

## **Karta grafiki.**

### **1.1. Wstęp.**

Jest to bardzo rozbudowane zagadnienie, na którego temat zostało napisane wiele książek. Aby zająć się programowaniem sterowników kart graficznych, należy zdać sobie sprawę z wielu zagadnień. Dlatego w niniejszym opracowaniu dokonano daleko idących uogólnień, a od strony programowej, ograniczono się do bardzo wąskiego zakresu. Oprogramowanie nawet najprostszego sterownika grafiki, byłoby tematem na napisanie obszernej książki. BIOS dostarcza bardzo wielu procedur do testowania, ustawiania i używania różnego rodzaju kart, oraz ich trybów pracy. W poniższym opracowaniu zwrócono uwagę na kartę VGA. Od strony programowej opisano tylko funkcje do przedstawiania danych na ekranie a nie konfiguracji sterownika. Nie należy zapominać, że karty graficzne posiadają także swój BIOS.

### **1.2. Rejestry sterownika VGA.**

Sterownik VGA posiada szereg rejestrów (odczytu, zapisu, stanu, sterowania - standard EGA posiadał ich kilkadziesiąt, natomiast VGA kilkaset). Można je podzielić na pewne grupy:

- Zespół rejestrów zewnętrznych.
- Rejestry sekwencyjne, dotyczące układu odpowiadającego za generowanie sygnału zegarowego, przesyłanie danych pomiędzy pamięcią obrazu, układem graficznym i układem określania atrybutu oraz za lokalizację wyświetlanego zbioru znaków.
- Rejestry graficzne, dotyczące układu odpowiadającego za przekazywanie danych pomiędzy pamięcią obrazu a procesorem i układem określania atrybutu.
- Rejestry sterownika atrybutów, dotyczące układu służącego do zmiany kolorów zapisanych w pamięci obrazu na indeksy kolorów zdefiniowanych w rejestrach wzorców kolorów.
- Rejestry sterowania wyświetlaczem. Czyli rejestry kontrolera CRT odpowiadającego za zachowanie zależności czasowych podczas wyświetlania obrazu oraz wyświetlającego kursor.
- Rejestry przetwornika cyfrowo-analogowego, który przetwarza wzorce cyfrowe na sygnał analogowy przesyłany do monitora.

### **1.3. Pamięć obrazu.**

Standardowy sterownik VGA posiada 256 KB pamięci RAM, natomiast w przestrzeni adresowej procesora, na pamięć obrazu przeznaczone jest 128 KB. Z tego powodu pamięć obrazu zorganizowana jest na różne sposoby, w zależności od trybu pracy sterownika.

#### **1.3.1. Odwzorowanie pamięci obrazu.**

Pamięć obrazu jest bezpośrednio odwzorowana w przestrzeni adresowej procesora, pomiędzy adresami A000:0000h a B000:FFFFh. Z punktu widzenia procesora, karta graficzna to zbiór portów we/wy i obszar pamięci obrazu, więc programowanie sterownika sprowadza się do zapisu oraz odczytu do i z pamięci oraz portów we/wy. Zapis lub odczyt wymaga dodatkowo programowania rejestrów układu graficznego. Spowodowane jest to właśnie organizacją pamięci obrazu. („Anatomia PC”, P. Metzger, A. Jałowicki).

#### **1.3.2. Metoda płatowa.**

W karcie VGA pamięć dzielona jest na 4 bloki po 64 KB. W każdym bloku jednemu bitowi odpowiada jeden punkt - dzięki takiej strukturze 256 KB pamięci obrazu zajmuje 64 KB przestrzeni adresowej. Metoda ta nazywana jest płatową. W trybach graficznych o płatowej organizacji pamięci obrazu, odczyt lub zapis komórek pamięci może być różnie zinterpretowany przez poszczególne układy karty. Z tego powodu wprowadzone zostały cztery tryby zapisu i dwa tryby odczytu.

#### **1.3.3. Stronicowanie pamięci.**

Innym sposobem organizacji pamięci jest stronicowanie. W trybach graficznych SVGA w obrębie banku, odwzorowanie pamięci obrazu jest identyczne jak w analogicznych trybach VGA. We wszystkich metodach stronicowania ważnym parametrem jest odstęp pomiędzy początkami kolejnych banków, może być on mniejszy niż rozmiar banku, co oznacza, że kolejne banki mają część wspólną.

