

ARHITECTURA CALCULATOARELOR



Tematica disciplinei Arhitectura Calculatoarelor:

Tema 1. Introducere. Bazele fundamentale ale Arhitecturii Calculatoarelor;

Tema 2. Microprocesoare și Microcontrolere. Limbajul de programare Assembler;

Tema 3. Dispozitive pentru achiziția datelor;

Tema 4. Dispozitive pentru afișarea și imprimarea datelor;

Tema 5. Dispozitive pentru stocarea datelor.

Tema 4. Dispozitive pentru afișarea și imprimarea datelor:

- 1. Clasificarea dispozitivelor pentru afișarea și imprimarea datelor.**
- 2. Sistemul video.**
- 3. Videomonitoare.**
- 4. Interfața video.**
- 5. Videoterminale.**
- 6. Imprimanta.**
- 7. Plotter-ul.**

Dispozitive de ieșire, afișare și imprimarea datelor

Sistemul Video

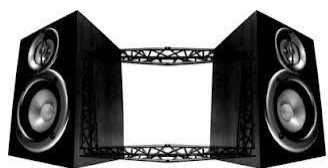
Imprimanta

Sistemul Audio

Plotterul

**Ecrane informaționale.
Proiector**

**Indicatoare:
Segmente, matrice**



Sistemul video al unui PC prezintă o integritate de dispozitive Hardware și produse Software destinate pentru: memorizarea, procesarea, organizarea schimbului de date și afișarea acestora în formă de imagini dinamice pe ecranul videomonitorului.

Componentele de bază ale Sistemului Video:

- **Dispozitive Hardware: Memoria Video (RAM Video), Interfața (Placa) Video și Videomonitorul.**
- **Produse Software: funcții BIOS și DOS, Drivere și OS.**
- **Regimul de funcționare: Text sau Grafic.**

Standarde pentru Sistemul video:

- Standarde digitale: MDA, CGA, EGA;
- Standarde analogice:

Standard	Semnificație	Rezoluție	Aspect
VGA	Video Graphics Array	640x480	4:3
SVGA	Super VGA	800x600	4:3
XGA	Extended Graphics Array	1024x768	4:3
SXGA	Super XGA	1280x1024	5:4
UXGA	Ultra XGA	1600x1200	4:3
HDTV	High Definition TV	1920x1080	16:9
QXGA	Quad XGA	2048x1536	4:3

Resursele rezervate de sistem pentru gestiunea Sistemului Video:

- RAM Video: B8000h-BFFFFh, A0000h-BFFFFh, E0000000h-FFFFFFFFh;
- Întreruperi: Hardware IRQ09h (INT 71h), BIOS INT 10h, DOS ...;
- Porturi I/O: 2C0h-2DFh, 3C0h-3CFh, 3C0h-2DFh, 3D0h-3DFh;
- Generatorul de simboluri – prezentarea grafică a caracterilor în raport cu codul ASCII;
- Porturi video generate: SVGA, HDMI, DVI;
- Producători: ASUS GeForce, NVIDIA, ...

Principalele caracteristici ale Sistemului Video:

- Amplasarea fizică a interfeței video: integrată pe placa de bază sau conectată pe Magistrala de Sistem;
- RAM Video: 32KB –2GB;
- Procesorul Video: Magistrala de Date – 8b-128b, Magistrala de Adrese – 32-64b, Numărul de nuclee – 64-128;
- Porturi standard generate: VGA, DVI, HDMI;
- Tehnologia CUDA, ...

GeForce



Fig. 8.1

NVIDIA

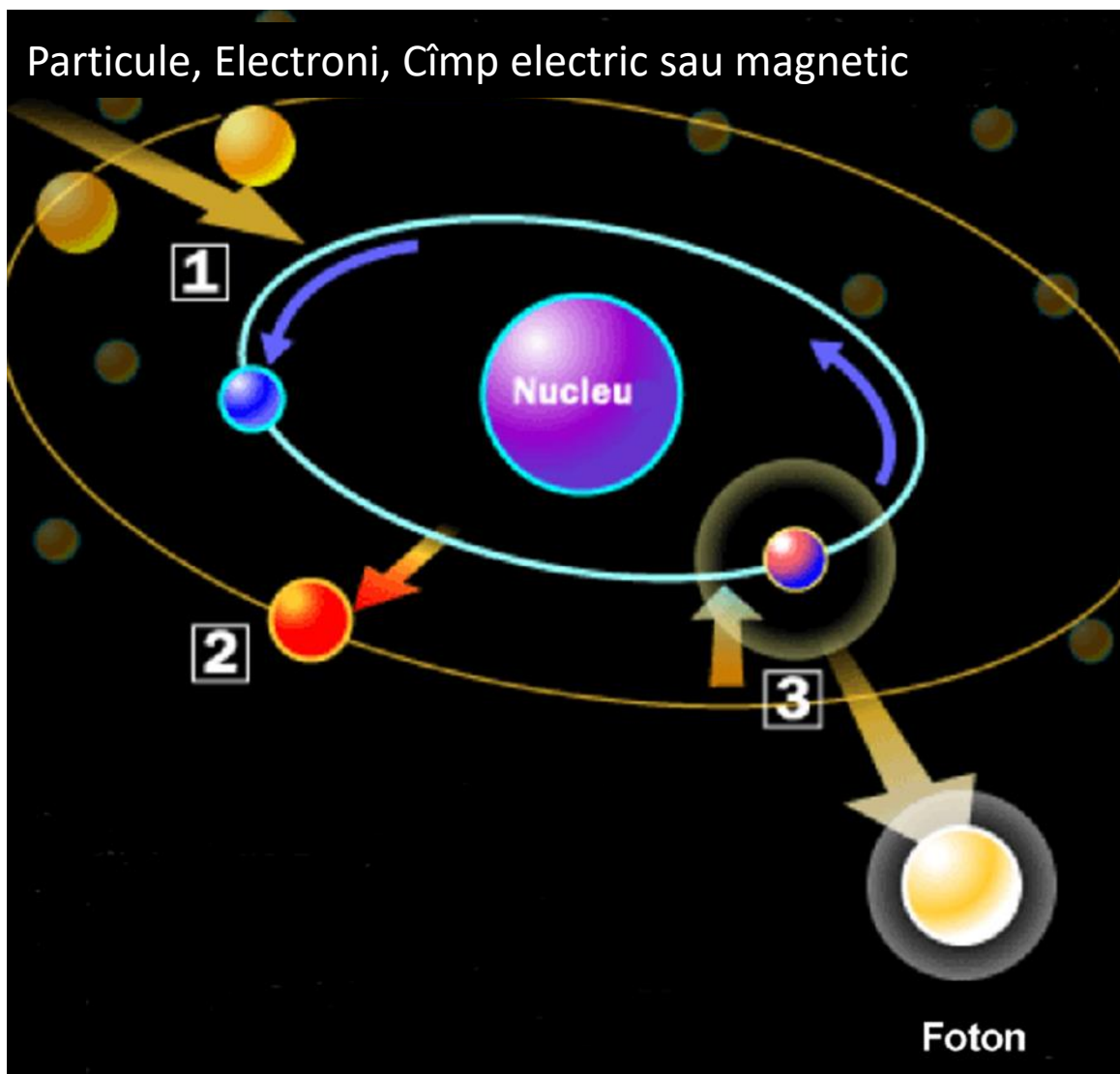


Fig. 8.2

Secvența de operații și interacțiunea dintre dispozitivele Sistemului Video pentru afișarea pe ecran a datelor.



Fig. 9.1



Tehnologia generării imaginilor pe ecranul videomonitorului.

Este prezentat un Atom format din Nucleu și Electroni amplasați pe orbitele sale.

Sub acțiunea factorilor externi are loc trecerea electronilor de pe un strat energetic mai inferior pe un strat energetic mai superior. Trecerea inversă duce la eliminarea unui Foton care și prezintă fluxul video.

Fig. 10.1

Videomonitorul este un dispozitiv destinat pentru afișarea grafică luminoasă a datelor și imaginilor. Principiul de funcționare se bazează pe recepția semnalelor de intensitate a culorii RGB, și de sincronizare VSYN și HSYN, și generarea în baza acestora a imaginii pe ecran (un flux de fotoni). Generarea imaginii are loc într-un mod secvențial unde în fiecare moment de timp este generată culoarea RGB a unui pixel coordonatele cărui sunt determinate de semnalele de sincronizare VSYN și HSYN.

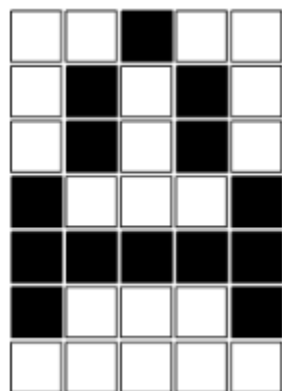


Fig. 11.1

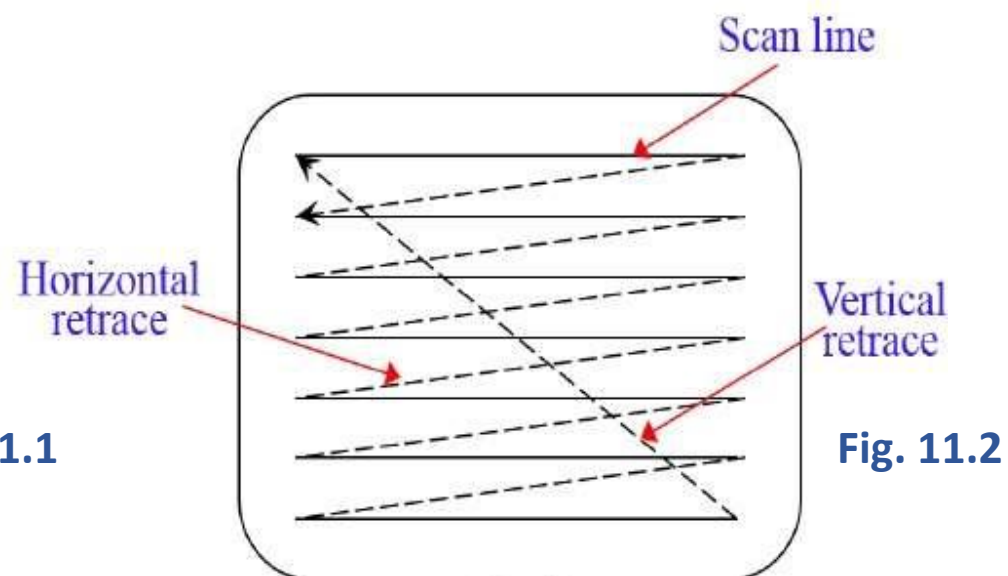
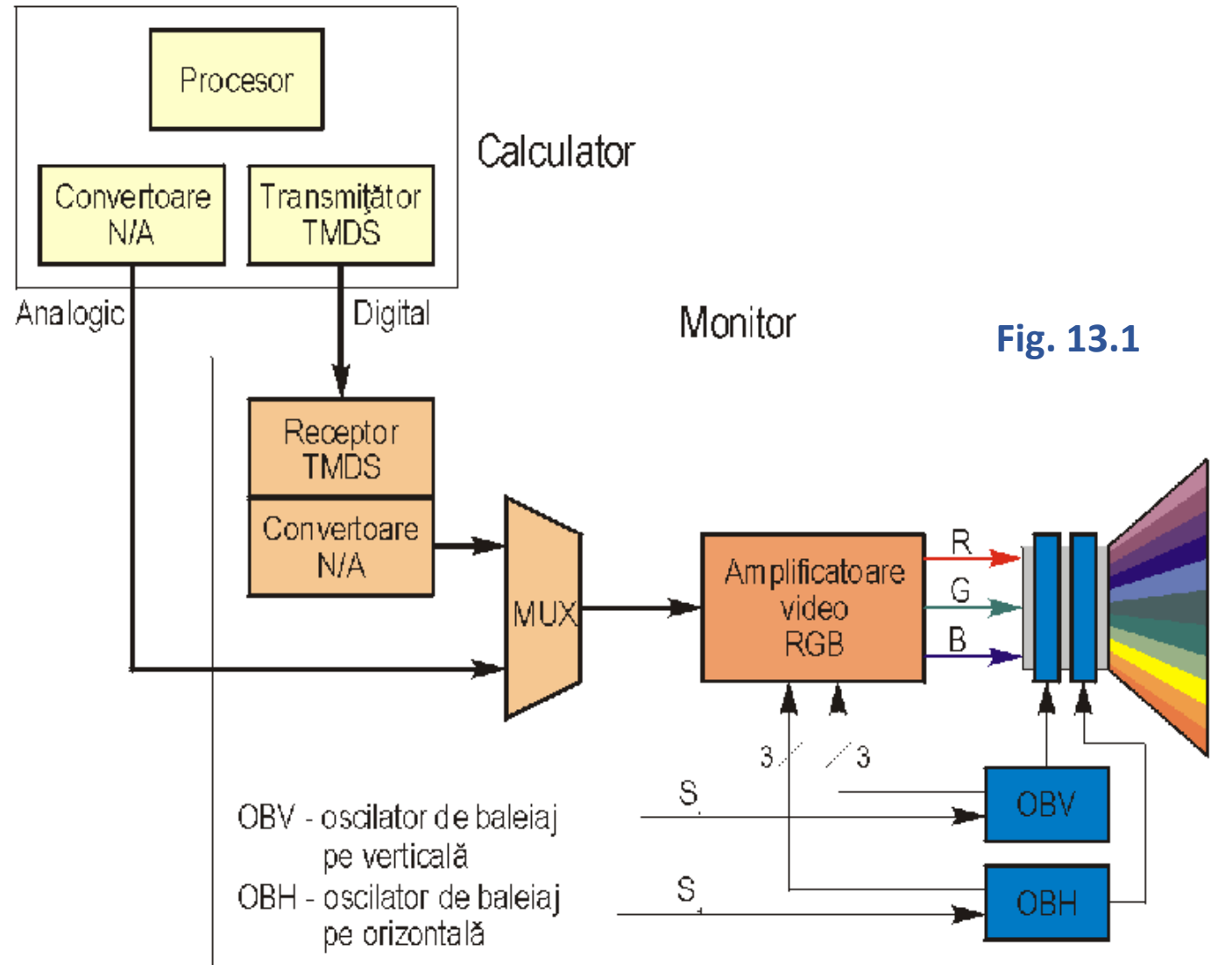


Fig. 11.2

Clasificarea și caracteristicile de bază ale Videomonitoroarelor:

- Standardul suportat: MDA, CGA, EGA, VGA, SVGA,;
- Tehnologia de generare a imaginii: CRT, LCD, TFT, Plasmă, LED, OLED;
- Interfața de comunicare cu PC: VGA, HDMI, DVI;
- Rezoluția: ...;
- Dimensiunea și raportul maximal al ecranului: ...;
- Culori generate: a/n, 16, 64, 256, 2^{16} , 2^{24} , 2^{32} ;
- Frecvența de regenerare a imaginii: 25Hz – 100Hz;
- Modul de control și setarea parametrilor funcționali;
- Producători.

Structura clasică a Videomonitoroarelor:



Modul de comunicare dintre PC și Videomonitor:

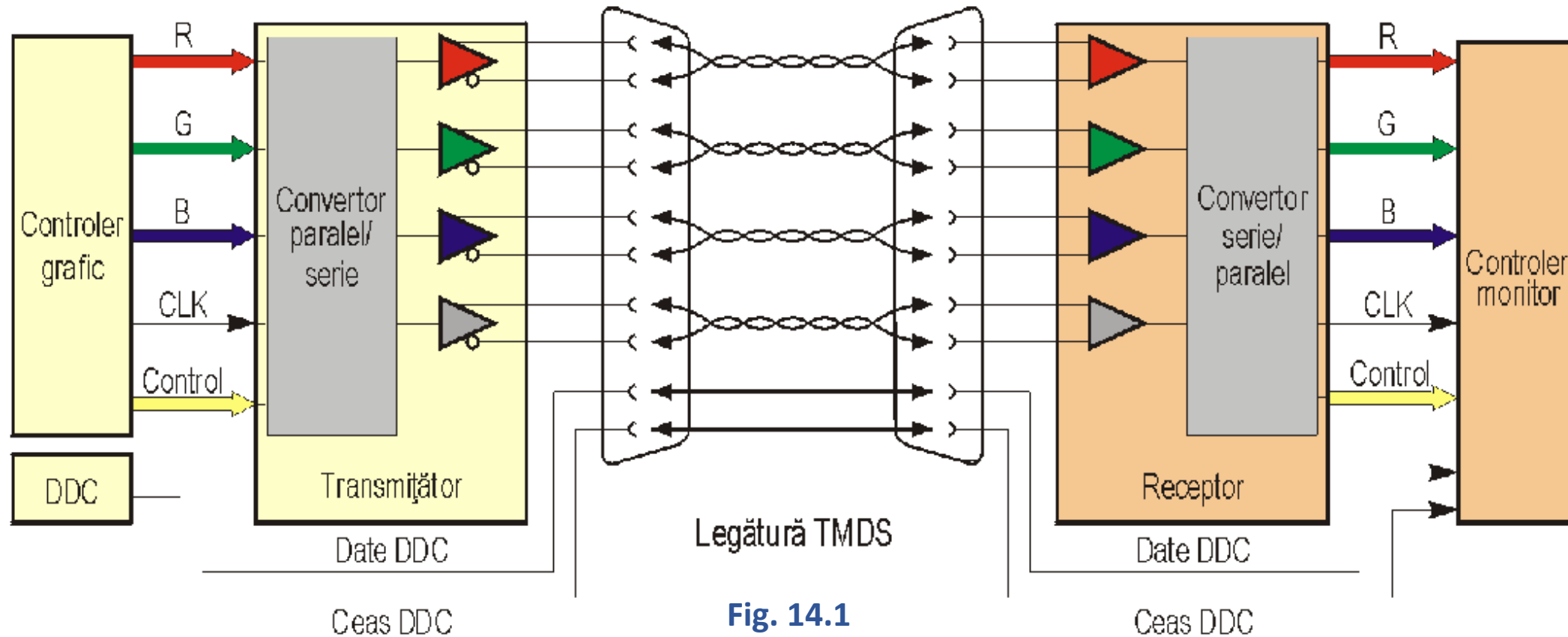


Fig. 14.1

Modelul de cegerare a imaginii pe ecran:

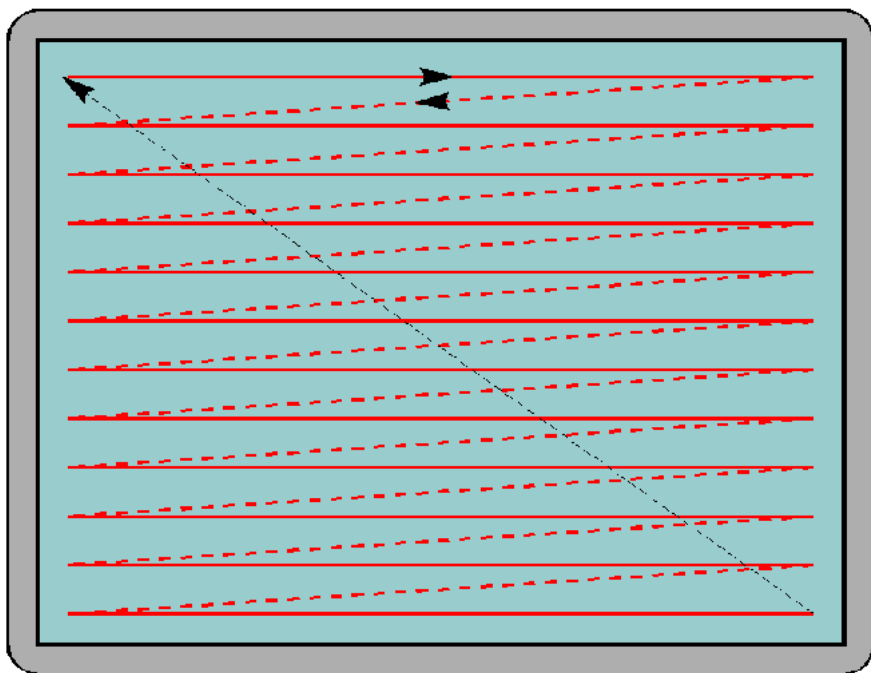


Fig. 15.1

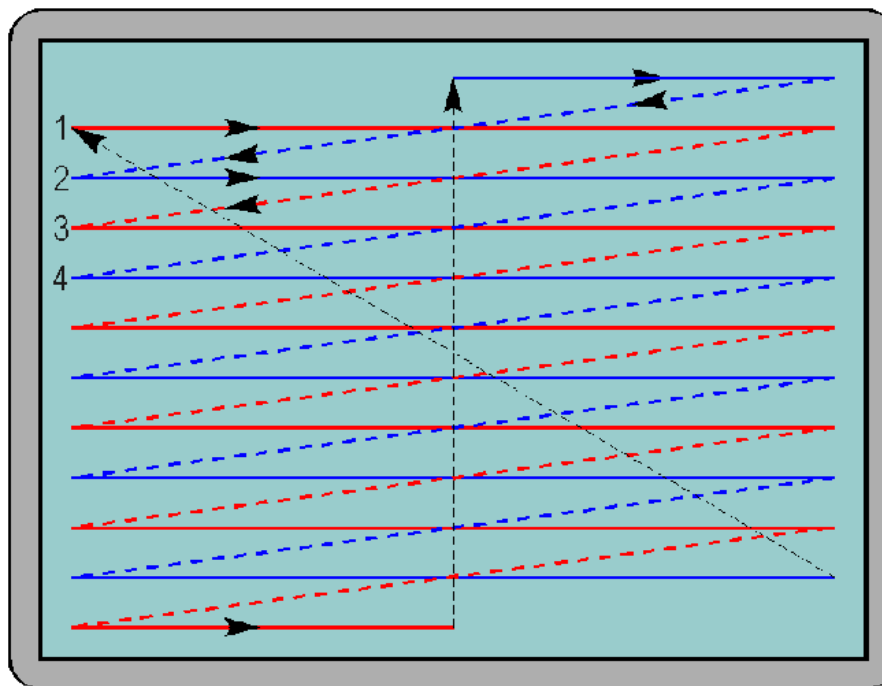


Fig. 15.2

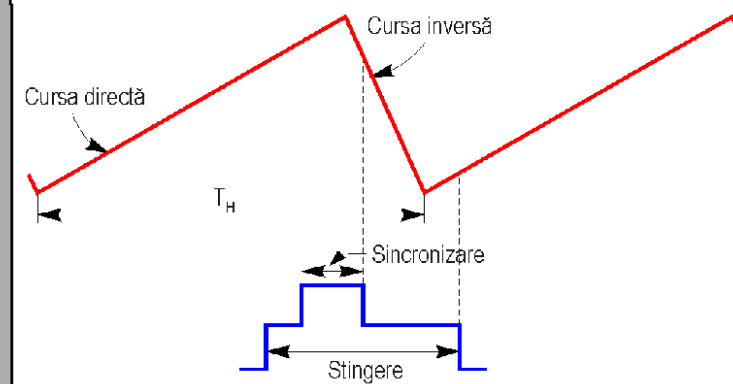


Fig. 13.3

Raportul dintre RAM Video și coordonatele caracterului pe ecran:

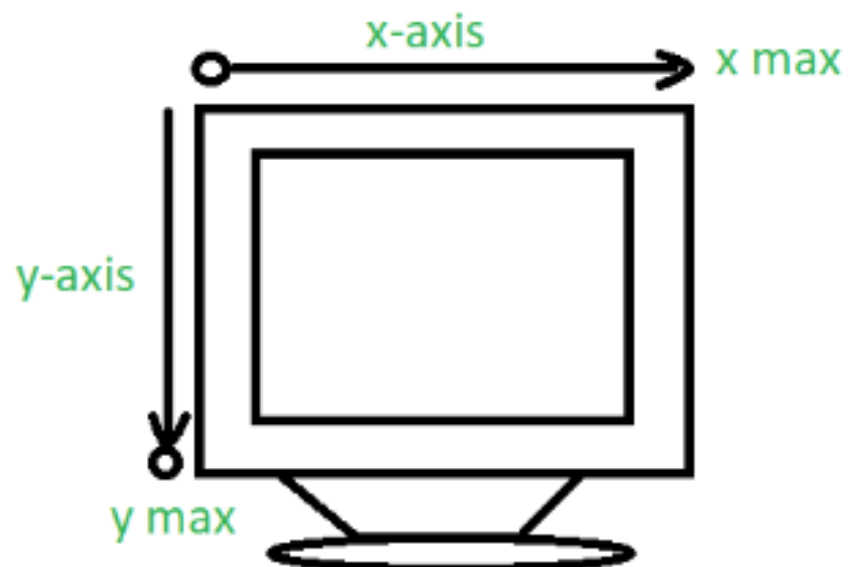


Fig. 16.1

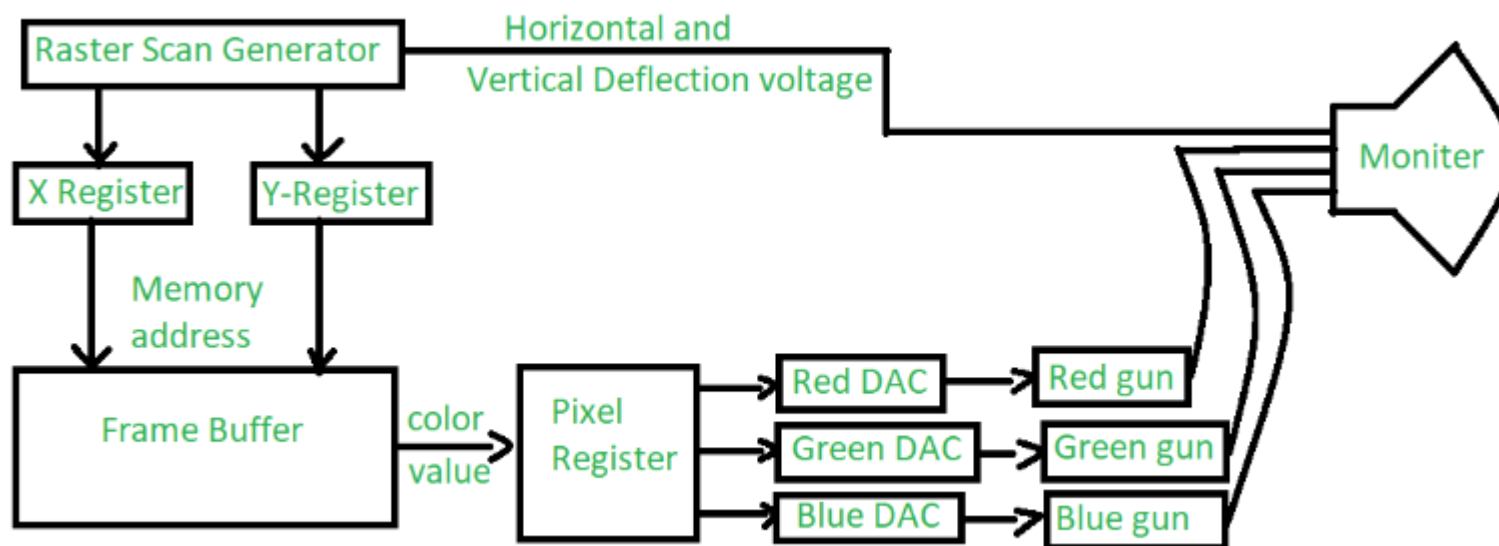


Fig. 16.2

Regin Text: B800:0000h + Y*160 + 2*X

Regim Grafic: A000:0000h + Y*320 + X

Videomonitoroarele CRT:

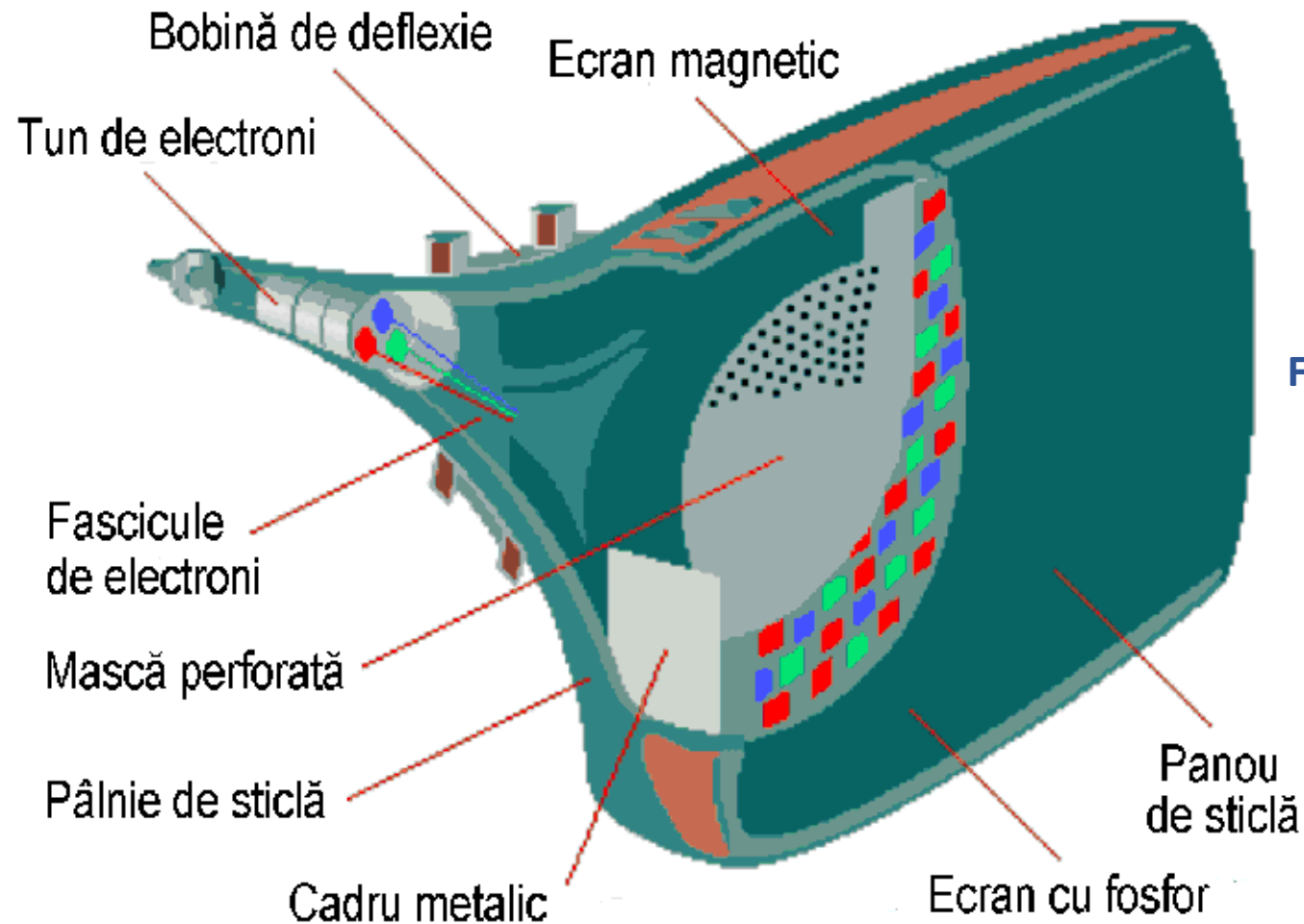


Fig. 17.1

Videomonitoroarele CRT. Generarea unui pixel. Culoarea pixel-ului:

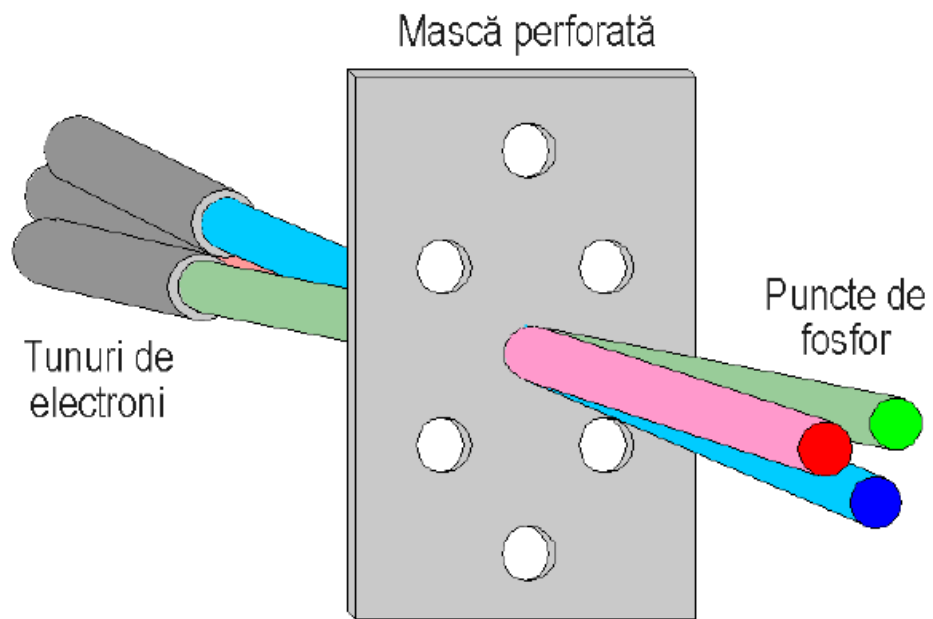


Fig. 18.1

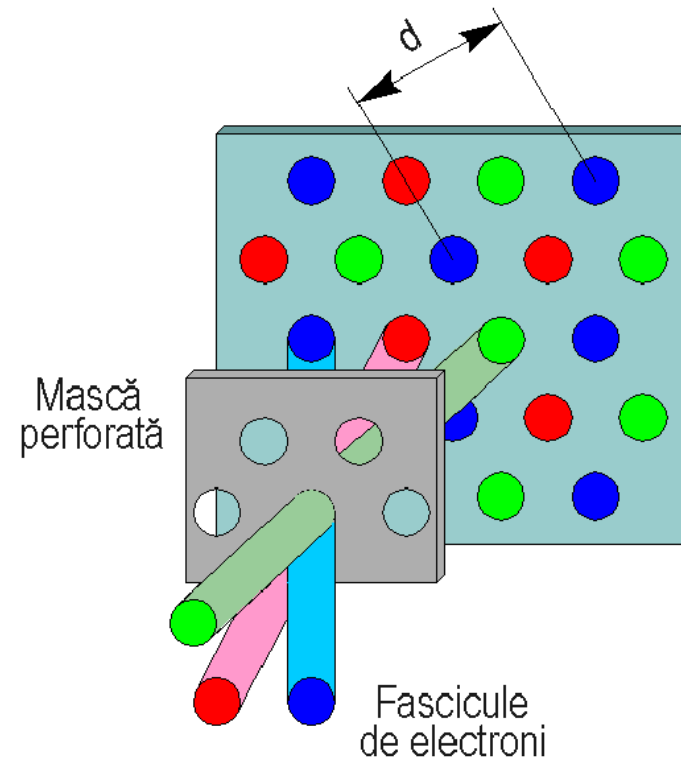


Fig. 18.2

Videomonitoroarele LCD (Liquid Crystal Display):

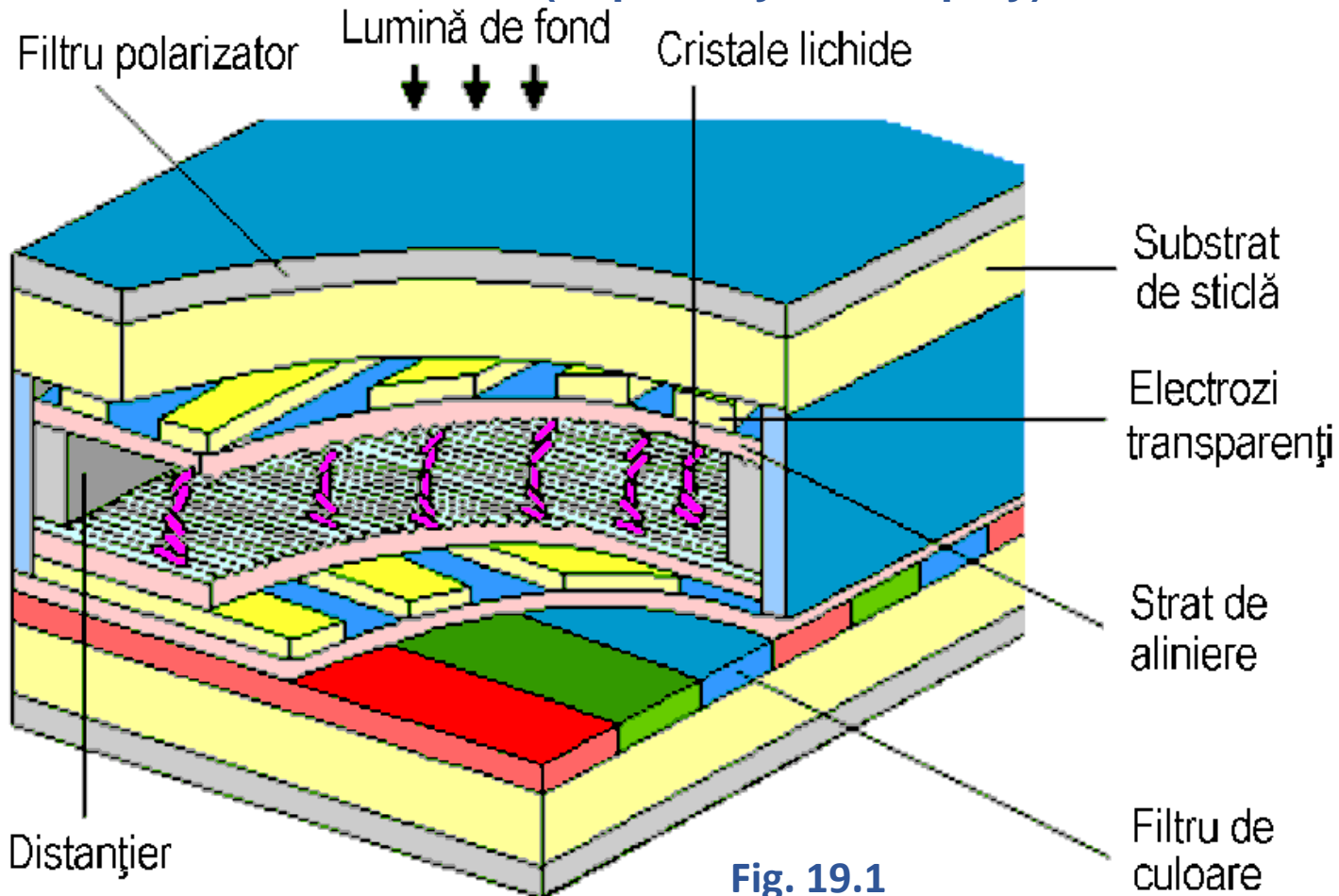


Fig. 19.1

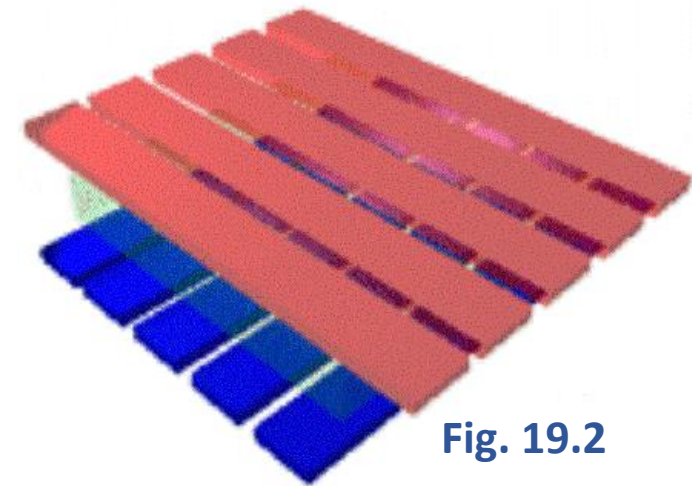


Fig. 19.2

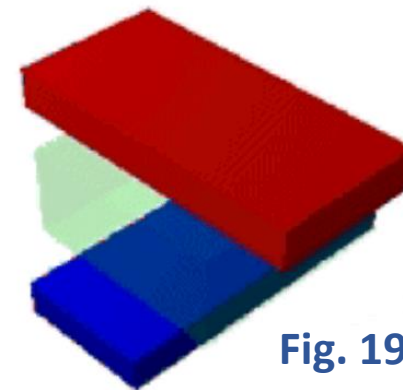


Fig. 19.3

Videomonitoroarele LCD-TFT (Thin-Film-Transistor):

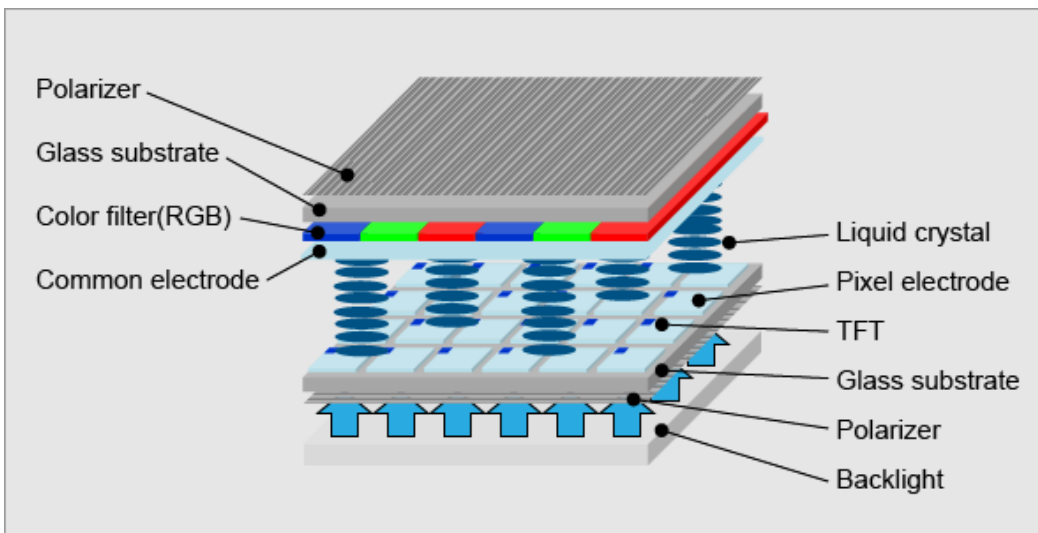


Fig. 20.1

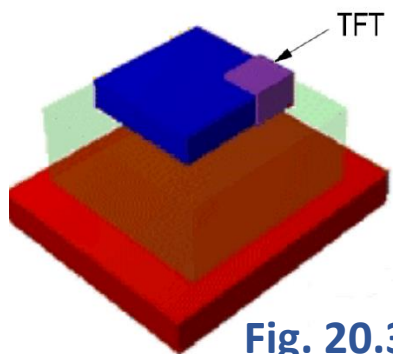


Fig. 20.3

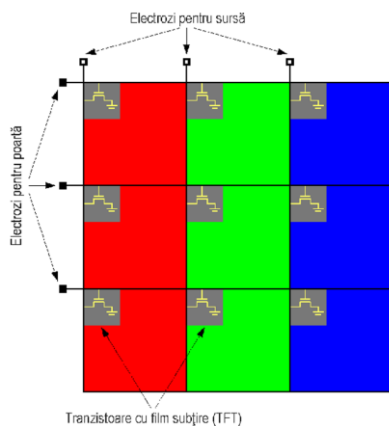


Fig. 20.4

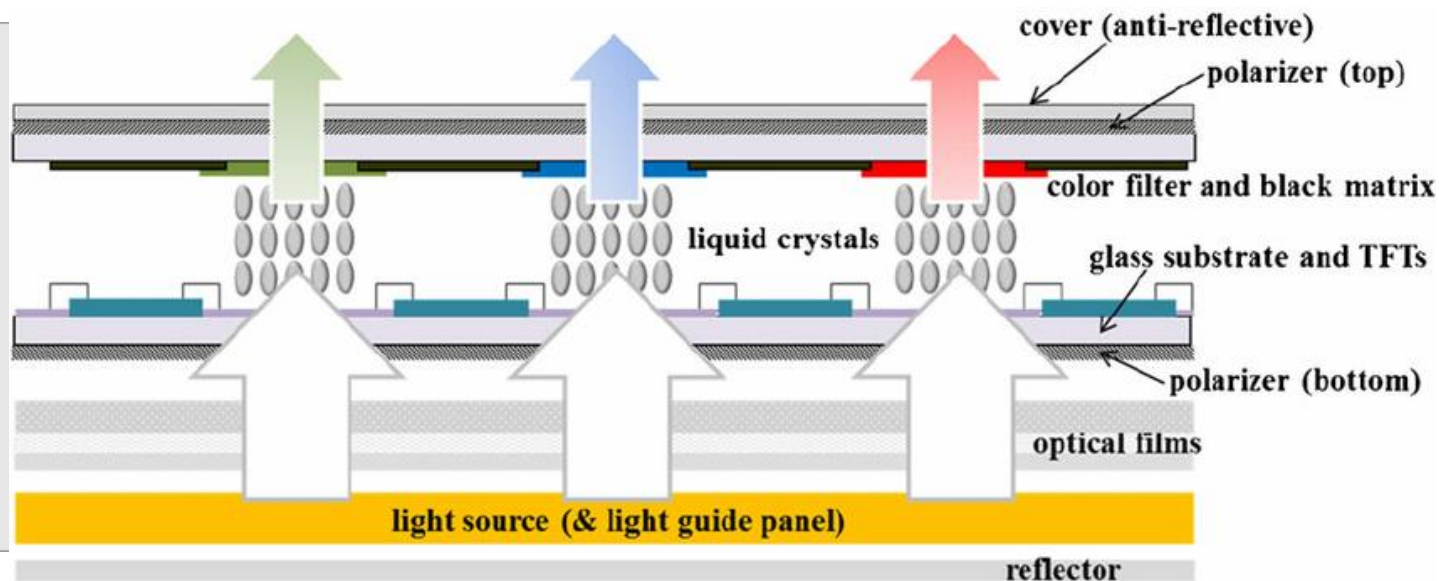


Fig. 20.2

Videomonitoroarele LCD-TFT. Modul de activare a unui pixel:

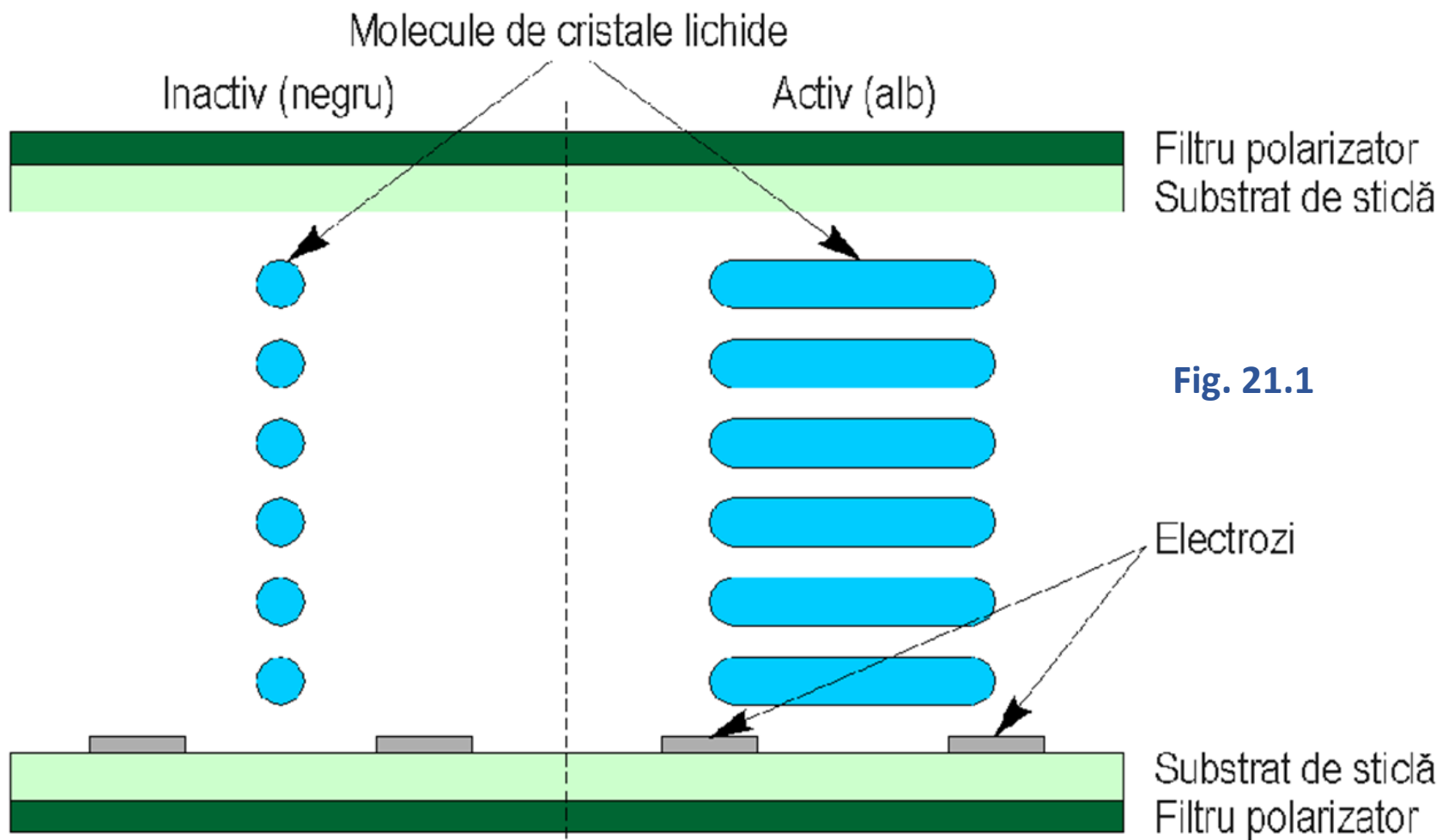
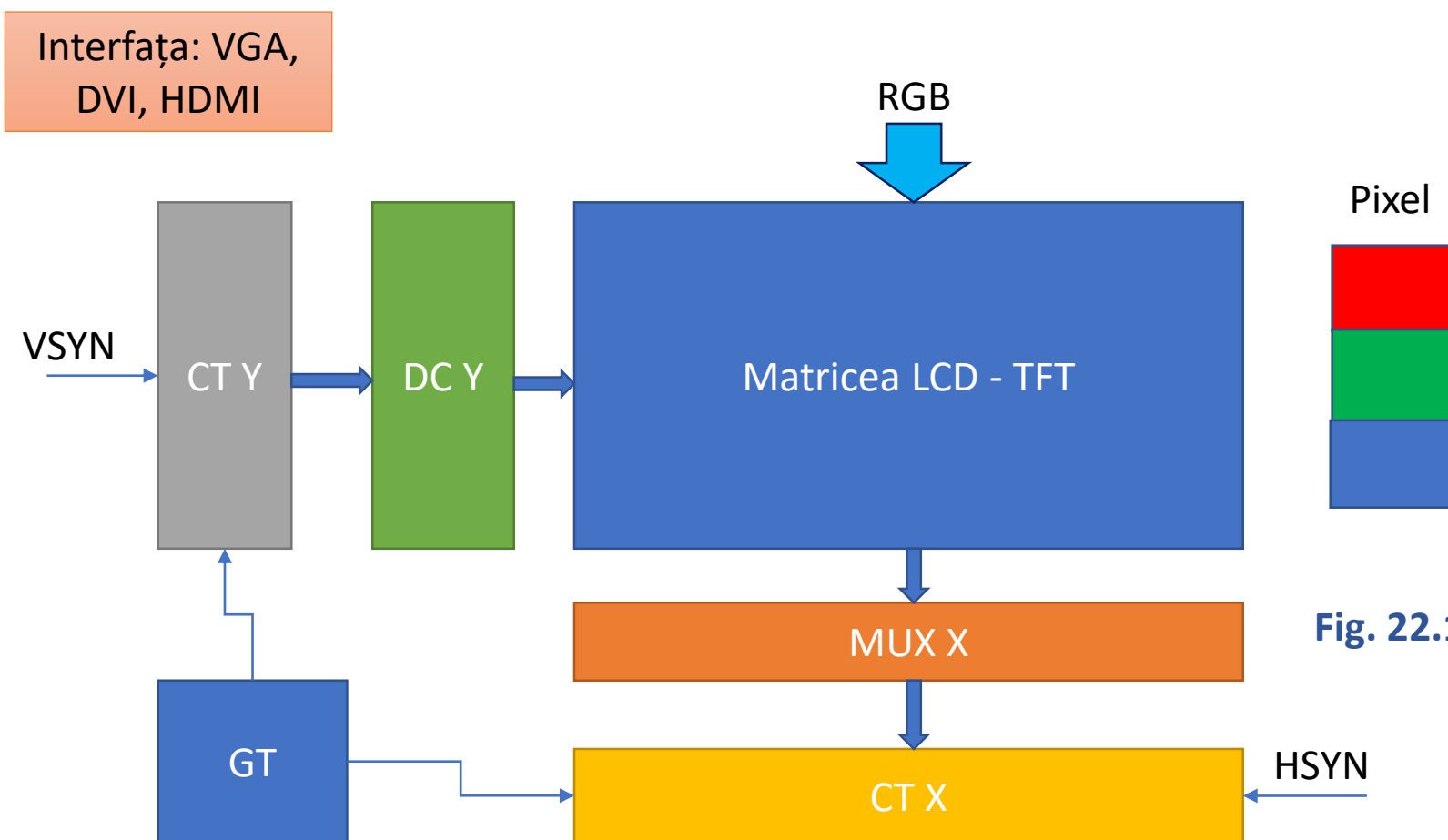


Fig. 21.1

Structura Videomonitoroarele LCD-TFT:



Videomonitoroarele cu Plasmă. Structura unui sub-pixel:

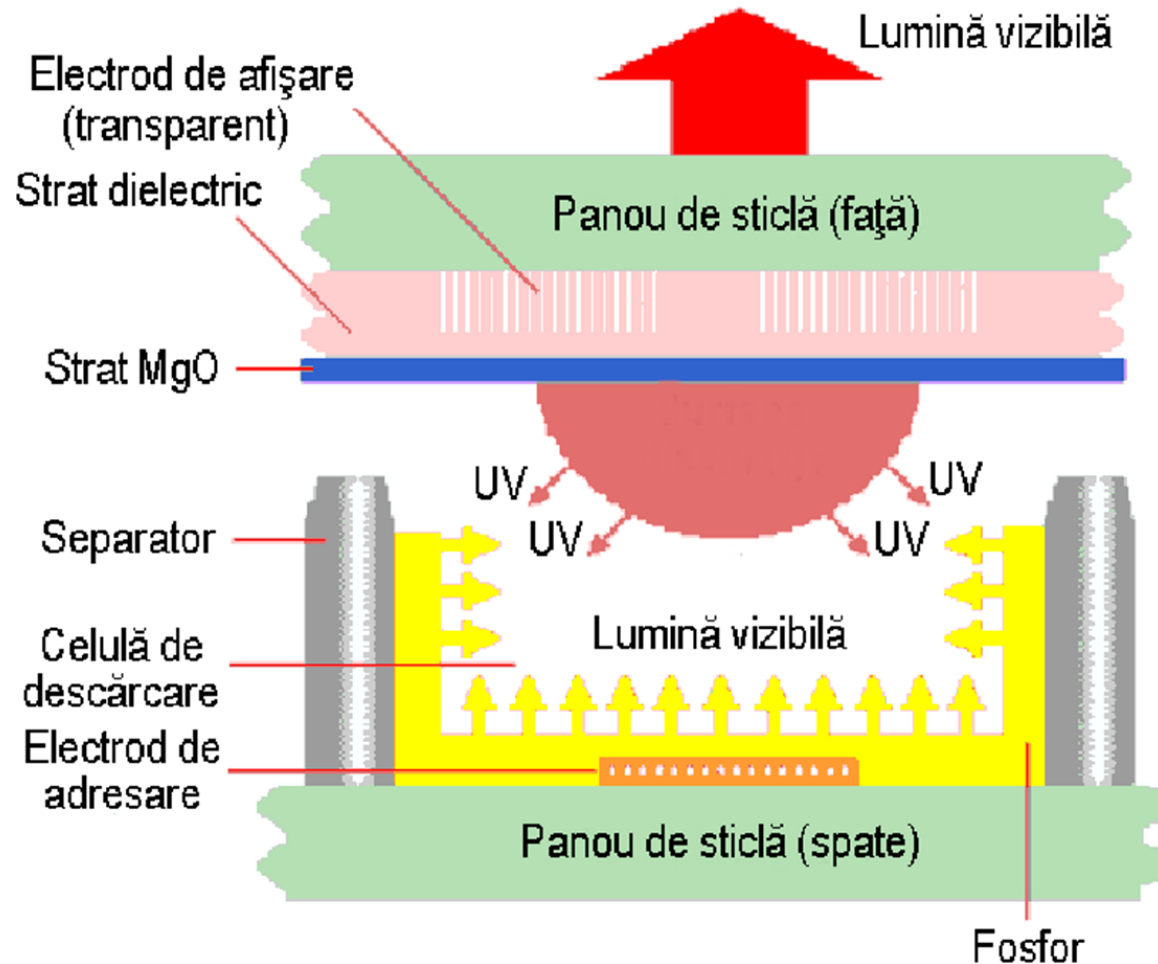


Fig. 23.1

Videomonitoroarele cu Plasmă. Structura matricei de pixeli:

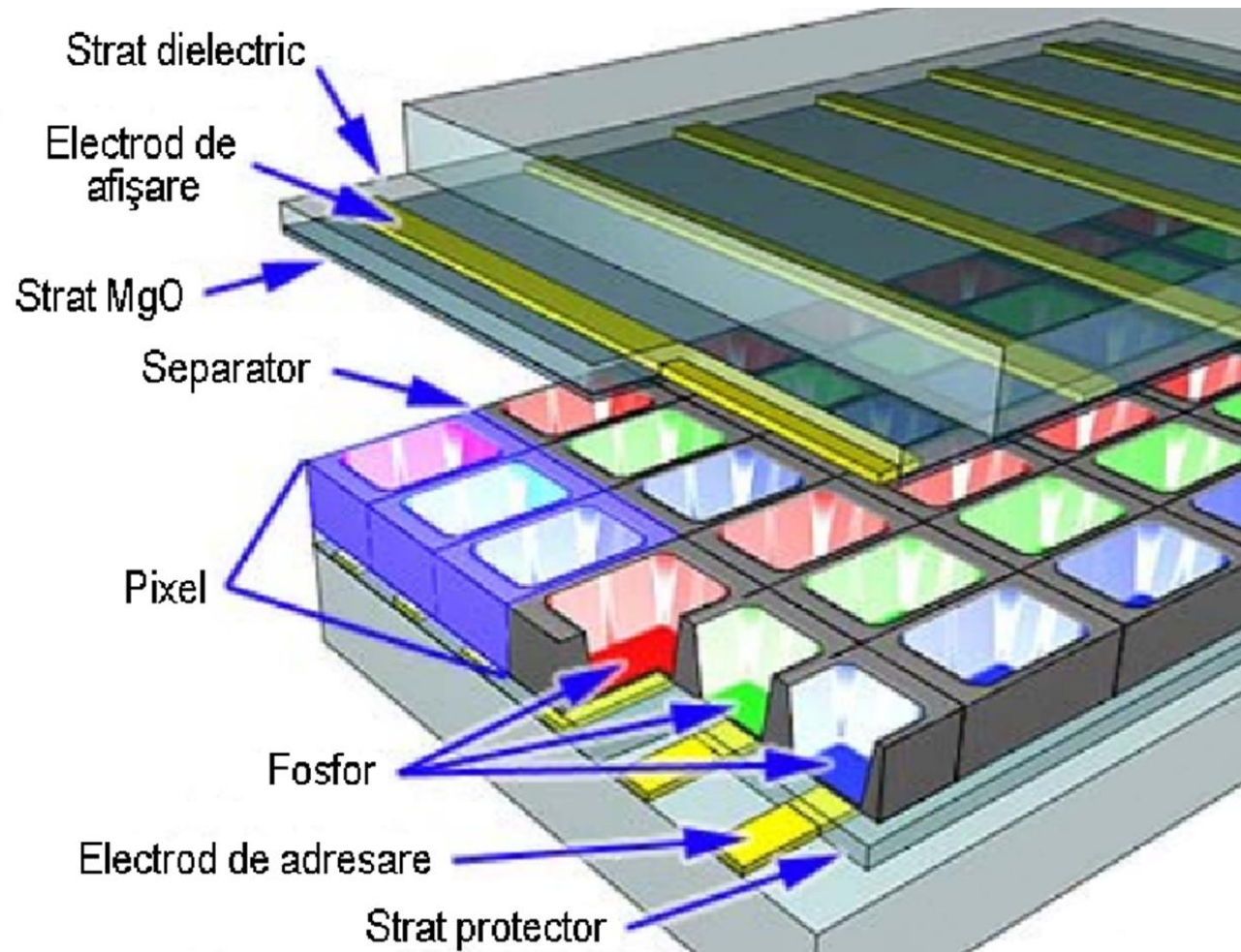


Fig. 24.1

Videomonitoroarele LED. Structura unui pixel:

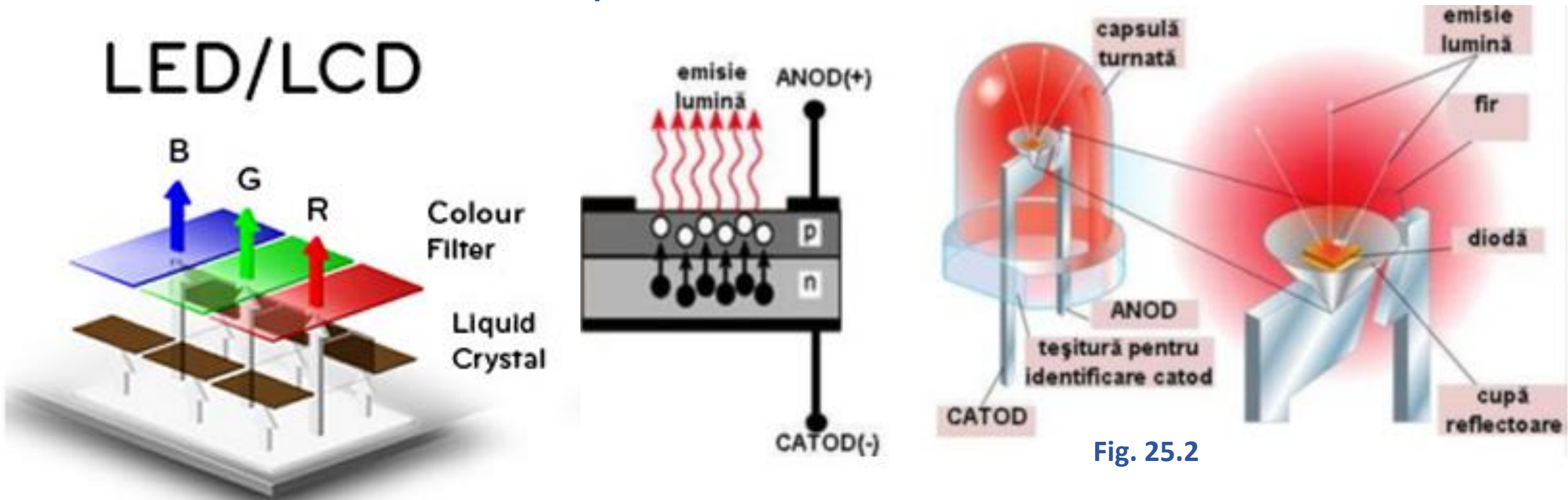


Fig. 25.1

Fig. 25.2

Videomonitoroarele OLED. Structura matricei de pixeli:

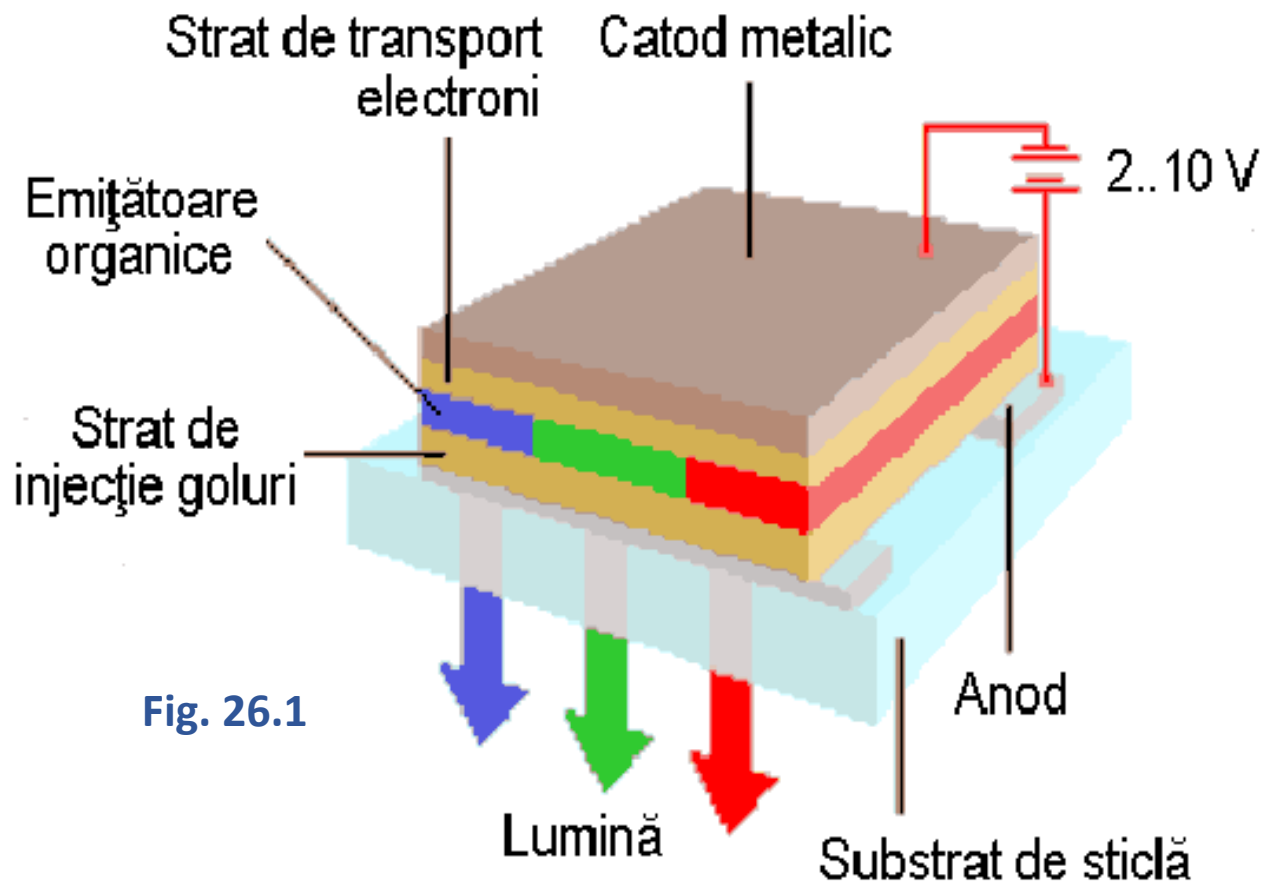


Fig. 26.1

Videomonitoroarele OLED. Structura matricei de pixeli:

LG OLED

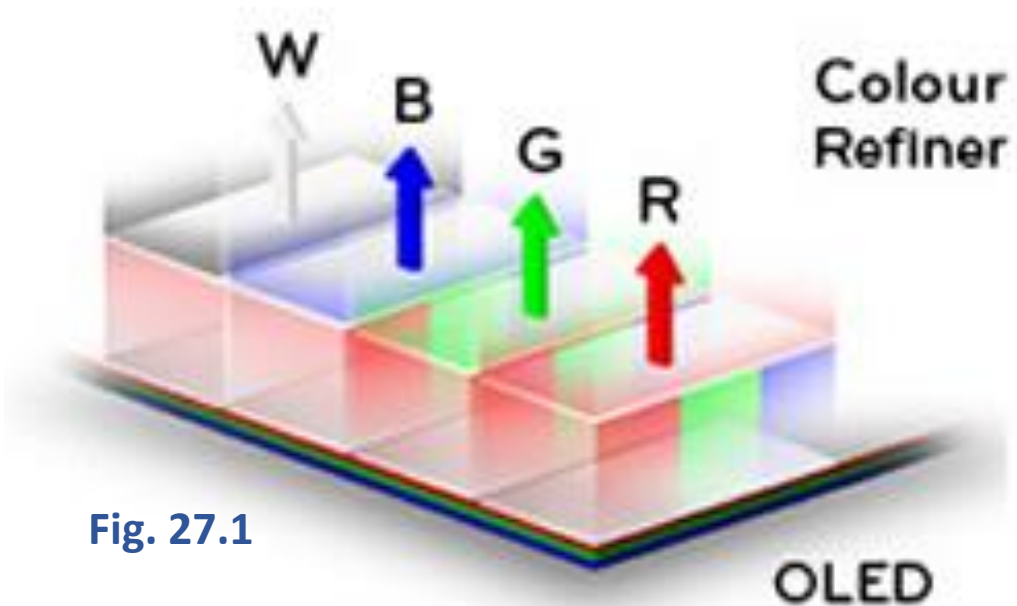


Fig. 27.1

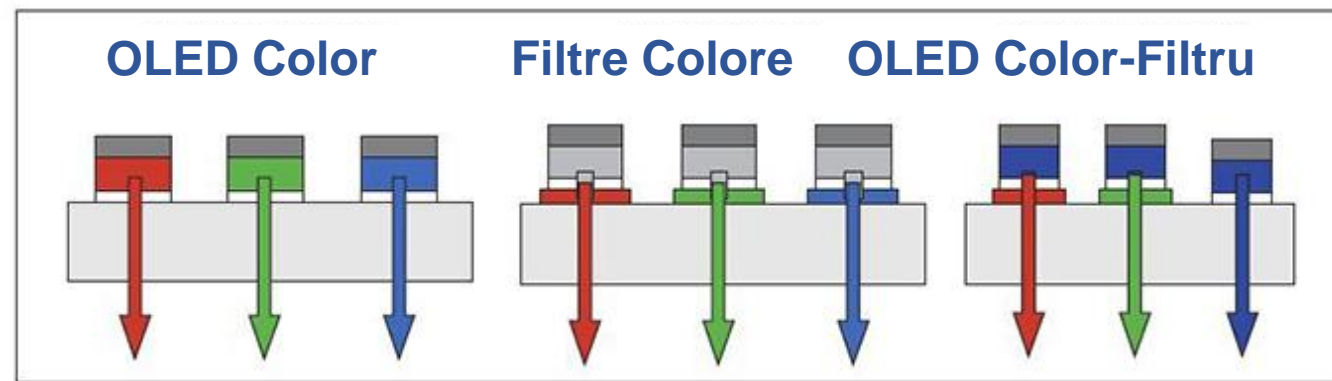


Fig. 27.2

Interfața video (Placa Video) este destinată pentru a transforma datele din memoria RAM Video în semnale electrice pentru a genera o imagine pe ecranul videomonitorului. Semnalele generate sunt: RGB care determină intensitatea culorii pixelului și semnalele de sincronizare a poziției pixelului pe ecranul videomonitorului VSYN și HSYN.

Interfața video poate fi integrată pe placa de baza a PC, sau este conectată pe Magistrala de Sistem (AGP).

Interfața video funcționează în două regimuri: Regim Text și Regim Grafic.

În Regim Text fiecare caracter ocupă în RAM Video 2 octeți: primul octet cu adresa impara determină codul ASCII al caracterului, al doilea octet, cu adresa para, prezintă Atributul, sau culoarea caracterului și culoarea fondalului. Biții D0-D3 indică culoarea caracterului și D4-D7 indică culoarea fondalului.

Forma de prezentare a caracterului este determinată de Generatorul de simboluri care face parte din structura Memoriei PC. Dimensiunea ecranului în regim standard: 80 caractere * 25 rânduri.

În Regim Text memoria RAM Video începe cu adresa: B800:0000h:

B800:0000h[ASCII1], B800:0001h[Atribut1];

B800:0002h[ASCII2], B800:0003h[Atribut2];...

Adresa caracterului cu coordonatele X, Y se calculează din formula:

$Ard_RAM = B800:0000h + 160 * Y + 2 * X.$

Interfața video (Placa Video). Regim Grafic:

Forma de prezentare a imaginii pe ecranul videomonitorului depinde de standardul de funcționare a Sistemului Video (MDA, CGA, EGA, VGA, SVGA, ...): MDA – 1 bit (2 culori), CGA – 4 biți (16 culori), EGA – 6 biți (64 culori), VGA – 8 biți (256 culori), SVGA – 16/24/32 biți.

Dimensiunea ecranului: EGA – 320*200, VGA – 640*480, ...

În Regim Grafic memoria RAM Video începe cu adresa: A000:0000h:

A000:0000h[pixel 0,0], A000:0001h[pixel 0,1]; A000:0002h[pixel 0,3]; ...

A000:0100h[pixel 0,256], A000:0101h[pixel 0,257], ...

Adresa pixelului cu coordonatele X, Y se calculează din formula:

$Ard_RAM = A000:0000h + 640 * Y + X.$

În Regim Grafic culoarea fiecărui pixel pe ecran este determinată de codul memorizat pe adresa respectivă a RAM Video.

Desenarea (afișarea) unui pixel pe ecran constă în înscrierea codului culorii acestuia pe adresa respectivă a RAM Video.

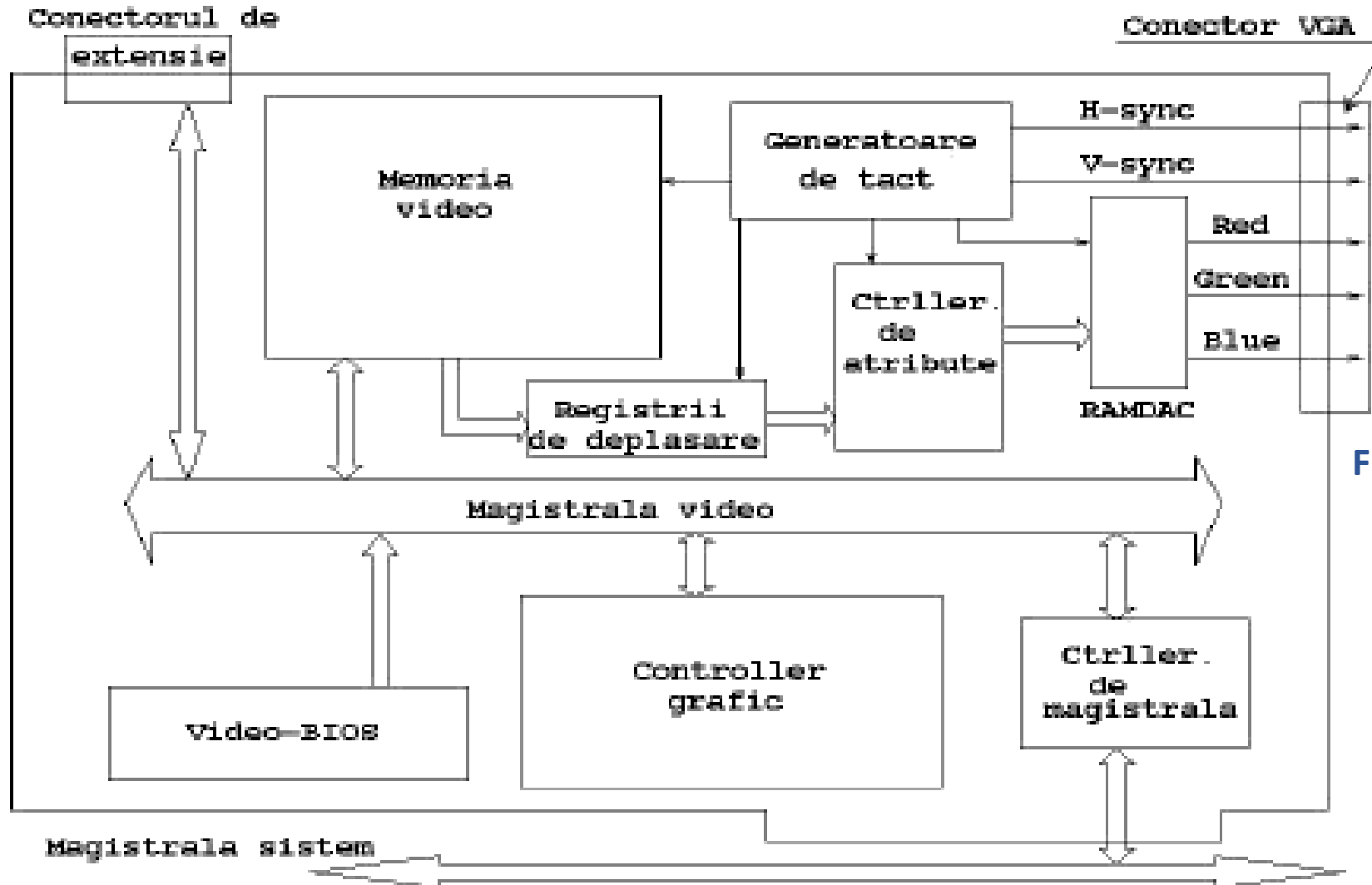


Fig. 30.1

Structura Interfaței video
(Placă Video):

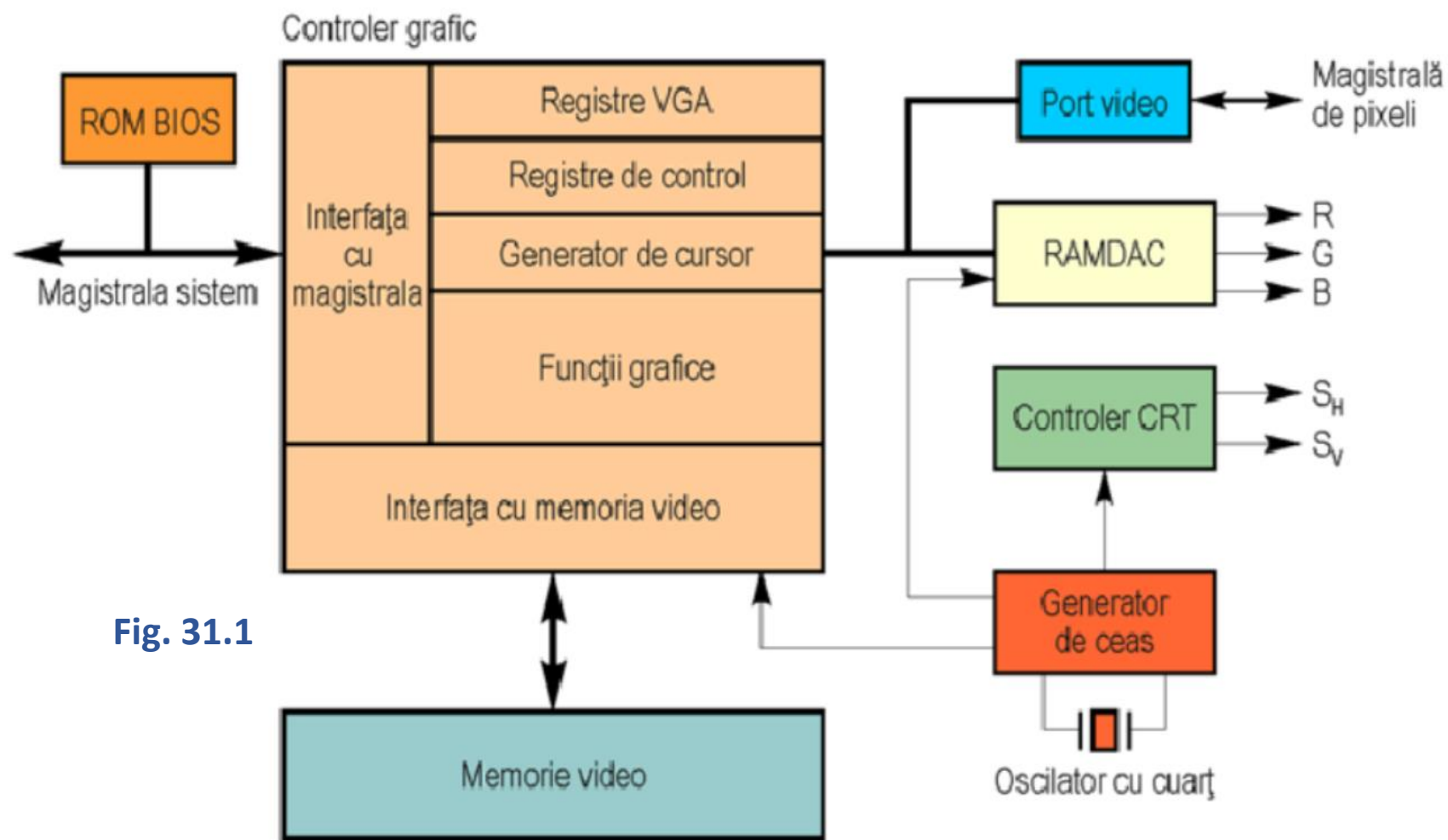


Fig. 31.1

**Structura Interfaței video.
Interacțiunea
componentelor:**

**Structura Interfaței video.
Funcționalitatea RAMDAC:**

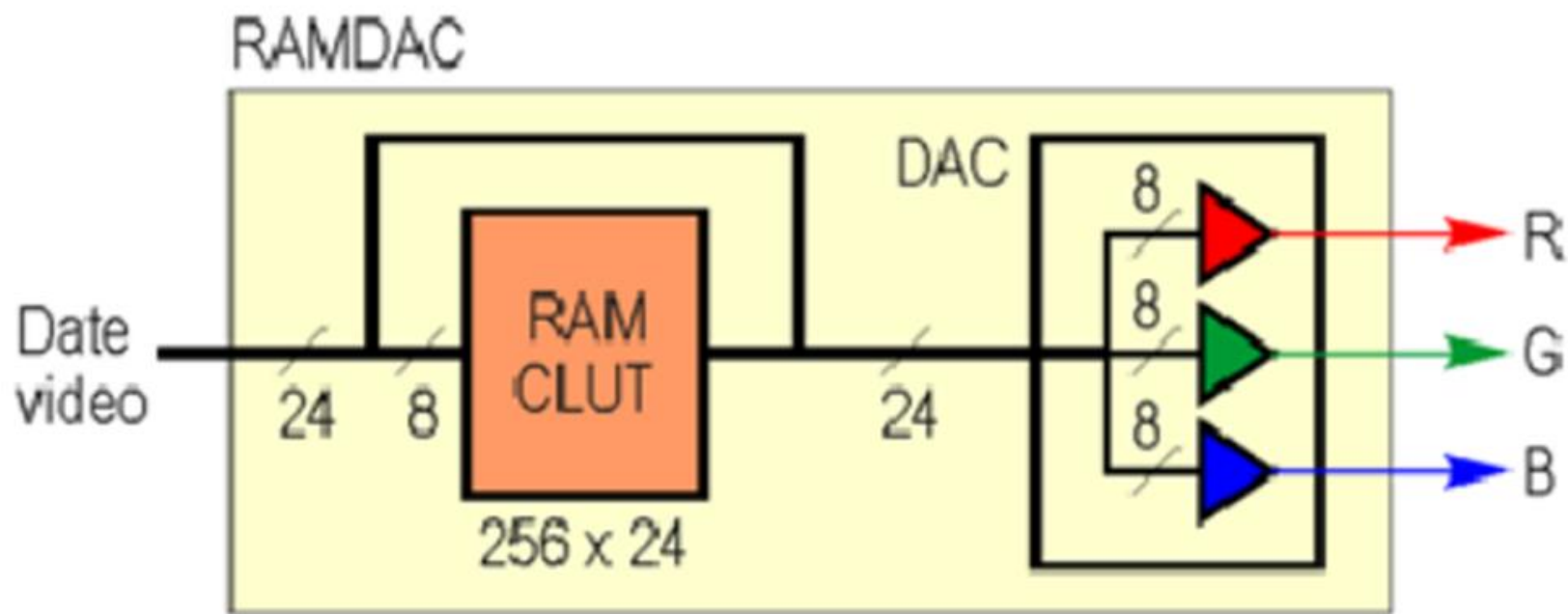


Fig. 32.1

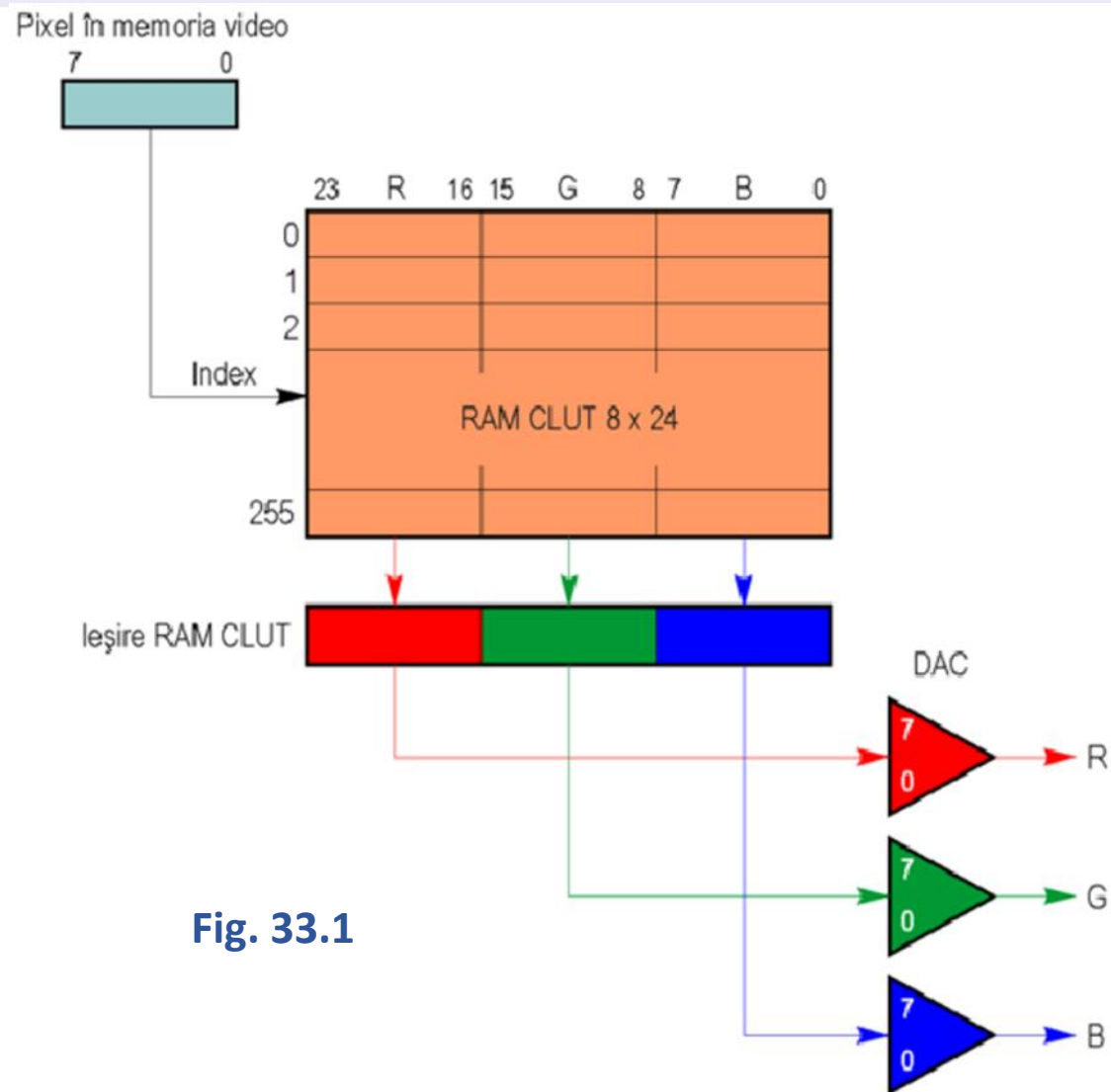


Fig. 33.1

**Structura Interfaței video.
Selectarea culorii în RAM CLUT:**

Tab. 34.1

Frecvența conversiei DAC (MHz)
în raport cu Standardul:

Rezoluție	60 Hz	75 Hz	85 Hz	100 Hz
800 x 600	38,0	47,5	53,9	63,4
1024 x 768	62,3	77,8	88,2	103,8
1280 x 1024	103,8	129,7	147,1	173,0
1600 x 1200	152,1	190,1	215,4	253,4
1920 x 1440	219,0	273,7	310,2	364,9
2048 x 1536	249,1	311,4	352,9	415,2

Standarde de prezentare a
culorii pixelelor în RAM Video:

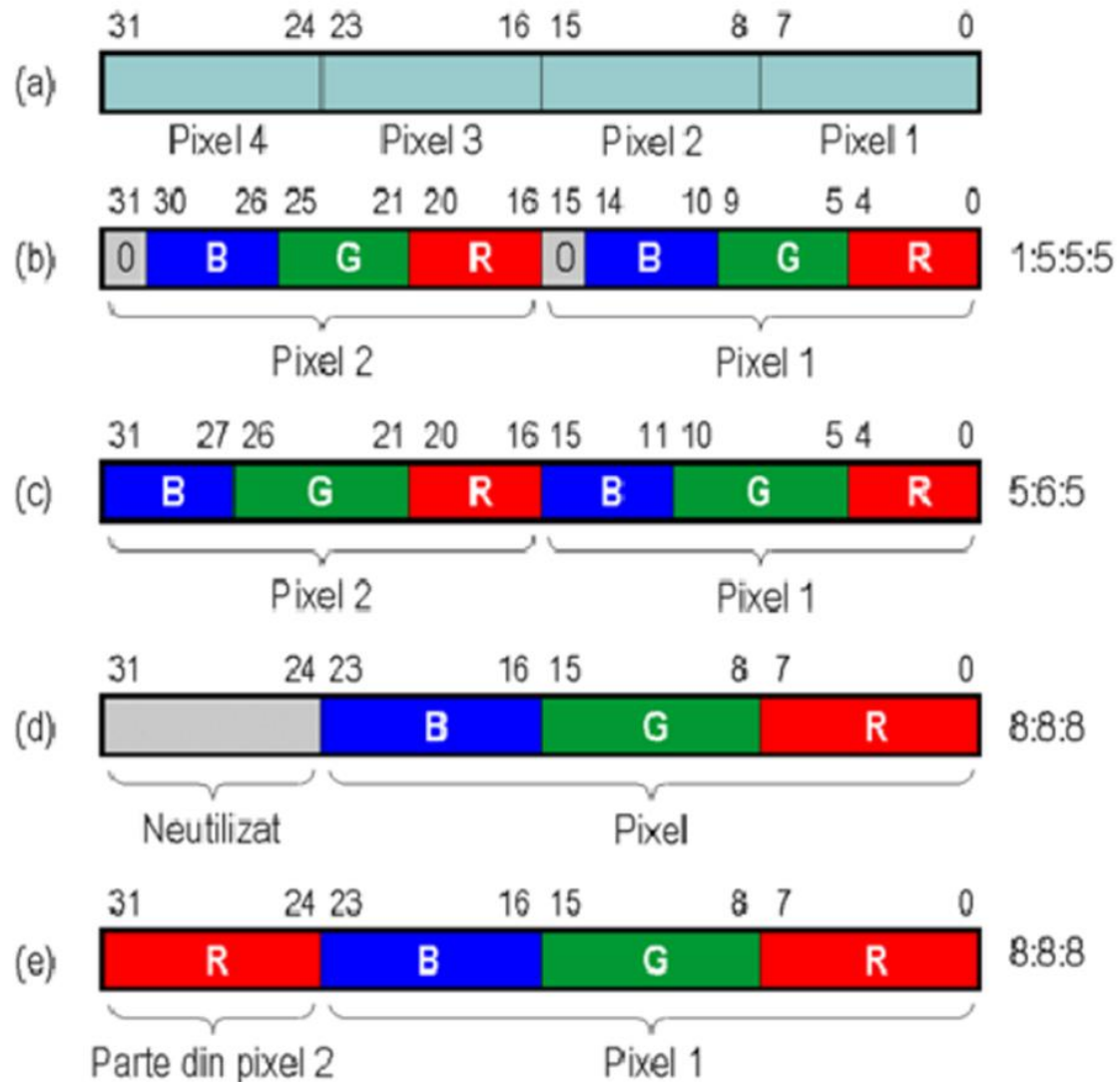


Fig. 35.1

Structura și accesarea Memoriei Video (RAM Video): Port unic, Port dual.

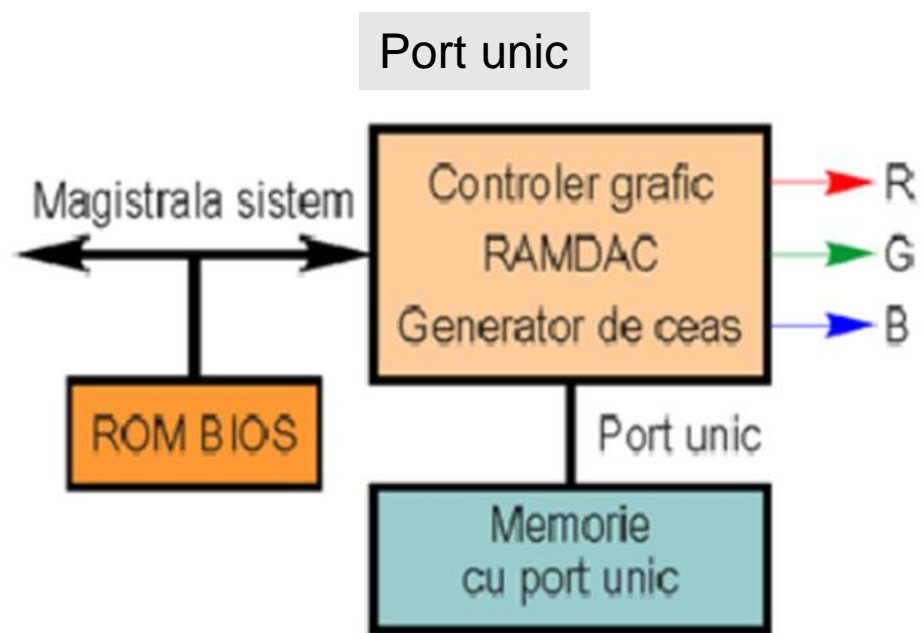


Fig. 36.1

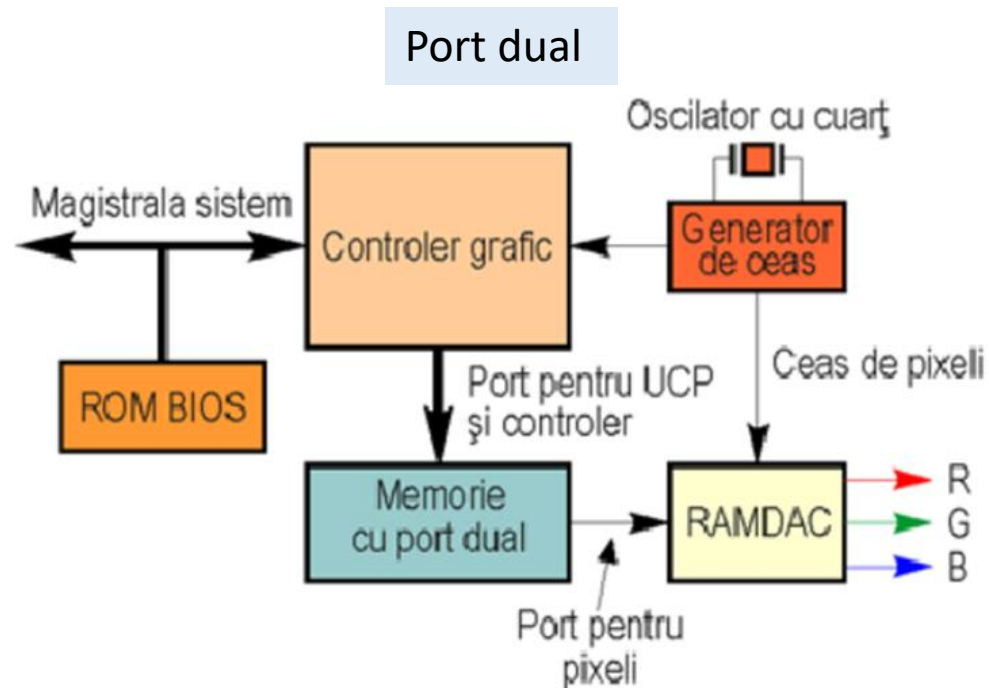


Fig. 36.2

Parametrii RAM Video în raport cu Standardul utilizat (MB):

Tab. 37.1

Rezoluție	8 biți	16 biți	24 biți	32 biți
800 x 600	0,46 (1 MB)	0,92 (1 MB)	1,37 (2 MB)	1,83 (2 MB)
1024 x 768	0,75 (1 MB)	1,50 (2 MB)	2,25 (4 MB)	3,00 (4 MB)
1280 x 1024	1,25 (2 MB)	2,50 (4 MB)	3,75 (4 MB)	5,00 (8 MB)
1600 x 1200	1,83 (2 MB)	3,66 (4 MB)	5,49 (8 MB)	7,32 (8 MB)
1920 x 1440	2,63 (4 MB)	5,27 (8 MB)	7,91 (8 MB)	10,54 (16 MB)
2048 x 1536	3,00 (4 MB)	6,00 (8 MB)	9,00 (16 MB)	12,00 (16 MB)

Rata de transfer MB/s în raport cu Standardul utilizat:

Tab. 38.1

Rezoluție	60 Hz	75 Hz	85 Hz	100 Hz
800 x 600	86,4	108,0	122,4	144,0
1024 x 768	141,6	176,9	200,5	235,9
1280 x 1024	235,9	294,9	334,2	393,2
1600 x 1200	345,6	432,0	489,6	576,0
1920 x 1440	497,6	622,1	705,0	829,4
2048 x 1536	566,2	707,8	802,1	943,7

Exemple de Interfețe video:



Fig. 39.1



Standarde de comunicare Intrefața Video - Videomonitor:



VGA



DVI



Mini DVI

Fig. 40.1



HDMI

Videoterminalul este o colecție de dispozitive Hardware și produse Software destinate pentru achiziția, memorizarea, afișarea și comunicarea în rețelele de calculatoare.

Structural un Videoterminal este format din: Tastatură, Videomonitor, Procesor, RAM, ROM BIOS și alte periferice la necesitate.

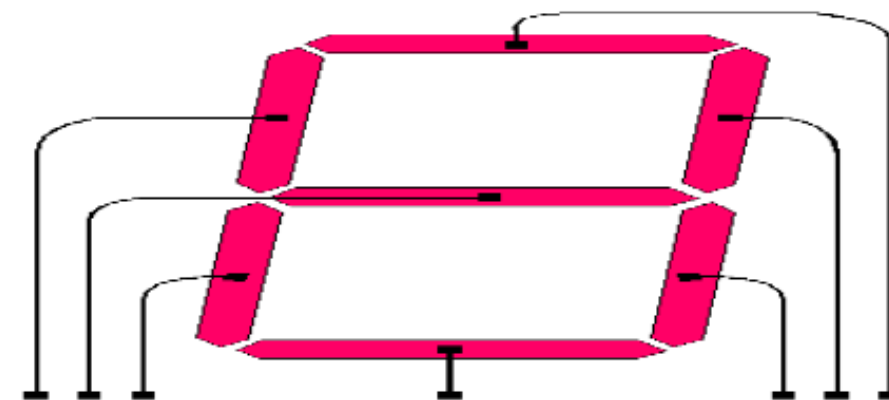
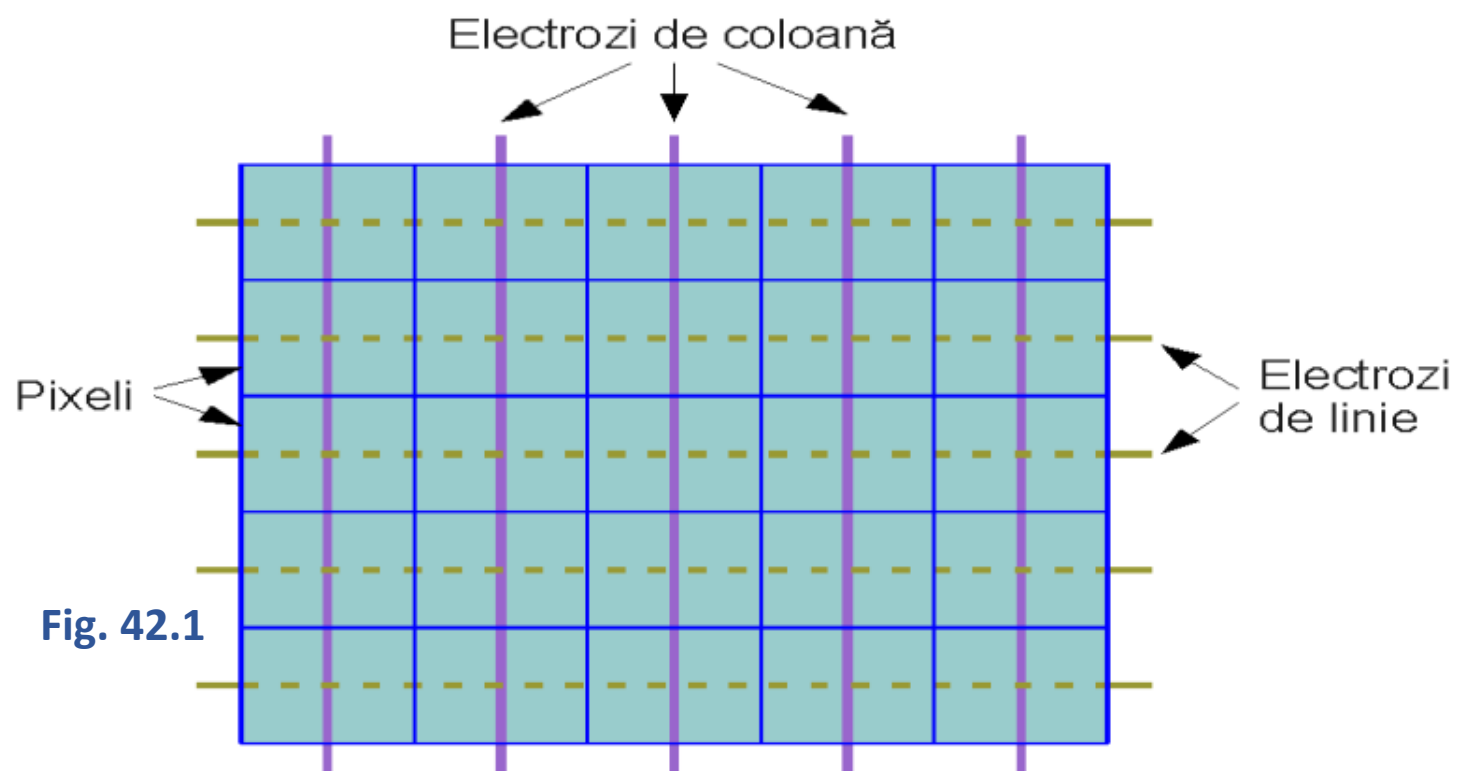
Videoterminalul este utilizat în arhitecturile de calcul Client-Server în calitate de stație Client.

Orice PC poate funcționa în regim de terminal la utilizare produselor program respective.

Avantajele utilizării de videoterminal:

- Preț redus pentru Hard și Soft,
- Procesarea distribuită a datelor,
- Utilizarea tehnologiilor Client-Server,
- Accesare servicii și resurse Cloud, etc.

Indicatoare informaționale matriciale și pe segmente:



Indicatoare informaționale matriciale și pe segmente. Exemple:



Fig. 43.1

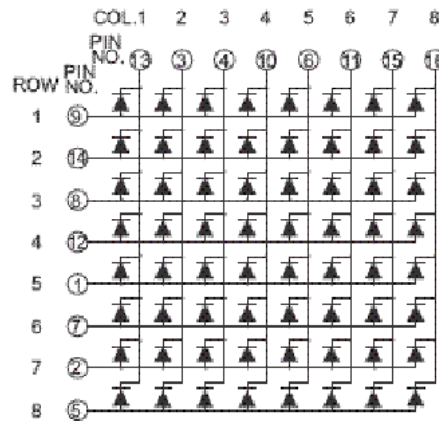


Fig. 43.2

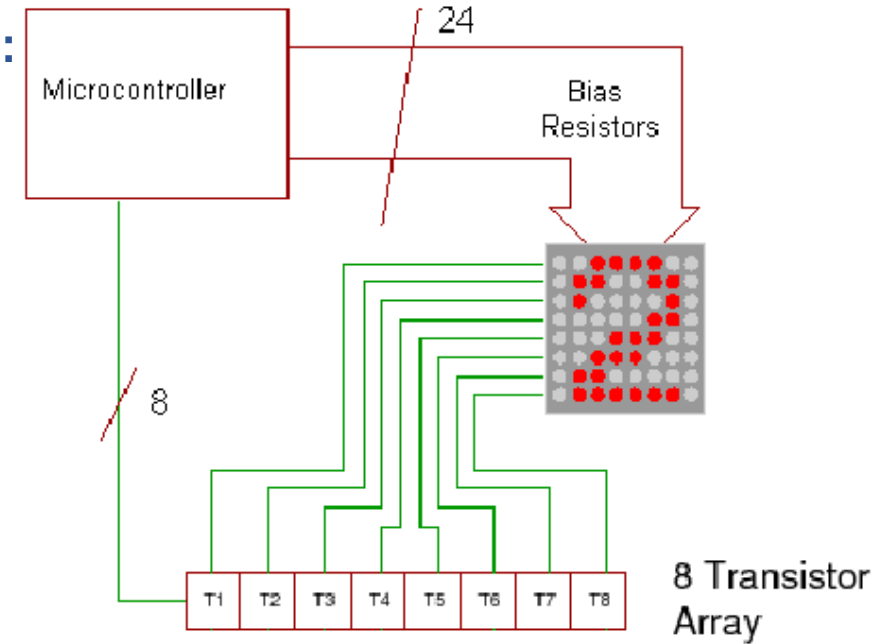


Fig. 43.3

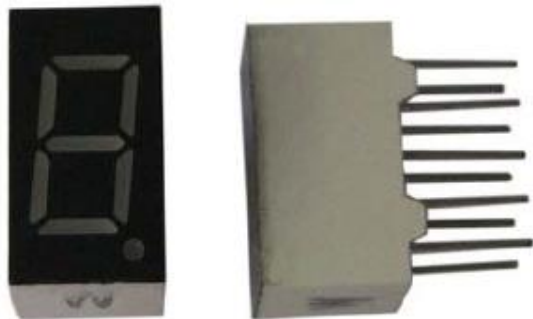


Fig. 43.4

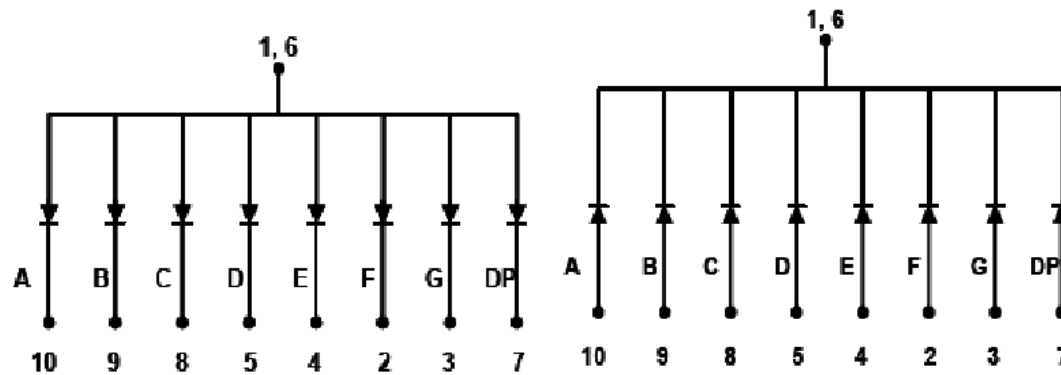


Fig. 43.5

Fig. 43.6

Indicatoare informaționale matriciale și pe segmente. Alfa-numerice:



Fig. 44.1

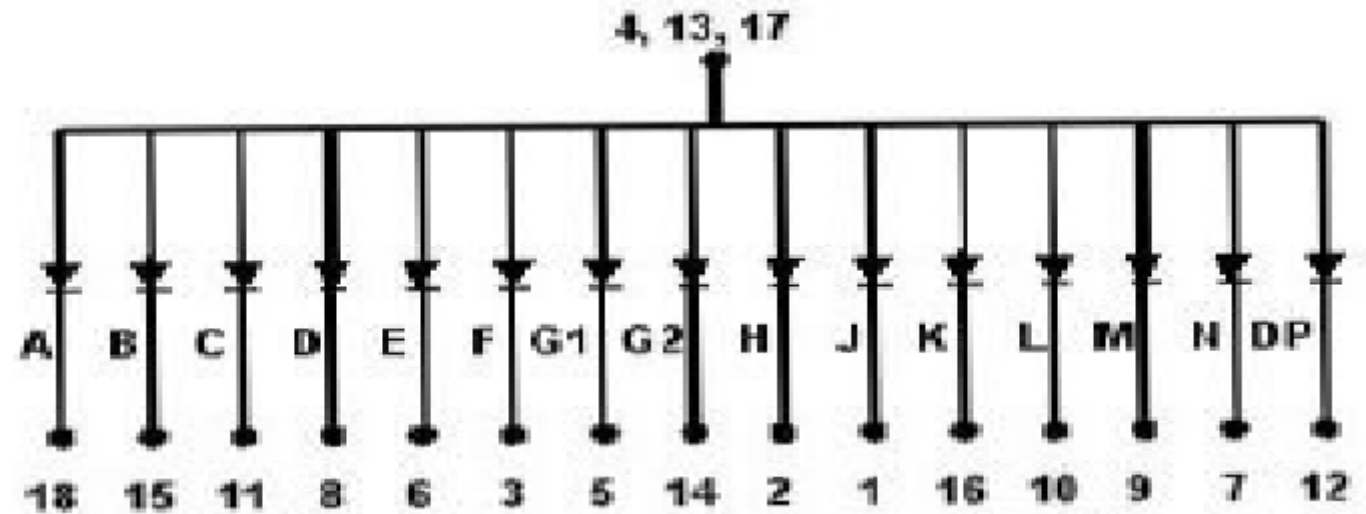


Fig. 44.2

Indicatoare informaționale matriciale și pe segmente. Panouri LED RGB:

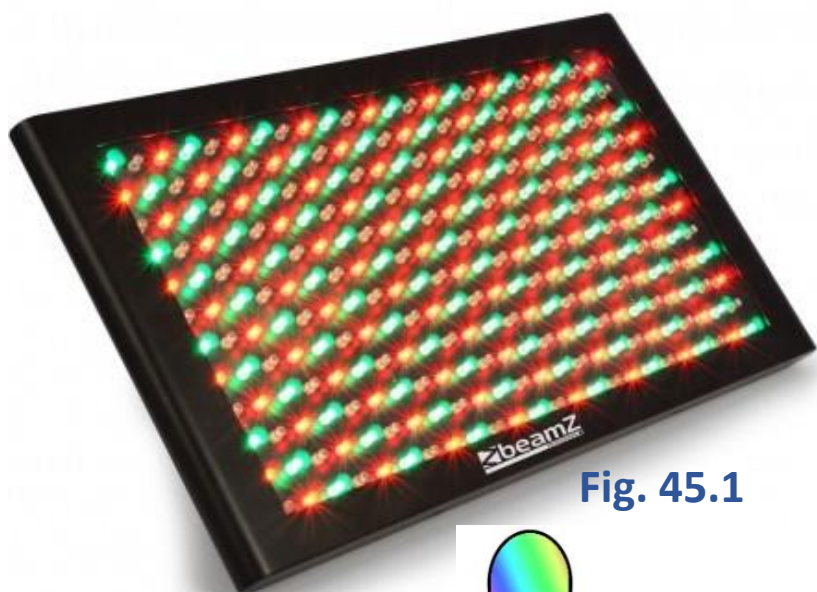


Fig. 45.1

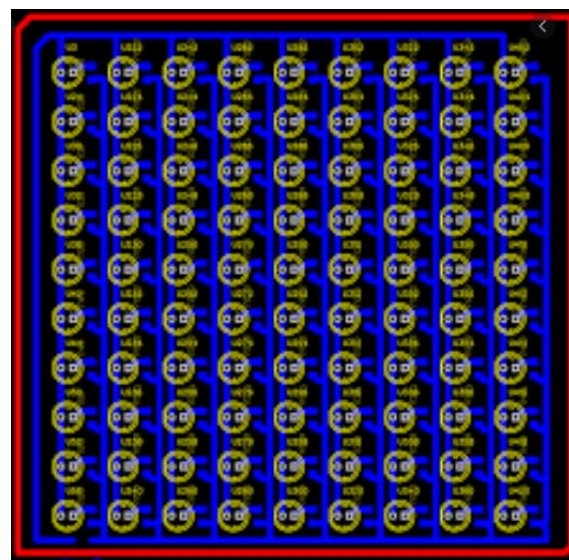


Fig. 45.2

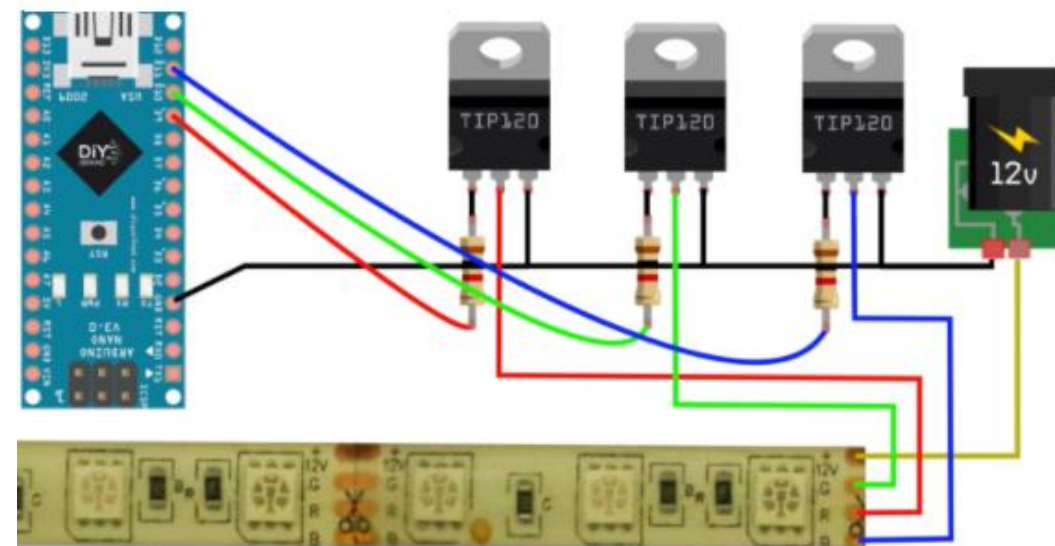


Fig. 45.3

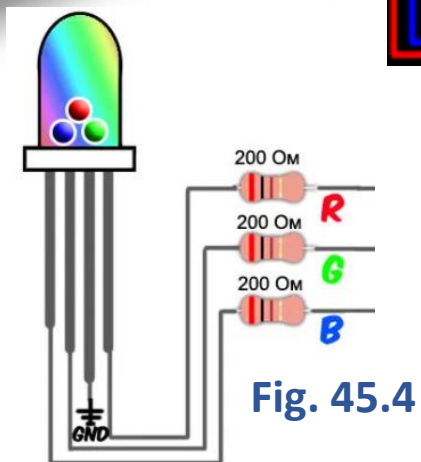


Fig. 45.4

Imprimanta este destinată pentru înregistrarea informației pe suport fizic (hârtie).
Imprimanta poate funcționa în două regimuri: Regim Text sau Regim Grafic:

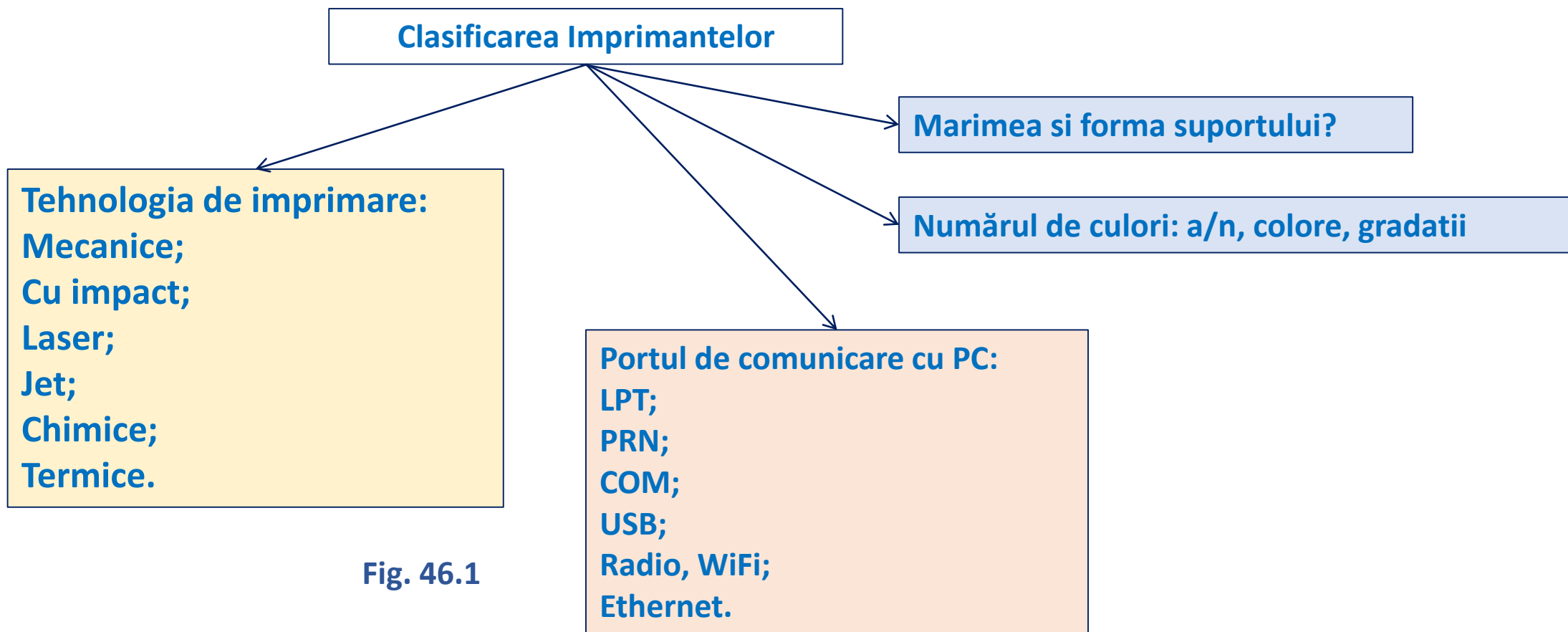


Fig. 46.1

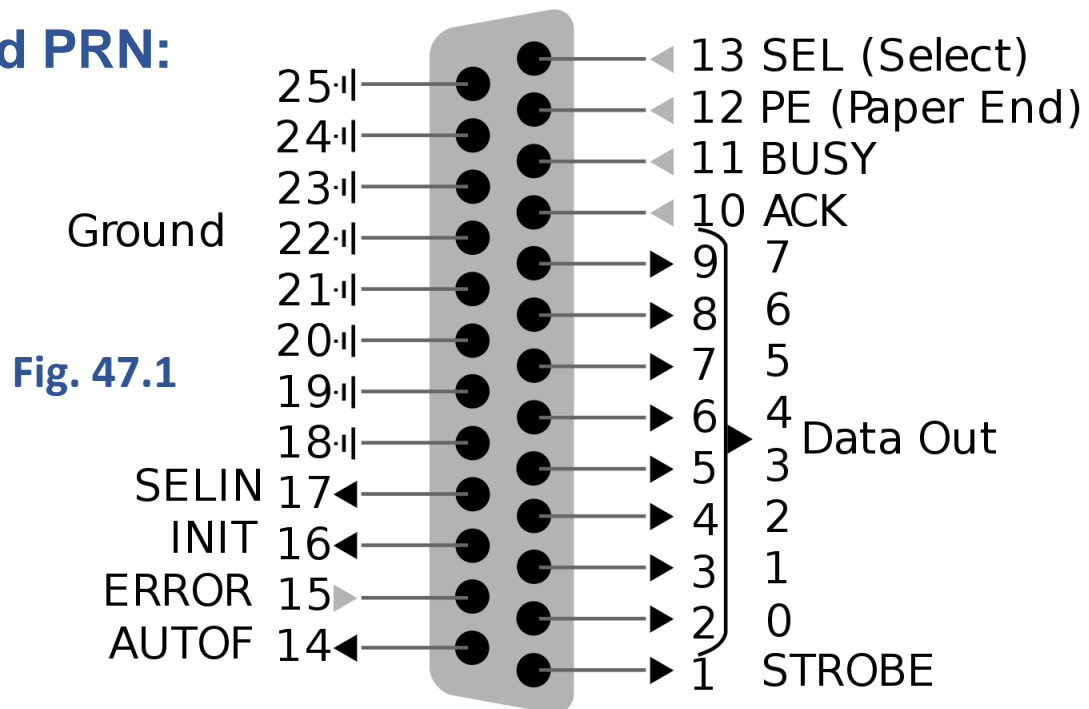
Resursele rezervate de sistem pentru gestiunea Imprimantei:

Întrerupere Hardware: IRQ05h/IRQ07h;

Întrerupere SoftWare: INT 17h;

Porturi I/O: 0x278-0x27F, 0x378-0x37F, 0x3BC-0x3BF.

Portul Standard PRN:



Imprimanta mecanică:



armătură mobilă

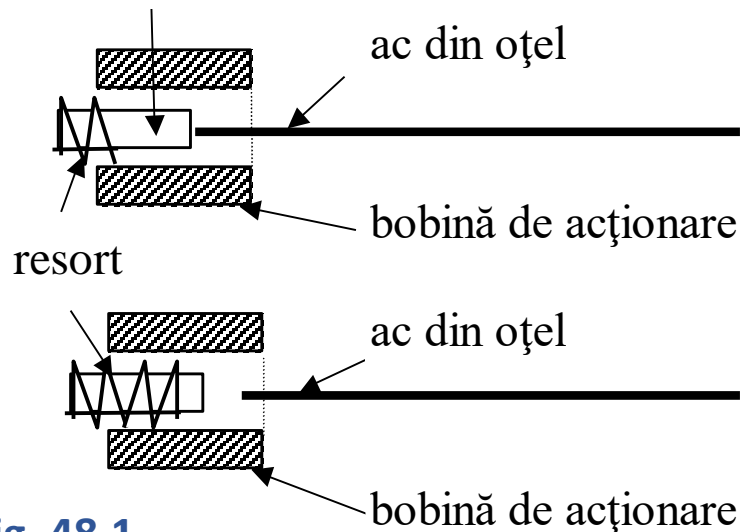


Fig. 48.1

Fig. 48.2

dispunerea bobinelor de acționare

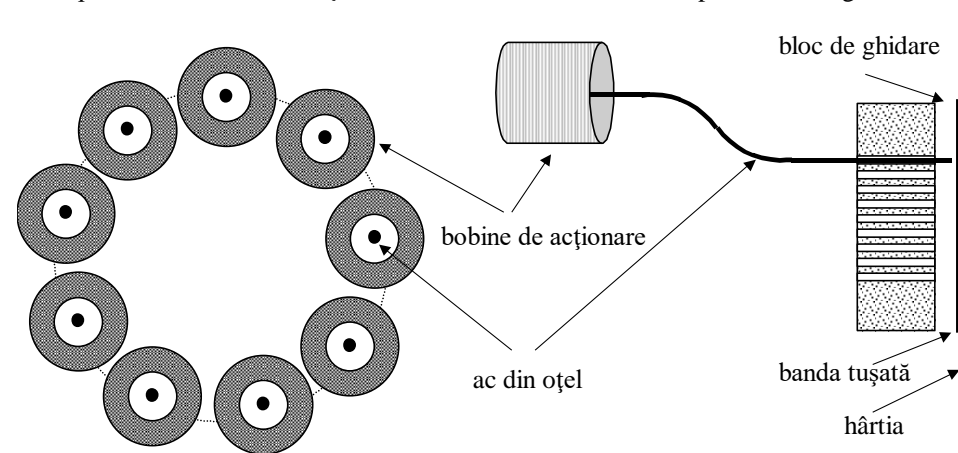


Fig. 48.3

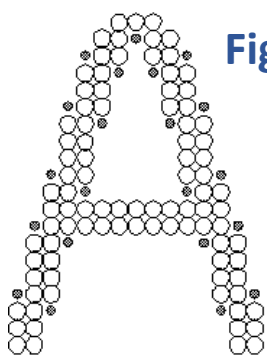
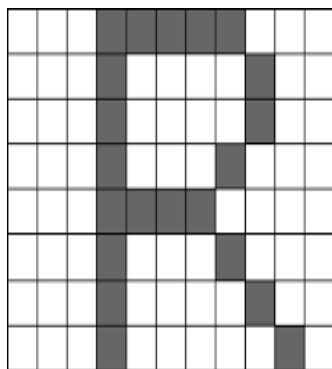


Fig. 48.4



Structura logică de comandă cu Imprimanta:

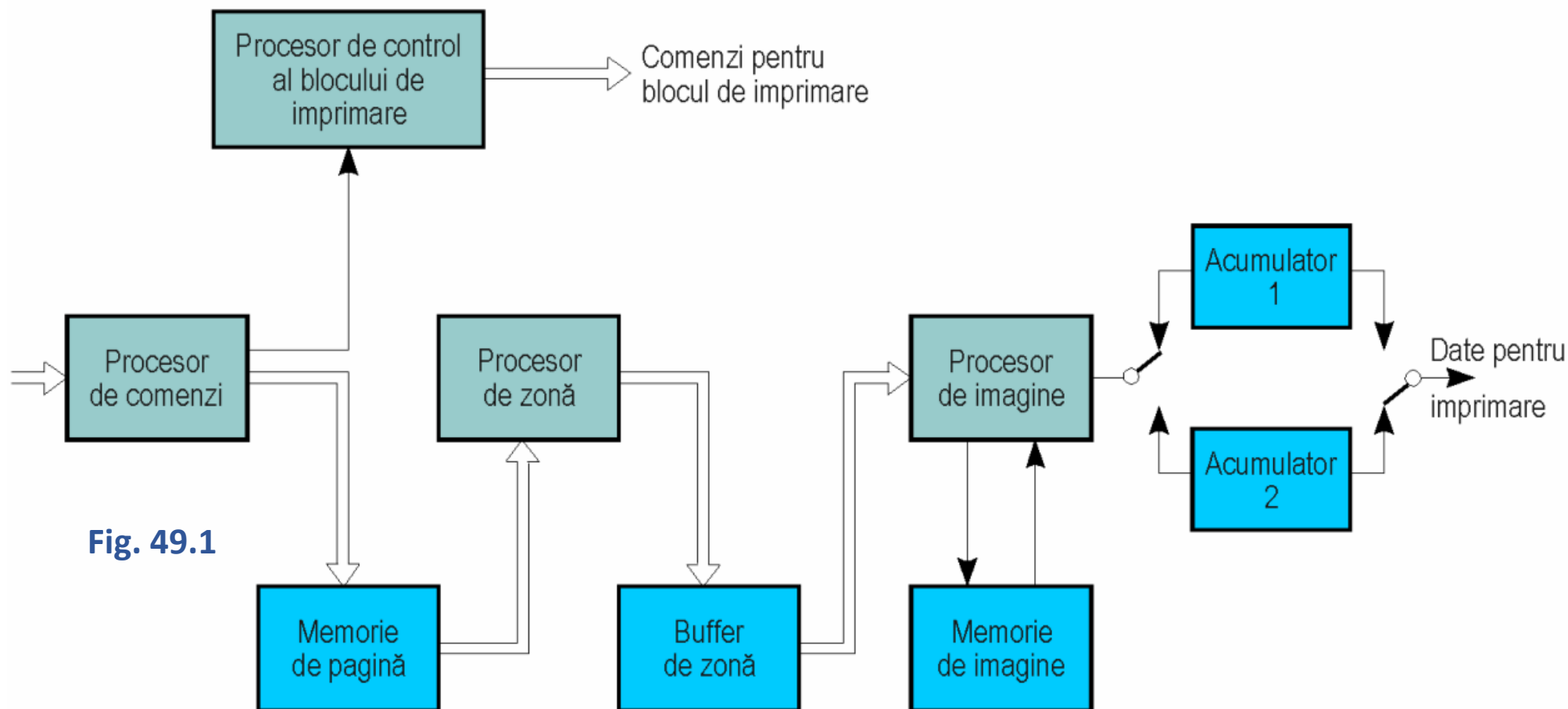


Fig. 49.1

Structura Imprimantei cu Jet:

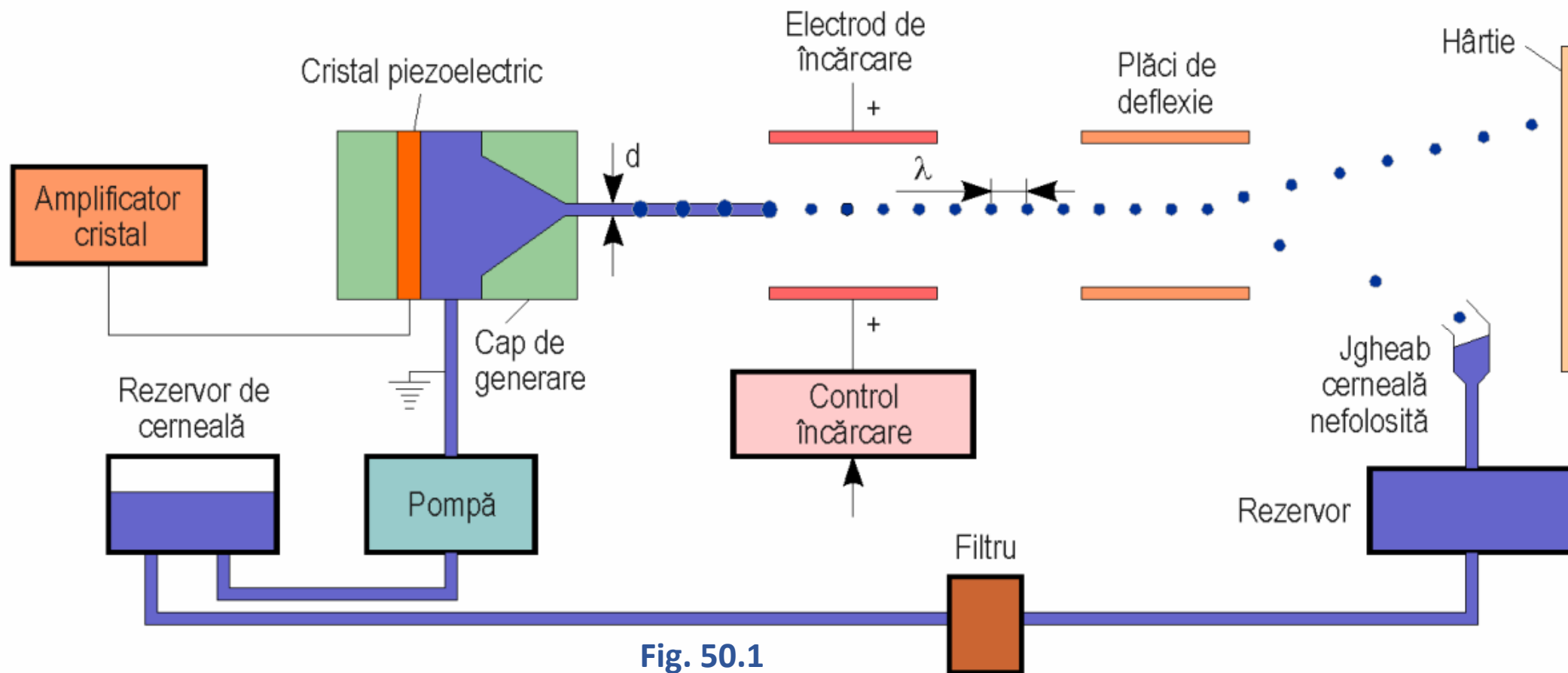


Fig. 50.1

Structura Imprimantei cu Jet. Capul de imprimare:

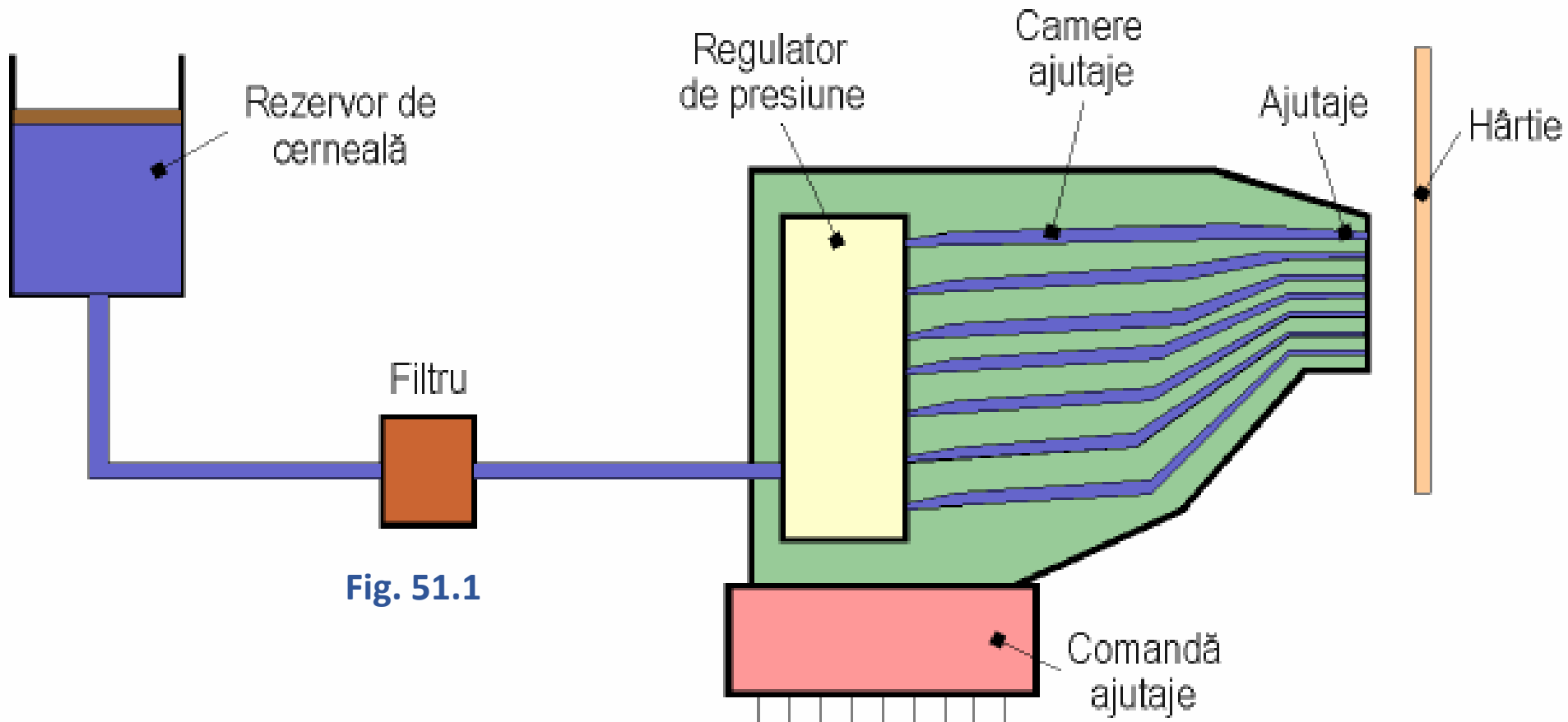


Fig. 51.1

Structura Imprimantei cu Jet. Generarea unei picături prin metoda termică:

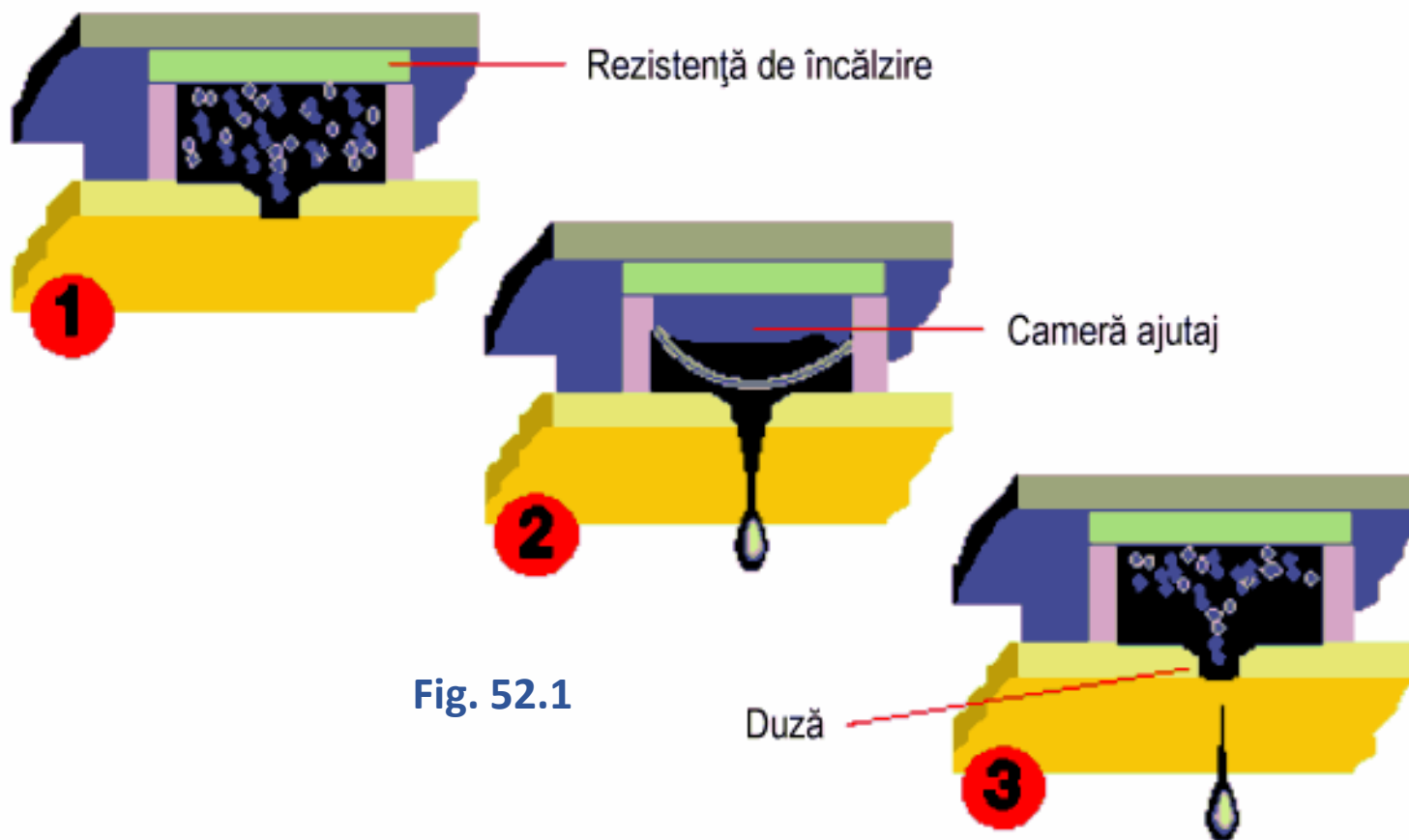


Fig. 52.1

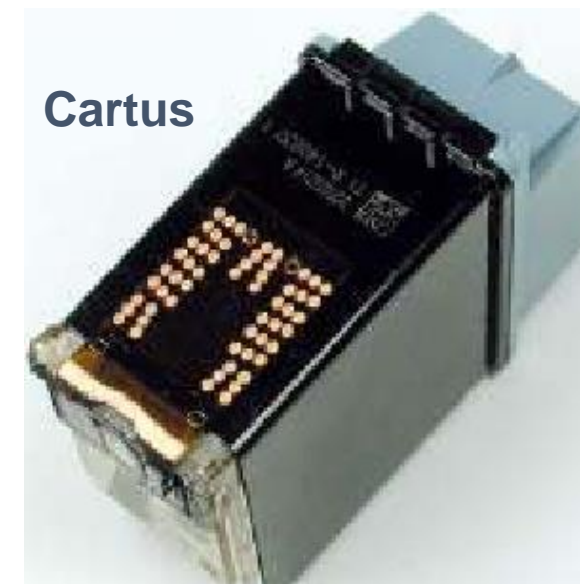
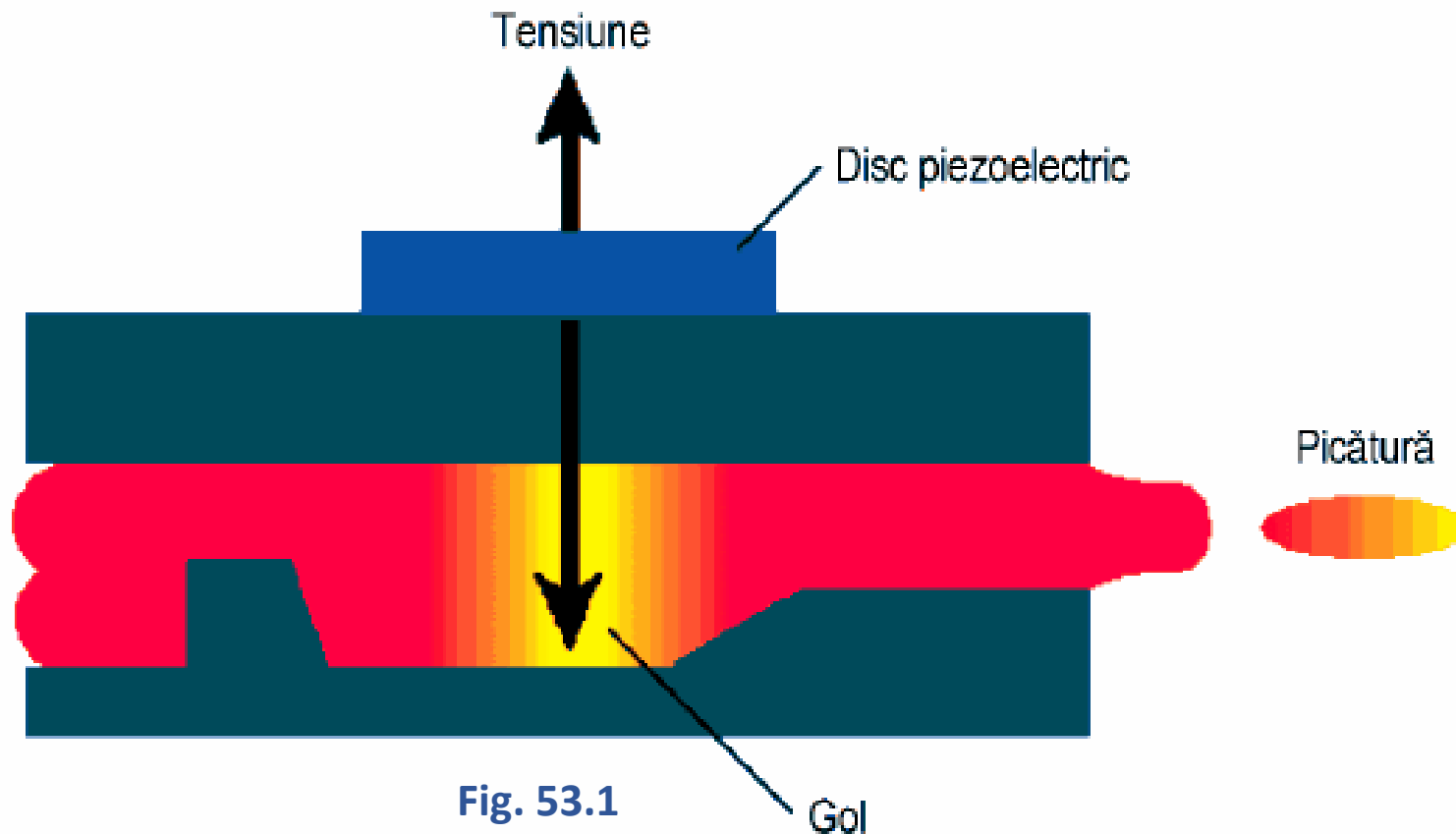


