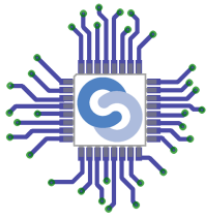


Calculatoare Numerice (2)

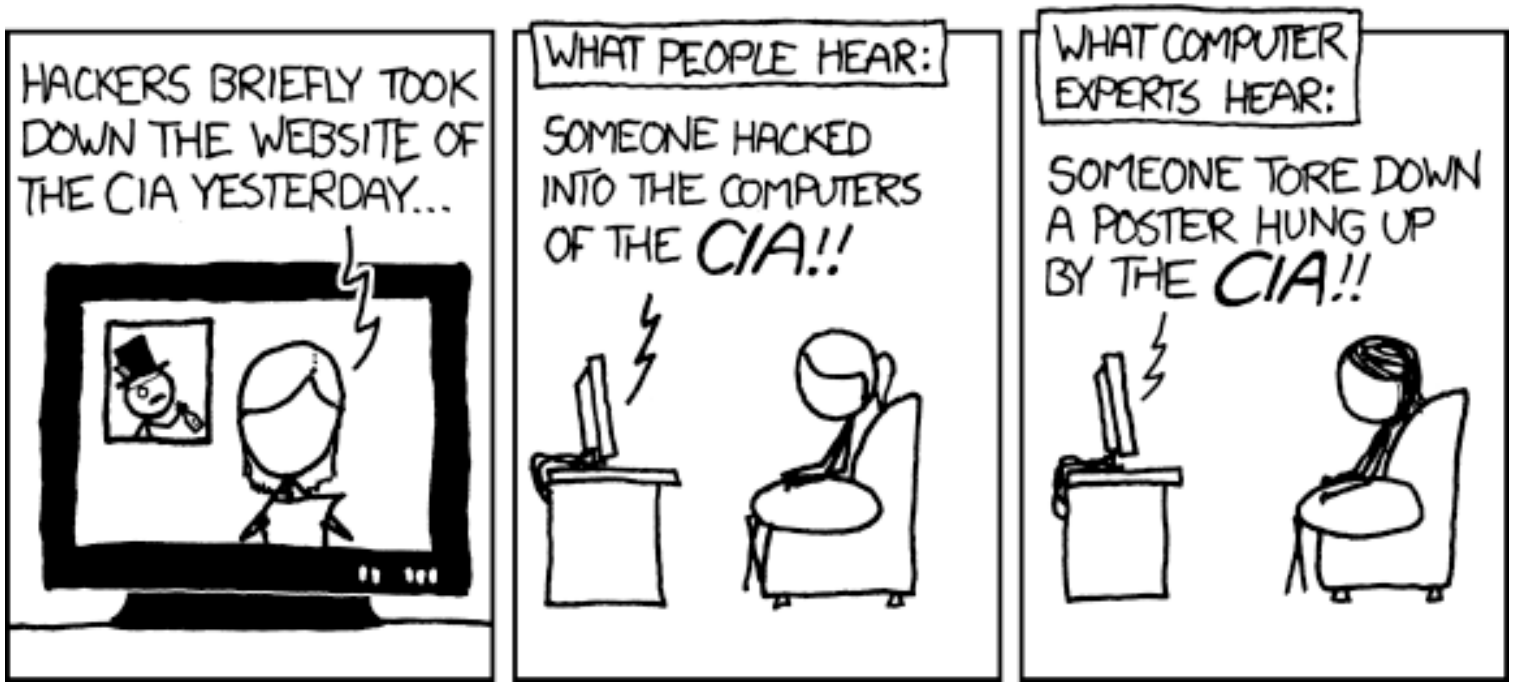
– Cursul 9 –

Interfațarea cu dispozitivele I/O

Dan Tudose

Facultatea de Automatică și Calculatoare
Universitatea Politehnica București

Comic of the day



http://xkcd.com/932/

Dispozitive Input/Output

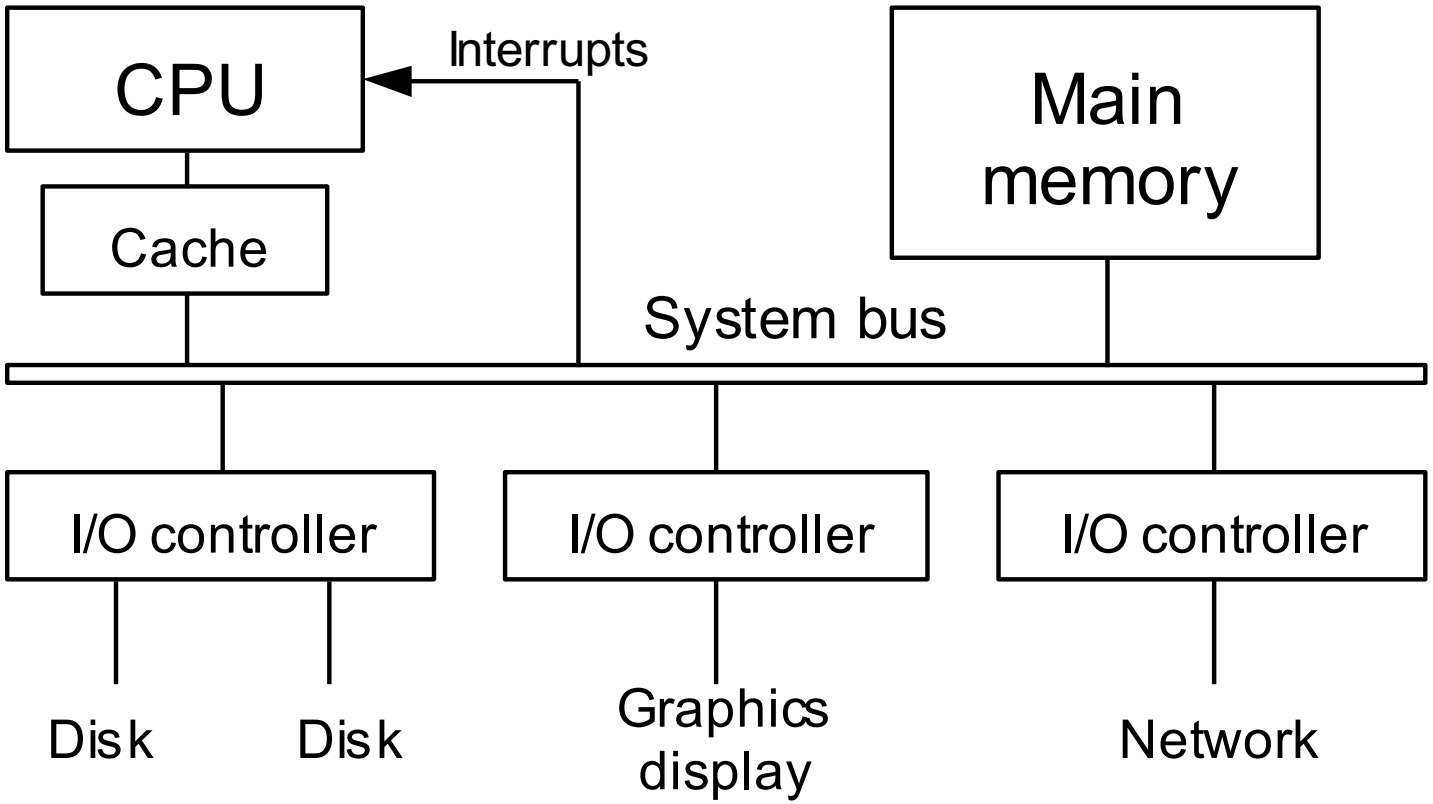
Cuprins
1. Dispozitive și controllere Input/Output
2. Keyboard and Mouse
3. Unități de display
4. Dispozitive de imprimare
5. Alte dispozitive Input/Output
6. Conectarea în rețea a dispozitivelor Input/Output


intel inside
HEWLETT
PACKARD

Dispozitive și controllere de intrare/ieșire

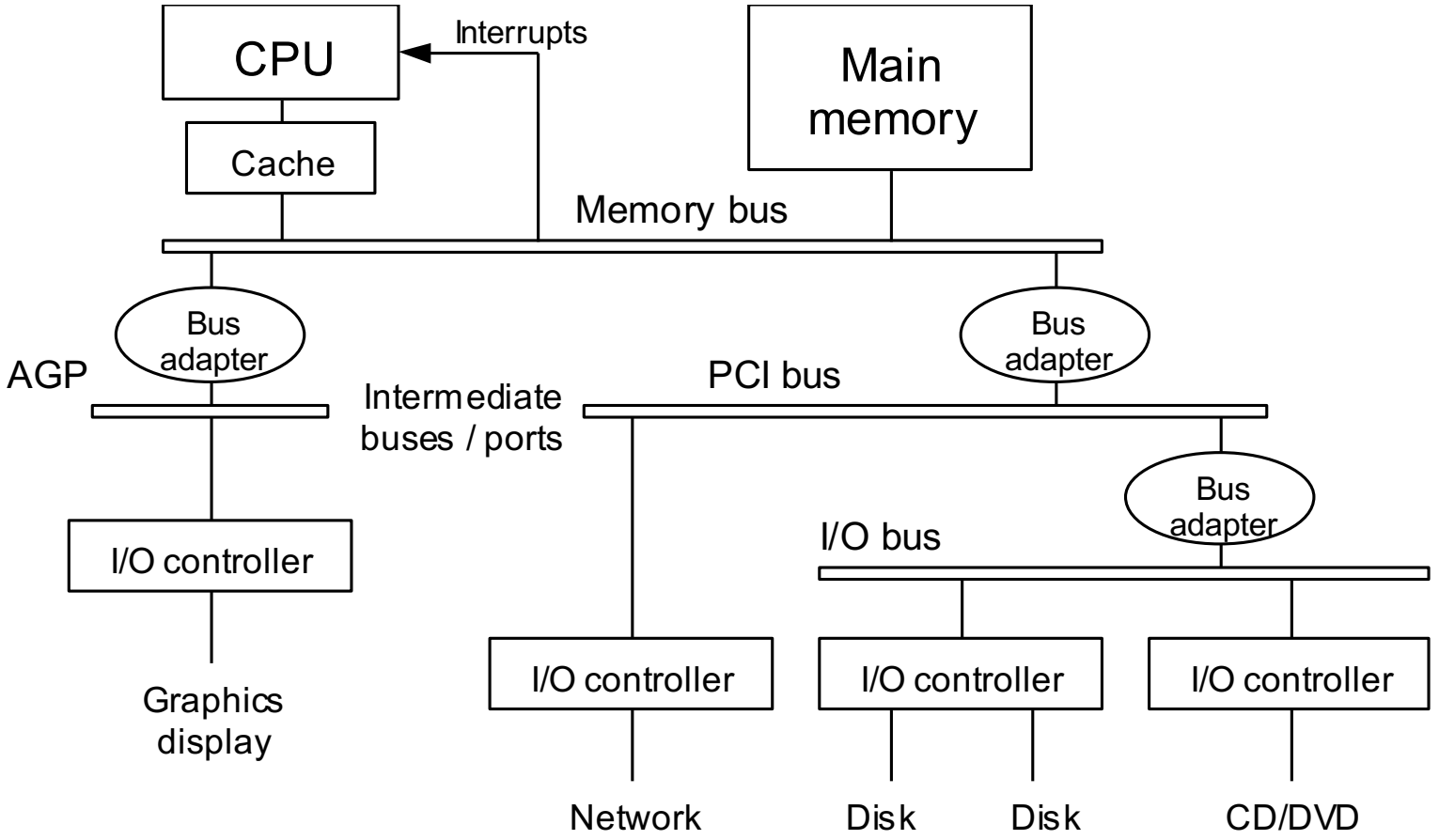
Input type	Prime examples	Other examples	Data rate (b/s)	Main uses
Symbol	Keyboard, keypad	Music note, OCR	10s	Ubiquitous
Position	Mouse, touchpad	Stick, wheel, glove	100s	Ubiquitous
Identity	Barcode reader	Badge, fingerprint	100s	Sales, security