#### Metody stronicowania:

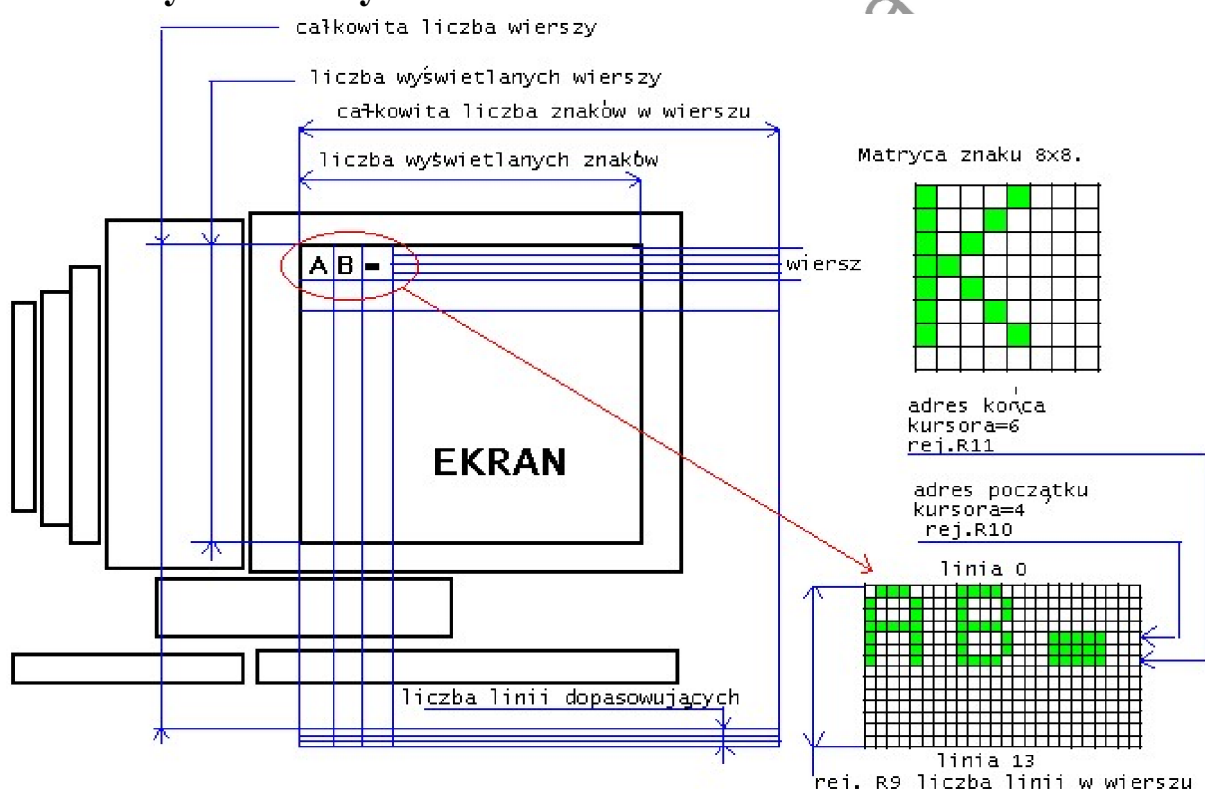
- Stronicowanie pamięci techniką pojedynczego okna.
- Stronicowanie pamięci techniką dwóch nakładających się okien.
- Stronicowanie pamięci techniką dwóch nie nakładających się okien.

## 1.4. Tryb tekstowy.

### 1.4.1. Matryce znaków.

Rozmiary tych matryc uzależnione są od rodzaju karty graficznej oraz ustawionego trybu, w jakim ta karta pracuje. Standardowe matryce zapisane są w pamięci stałej sterownika graficznego. W kartach EGA i VGA można zmienić zestaw znaków oferowanych przez sterownik. Podaje się inne miejsce w pamięci, z którego pobierane będą wyświetlane znaki, dzięki funkcji 11h (podfunkcje 00h i 10h). W innych kartach trzeba zmieniać pamięć stałą na karcie grafiki. (*Rys. 1.1.*)

**Rys. 1.1. Tryb tekstowy na ekranie monitora.**



CGA	w trybach od 0 do 3	matryca 8x8.
EGA	w trybach od 0 do 3 w trybie 7	matryca 8x14, matryca 9x14.
MDA	w trybie 7	matryca 9x14.
VGA	we wszystkich trybach	matryca 9x16.

### 1.4.2. Tekstowe tryby pracy kart.

Tryb numer: 0h.

Rozdzielczość: 40x25.  
Liczba kolorów: 16/8 szarości.

Karta graficzna: CGA, EGA, VGA.

Adres początku pamięci obrazu: B8000h.

Tryb numer: 1h.

Rozdzielczość: 40x25.

Liczba kolorów: 16/8.

Karta graficzna: CGA, EGA, VGA.

Adres początku pamięci obrazu: B8000h.

Tryb numer: 2h.

Rozdzielczość: 80x25.

Liczba kolorów: 16/8 szarości.

Karta graficzna: CGA, EGA, VGA.

Adres początku pamięci obrazu: B8000h.

Tryb numer: 3h.

Rozdzielczość: 80x25.

Liczba kolorów: 16/8.

Karta graficzna: CGA, EGA, VGA.

Adres początku pamięci obrazu: B8000h.

Tryb numer: 7h.

Rozdzielczość: 80x25.

Liczba kolorów: 3.

Karta graficzna: MDA, EGA, VGA.