Fig. 52.2

Structura Imprimantei cu Jet. Generarea unei picături prin metoda piezoelectrică:



Structura Imprimantei Laser:

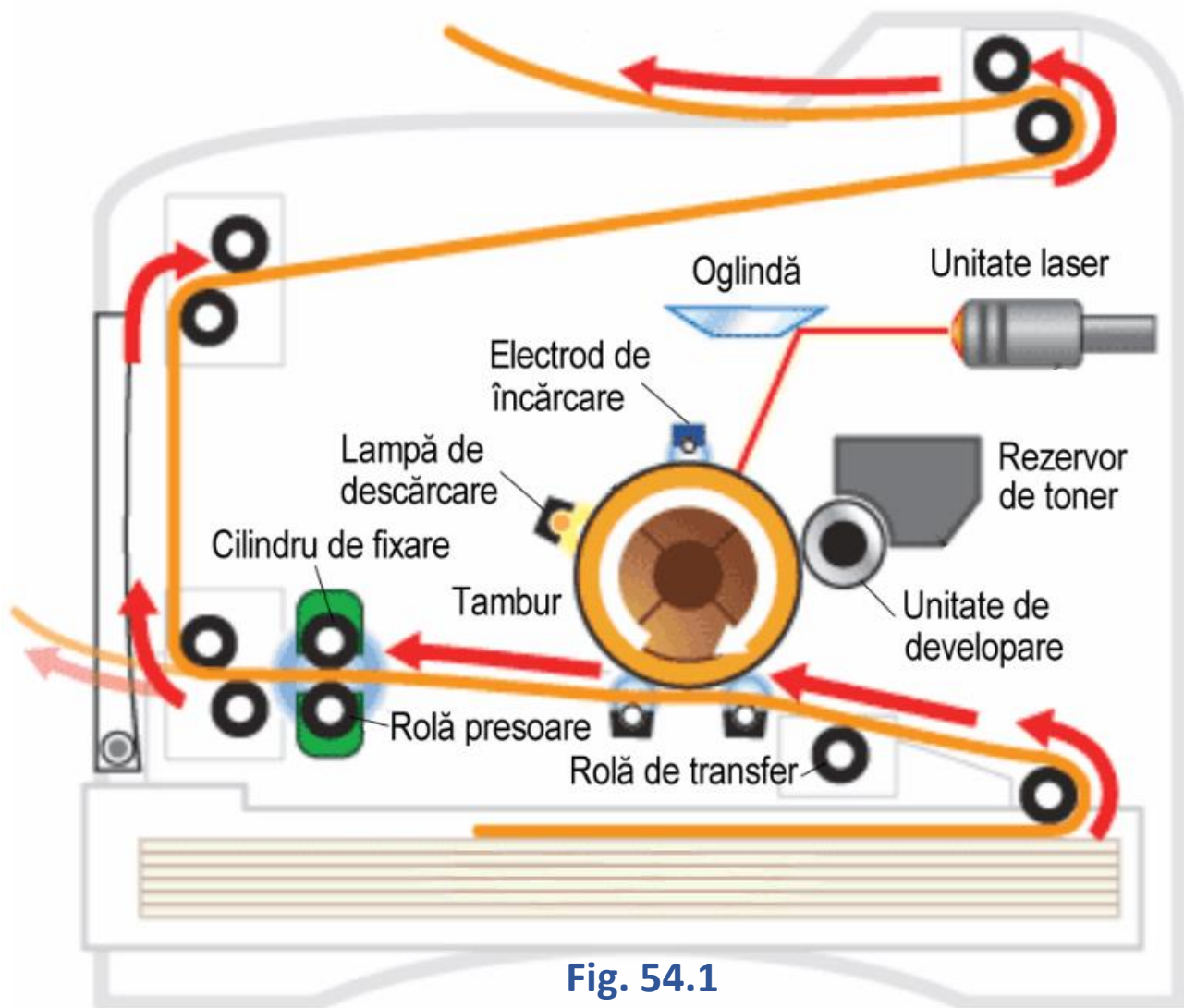
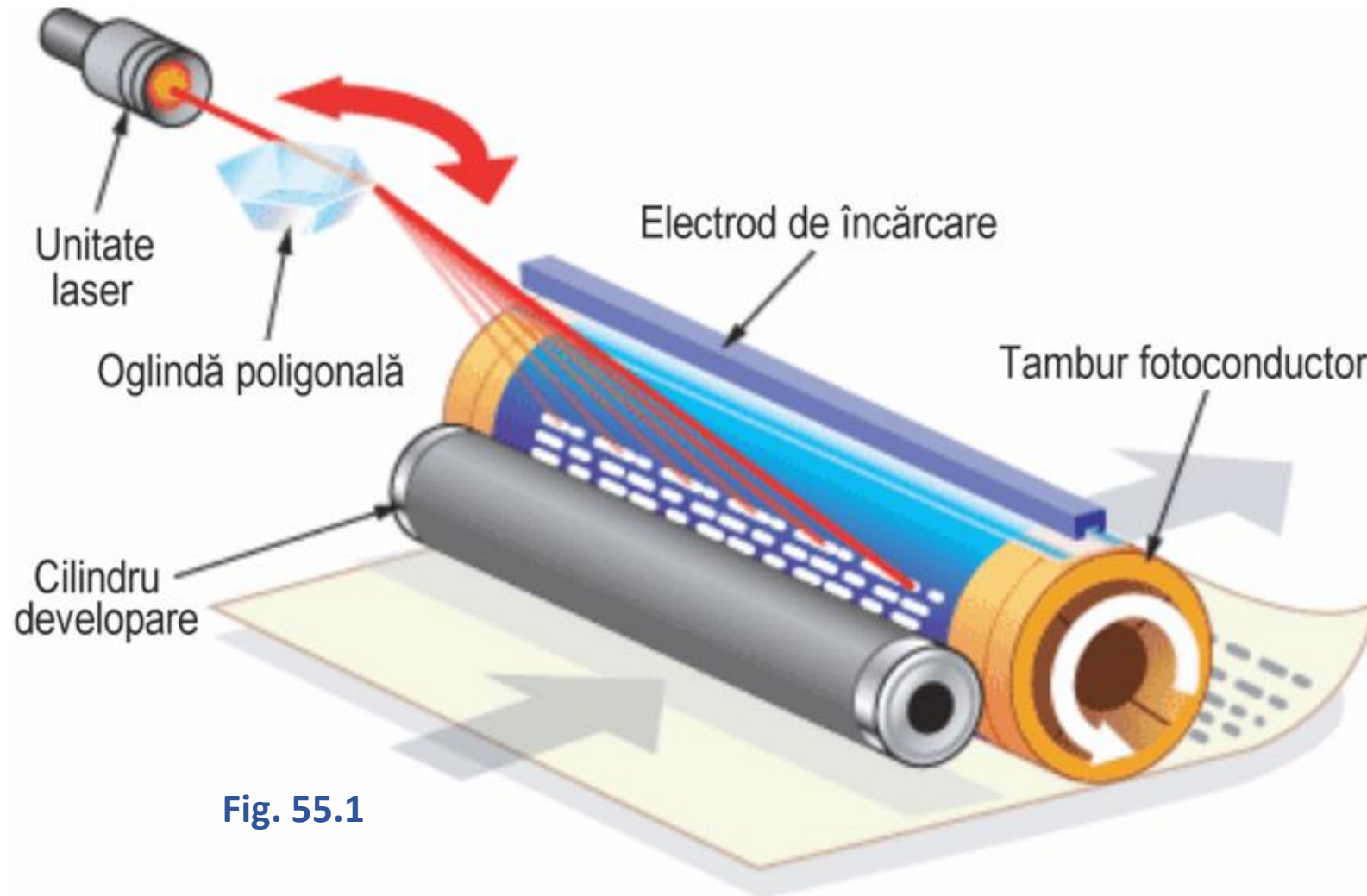


Fig. 54.1



**Structura Imprimantei Laser.
Generarea imaginii pentru
imprimare:**

Fig. 55.1

Structura Imprimantei Laser.
Fixarea imaginii:

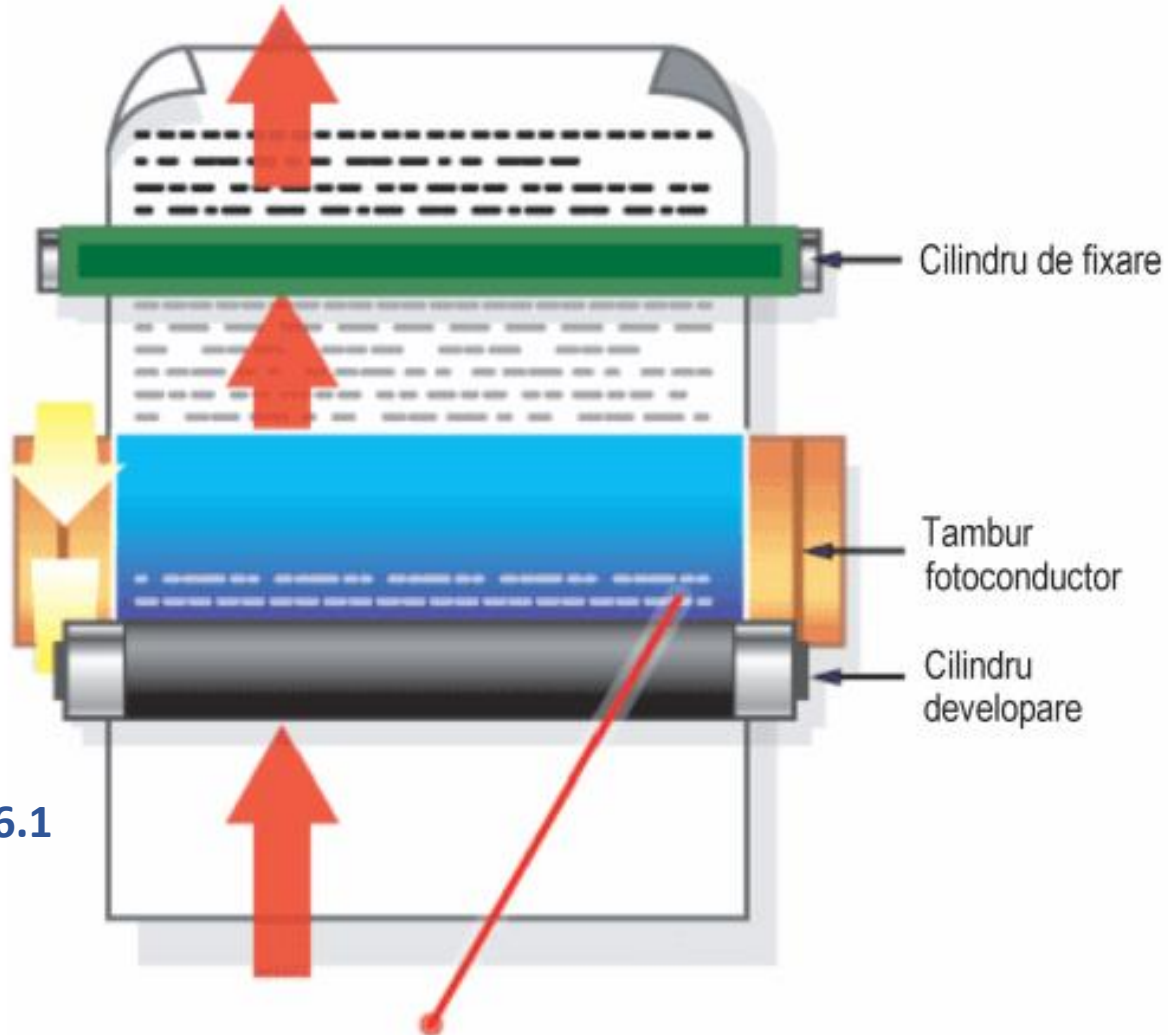


Fig. 56.1

Structura Imprimantei Laser
Coloră:

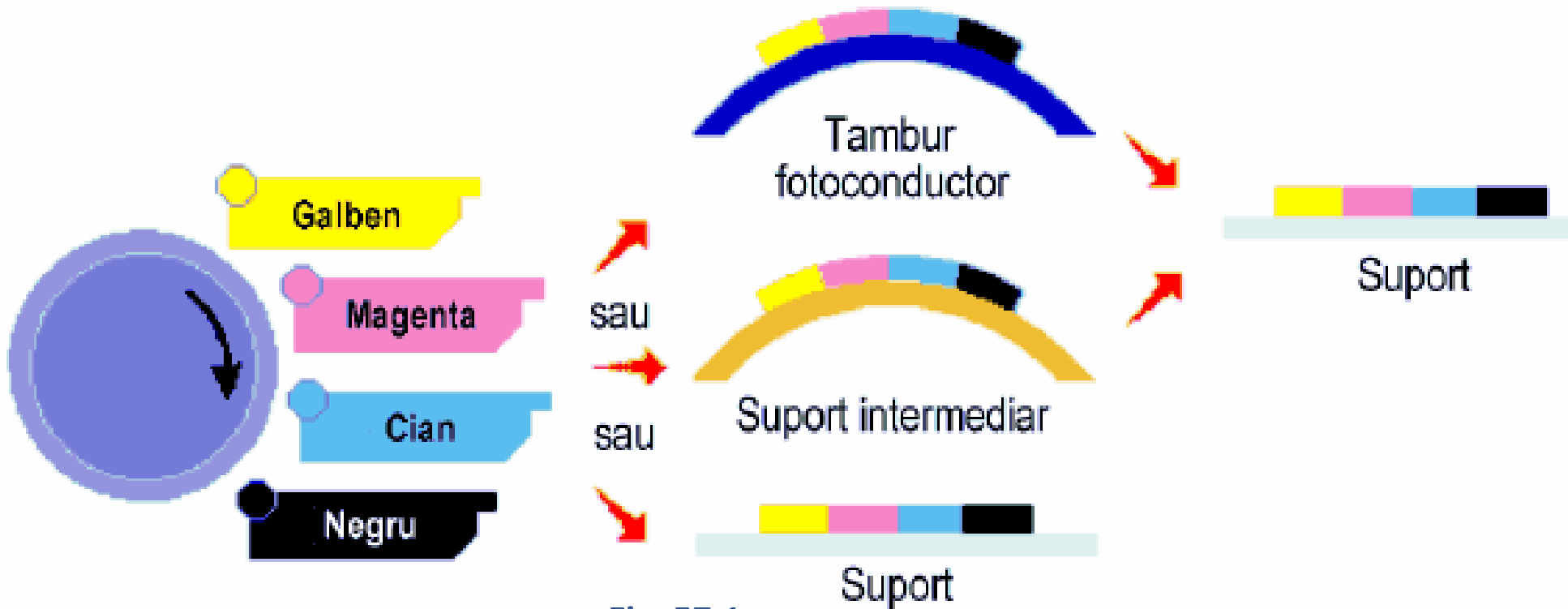


Fig. 57.1

Plotter-ul este un dispozitiv destinat pentru înregistrarea informației pe suport fizic (Hârtie) prin metoda vectorială. Este un dispozitiv electromecanic, principiul său de funcționare fiind combinarea a două deplăsări, axa X și axa Y. Mișcările pot fi făcute ori de capul de imprimare pe ambele axe, ori pe o axă de capul de imprimare iar pe cealaltă axă de hârtie.

Principalele caracteristici ale ploterelor:

- **Modul de amplasare a suportului: orizontal sau vertical;**
- **Numărul de culori generate: a/n, color;**
- **Domeniul de aplicare: imprimare suport hârtie, suport peliculă, panouri de reclamă, imprimare PCB, aşchiere, frezare, operații cu sursă laser;**
- **Dimensiunea suportului,**

Plotter-ul. Exemple:



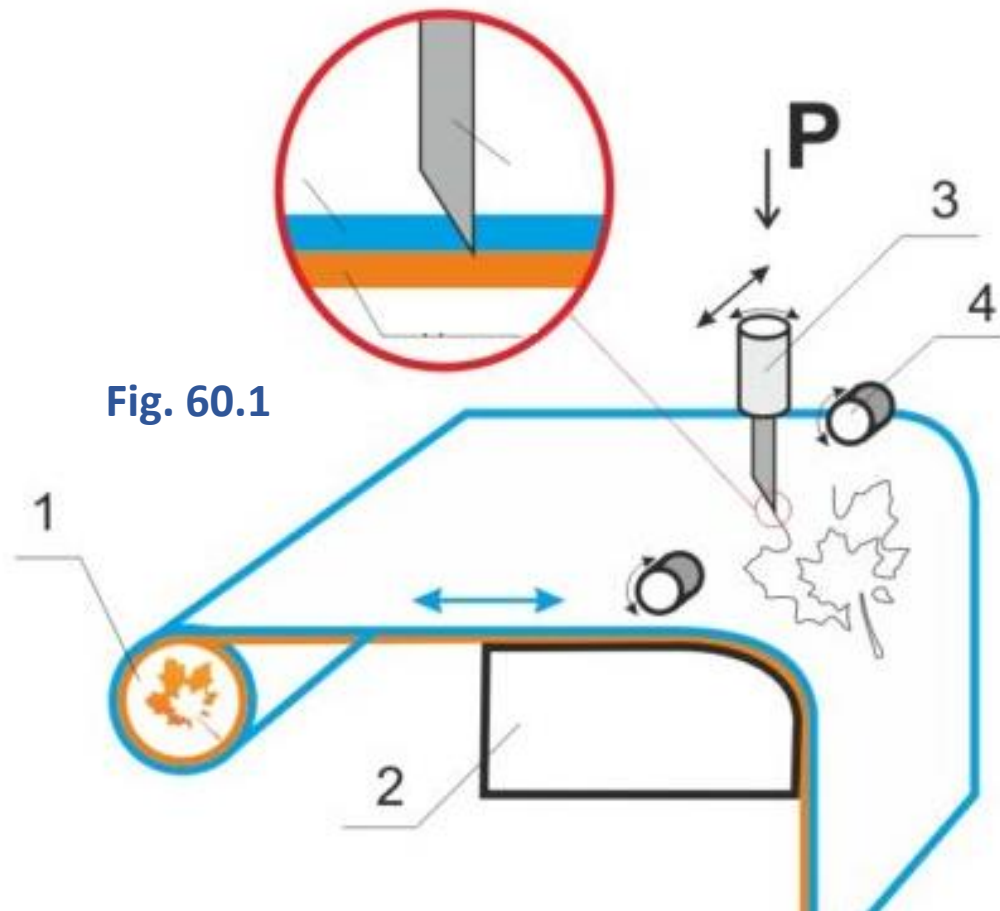
Fig. 59.1



Fig. 59.2



Plotter-ul. Procesul de inregistrare a informației pe suport:



Plotter-ul. Schema de structură clasică:

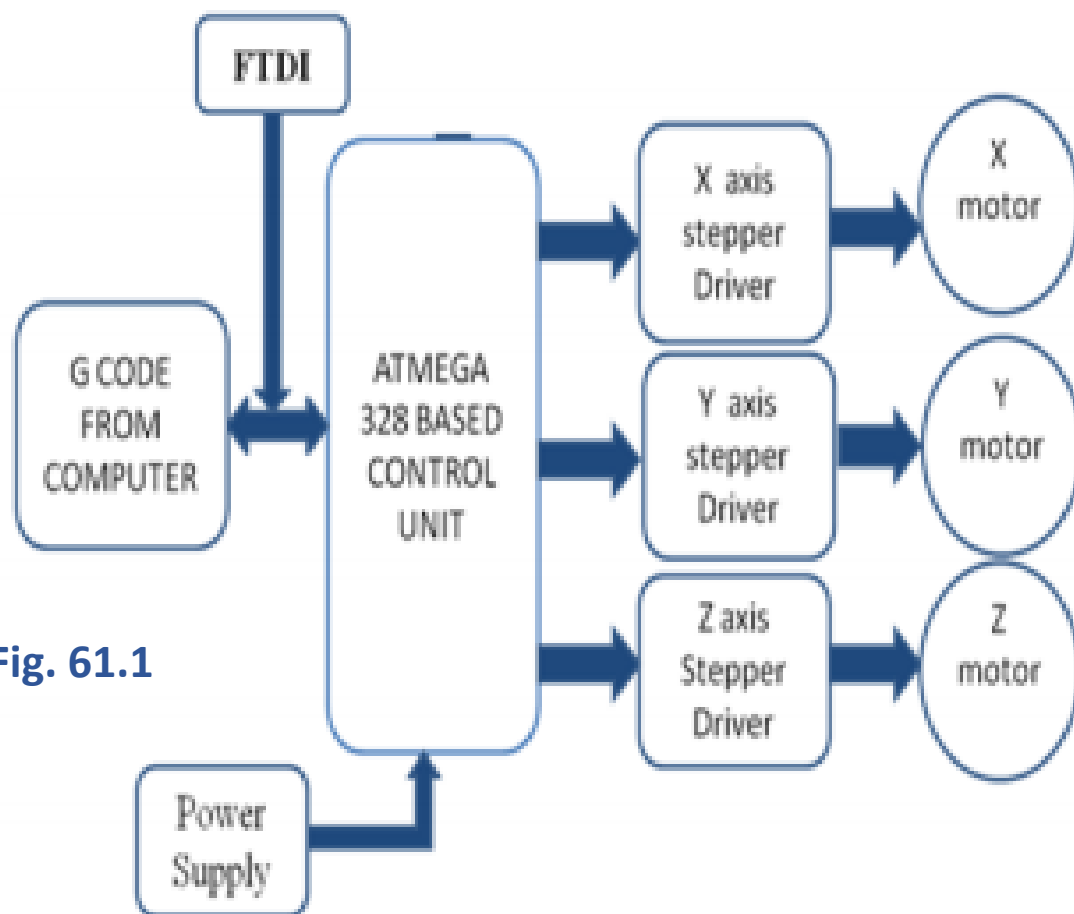


Fig. 61.1

Plotter-ul. Aplicarea materialelor pe suprafața suportului:

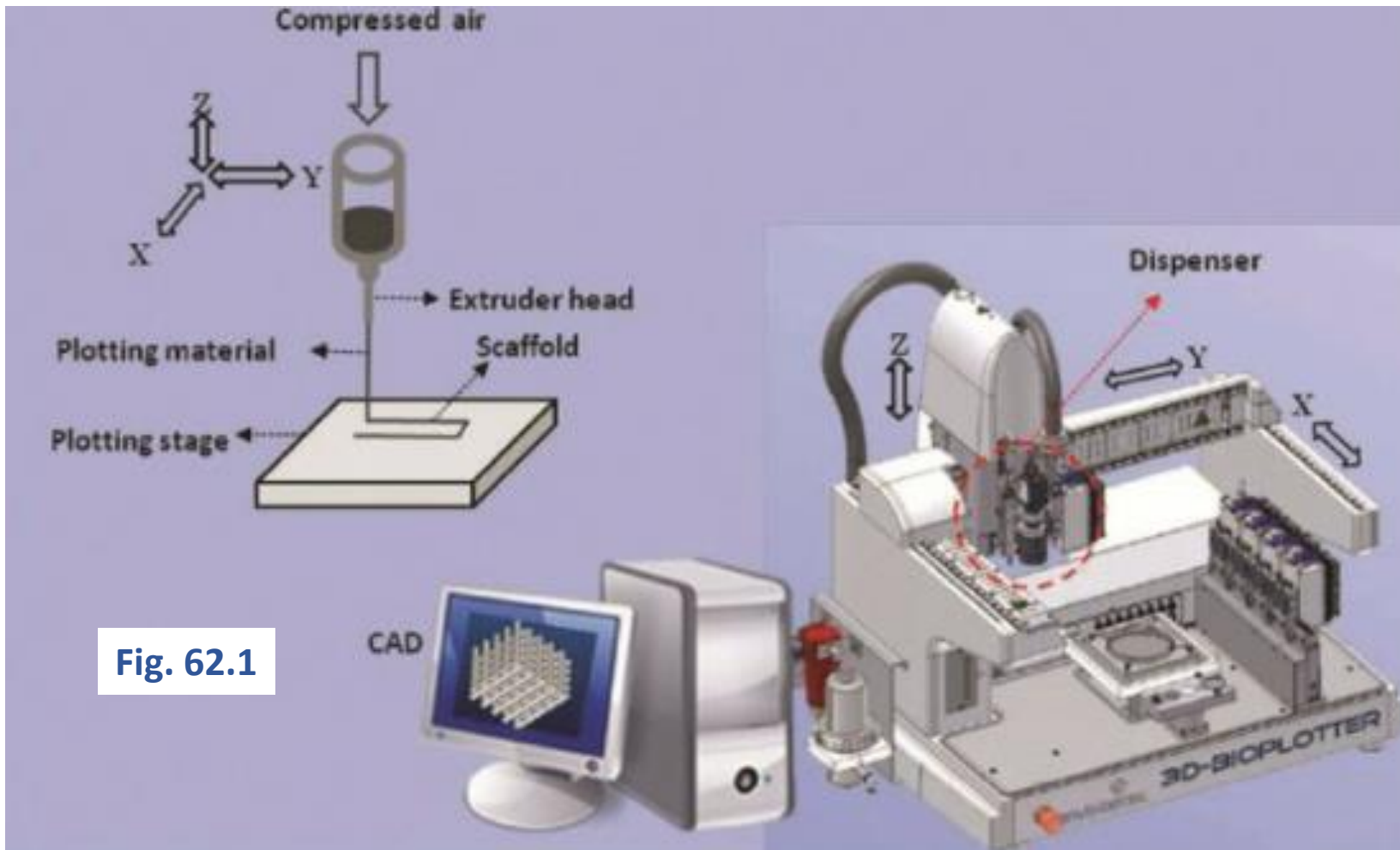


Fig. 62.1

Imprimanta 3D:

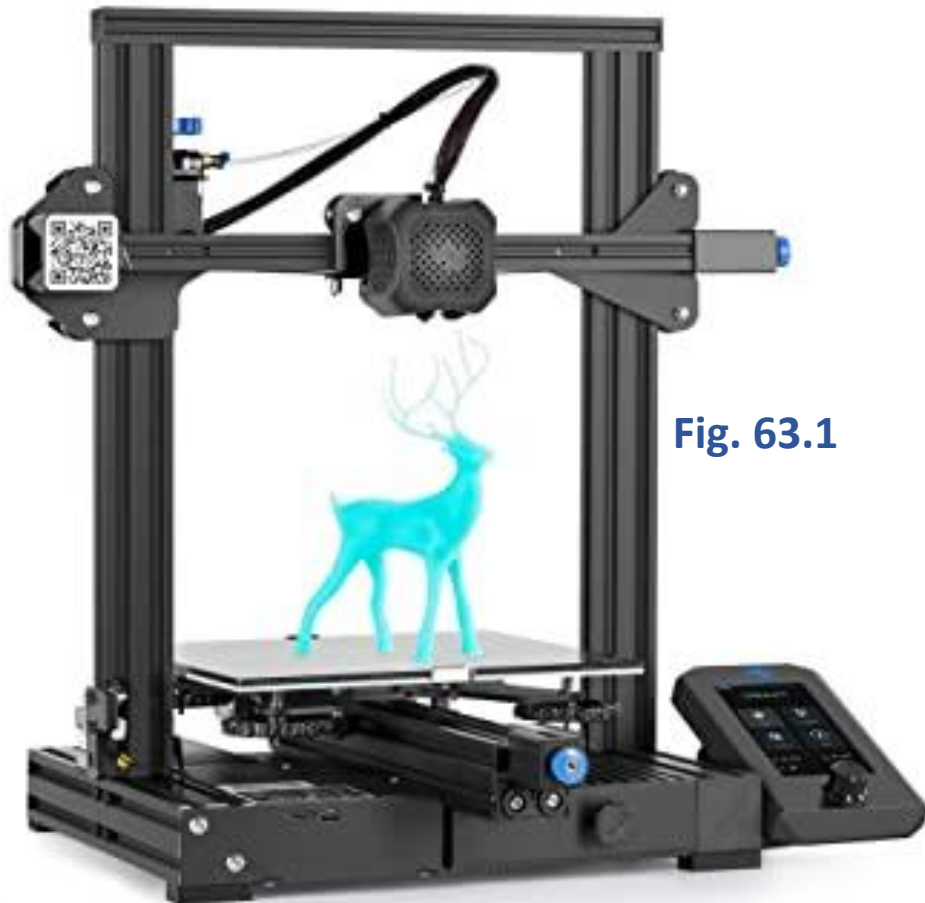


Fig. 63.1

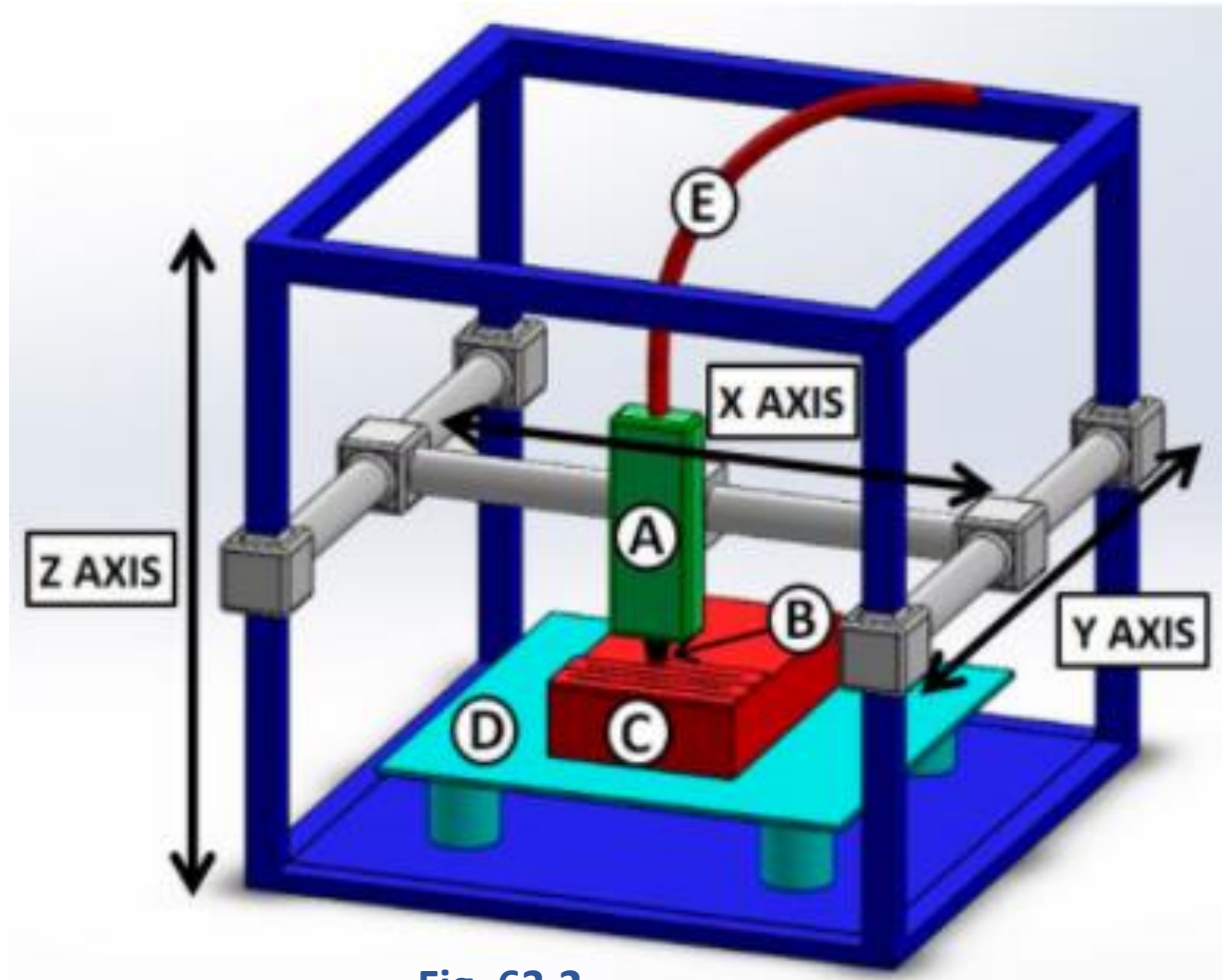


Fig. 63.2

Imprimanta Termică:

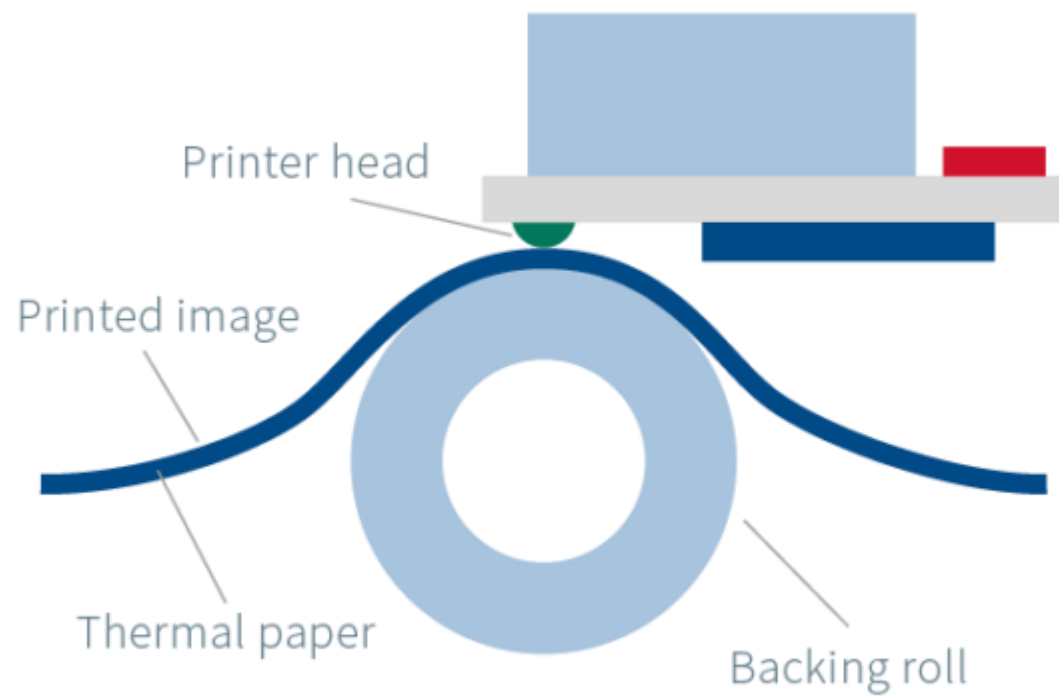


Fig. 64.1

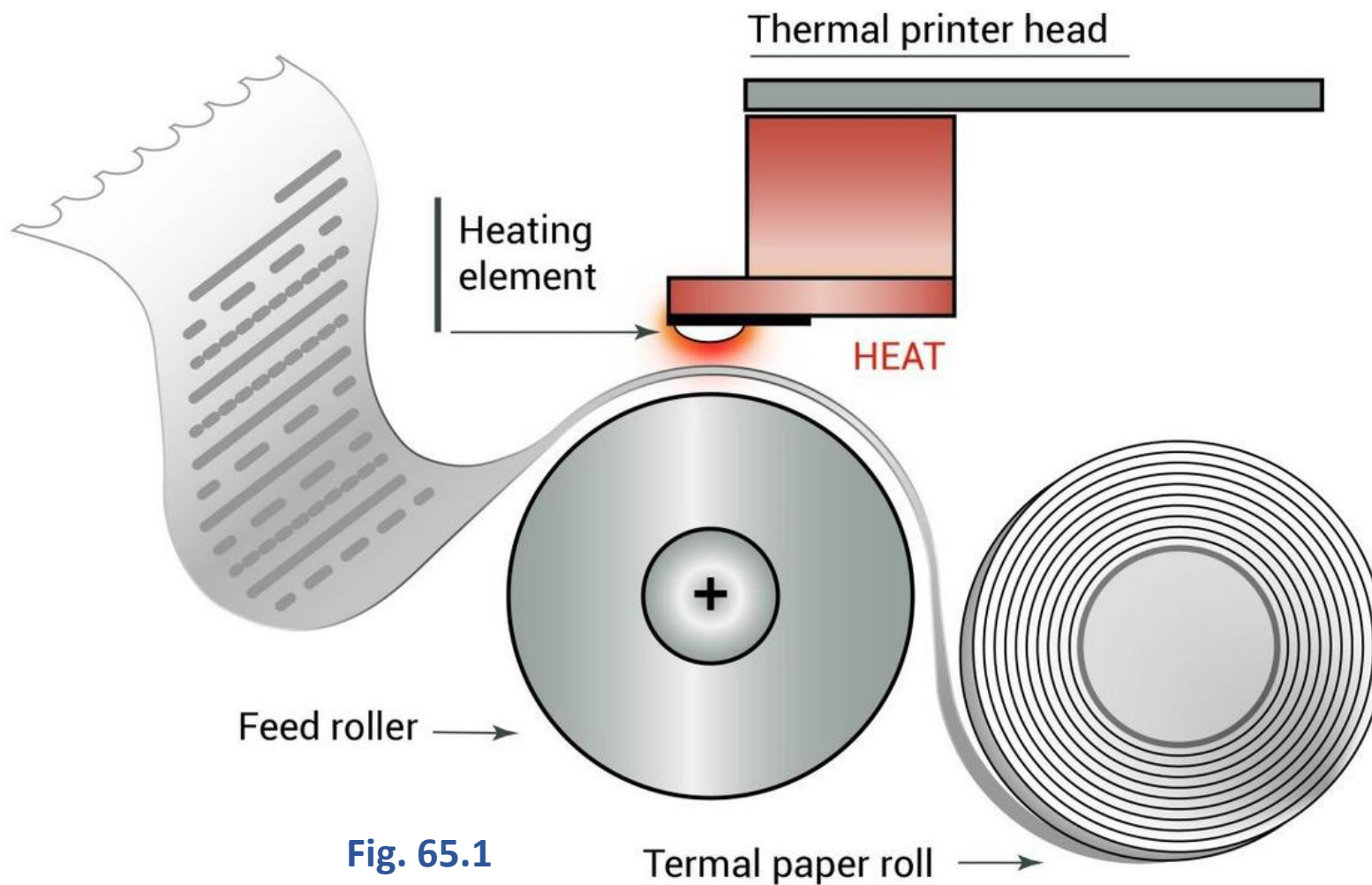


Fig. 65.1

Thermal paper roll

Imprimante Termice:



Fig. 66.1

Pentru programarea Sistemului Video sunt utilizate întreruperea BIOS INT 10h cu tot setul de funcții disponibile și unele funcții ale întreruperii DOS INT 21h.

Funcțiile întreruperii BIOS INT 10h:

INT 10h / AH = 0 - set video mode.

input:

AL = desired video mode.

These video modes are supported:

00h - text mode. 40x25. 16 colors. 8 pages.

03h - text mode. 80x25. 16 colors. 8 pages.

13h - graphical mode. 40x25. 256 colors. 320x200 pixels. 1 page.

Example:

```
mov al, 13h
```

```
mov ah, 0
```

```
int 10h
```

Modul de funcționare a Sistemului Video:

Mod	Regim	Rezoluție text		Rezoluție grafică	Culori	Nr. pagini	RAM video	Standard SVGA
00H	Text	40x25	8x8	320x200	16	8	B8000H	CGA
			8x14	320x350				EGA
			9x16	360x400				VGA
01H	Text	40x25	8x8	320x200	16	8	B8000H	CGA
			8x14	320x350				EGA
			9x16	360x400				VGA
02H	Text	80x25	8x8	640x200	16	8	B8000H	CGA
			8x14	640x350				EGA
			9x16	720x400				VGA
03H	Text	80x25	8x14	640x350	16	8	B8000H	CGA
			9x14	720x350				EGA
			9x16	720x400				VGA
04H	Grafic	40x25	8x8	320x200	4	1	B8000H	CGA
05H	Grafic	40x25	8x8	320x200	4	1	B8000H	CGA
06H	Grafic	80x25	8x8	640x200	a/n	1	B8000H	CGA
07H	Text	40x25	9x14	720x350	mono	1	B8000H	MDA
08H	Grafic	20x25	8x8	160x200	16	1	B8000H	PCjr
09H	Grafic	40x25	8x8	320x200	16	1	B8000H	PCjr
0AH	Grafic	80x25	8x8	640x200	4	1	B8000H	PCjr
0DH	Grafic	40x25	8x8	320x200	16	8	A0000H	EGA
0EH	Grafic	80x25	8x8	640x200	16	4	A0000H	EGA
0FH	Grafic	80x25	8x14	640x350	mono	2	A0000H	EGA
10H	Grafic	80x25	8x14	640x350	16	2	A0000H	EGA
11H	Grafic	80x30	8x16	640x480	mono	1	A0000H	VGA
12H	Grafic	80x30	8x16	640x480	16	1	A0000H	VGA
13H	Grafic	40x30	8x8	320x200	256	1	A0000H	VGA
4EH	Text	80x60	8x8	640x480	16	1	B8000H	SVGA
4FH	Text	132x60	8x8	1056x480	16	1	B8000H	SVGA
50H	Text	132x25	8x14	1056x350	16	1	B8000H	SVGA
51H	Text	132x43	8x8	1056x344	16	1	B8000H	SVGA
52H	Grafic	100x37	8x16	800x600	16	1	A0000H	SVGA
53H	Grafic	80x30	8x16	640x480	256	1	A0000H	SVGA
54H	Grafic	100x37	8x16	800x600	256	1	A0000H	SVGA
55H	Grafic	128x48	8x16	1024x768	4	1	A0000H	SVGA
56H	Grafic	128x48	8x16	1024x768	16	1	A0000H	SVGA
57H	Grafic	96x64	8x16	768x1024	16	1	A0000H	SVGA
58H	Grafic	160x64	8x16	1280x1024	16	1	A0000H	SVGA
59H	Grafic	128x48	8x16	1024x768	256	1	A0000H	SVGA

Tab. 68.1

Funcțiile întreprerii BIOS INT 10h:

INT 10h / AH = 2 - set cursor position.

input:

DH = row.

DL = column.

BH = page number (0..7).

Example:

```
mov dh, 10
```

```
mov dl, 20
```

```
mov bh, 0
```

```
mov ah, 2
```

```
int 10h
```

Funcțiile întreprerii BIOS INT 10h:

INT 10h / AH = 08h - read character and attribute at cursor position.

input:

BH = page number.

return:

AH = attribute.

AL = character.

INT 10h / AH = 09h - write character and attribute at cursor position.

input:

AL = character to display.

BH = page number.

BL = attribute.

CX = number of times to write character.

Funcțiile întreprerii BIOS INT 10h:

INT 10h / AH = 0Ch - change color for a single pixel.

input:

AL = pixel color

CX = column.

DX = row.

Example:

```
mov al, 13h
mov ah, 0
int 10h ; set graphics video mode.
mov al, 1100b
mov cx, 10
mov dx, 20
mov ah, 0ch
int 10h ; set pixel.
```

Funcțiile întreprerii BIOS INT 10h:

INT 10h / AH = 0Dh - get color of a single pixel.

input:

CX = column.

DX = row.

output:

AL = pixel color

Funcțiile întreprerii BIOS INT 10h:

INT 10h / AH = 13h - write string.

input:

AL = write mode:

bit 0: update cursor after writing;

bit 1: string contains attributes.

BH = page number.

BL = attribute if string contains only characters (bit 1 of AL is zero).

CX = number of characters in string (attributes are not counted).

DL,DH = column, row at which to start writing.

ES:BP points to string to be printed.

Example:

```
mov al, 1
mov bh, 0
mov bl, 0011_1011b
mov cx, msg1end - offset msg1 ; calculate message size.
```

```
mov dl, 10
mov dh, 7
push cs
pop es
mov bp, offset msg1
mov ah, 13h
int 10h
jmp msg1end
msg1 db " hello, world! "
msg1end:
RET
```

Funcțiile întreprerii DOS INT 21h:

INT 21h / AH=2 - write character to standard output.
entry: **DL** = character to write, after execution **AL = DL**.

Example:

```
mov ah, 2  
mov dl, 'a'  
int 21h
```

INT 21h / AH=5 - output character to printer.
entry: **DL** = character to print, after execution **AL = DL**.

Example:

```
mov ah, 5  
mov dl, 'a'  
int 21h
```

Funcțiile întreprerii DOS INT 21h:

INT 21h / AH=9 - output of a string at **DS:DX**.
String must be terminated by '\$'.

Example:

```
org 100h
mov dx, offset msg
mov ah, 9
int 21h
ret
msg db "hello world $"
```

Afișarea datelor pe ecranul videomonitorului în Regim Text poate fi efectuată în 2 moduri:

- Prin utilizarea metodelor de adresare directă a memoriei RAM Video;
- Cu ajutorul funcțiilor întreruperii INT 10h.

Să se elaboreze programul care afișează informația în Regim Text pe ecranul videomonitorului utilizând adresarea directă a memoriei RAM Video:

- RAM Video = B8000h
- Fiecare caracter în memorie ocupă 2 octeți: Codul ASCII și Atributul;
- Introducerea caracterului de la KBD;
- Manipularea cu poziția cursorului:

O – în sus, m – în jos, k = în stânga, l – în dreapta, c- schimbă culoarea.



name "int10h"

JMP Start

Charr DB ?

Start:

org 100h

MOV AL, 00h

MOV AH, 03h

int 10h

MOV BX, 0B800h

MOV ES, BX

MOV BX, 00h ; Addressa RAM Video

MOV DL, 0F0h ; Color Char + BG

MOV CX, 020h

Label:

MOV AH, 00h

int 16h ; Input Char

MOV Charr, AL

CMP AL, "o"

JNZ Sus ; Deplasare in sus

SUB BX, 160

JMP Label

Sus:

MOV AL, Charr

CMP AL, "m"

JNZ Jos ; Deplasare in jos

ADD BX, 160

JMP Label

Jos:

MOV AL, Charr

CMP AL, "k"

JNZ L ; Deplasare in stinga

SUB BX, 4

JMP Label

L:

MOV AL, Charr

CMP AL, "I"

JNZ R ; Deplasare in dreapta

ADD BX, 2

JMP Label

R:

MOV AL, Charr

CMP AL, "c"

JNZ C ; Schimba culoarea caracterului

INC DL

JMP Label

C:

MOV AL, Charr

MOV ES:[BX], AL ; Write Char RAM

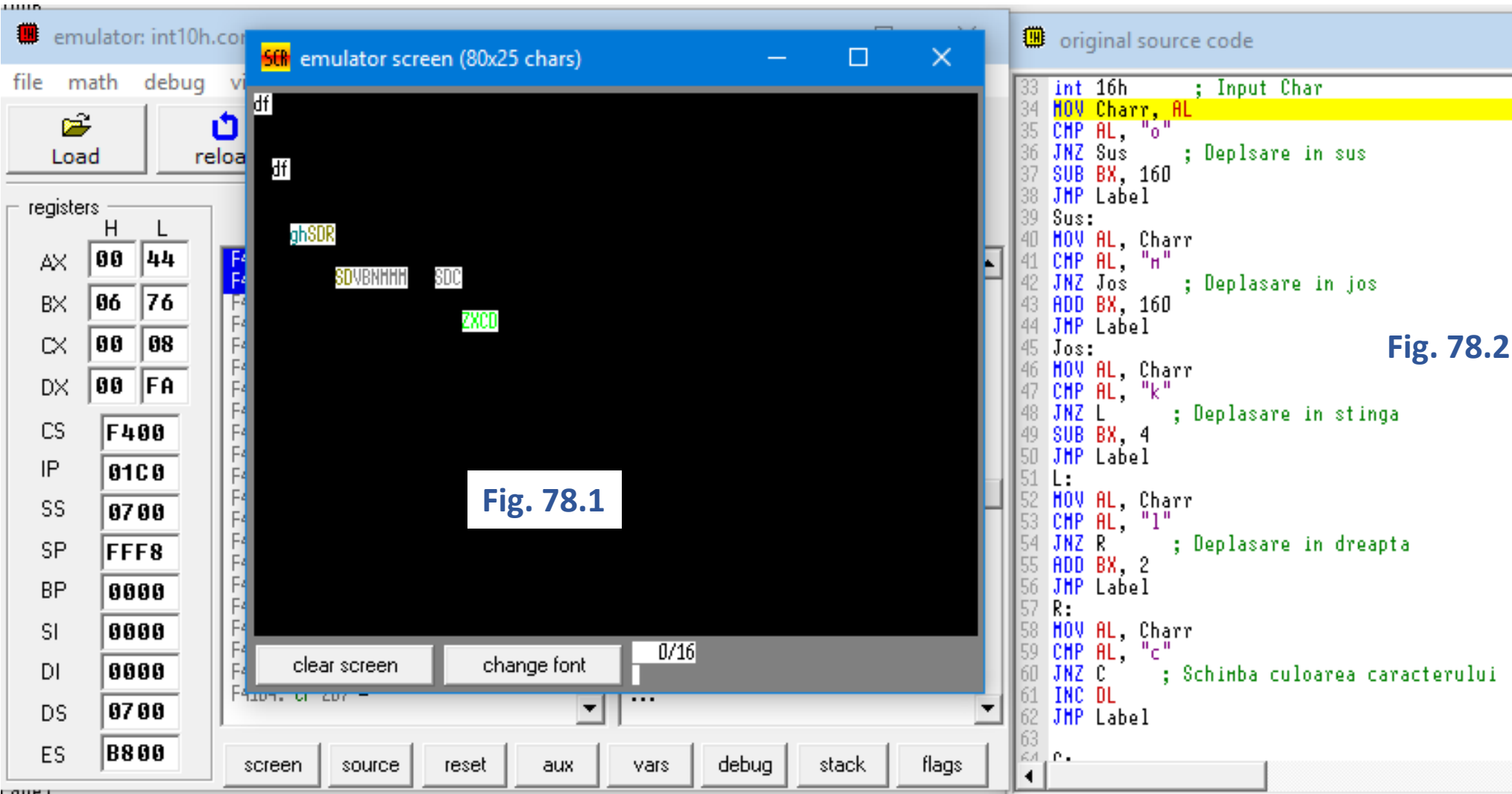
INC BX

MOV ES:[BX], DL ; Write Attribut RAM

INC BX

Loop Label

RET



The image shows a screenshot of an x86 emulator interface. On the left, a 'registers' window displays the state of various registers:

Register	H	L
AX	00	44
BX	06	76
CX	00	08
DX	00	FA
CS	F400	
IP	01C0	
SS	0700	
SP	FFF8	
BP	0000	
SI	0000	
DI	0000	
DS	0700	
ES	B800	

The main window, titled 'emulator screen (80x25 chars)', displays a video screen with the following text:

```

df
df
ghSDR
SDVBNHHH SOC
EXCD
  
```

Below the screen are buttons for 'clear screen' and 'change font', along with a font size indicator '0/16'. At the bottom, there are tabs for 'screen', 'source', 'reset', 'aux', 'vars', 'debug', 'stack', and 'flags'.

