 (1m) F .....	G0 DRCS 94-znakowy	ESC 2/11 2/0 (1m) F .....	G3 DRCS 94-znakowy
ESC 2/8 (1n 1m) F .....	G0 94-znakowy	ESC 2/11 (1n 1m) F .....	G3 94-znakowy
ESC 2/9 2/10 (1m) F .....	G1 DRCS 94-znakowy	ESC 2/13 2/0 (1m) F .....	G1 DRCS 96-znakowy
		ESC 2/13 (1n 1m) F .....	G1 96-znakowy
		ESC 2/14 2/0 (1m) F .....	G2 DRCS 96-znakowy
		ESC 2/14 (1n 1m) F .....	G2 96-znakowy
		ESC 2/15 2/0 (1m) F .....	G3 DRCS 96-znakowy
		ESC 2/15 (1n 1m) F .....	G3 96-znakowy
		ESC 6/4 .....	Ogranicznik metody kodowania

<sup>1)</sup> Z wyjątkiem sekwencji ESC 2/5 4/0, która reprezentuje znormalizowane środki powrotu.

## ZAŁĄCZNIK 3

## TERMINY I DEFINICJE UŻYWANE W NINIEJSZEJ NORMIE

**1. bajt** — ciąg złożony z bitów, traktowany jako jednostka, której długość nie zależy od nadmiaru ani od metod kodowania.

**2. przesunięcie blokujące** — funkcja przesunięcia, przywołująca cały oznaczony zbiór znaków graficznych, np. funkcje SI, SO, LS3.

**3. wersja kodu 7- lub 8-bitowego** — tablica kodu, dla której wszystkie reguły tworzenia zbioru znaków są zgodne odpowiednio z PN-89/T-42109/01 i PN-79/T-42112/01, a której każdy znak powinien być przydzielony jednej pozycji, przy czym co do tego istnieje wolny wybór lub dana pozycja powinna być ujawniona jako nie wykorzystywana.

**4. przywołać** — spowodować, aby oznaczony zbiór znaków był reprezentowany przez założone kombinacje kodowe za każdym razem, gdy te kombinacje wystąpią aż do chwili, gdy to zostanie przerwane przez inną funkcję rozszerzenia kodu.

**5. znak graficzny** — wg PN-89/T-42108.

**6. zbiór znaków określany dynamicznie (DRCS)** — zbiór znaków, których reprezentacje graficzne można określić za pomocą danych przesyłanych w strumieniu informacji.

**7. powiadomić** — poinformować o stosowanych środkach rozszerzania kodu za pomocą sekwencji ESC.

**8. klasa sekwencji ESC** — zbiór sekwencji ESC o tym samym pierwszym znaku wewnętrznym.

**9. kodowany zestaw znaków, kod** — wg PN-88/T-42109/01.

**10. tablica kodu** — wg PN-88/T-42109/01.

**11. kombinacja kodowa** — wg PN-88/T-42109/01.

**12. znak finalny** — znak, którego kombinacja kodowa kończy działanie sekwencji ESC.

**13. oznaczać** — identyfikować zbiór znaków, które powinny być reprezentowane w założony sposób w pew-

nych przypadkach bezpośrednio, w innych — podczas wystąpienia innej funkcji sterującej.

**14. pozycja w tablicy kodu** — wg PN-88/T-42109/01.

**15. sekwencja ESC** — sekwencja kombinacji kodowych wykorzystywana w celu sterowania w procedurach rozszerzania kodu, zawierająca dwie lub więcej kombinacji kodowych.

**16. reprezentować** — 1) wykorzystać założoną kombinację kodową o znaczeniu znaku ze zbioru znaków, który został oznaczony i przywołany lub 2) wykorzystać sekwencję ESC o znaczeniu dodatkowej funkcji sterującej.

**17. znak wewnętrzny** — znak, którego kombinacja kodowa występuje między kombinacją kodową znaku ESC a kombinacją kodową znaku finalnego w sekwencji ESC składającej się z więcej niż dwóch kombinacji kodowych.

**18. rozszerzenie kodu** — metody kodowania znaków nie występujących w zbiorze znaków danego kodu.

**19. znak** — wg PN-89/T-42108<sup>2)</sup>.

**20. znak niespacjujący** — znak, który po zobrazowaniu danego znaku nie powoduje przemieszczenia czynnej pozycji znakowej.

**21. przesunięcie jednostkowe** — funkcja przesunięcia, np. SS2, przywołująca jeden znak graficzny ze zbioru oznaczonego.