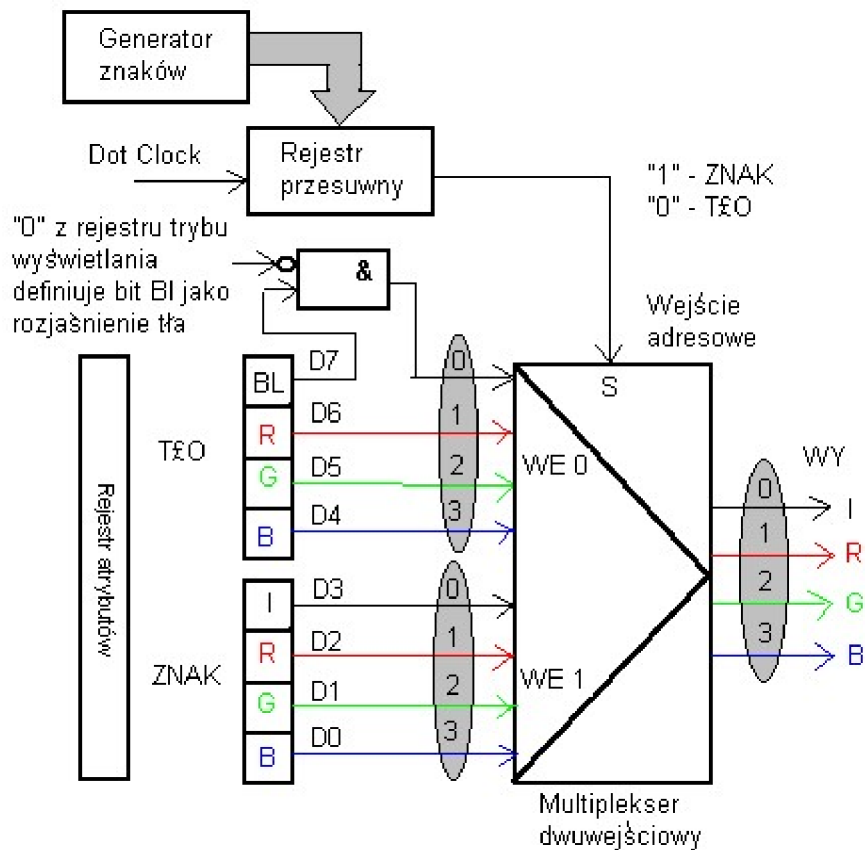
Adres początku pamięci obrazu: B0000h.

Sterowniki SVGA posiadają dodatkowe tryby tekstowe (np. 80x30, 80x43, 80x60, 132x25, 132x30, 132x43, 132x60, wszystkie w 16 kolorach).

### 1.4.3. Reprezentacja znaku.

Każdy znak reprezentowany jest przez dwa bajty pamięci obrazu. Pierwszy z nich zawiera kod ASCII wyświetlanego znaku, drugi zaś określa jego atrybut. Wyświetlanie znaków odbywa się bardzo szybko, gdyż do zapisania jednego znaku wymagane jest przesłanie tylko dwóch bajtów. W pamięci obrazu kody znaków zapisane są na przemian z atrybutami. W pamięci obrazu można przechowywać do 8 stron tekstu, w zależności od ilości zainstalowanej pamięci i trybu pracy. (Rys. 1.2.)

**Rys. 1.2. Układ kodera kolorów – wyświetlenie tekstu w kolorze.**



## 1.5. Tryb graficzny.

### 1.5.1. Podstawowe tryby pracy.

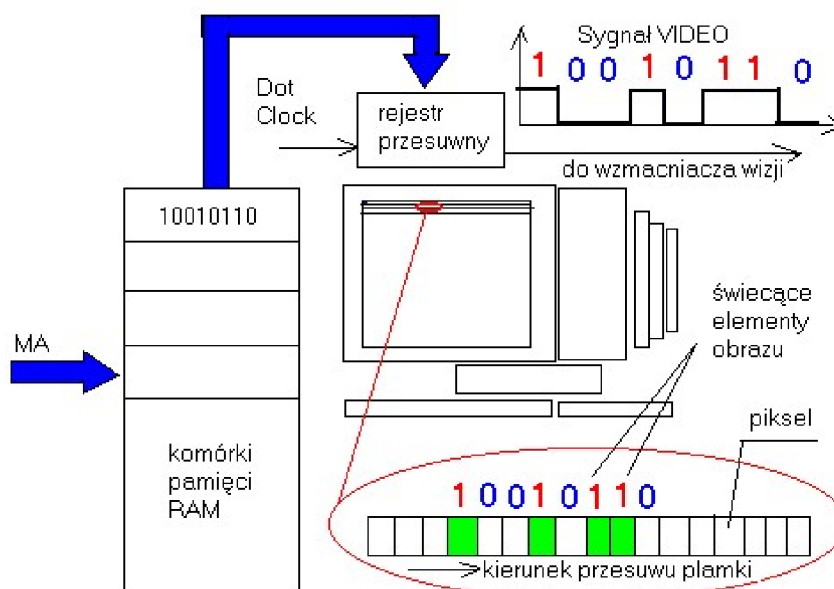
numer trybu	rozdzielczość	liczba kolorów	rozmiar znaku	segment pamięci obrazu	zgodność ze sterownikiem
04h	320x200	4	8x8	B800	CGA, EGA
05h	320x200	4 (szarości)	8x8	B800	CGA, EGA
06h	640x200	2	8x8	B800	CGA, EGA
0Dh	320x200	16	8x8	A000	EGA, VGA
0Eh	640x200	16	8x8	A000	EGA, VGA
0Fh	640x350	3	8x14	A000	EGA, VGA
10h	640x350	4	8x14	A000	EGA, VGA
11h	640x480	2	8x16	A000	VGA
12h	640x480	16	8x16	A000	VGA
13h	320x200	256	8x16	A000	VGA

Sterowniki SVGA oferują dodatkowe tryby graficzne (np. 640x480 256kol, 800x600 16/256kol, 1024x768 16/256kol).