Sensory	Touch, motion, light	Scent, brain signal	100s	Control, security
Audio	Microphone	Phone, radio, tape	1000s	Ubiquitous
Image	Scanner, camera	Graphic tablet	1000s-10^6s	Photos, publishing
Video	Camcorder, DVD	VCR, TV cable	1000s-10^9s	Entertainment
Output type	**Prime examples**	**Other examples**	**Data rate (b/s)**	**Main uses**
Symbol	LCD line segments	LED, status light	10s	Ubiquitous
Position	Stepper motor	Robotic motion	100s	Ubiquitous
Warning	Buzzer, bell, siren	Flashing light	A few	Safety, security
Sensory	Braille text	Scent, brain stimulus	100s	Personal assistance
Audio	Speaker, audiotape	Voice synthesizer	1000s	Ubiquitous
Image	Monitor, printer	Plotter, microfilm	1000s	Ubiquitous
Video	Monitor, TV screen	Film/video recorder	1000s-10^9s	Entertainment
Two-way I/O	**Prime examples**	**Other examples**	**Data rate (b/s)**	**Main uses**
Mass storage	Hard/floppy disk	CD, tape, archive	10^6s	Ubiquitous
Network	Modem, fax, LAN	Cable, DSL, ATM	1000s-10^9s	Ubiquitous