**22. znak spacjujący** — znak, który powoduje przemieszczenie czynnej pozycji o jedną pozycję znakową do przodu po zobrazowaniu danego znaku graficznego.

**23. środowisko** — cecha wskazująca na liczbę bitów wykorzystywanych do reprezentowania znaku w systemie

<sup>2)</sup> Definiowany jako suma zakresów terminów „znak graficzny” i „znak funkcyjny”, określanych tamże na podstawie ich przeznaczenia.

przetwarzania lub transmisji danych, lub w części takiego systemu.

**24. funkcja sterująca** — oddziaływanie na zapis, przetwarzanie, przesyłanie lub interpretację danych, mające zakodowaną reprezentację w postaci dwóch lub większej liczby kombinacji kodowych.

**25. znak funkcyjny** — wg PN-89/T-42108.

**26. funkcja przesunięcia** — ogólna nazwa funkcji sterujących, np. SI, SO, LS2, SS3, które przywołują zbiór znaków graficznych lub jeden znak graficzny.

## ZAŁĄCZNIK 4

## PROCEDURY REJESTRACJI ZBIORÓW ZNAKÓW

1. W ramach RWPG rejestracją zbiorów znaków funkcyjnych i graficznych (odpowiednio C0, C1, G0, G1, G2, G3) oraz jednostkowych funkcji sterujących zajmuje się Komisja Międzyrządowa ds. Elektronicznej Techniki Obliczeniowej, działając zgodnie z odpowiednią procedurą określoną w materiale normatywnym 33-80 „Komputery i systemy przetwarzania danych. Znaki alfanumeryczne. Procedury rejestracji”. Funkcję rejestratora zbiorów znaków pełni Centrum Koordynacyjne tej Komisji.
2. W ramach ISO rejestracja zbiorów znaków jak wyżej jest realizowana zgodnie z normą ISO 2375 przez

International Registration Authority, którego funkcję pełni ECMA [114, Rue du Rhone, CH-1204, Geneve].

3. Sekwencje ESC  $F_s$  są rejestrowane w Rejestrze RWPG i w Międzynarodowym Rejestrze ISO obejmującym zbiory znaków przeznaczone do wykorzystania wraz z oznaczającymi je sekwencjami ESC. Każda sekwencja ESC  $F_s$  przeznaczona do zarejestrowania powinna być zatwierdzona przez ISO TC 97/SC2. Kodowanie znaku finalnego  $F_s$  jest realizowane przez organizację rejestrującą zgodnie z procedurą określoną w normie ISO 2375.

## ZAŁĄCZNIK 5

## FORMY ZAPISU I OZNACZANIA POZYCJI TABLICY KODU

1. W niniejszej normie obowiązuje następujący zapis i oznaczenie pozycji tablicy kodu:

Bit y kombinacji kodowych kodu 7-bitowego	—	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$
Bit y kombinacji kodowych kodu 8-bitowego	$b_8$	$b_7$	$b_6$	$b_5$	$b_4$	$b_3$	$b_2$	$b_1$
Wagi bitów względem kolumn i wierszy w tablicy kodowej	$2^3$	$2^2$	$2^1$	$2^0$	$2^3$	$2^2$	$2^1$	$2^0$
	Kolumna				Wiersz			

2. Do kombinacji kodowej można odsyłać przez wskazanie numeru kolumny i wiersza tej pozycji w tabli-

cy kodu. Numer kolumny jest dziesiętnym ekwiwalentem bitów  $b_7 \div b_5$  (lub  $b_8 \div b_5$ ), numer wiersza jest dziesiętnym ekwiwalentem bitów  $b_4 \div b_1$ , przy czym bitom są przypisane wagi podane wyżej w p. 1. Numery kolumn i wierszy są oddzielone kreską ułamkową (“/”).

3. W kodzie 8-bitowym dziesiętne ekwiwalenty bitów są zwyczajowo poprzedzane zerem dopisywanym dla numerów kolumn i wierszy od 00 ÷ 09. Przykład: pozycja znaku SPACJA w kodzie 7-bitowym jest zapisywana jako 2/0, w kodzie 8-bitowym — jako 02/00.