### 1.5.2. Opis punktu na ekranie.

We wszystkich trybach graficznych kolejne komórki pamięci obrazu reprezentują kolejne punkty położone w kolejnych liniach poziomych, odczytywane z lewa na prawo i od góry do dołu (**Rys. 1.3.**). W trybach dwukolorowych pojedynczy punkt opisywany jest przez 1 bit. Osiem kolejnych punktów opisywanych jest przez jeden bajt pamięci obrazu, przy czym najbardziej znaczący bit znajduje się z lewej strony, zaś najmniej znaczący z prawej. W trybach czterokolorowych pojedynczy punkt opisywany jest dwoma bitami, a w trybach szesnastokolorowych-czterema. Specyficznym trybem jest tryb 13h, który był często stosowany w grach pomimo niskiej rozdzielczości. Jest to tryb 256-kolorowy, w którym pojedynczy punkt opisany jest jednym bajtem. Do umieszczenia punktu na ekranie nie trzeba wykonywać żadnych złożonych operacji, wystarczy policzyć przesunięcie względem początku pamięci obrazu.

**Rys. 1.3. Tryb graficzny na ekranie monitora.**





## 1.6. Zagadnienia praktyczne.

Do wszystkich funkcji BIOS-u sterownika można odwołać się poprzez przerwanie programowe **INT 10h**. („Anatomia PC”, P Metzger A. Jałowicki, „Encyklopedia informatyki”, S. Kruk).

Funkcje te można podzielić tematycznie na:

- funkcje określające tryb pracy i ogólne parametry sterownika (00h-07h oraz 0Fh),
- funkcje dostępu do ekranu (08h-0Eh),
- funkcje służące do definiowania kolorów (10h) (Rys. 1.2. i Rys. 1.3.),
- funkcje generatora znaków (11h),
- funkcje konfigurujące sterownik (12h),
- funkcje uzupełniające (13h, 1Ah-1Ch).

### 1.6.1. Funkcje określające tryb pracy i ogólne parametry sterownika.

**INT 10 00h** - Rozkaz ten ustawia tryb graficzny

AH 00h

AL ustawiany tryb video

Bit 7      0 – przy zmianie trybu pamięć zostanie wyzerowana

Bit 7      1 – zawartość pamięci obrazu pozostanie bez zmian

Bity 6-0    numer trybu (np. 13h to 320x200 przy 256 kolorach)

**INT 10 02h** - Rozkaz ten ustawia kursor w danej pozycji na ekranie

AH 02h

BH numer strony

Od 0 do 1 w trybach 0Fh-10h

Od 0 do 3 w trybie 0Eh

Od 0 do 7 w trybach 00h-03h, 07h i 0Dh

W trybach 04h-06h i 11h-13h istnieje tylko jedna strona graficzna o numerze 0.

DH Y

DL X

**INT 10 03h** - Rozkaz ten podaje pozycję kursora

AH 03h

BH numer strony

Wynik:

AX 0000h (Phoenix BIOS)

CH numer górnej linii kursora

CL numer dolnej linii kursora

DH Y kursora

DL X kursora

**INT 10 05h** - Rozkaz ten ustawia obecnie wyświetlaną stronę AH  
05h

AL numer nowej strony (00h - numer pierwszej strony)

**INT 10 06h** - Rozkaz ten przewija tekst w górę

AH 06h

AL wartość przesunięcia w wierszach

BH atrybut spacji wypełniających puste linie

CH numer wiersza lewego górnego rogu obszaru

CL numer kolumny lewego górnego rogu obszaru

DH numer wiersza prawego dolnego rogu obszaru

DL numer kolumny prawego dolnego rogu obszaru

Wiersze przewinięte za górną krawędź obszaru zostają utracone. Podanie wartości AL=0 powoduje wypełnienie spacjami (wymazanie) całego podanego obszaru. Pozycja kursora pozostaje bez zmian.

**INT 10 07h** - Rozkaz ten przewija tekst w dół AH

07h

AL wartość przesunięcia w wierszach

BH atrybut spacji wypełniających puste linie

CH numer wiersza lewego górnego rogu obszaru

CL numer kolumny lewego górnego rogu obszaru

DH numer wiersza prawego dolnego rogu obszaru

DL numer kolumny prawego dolnego rogu obszaru

Wiersze przewinięte za dolną krawędź obszaru zostają utracone. Podanie wartości AL=0 powoduje wypełnienie spacjami (wymazanie) całego podanego obszaru. Pozycja kursora pozostaje bez zmian.