Diagrama unui sistem cu Input/Output



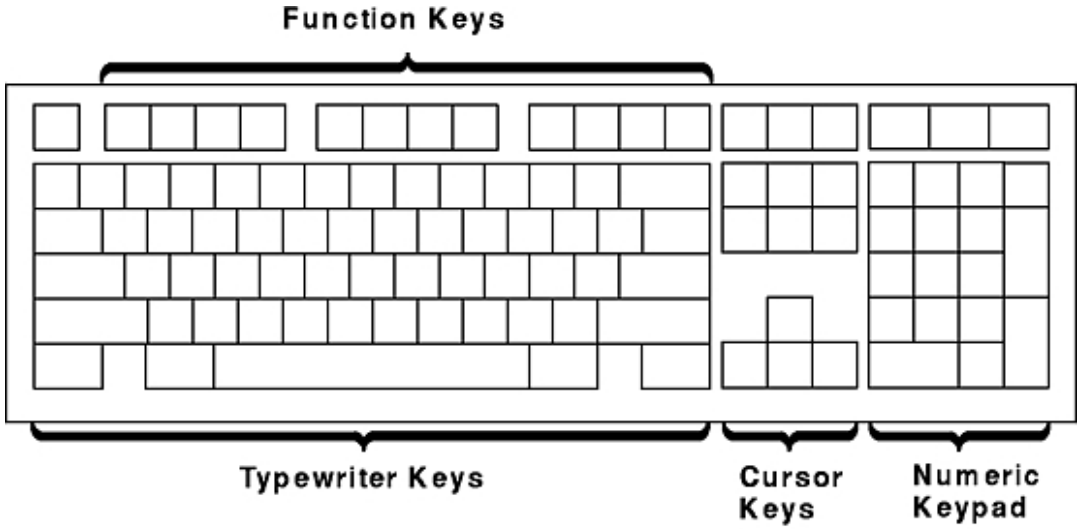
Sistem de input/output printr-un singur bus de date comun.

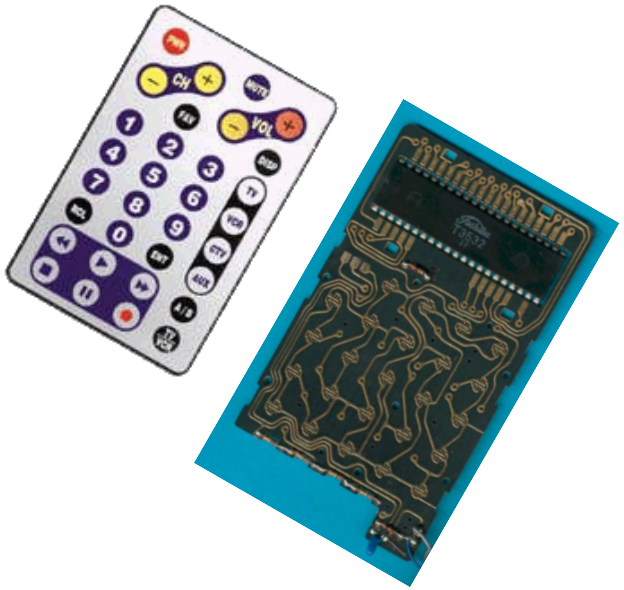
Organizarea I/O pentru performanță mărită



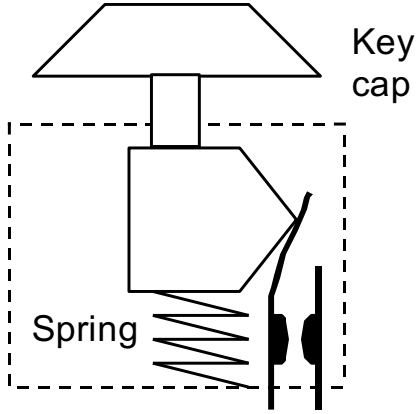
Input/output prin bus-uri de date intermediare sau dedicate