## ZAŁĄCZNIK 6

## STRUKTURA KODU 7-BITOWEGO

Tablica kodu 7-bitowego wg PN-89/T-42109/01 składa się z następujących obszarów przewidzianych dla uporządkowanych zbiorów znaków funkcyjnych i graficznych przedstawionych na rysunku:

1) kolumny 0 i 1 zawierają zbiór 32 znaków funkcyjnych,

2) znak SPACJA (SP) na pozycji 2/0, który może być traktowany jako znak funkcyjny albo jako znak graficzny,

3) kolumny 2 ÷ 7, pozycje 2/1 ÷ 7/14 zawierają 94 znaki graficzne,

4) znak KASOWANIE (DEL) umieszczony na pozycji 7/15.

	0	1	2	3	4	5	6	7
0			SP					
1								
2								
3								
4								
5								
6								
7								
8								
9								
10								
11								
12								
13								
14								
15								DEL

## STRUKTURA KODU 8-BITOWEGO

Tablica  $16 \times 16$  złożona z kolumn ponumerowanych od 00 do 15 i wierszy ponumerowanych od 00 do 15 zawiera 256 pozycji kodowych (zobacz rysunek).

Kolumny 00 ÷ 07 zawierają 128 pozycji dla znaków pozostających w jednoznacznej odpowiedzialności ze znakami kodu 7-bitowego. Ich reprezentacje kodowe są takie same, jak w środowisku kodu 7-bitowego, z tym że dołączono ósmy bit o największej wadze i znaczeniu ZERO.

Kolumny 08 ÷ 15 tablicy zawierają 128 pozycji — ósmym bitem ich reprezentacji kodowych jest JEDEN.

Kolumny 08 i 09 są przewidziane dla znaków funkcyjnych, a kolumny 10 ÷ 15 — dla znaków graficznych.

Wśród znaków funkcyjnych w kolumnach 08 i 09 kodu 8-bitowego nie powinny występować znaki sterowania transmisją.

	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15
00			SP								10/00					
01																
02																
03																
04																
05																
06																
07																
08																
09																
10																
11																
12																
13																
14																
15								DEL								15/15

BN-89/3101-06-77

Struktura kodu 8-bitowego

KONIEC NORMY MIĘDZYNARODOWEJ

### INFORMACJE DODATKOWE

**1. Instytucja opracowująca normę** — Instytut Maszyn Matematycznych.

**2. Istotne zmiany w stosunku do BN-76/3101-06.** Wprowadzono nowe funkcje sterujące rozszerzeniem kodu: LS0, LS1, LS2, LS3, LS1R, LS2R, LS3R, SS2 i SS3, w związku z czym zwiększono możliwości wprowadzania do kodów 7- i 8-bitowego nowych znaków funkcyjnych i graficznych. Zwiększono repertuar sekwencji ESC jako sekwencji oznaczających (i/lub przywołujących) oraz jako powiadomień o stosowanych środkach rozszerzania kodu. Wprowadzono sekwencje ESC oznaczające i przywołujące inne systemy kodowania ze znormalizowanym powrotem do systemu kodowania zgodnego z niniejszą normą, a także sekwencję ESC 6/4 działającą jak rozkaz powrotu od metody kodowania zgodnej z niniejszą normą do metody kodowania Protokołu Prezentacji lub innego ogólnego poziomu sterowania.

#### 3. Normy związane

PN-89/T-42108 Przetwarzanie informacji i komputery. Znaki alfanumeryczne. Klasyfikacja, nazwy i symbole (eqv CT CЭB 359-86)  
 PN-88/T-42109/01 Przetwarzanie informacji i komputery. Kod 7-bitowy. Tablica kodu i zestawy znaków ISO i RWPG (eqv CT CЭB 356-86)

PN-84/T-42109/02 Przetwarzanie informacji i komputery. Kod 7-bitowy. Krajowe zestawy znaków

PN-86/T-42109/03 Przetwarzanie informacji i komputery. Kod 7-bitowy. Krajowy zestaw znaków wprowadzany techniką rozszerzenia kodu

PN-79/T-42112/01 Przetwarzanie informacji i komputery. Kod 8-bitowy. Tablica kodu i zestawy znaków ISO i RWPG (eqv CT CЭB 358-76)

#### 4. Normy międzynarodowe

RWPG CT CЭB 360-86 Системы обработки информации. Наборы символов в 7- и 8-битных кодах. Методы расширения кодов — норма zgodna, z wyjątkami omówionymi w „Przedmowie

ISO 2022 (1986) Information processing — ISO 7-bit and 8-bit coded character sets — Code extension techniques — norma zgodna w przedstawionym zakresie.

**5. Autorka projektu normy** — mgr Krystyna Radzimowska — Instytut Maszyn Matematycznych.