**INT 10 0Fh** - Rozkaz ten podaje bieżący tryb graficzny

AH 0Fh

Wynik:

AH liczba kolumn znakowych na ekranie

AL aktualny tryb

BH aktywna strona



### Przykład:

Funkcja zwraca dodatkowo liczbę wierszy na ekranie pobraną spod adresu w (BDA) obszarze danych BIOS. („*BIOS dla programujących w językach C i C++*”, B. Taylor).

```
void odczytajParametry(unsigned char *tryb,
unsigned char *strona,
unsigned char *kolumny,
unsigned char *wiersze ) {
#define BDA_rows (unsigned int far *)0x00400084
union REGS regs;

regs.h.ah = 0x0f;
/* AH = 0x0F - odczytaj parametry trybu pracy */
int86(0x10, &regs, &regs);
/* Wywołanie funkcji BIOS */
*tryb = regs.h.al; /* AL = bieżący tryb */
*strona = regs.h.bh; /* BH = wyświetlana strona */
*kolumny = regs.h.ah; /* AH = liczba
wyświetlanych kolumn */
*wiersze = *BDA_rows + 1;
/* Liczba wierszy z BDA */ }

```

### 1.6.2. Funkcje dostępu do ekranu.

**INT 10 08h** - Rozkaz ten odczytuje znak z ekranu i jego atrybut z pozycji kursora

AH 08h

BH numer strony (00h numer pierwszej strony) Wynik:

AH atrybut znaku (tylko tryb tekstowy)

AL znak (jego kod ASCII)

**INT 10 09h** - Rozkaz ten pisze znak na pozycji kursora z uwzględnieniem atrybutu

AH 09h

AL znak do wyświetlenia (kod ASCII)

BH numer strony (00h - numer strony pierwszej)

BL atrybut (tryb tekstowy) lub kolor (tryb graficzny), jeśli 7 bit jest ustawiony w <256-kolorowym trybie graficznym, znak jest XOR'owany z obrazem

CX liczba znaków - ile razy ma być powtórzony znak

**INT 10 0Ah** - Rozkaz ten pisze znak na pozycji kursora

AH 0Ah

AL wyświetlany znak (kod ASCII)

BH numer strony (00h - numer strony pierwszej)

BL atrybut (PCjr, Tandy 1000 tylko) lub kolor (tryb graficzny), jeśli bit 7 jest zapalony w <256-kolorowym trybie graficznym, znak jest XOR'owany z obrazem

CX liczba znaków - ile razy ma być powtórzony znak

**INT 10 0Bh** - Rozkaz ten ustawia kolor ekranu lub tła tekstu

AH 0Bh

BH 00h

BL kolor ekranu (tryb graficzny 0-15) lub tło tekstu (tryb tekstowy 0-31) lub

BH 01h wybór palety kolorów

BL numer palety

**INT 10 0Ch** - Rozkaz ten zapala piksel graficzny

AH 0Ch

BH numer strony

AL kolor piksela, jeśli bit 7(ostatni) jest zapalony, wartość jest XOR'owana z obrazem w <256-kolorowych trybach

CX Y

DX X

**INT 10 0Dh** – Rozkaz ten odczytuje kolor piksela na ekranie graficznym

AH 0Dh

BH numer strony

CX Y

DX X

Wynik:

AL odczytany kolor piksela

**INT 10 0Eh** – Rozkaz ten wyświetla znak i przemieszcza kursor na następną pozycję

AH 0Eh

AL kod ASCII znaku

BH numer strony

BL w graficznych trybach pracy numer koloru znaku

Funkcja działa zarówno w trybach graficznych jak i tekstowych. Brane są pod uwagę niektóre kody sterujące: 07h-sygnal dźwiękowy BEL. 08h - cofnięcie kursora BS, 0Dh-powrót na początek linii CR i 0Ah-przejsie do nowej linii LF, natomiast pozostałe kody sterujące wyświetlane są jak normalne znaki.

### 1.6.3. Funkcje służące do definiowania kolorów.

**INT 10 10h** – operacje na paletce kolorów (AH = 10h)

**Podfunkcje (AL = ...):**

**00h** – ustawienie jednego koloru palety.

AH 10h

AL 00h

BH numer wzorca koloru

BL numer koloru w paletce

Zmiana koloru następuje przez wpisanie numeru wzorca koloru do określonego rejestru palety.

**01h** – zmiana koloru krawędzi ekranu.

AH 10h

AL 01h

BH numer wzorca koloru

**02h** – zmiana kolorów palety i krawędzi ekranu.

AH 10h

AL 02h

ES:DX adres obszaru zawierającego tablicę kolorów (17 bajtów=16 kolorów + krawędź).

Zmiana kolorów następuje przez wpisanie numeru wzorca kolorów do kolejnych rejestrów palety oraz do rejestru koloru krawędzi ekranu. Pierwsze 16 bajtów wskazywanego obszaru powinno zawierać numery kolejnych wzorców kolorów (0-15), a bajt o adresie ES:DX+16 powinien zawierać numer wzorca koloru krawędzi ekranu. Standardowe kolory w trybach 16-kolorowych:

0-czarny, 1-niebieski, 2-zielony, 3-siny, 4-czerwony, 5-fioletowy, 6-brązowy, 7-jasnoszary, 8-szary, 9-jasnoniebieski, 10-jasnozielony, 11-jasnosiny, 12-jasnoczerwony, 13-różowy, 14-żółty, 15-biały.

**03h** – ustawienie sposobu interpretacji atrybutu znaku.

Funkcja używana tylko w trybach tekstowych.