Tastatură și mouse

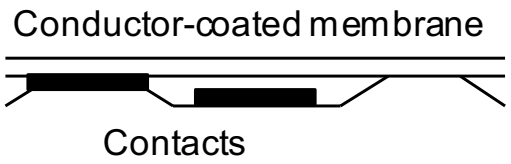




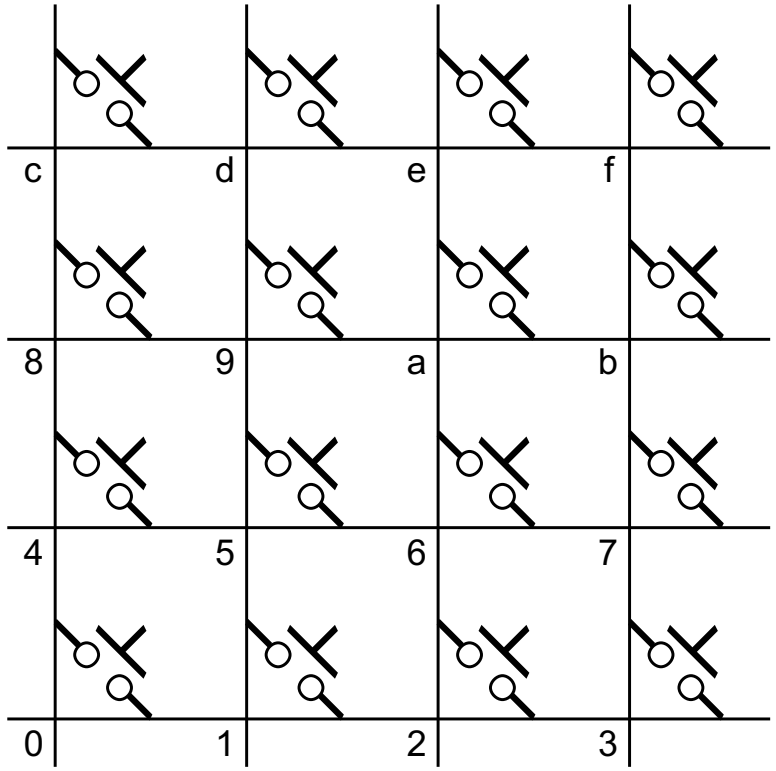
Butoanele tastaturii și codificarea lor



(a) Mechanical switch with a plunger



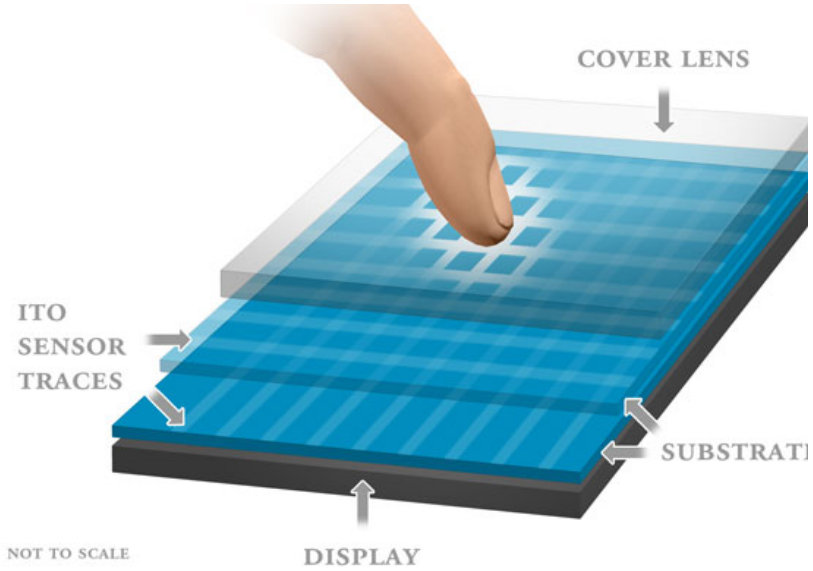
(b) Membrane switch



(c) Logical arrangement of keys

Două tipuri de butoane și schema logică a conectării lor într-o tastatură hexazecimală