On the right, the 'original source code' window shows the following assembly code:

```

33 int 16h ; Input Char
34 MOV Charr, AL
35 CMP AL, "o"
36 JNZ Sus ; Deplasare in sus
37 SUB BX, 160
38 JMP Label
39 Sus:
40 MOV AL, Charr
41 CMP AL, "h"
42 JNZ Jos ; Deplasare in jos
43 ADD BX, 160
44 JMP Label
45 Jos:
46 MOV AL, Charr
47 CMP AL, "k"
48 JNZ L ; Deplasare in stinga
49 SUB BX, 4
50 JMP Label
51 L:
52 MOV AL, Charr
53 CMP AL, "l"
54 JNZ R ; Deplasare in dreapta
55 ADD BX, 2
56 JMP Label
57 R:
58 MOV AL, Charr
59 CMP AL, "c"
60 JNZ C ; Schimba culoarea caracterului
61 INC DL
62 JMP Label
63
  
```

Fig. 78.1

Fig. 78.2

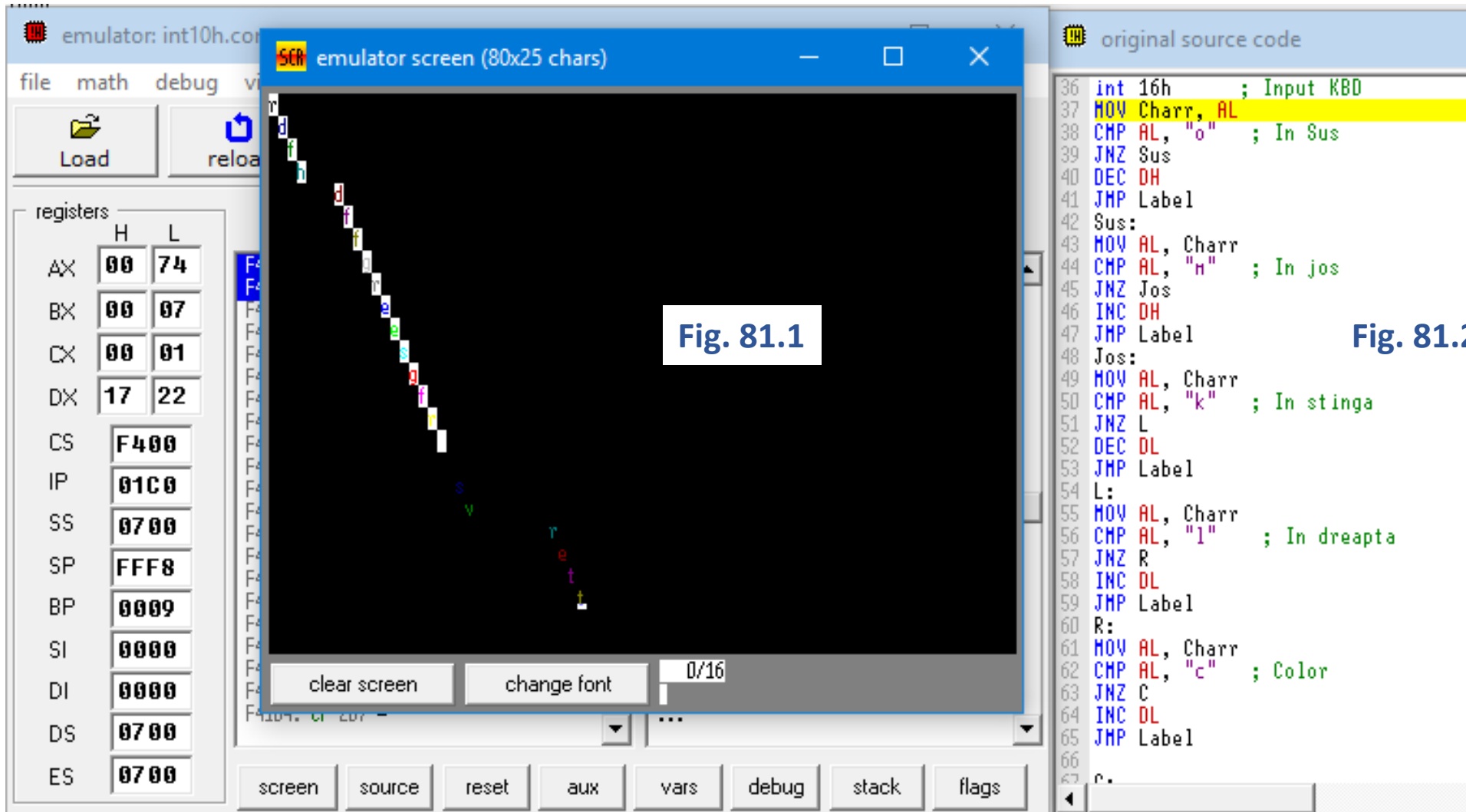
Să se elaboreze programul care afișează informația in Regim Text pe ecranul videomonitorului utilizând funcțiile întreruperii INT 10h:

- RAM Video = B8000h
- Fiecare caracter în memorie ocupă 2 octeți: Codul ASCII și Atributul;
- Introducerea caracterului de la KBD;
- Manipularea cu poziția cursorului:

O – în sus, m – în jos, k = în stânga, l – în dreapta, c- schimbă culoarea.



```
name "int10h"
JMP Start
Charr DB ?
Start:
org 100h
    MOV AL, 03h
    MOV AH, 00h ; Setare regim Sistem Video
int 10h
    MOV DH, 00h ; Row
    MOV DL, 00h ; Col
    MOV BH, 0
    MOV AH, 2 ; Pozitia cursorului
int 10h
    MOV BL, 0F0h ; Atributul
    MOV BP, 020h
Label:
    MOV AH, 00h
int 16h ; Input KBD
    MOV Charr, AL
    CMP AL, "o" ; In Sus
    JNZ Sus
    DEC DH
    JMP Label
Sus:
    MOV AL, Charr
    CMP AL, "m" ; In jos
    JNZ Jos
    INC DH
    JMP Label
Jos:
    MOV AL, Charr
    CMP AL, "k" ; In stinga
    JNZ L
    DEC DL
    JMP Label
L:
    MOV AL, Charr
    CMP AL, "I" ; In dreapta
    JNZ R
    INC DL
    JMP Label
R:
    MOV AL, Charr
    CMP AL, "c" ; Color
    JNZ C
    INC DL
    JMP Label
C:
    MOV BH, 0
    MOV AH, 2
int 10h
    MOV AL, Charr
    MOV AH, 09h
    MOV BH, 0
    MOV CX, 1
int 10h
    INC DL
    INC DH
    INC BL
    DEC BP
    JNZ Label
RET
```

The image shows a software emulator interface. On the left, a 'registers' window displays the state of various CPU registers. In the center, a window titled 'emulator screen (80x25 chars)' shows a black background with a diagonal line of multi-colored characters (a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z) starting from the top-left corner. Below the screen are buttons for 'clear screen' and 'change font', along with a '0/16' indicator. On the right, a window titled 'original source code' displays assembly code. Line 37, 'MOV Charr, AL', is highlighted in yellow. At the bottom, a navigation bar contains buttons for 'screen', 'source', 'reset', 'aux', 'vars', 'debug', 'stack', and 'flags'.

Fig. 81.1

Fig. 81.2

```

36 int 16h      ; Input KBD
37 MOV Charr, AL
38 CMP AL, "o"  ; In Sus
39 JNZ Sus
40 DEC DH
41 JMP Label
42 Sus:
43 MOV AL, Charr
44 CMP AL, "h"  ; In jos
45 JNZ Jos
46 INC DH
47 JMP Label
48 Jos:
49 MOV AL, Charr
50 CMP AL, "k"  ; In stinga
51 JNZ L
52 DEC DL
53 JMP Label
54 L:
55 MOV AL, Charr
56 CMP AL, "l"  ; In dreapta
57 JNZ R
58 INC DL
59 JMP Label
60 R:
61 MOV AL, Charr
62 CMP AL, "c"  ; Color
63 JNZ C
64 INC DL
65 JMP Label
66
67 C:

```

Să se elaboreze programul care afișează informația in Regim Grafic pe ecranul videomonitorului utilizând Adresarea directă a mamoriei RAM Video:

- RAM Video = A0000h
- Standardul de funcționare a Sistemului Video: VGA, 1 Pixel = 8 biți;
- Coordonatele Pixel-ului sunt obținute cu ajutorul INT 33h, funcția 0003h;
- La tastarea butonului din stânga are loc desenarea Pixelu-lui;
- La tastarea butonului din dreapta are loc ștergerea Pixel-ului.



name "int10h_13"	MOV X, CX	ADD BX, AX
org 100h	MOV Y, DX	MOV AL, 0Fh ; Culoare Pixel Alba
JMP Start	MOV Button, BX	MOV ES:[BX], AL ; Write RAM Video
X DW ?	CMP BX, 0 ; Comparare cu 0	DEC BP
Y DW ?	JZ Label	JNZ Label
Button DW ?	MOV BX, Button	Stergere:
Start:	CMP BX, 1 ; Desenare	MOV BX, 00h
MOV AH, 0h	JZ Desenare	ADD BX, X
MOV AL, 13h	MOV BX, Button	MOV AX, 320
int 10h	CMP BX, 2 ; Stergere	MUL Y
MOV BX, 0A000h ; RAM Video Regim Grafic	JZ Stergere	ADD BX, AX
MOV ES, BX	Desenare: ; Desenare Pixel	MOV AL, 00h ; Culoare Pixel Negru
MOV BP, 200h ; Iteratii	MOV BX, 00h	MOV ES:[BX], AL ; Write RAM Video
Label:	ADD BX, X	DEC BP
MOV AX, 3	MOV AX, 320	JNZ Label
int 33h ; Read Mouse	MUL Y	RET

The image shows a screenshot of an 8086 emulator interface. On the left, there is a 'registers' window displaying the state of various registers:

Register	H	L
AX	00	03
BX	00	00
CX	02	7C
DX	00	A3
CS	0700	
IP	0116	
SS	0700	
SP	FFFE	
BP	0190	
SI	0000	
DI	0000	
DS	0700	
ES	A000	

In the center, a window titled 'emulator screen (40x25 char...)' displays a black screen with white stars and the text 'Fig. 84.1' in a white box. Below the screen are buttons for 'clear screen' and 'change font', and a dropdown menu showing '0/16'.

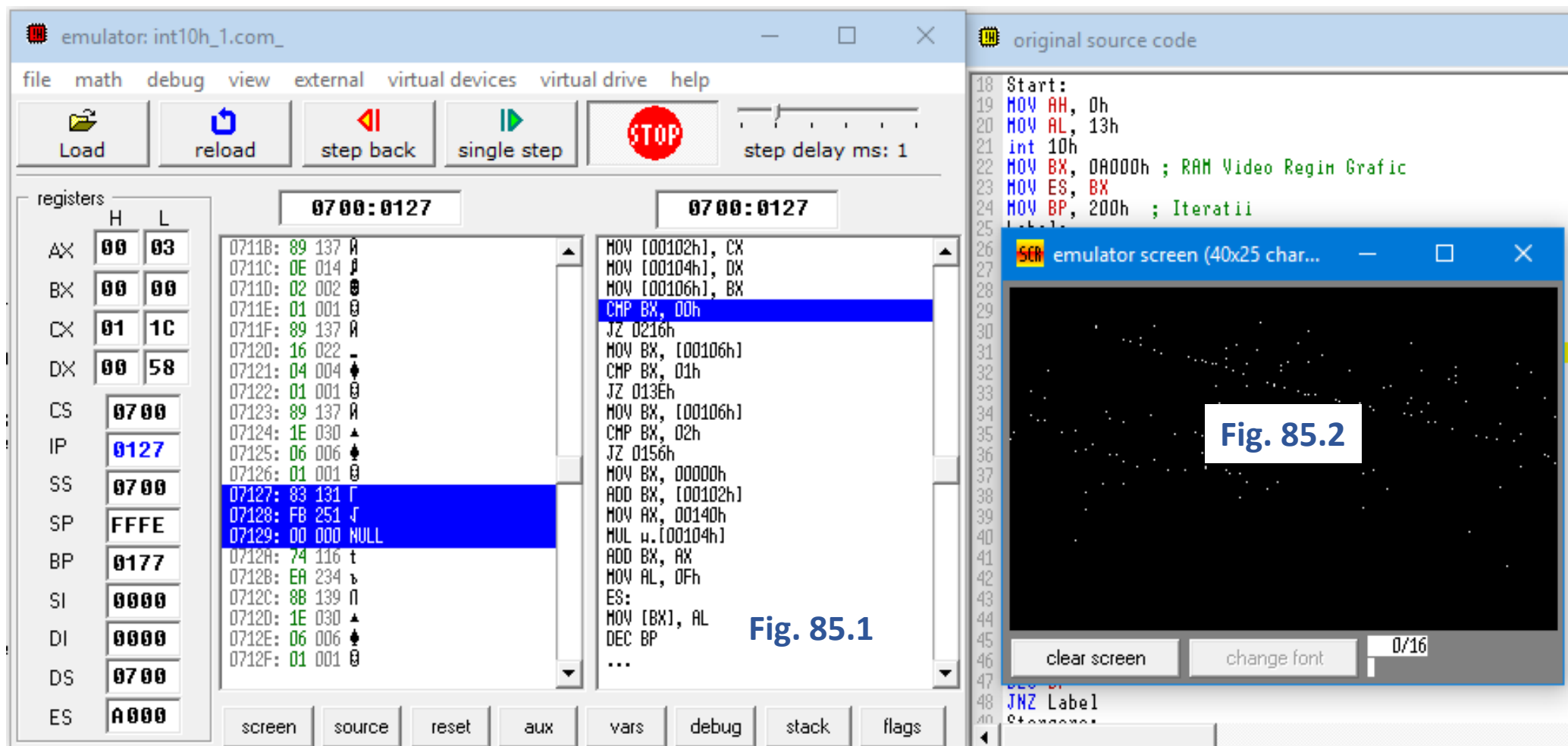
On the right, a window titled 'original source code' displays assembly code. Line 26, 'MOV AX, 3', is highlighted in yellow. The code includes comments in Romanian for video RAM initialization and pixel drawing logic.

```

18 Start:
19 MOV AH, 0h
20 MOV AL, 13h
21 int 10h
22 MOV BX, 0A000h ; RAM Video Regim Grafic
23 MOV ES, BX
24 MOV BP, 200h ; Iteratii
25 Label:
26 MOV AX, 3
27 int 33h ; Read Mouse
28 MOV X, CX
29 MOV Y, DX
30 MOV Button, BX
31 CMP BX, 0 ; Comparare cu 0
32 JZ Label
33 MOV BX, Button
34 CMP BX, 1 ; Desenare
35 JZ Desenare
36 MOV BX, Button
37 CMP BX, 2 ; Stergere
38 JZ Stergere
39 Desenare: ; Desenare Pixel
40 MOV BX, 00h
41 ADD BX, X
42 MOV AX, 320
43 MUL Y
44 ADD BX, AX
45 MOV AL, 0Fh ; Culoarea Pixel Alba
46 MOV ES:[BX], AL ; Write RAM Video
47 DEC BP
48 JNZ Label
49 Stop:

```

The label 'Fig. 84.2' is placed to the right of the code window.



The image shows a screenshot of an 8086 emulator interface. The main window is titled "emulator: int10h_1.com_". It features a menu bar (file, math, debug, view, external, virtual devices, virtual drive, help) and a toolbar with buttons for Load, reload, step back, single step, and a STOP button. Below the toolbar is a "registers" section with a table of registers and their values:

Register	H	L
AX	00	03
BX	00	00
CX	01	1C
DX	00	58
CS	0700	
IP	0127	
SS	0700	
SP	FFFE	
BP	0177	
SI	0000	
DI	0000	
DS	0700	
ES	A000	

To the right of the registers is a memory dump showing addresses from 07118 to 0712F. The instruction at 07127 is highlighted in blue: "MOV [00102h], CX". Below the memory dump is a disassembled instruction list, with "MOV [00102h], CX" also highlighted. At the bottom of the emulator window are buttons for screen, source, reset, aux, vars, debug, stack, and flags.

Overlaid on the right side is a window titled "original source code" showing assembly code:

```

18 Start:
19 MOV AH, 0h
20 MOV AL, 13h
21 int 10h
22 MOV BX, 0A000h ; RAM Video Regim Grafic
23 MOV ES, BX
24 MOV BP, 200h ; Iteratii
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48 JNZ Label

```

Overlaid on the bottom right is a window titled "emulator screen (40x25 char...)" showing a black screen with white speckles, representing a video mode. Below the screen are buttons for "clear screen" and "change font", and a font size indicator set to "0/16".

Fig. 85.1

Fig. 85.2

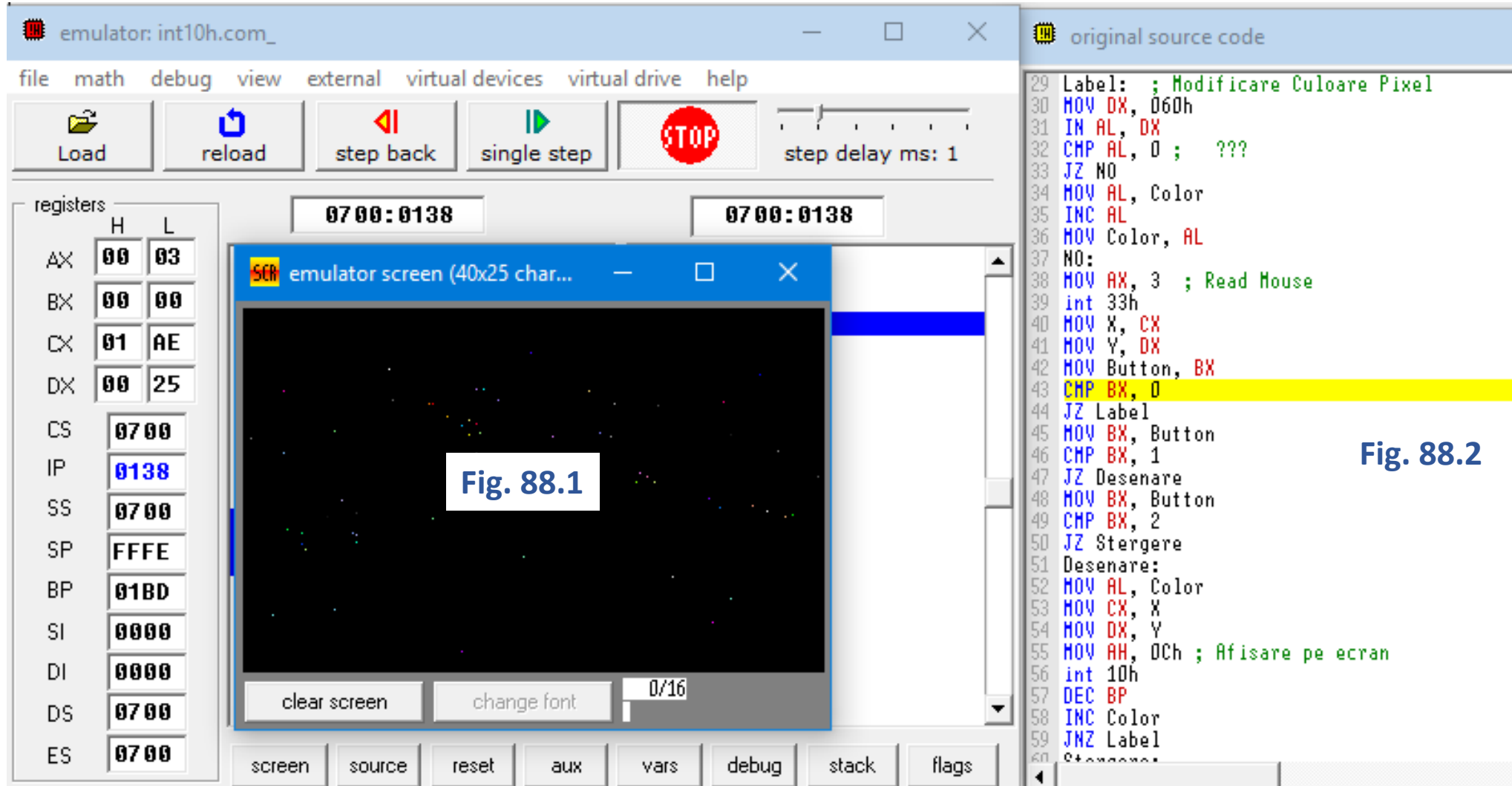
Să se elaboreze programul care afișează informația in Regim Grafic pe ecranul videomonitorului utilizând Funcțiile întreruperii BIOS INT 10h:

- RAM Video = A0000h
- Standardul de funcționare a Sistemului Video: VGA, 1 Pixel = 8 biți;
- Coordonatele Pixel-ului sunt obținute cu ajutorul INT 33h, funcția 0003h;
- La tastarea butonului din stânga are loc desenarea Pixelu-lui;
- La tastarea butonului din dreapta are loc ștergerea Pixel-ului.

```
name "int10h"  
org 100h  
JMP Start  
X DW ?  
Y DW ?  
Button DW ?  
Color DB ?  
Start:  
MOV AH, 0h  
MOV AL, 13h ; Setare Regim Grafic  
int 10h  
    MOV BP, 200h ; Iteratii  
    MOV AL, 0Fh ; Culoarea Pixel  
    MOV Color, AL  
Label: ; Modificare Culoare Pixel  
    MOV DX, 060h  
    IN AL, DX  
    CMP AL, 0 ; ???
```

```
JZ NO  
MOV AL, Color  
INC AL  
MOV Color, AL  
NO:  
MOV AX, 3 ; Read Mouse  
int 33h  
MOV X, CX  
MOV Y, DX  
MOV Button, BX  
CMP BX, 0  
JZ Label  
MOV BX, Button  
CMP BX, 1  
JZ Desenare  
MOV BX, Button  
CMP BX, 2  
JZ Stergere
```

```
Desenare:  
MOV AL, Color  
MOV CX, X  
MOV DX, Y  
MOV AH, 0Ch ; Afisare pe ecran  
int 10h  
INC Color  
DEC BP  
JNZ Label  
Stergere:  
MOV AL, 00h  
MOV CX, X  
MOV DX, Y  
MOV AH, 0Ch ; Sterge Pixel  
int 10h  
DEC BP  
JNZ Label  
RET
```



The image shows a screenshot of an x86 emulator interface. The main window is titled "emulator: int10h.com_". It features a menu bar (file, math, debug, view, external, virtual devices, virtual drive, help) and a toolbar with buttons for Load, reload, step back, single step, and a STOP button. Below the toolbar is a registers window showing the state of various registers:

registers	H	L
AX	00	03
BX	00	00
CX	01	AE
DX	00	25
CS	0700	
IP	0138	
SS	0700	
SP	FFFE	
BP	01BD	
SI	0000	
DI	0000	
DS	0700	
ES	0700	

The central area displays a window titled "emulator screen (40x25 char...)" which shows a black screen with scattered colored pixels (red, green, blue, yellow). A white box with the text "Fig. 88.1" is overlaid on the screen. Below the screen are buttons for "clear screen" and "change font", along with a font size indicator "0/16".

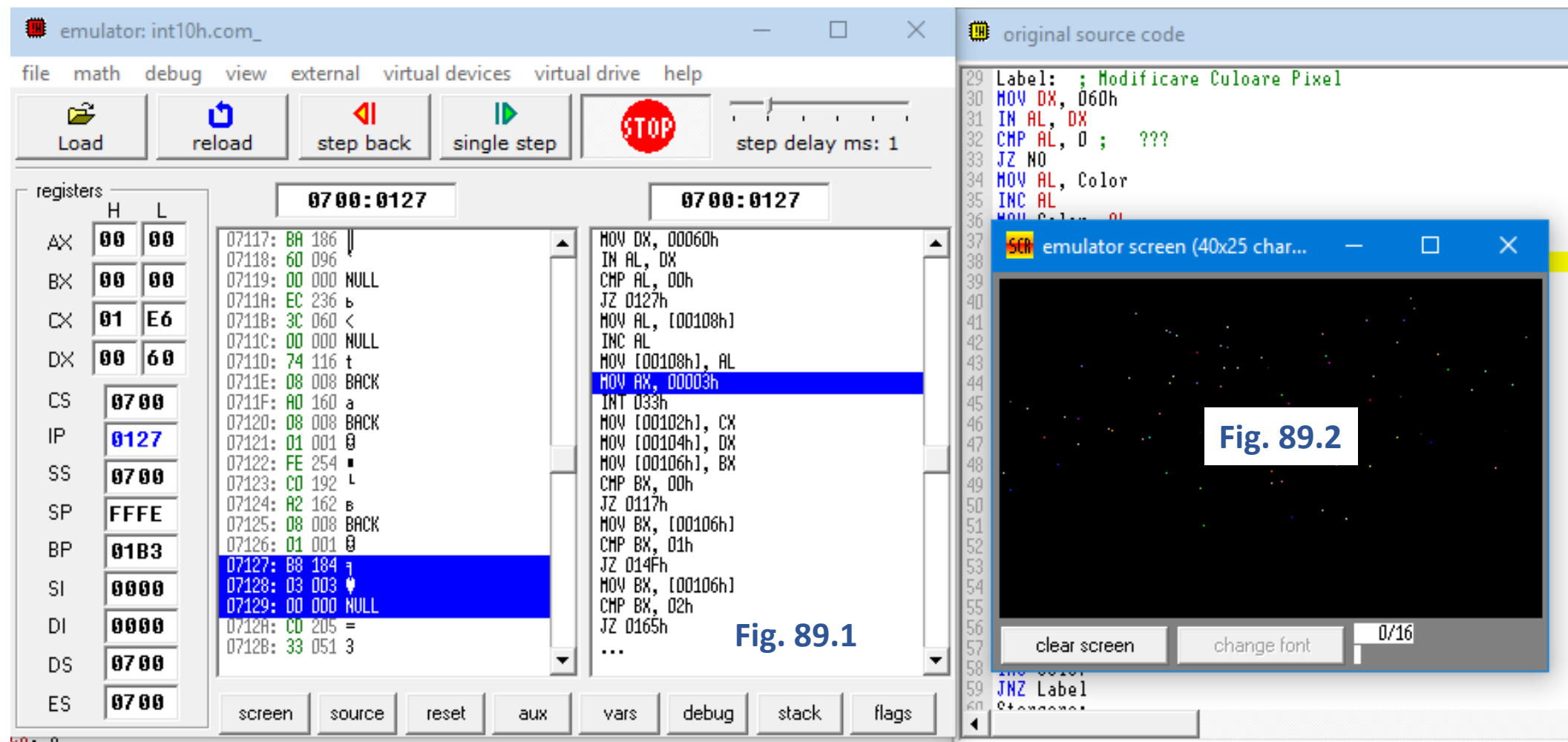
On the right side, there is a window titled "original source code" showing assembly instructions:

```

29 Label: ; Modificare Culoare Pixel
30 MOV DX, 060h
31 IN AL, DX
32 CMP AL, 0 ; ???
33 JZ NO
34 MOV AL, Color
35 INC AL
36 MOV Color, AL
37 NO:
38 MOV AX, 3 ; Read Mouse
39 int 33h
40 MOV X, CX
41 MOV Y, DX
42 MOV Button, BX
43 CMP BX, 0
44 JZ Label
45 MOV BX, Button
46 CMP BX, 1
47 JZ Desenare
48 MOV BX, Button
49 CMP BX, 2
50 JZ Stergere
51 Desenare:
52 MOV AL, Color
53 MOV CX, X
54 MOV DX, Y
55 MOV AH, 0Ch ; Afisare pe ecran
56 int 10h
57 DEC BP
58 INC Color
59 JNZ Label
60 Stergere:

```

The instruction `CMP BX, 0` on line 43 is highlighted in yellow. A white box with the text "Fig. 88.2" is overlaid on the code window.



The image shows a screenshot of an x86 emulator interface. The main window is titled "emulator: int10h.com_". It features a menu bar (file, math, debug, view, external, virtual devices, virtual drive, help) and a toolbar with buttons for Load, reload, step back, single step, and a STOP button. Below the toolbar, there are registers (AX, BX, CX, DX, CS, IP, SS, SP, BP, SI, DI, DS, ES) and a memory dump window showing addresses and hex values. The assembly code window displays the following code:

```

29 Label: ; Modificare Culoare Pixel
30 MOV DX, 060h
31 IN AL, DX
32 CMP AL, 0 ; ???
33 JZ NO
34 MOV AL, Color
35 INC AL
36 MOV Color, AL
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59 JNZ Label
60

```

The graphical output window, titled "emulator screen (40x25 char...", shows a black screen with a few scattered colored pixels. A white box with the text "Fig. 89.2" is overlaid on the screen. The bottom of the emulator window has buttons for "screen", "source", "reset", "aux", "vars", "debug", "stack", and "flags".

Fig. 89.1

Fig. 89.2

Tema 4. Dispozitive pentru afișarea și imprimarea datelor:

- 1. Clasificarea dispozitivelor pentru afișarea și imprimarea datelor.**
- 2. Sistemul video.**
- 3. Videomonitoare.**
- 4. Interfața video.**
- 5. Videoterminale.**
- 6. Imprimanta.**
- 7. Plotter-ul.**

Tematica disciplinei Arhitectura Calculatoarelor:

Tema 1. Introducere. Bazele fundamentale ale Arhitecturii Calculatoarelor;

Tema 2. Microprocesoare și Microcontrolere. Limbajul de programare Assembler;

Tema 3. Dispozitive pentru achiziția datelor;

Tema 4. Dispozitive pentru afișarea și imprimarea datelor;

Tema 5. Dispozitive pentru stocarea datelor.

Mulțumesc pentru atenție!