AH 10h

AL 03h

Gdy BH = 1

Bit 7 = 1 - znak będzie migotał

Bit 7 = 0 - znak nie będzie migotał

Bity 6-4 - numer koloru tła (0-7)

Bity 3-0 numer koloru znaku (0-15)

Gdy BH = 0

Bity 7-4 numer koloru tła (0-15)

Bity 3-0 numer koloru znaku (0-15)

**07h** – pobranie pojedynczego koloru palety.

Zwracana informacja jest odczytywana z odpowiedniego rejestru palety.

AH 10h

AL 07h

Wyjście:

BH numer wzorca koloru

**08h** – pobranie koloru krawędzi ekranu.

Zwracana informacja jest odczytywana z rejestru koloru krawędzi ekranu.

AH 10h

AL 08h

Wyjście:

BH numer wzorca koloru krawędzi ekranu **09h** –  
pobranie kolorów palety i krawędzi ekranu.

AH 10h

AL 09h

ES:DX adres obszaru pamięci do przechowania informacji o  
kolorach (17-bajtów)

Zwracana informacja jest odczytywana z kolejnych rejestrów palety oraz rejestru koloru krawędzi ekranu i umieszczana we wskazanym obszarze pamięci. Pierwsze 16 bajtów zawiera numery kolejnych wzorców kolorów

(0-15), a bajt o adresie ES:DX+16 zawiera numer wzorca koloru krawędzi ekranu.

**10h** – określenie pojedynczego wzorca koloru.

AH 10h

AL 10h

BX numer wzorca koloru (0-255)

DH składowa R czerwona koloru (0-63)

CH składowa G zielona koloru (0-63)

CL składowa B niebieska koloru (0-63)

Podfunkcja służy do określenia wzorca koloru poprzez podanie trzech składowych koloru składających się na 18-bitowy wzorec. Wpisywane składowe powinny zawierać się w przedziale 0-63, wartości większe w standardowym sterowniku VGA zostaną obcięte do 6 bitów. Jeśli zostało włączone przeliczanie barw do odcieni szarości, to wzorec koloru zostanie przeliczony do odpowiedniego odcienia szarości, wybranego z 64 poziomów.

**12h** – określenie bloku wzorców kolorów.

AH 10h

AL 12h

BX numer pierwszego wzorca koloru w bloku (0-255)

CX liczba ustawianych wzorców

ES:DX adres obszaru zawierającego kolejne wzorce kolorów

**13h(BL=00h)** – ustawienie liczby bloków wzorców kolorów.

AH 10h

AL 13h

BL 00h

BH 1-16 bloków po 16 wzorców kolorów

0-4 bloki po 64 wzorce kolorów

**13h(BL=01h)** – ustawienie domyślnego bloku wzorców kolorów.

AH 10h

AL 13h

BL 01h

BH numer domyślnego bloku wzorców kolorów

**15h** – pobranie pojedynczego wzorca koloru.

AH 10h

AL 15h

Wyjście:

DH składowa R,  
CH składowa G,  
CL składowa B,

**17h** – pobranie bloku wzorców kolorów.

AH 10h

AL 17h

BX numer pierwszego pobieranego wzorca koloru

CX liczba pobieranych wzorców koloru

ES:DX adres obszaru pamięci, w którym zapisane zostaną składowe kolorów

**1Ah** – pobranie informacji o blokach wzorców kolorów.

AH 10h

AL 1Ah

Wyjście:

BH numer aktywnego bloku wzorców koloru

BL 1 - bloki wzorców kolorów zgrupowane są w 16 bloków po 16 wzorców

0 - bloki wzorców kolorów zgrupowane są w 4 bloki po 64 wzorce

**1Bh** – przekształcenie do poziomów szarości.

AH 10h

AL 1Bh

BX numer pierwszego przeliczanego wzorca kolorów

CX liczba przeliczonych wzorców kolorów

Dla wszystkich wzorców w bloku zostaje policzona wartość poziomu szarości. Każda barwa składowa zastępowana jest sumą ważoną trzech barw, składającą się z 0.3 składowej R, 0.59 –G, oraz 0.11 składowej B. Pierwotne wartości składowych zostają utracone.

#### **1.6.4. Funkcje generatora znaków. (Rys. 1.1.)**

**INT 10 11h** – działania na generatorze znaków (AH = 11h)

***Podfunkcje (AL = ...):***

**00h** – definiowanie znaków tekstowego trybu pracy

Podfunkcja służy do zastąpienia zbioru znaków lub jego części znakami zdefiniowanymi przez użytkownika.



AH 11h

AL 00h

BH liczba bajtów opisujących pojedynczy znak

BL numer zastępowanego zbioru znaków

CX liczba zastępowanych znaków

DX kod ASCII pierwszego zmienianego znaku

ES:BP adres obszaru zawierającego definicje znaków użytkownika

Liczba bajtów opisujących pojedynczy znak waha się w zależności od trybu od 8 do 16, numer zastępowanego zbioru znaków należy do przedziału 0-7. Podczas wywołania funkcji może nastąpić zmiana trybu pracy sterownika, jednak zawartość pamięci obrazu pozostaje nienaruszona.