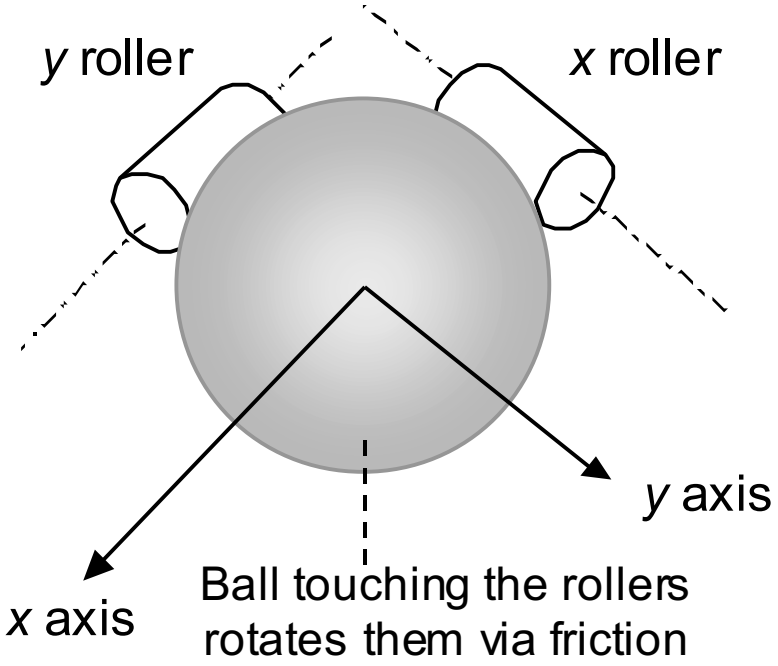
Tastatură cu proiecție sau capacitivă



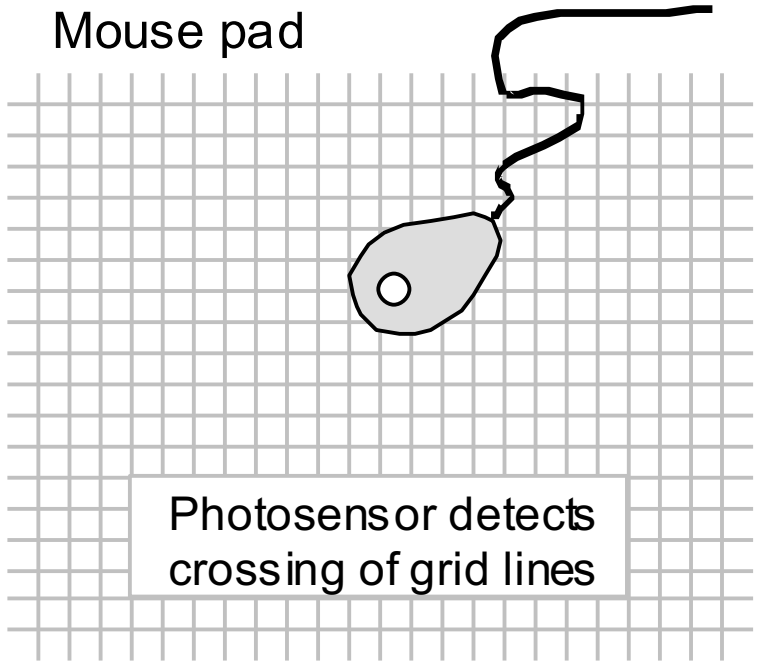
2. A capacitive-type touchscreen (self or absolute capacitive type)—a laminate of multiple substrates (either PET, or glass as shown) or, in some cases a single substrate—contains a matrix of clear conductive electrodes (made by patterning a clear conductor like ITO). This substrate(s) is then laminated to a top cover lens (either glass or PET). When a user's finger touches the cover lens, it increases the measured capacitance of the electrodes closest to the finger. This change in capacitance is used to compute the location of the finger within the capacitive touchscreen.

Dispozitive pentru indicat

Cum funcționează un mouse



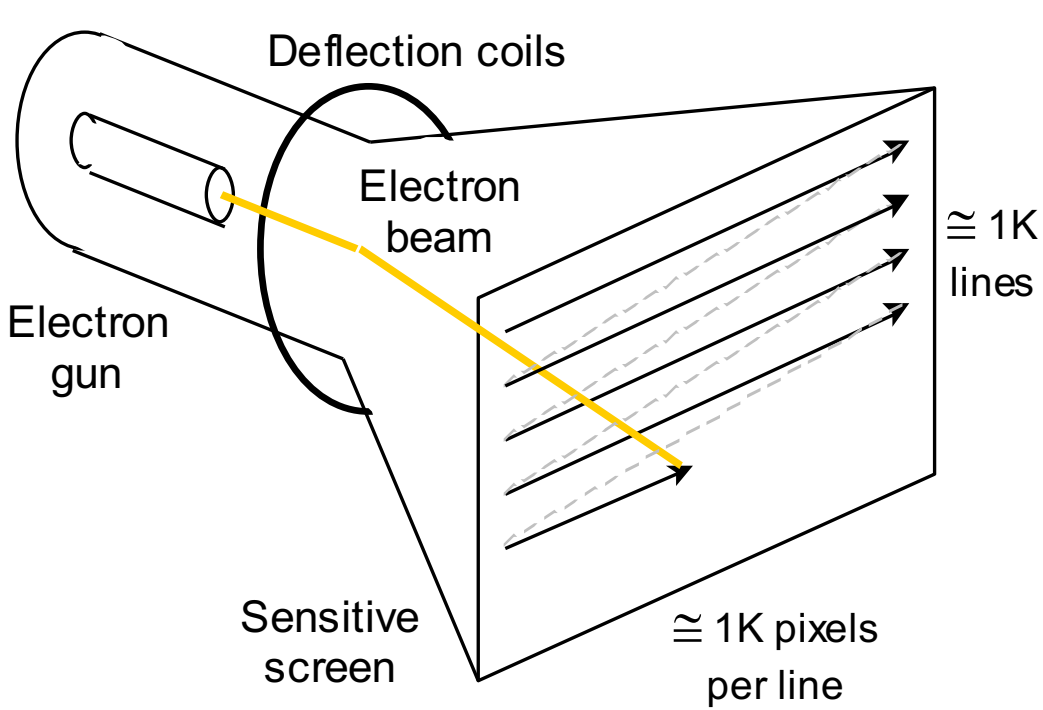
(a) Mechanical mouse



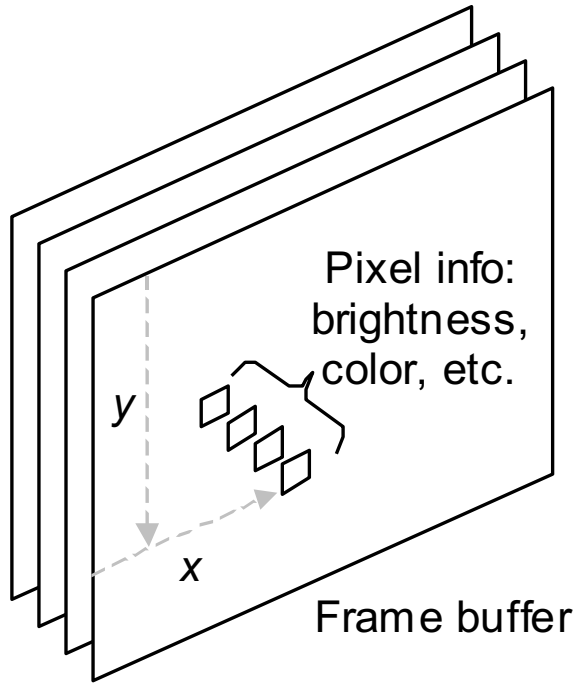
(b) Optical mouse

Mouse mecanic (cu bilă) și optic.

Unități de afișare



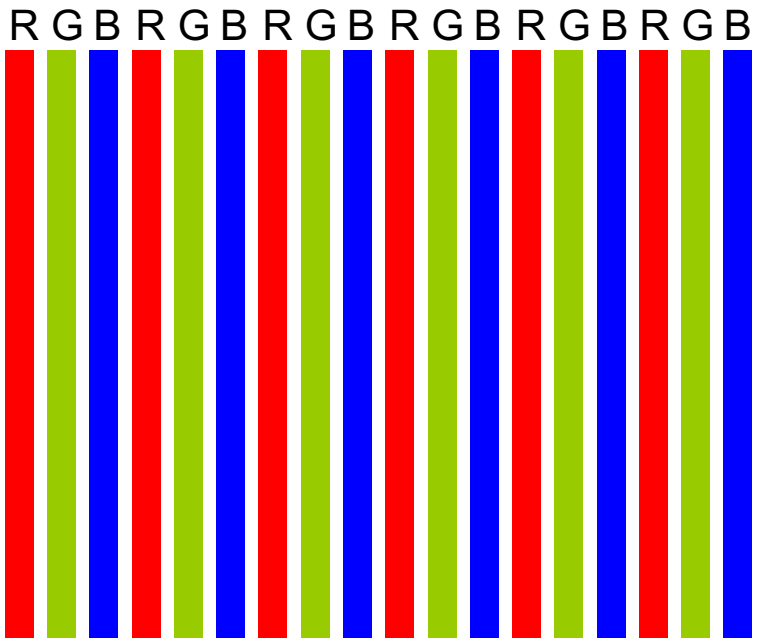
(a) Image formation on a CRT



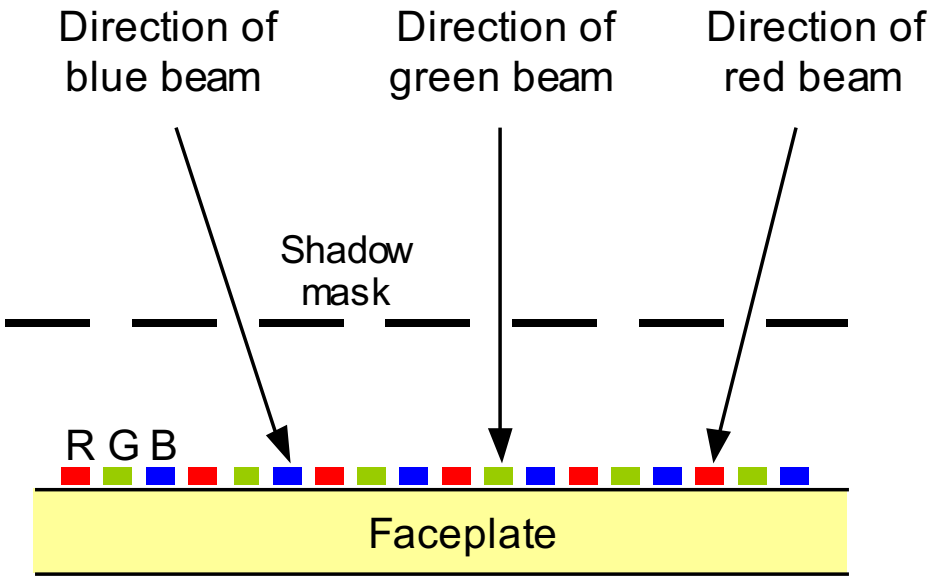
(b) Data defining the image

Display CRT și imaginea stocată în frame buffer.