**01h** – załadowanie znaków standardowych 8x14 (zapisanym w pamięci ROM ster. VGA)

AH 11h

AL 01h

BL numer zbioru znaków

Podczas wywołania funkcji może nastąpić zmiana trybu pracy sterownika, jednak zawartość pamięci obrazu pozostaje nienaruszona.

**02h** – załadowanie znaków standardowych 8x8

AH 11h

AL 02h

BL numer zbioru znaków

Podczas wywołania funkcji może nastąpić zmiana trybu pracy sterownika, jednak zawartość pamięci obrazu pozostaje nienaruszona.

**03h** – ustawianie domyślnego zbioru znaków

AH 11h

AL 03h

BL bity 4, 1, 0 – numer pierwszego zbioru znaków

bity 5, 3, 2 – numer drugiego zbioru znaków

Wybrany zbiór musi zostać przedtem załadowany podfunkcjami 00h, 01h, 02h, 10h, 11h, 12h, lub 14h funkcji 11h. W tekstowych trybach pracy można stosować zbiory 256-znakowe lub 512-znakowe. Dla zbioru 256znakowego numery pierwszego i drugiego zestawu znaków muszą być takie same. W zbiorach 512-znakowych pierwsze 256 znaków zostanie wybrane, gdy 3 bit atrybutu wynosi 0, a pozostałe 256 znaków – gdy 3 bit=1.

**04h** – załadowanie znaków standardowych 8x16

AH 11h

AL 04h

BL numer zbioru znaków

Podczas wywołania funkcji może nastąpić zmiana trybu pracy sterownika, jednak zawartość pamięci obrazu pozostaje nienaruszona.

**10h** – definicja znaków dla tekstowych trybów pracy

AH 11h

AL 10h

BH liczba bajtów opisujących pojedynczy znak

BL numer modyfikowanego zbioru znaków

CX liczba zastępowanych znaków

DX kod ASCII pierwszego zastępowanego znaku

ES:BP adres obszaru pamięci zawierającego definicję znaków

Podfunkcja 10h służy do modyfikowania zestawów znaków o różnych wymiarach (w odróżnieniu od podfunkcji 00h). Powinna być wywołana zaraz po ustawieniu tekstowego trybu pracy i podczas wyświetlania strony zerowej.

**11h** – załadowanie znaków standardowych 8x14

AH 11h

AL 11h

BL numer zbioru znaków

Załadowany zbiór znaków odpowiada rozmiarami zbiorowi znaków sterownika MDA, ale może on zastąpić zbiory znaków o innych rozmiarach. Podfunkcja powinna być wywołana zaraz po ustawieniu tekstowego trybu pracy i podczas wyświetlania strony 0.

**12h** – załadowanie znaków standardowych 8x8

AH 11h

AL 12h

BL numer zbioru znaków

Podfunkcja służy do zastąpienia zbioru znaków o wymiarach innych niż 8x8. Podfunkcja powinna być wywołana zaraz po ustawieniu tekstowego trybu pracy i podczas wyświetlania strony 0.

**14h** – załadowanie znaków standardowych 8x16

AH 11h

AL 14h

BL numer zbioru znaków

Podfunkcja służy do zastąpienia zbioru znaków o wymiarach innych niż 8x16. Podfunkcja powinna być wywołana zaraz po ustawieniu tekstowego trybu pracy i podczas wyświetlania strony 0.

**20h** – modyfikowanie zestawu w graficznych trybach pracy (>127)

AH 11h

AL 20h

ES:BP adres obszaru zawierającego opis znaków

Adres wskazywany przez rejestry ES:BP jest zapisywany w tablicy wektorów przerwań jako wektor 1Fh. Zmodyfikowane znaki są dostępne w graficznych trybach pracy 04h-06h. Podfunkcja powinna być wywołana zaraz po ustawieniu graficznego trybu pracy.

**21h** - modyfikowanie zestawu w graficznych trybach pracy

AH 11h

AL 21h

BL liczba wierszy na ekranie zakodowana następująco:

0 - liczba wierszy podana w DL

1 - 14 wierszy

2 – 25 wierszy

3 – 43 wiersze

CX liczba bajtów opisujących pojedynczy znak

DL gdy BL=0 zawiera liczbę wierszy na ekranie

ES:BP adres obszaru zawierającego opis znaków

Adres wskazywany przez rejestry ES:BP jest zapisywany w tablicy wektorów przerwań jako wektor 43h. Podfunkcja powinna być wywołana zaraz po ustaleniu graficznego trybu pracy.

**22h** - załadowanie znaków standardowych 8x14

AH 11h

AL 22h

BL liczba wierszy na ekranie zakodowana następująco:

0 - liczba wierszy podana w DL

1 - 14 wierszy

2 – 25 wierszy

3 – 43 wiersze

DL gdy BL=0 zawiera liczbę wierszy na ekranie

Adres zestawu znaków 8x14 jest zapisywany w tablicy wektorów przerwań jako wektor 43h. Podfunkcja powinna być wywołana zaraz po ustanowieniu graficznego trybu pracy.

**23h** - załadowanie znaków standardowych 8x8

AH 11h

AL 23h

BL liczba wierszy na ekranie zakodowana następująco:

0 - liczba wierszy podana w DL

1 - 14 wierszy

2 - 25 wierszy

3 - 43 wiersze

DL gdy BL=0 zawiera liczbę wierszy na ekranie

Adres zestawu znaków 8x8 jest zapisywany w tablicy wektorów przerwań jako wektor 43h. Podfunkcja powinna być wywołana zaraz po ustanowieniu graficznego trybu pracy.