Cum funcționează un display CRT



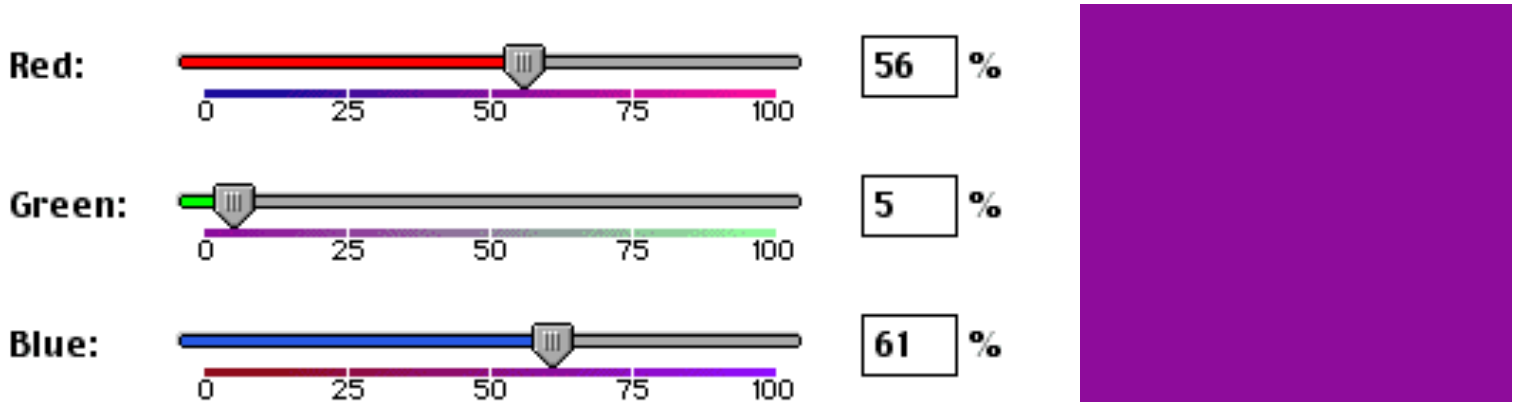
(a) The RGB color stripes



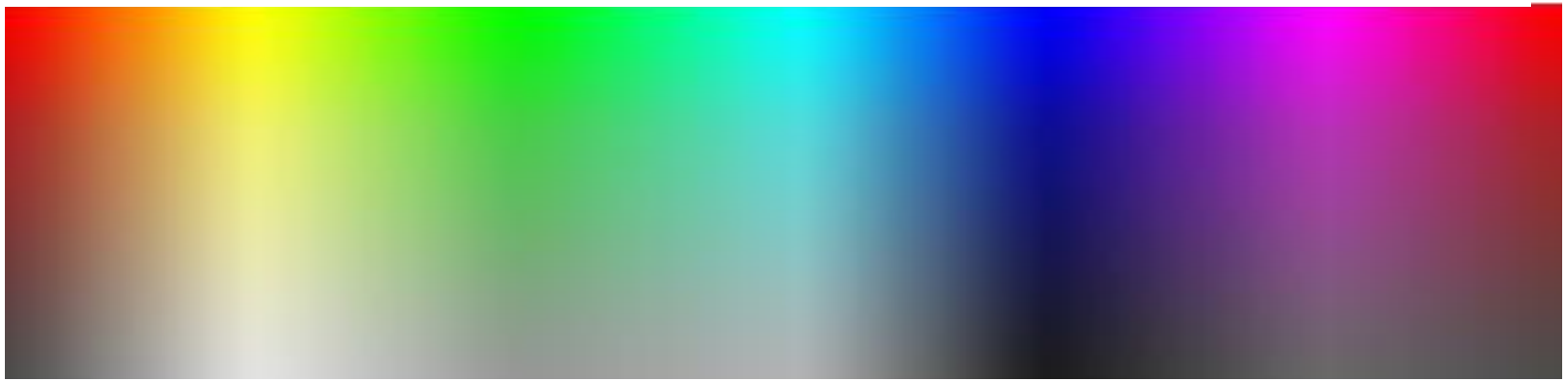
(b) Use of shadow mask

Schema e culori RGB pentru un display CRT modern.

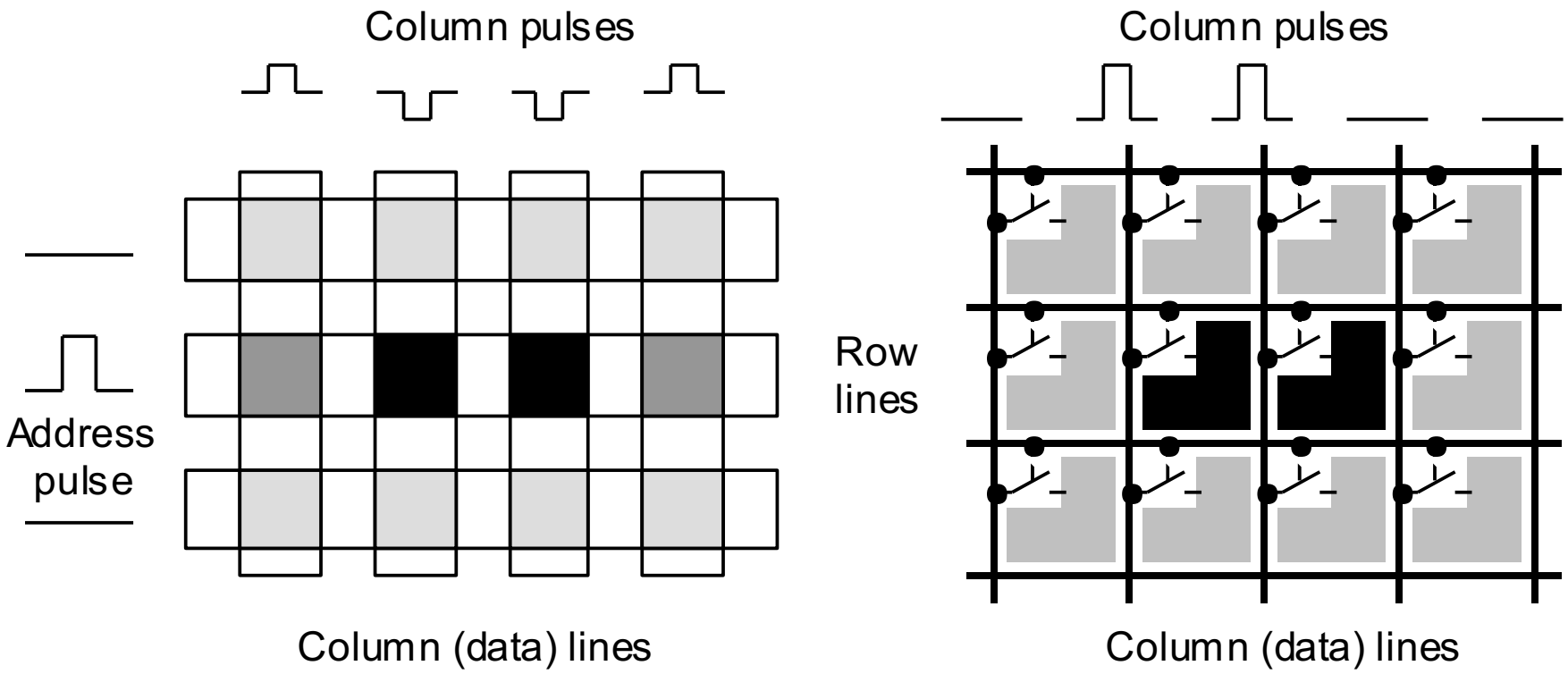
Codificarea culorilor în formatul RGB



Nuanța și saturația afectează aspectul unei culori (saturație maximă sus, saturație minimă la bază)