**24h** - załadowanie znaków standardowych 8x16

AH 11h

AL 24h

BL liczba wierszy na ekranie zakodowana następująco:

0 - liczba wierszy podana w DL

1 - 14 wierszy

2 - 25 wierszy

3 - 43 wiersze

DL gdy BL=0 zawiera liczbę wierszy na ekranie

Adres zestawu znaków 8x16 jest zapisywany w tablicy wektorów przerwań jako wektor 43h. Podfunkcja powinna być wywołana zaraz po ustanowieniu graficznego trybu pracy.

**30h** - pobranie informacji o zestawach znaków

AH 11h

AL 30h

BH zbiór znaków

0 - zbiór znaków o adresie wskazywanym przez wektor przerwania 1Fh

1 - zbiór znaków o adresie wskazywanym przez wektor przerwania 43h

2 - zbiór znaków 8x14 z pamięci ROM sterownika

- 3– zbiór znaków 8x8 z pamięci ROM sterownika
- 4– zbiór znaków 8x8 o kodach powyżej 127 z pamięci ROM sterownika
- 5– dodatkowy zbiór znaków 8x14 z pamięci ROM sterownika
- 6– zbiór znaków 8x16 z pamięci ROM sterownika
- 7– dodatkowy zbiór znaków 8x16 z pamięci ROM sterownika

Wynik:

CX liczba bajtów opisujących pojedynczy znak

DL liczba wierszy możliwych do wyświetlenia na ekranie

ES:BP adres początku obszaru zawierającego definicje znaków.

### 1.6.5. Funkcje konfigurujące sterownik.

**INT 10 12h** – konfiguracja sterownika (AH = 12h) *Podfunkcje* (AL = ...):

**10h** – informacja o konfiguracji aktywnego sterownika.

**20h** – rozszerzenie operacji drukowania zawartości ekranu.

**30h** – ustawienie rozdzielczości pionowej tekstowych trybów pracy.

**31h** – powrót do standardowych kolorów.

**32h** – odłączenie sterownika.

**33h** – przełączenie do poziomów szarości.

**34h** – zezwolenie na emulację kursora.

**35h** – wybór aktywnego sterownika. **36h**

– wygaszanie ekranu.

**1.6.6. Funkcje uzupełniające. INT 10 13h** – wypisanie ciągu znaków (AH = 13h) *Podfunkcje* (AL = ...):

**00h** – wypisanie ciągu znaków bez przesuwania kursora

AH 13h

AL 00h

BH numer strony

BL atrybut wyświetlanych znaków

CX liczba znaków w wyświetlanym ciągu

DH numer wiersza

DL numer początkowej kolumny

ES:BP adres początku ciągu znaków

Podany ciąg musi zawierać tylko kody ASCII znaków. Użycie atrybutów jest niedopuszczalne.

**01h** – wypisanie ciągu znaków z przesunięciem kursora

AH 13h

AL 01h

BH numer strony

BL atrybut wyświetlanych znaków

CX liczba znaków w wyświetlanym ciągu

DH numer wiersza

DL numer początkowej kolumny

ES:BP adres początku ciągu znaków

Podany ciąg musi zawierać tylko kody ASCII znaków. Użycie atrybutów jest niedopuszczalne.

**02h** – wypisanie ciągu znaków z atrybutami bez przesunięcia kursora

AH 13h

AL 02h

BH numer strony

CX liczba par znak-atrybut w wyświetlanym ciągu

DH numer wiersza

DL numer początkowej kolumny

ES:BP adres początku ciągu par

**03h** – wypisanie ciągu znaków z atrybutami oraz przesunięcia kursora

AH 13h

AL 03h

BH numer strony

CX liczba par znak-atrybut w wyświetlanym ciągu

DH numer wiersza

DL numer początkowej kolumny

ES:BP adres początku ciągu par

**INT 10 1Ah** – pobranie informacji o sterowniku graficznym (AH = 1Ah)

***Podfunkcje (AL = ...):***

**00h** - pobranie informacji o rodzaju sterownika graficznego.

**01h** – wybranie rodzaju sterownika.

**INT 10 1Bh** – informacja o stanie i funkcjach aktywnego sterownika (AH = 1Bh)

**INT 10 1Ch** – zachowanie / odtworzenie stanu sterownika (AH = 1Ch)

***Podfunkcje (AL = ...):***

**01h** – zachowanie stanu sterownika



**02h** – odtworzenie stanu sterownika

**INT 10 4Fh** - Rozkaz ten ustawia tryb graficzny SVGA i zawiera wiele podfunkcji zapisywanych do AL., np.

AH 4Fh

AL. 02h (podfunkcja zmiany trybu graficznego)

BX tryb

100h 640x400      256 kolorów.

101h 640x480      256 kolorów.

102h 800x600      16 kolorów.

103h 800x600      256 kolorów.

104h 1024x768      16 kolorów.

105h 1024x768      256 kolorów.

Politechnika Rzeszowska  
Katedra Informatyki i Automatyki