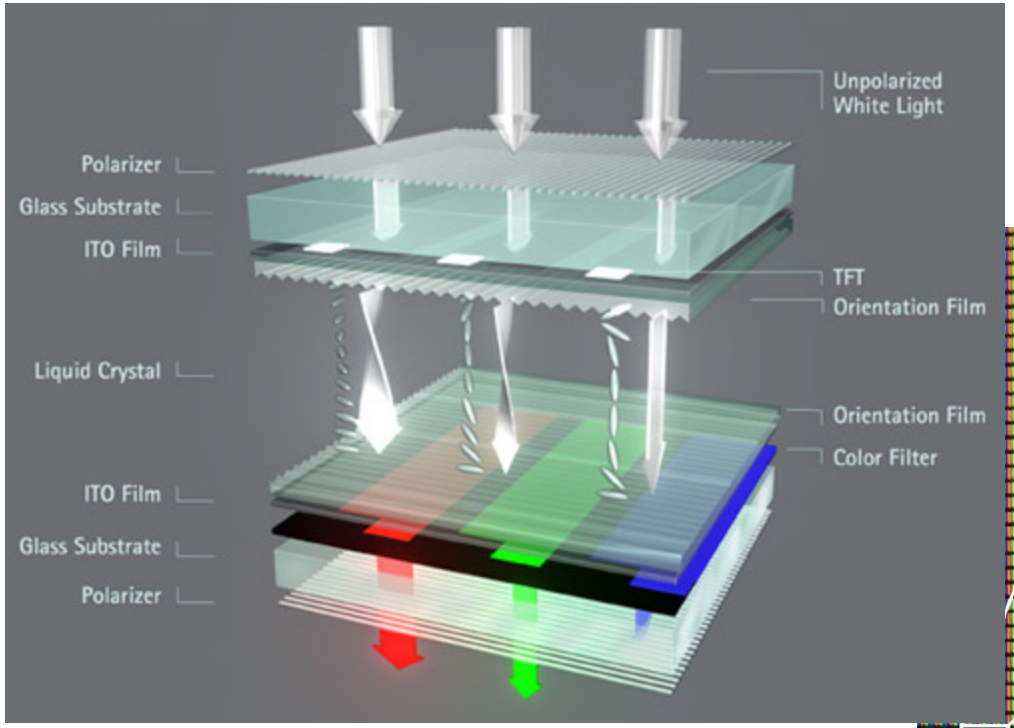


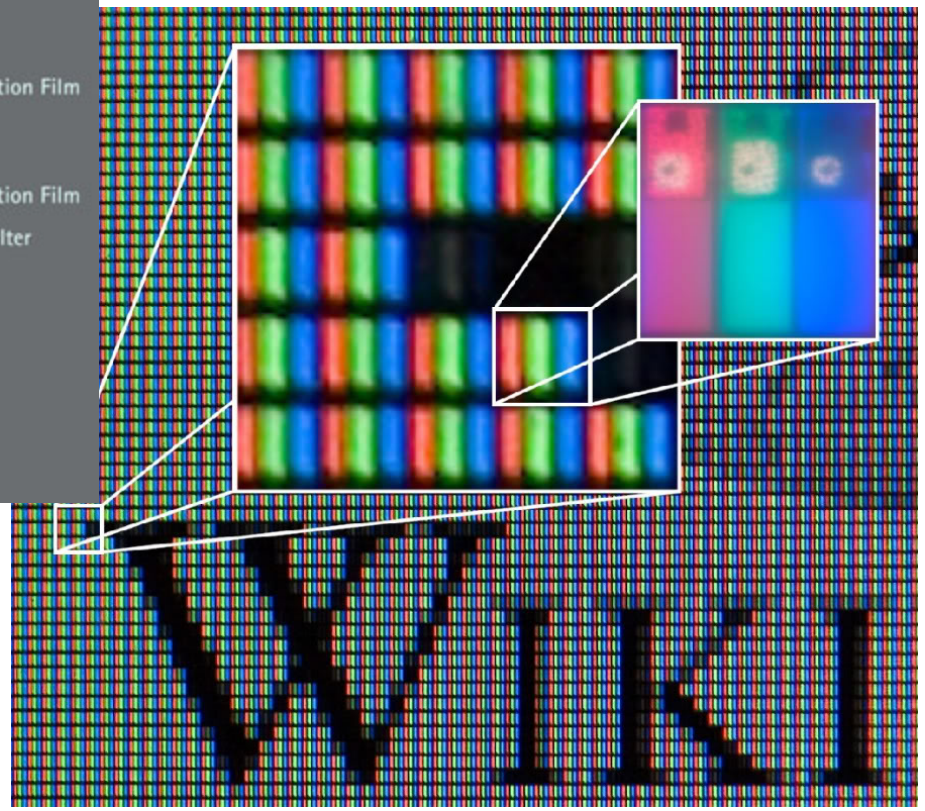
Display-uri LCD



Display-uri LCD active și pasive

Pixelii într-un ecran LCD modern



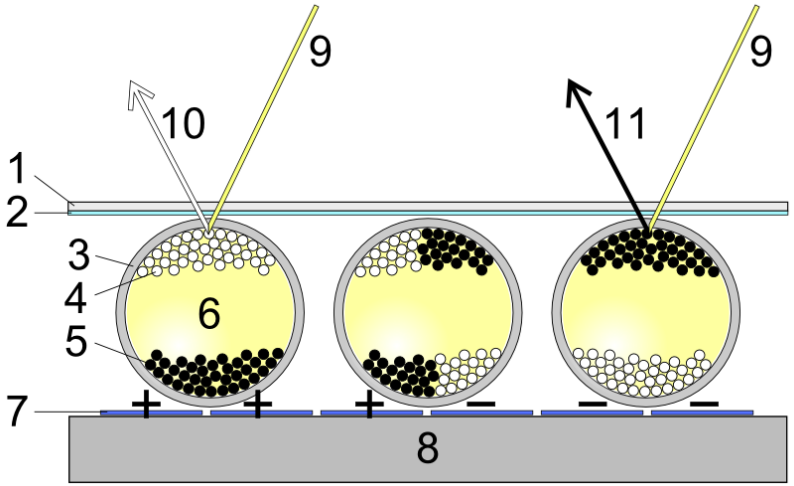


Unități de display flexibile

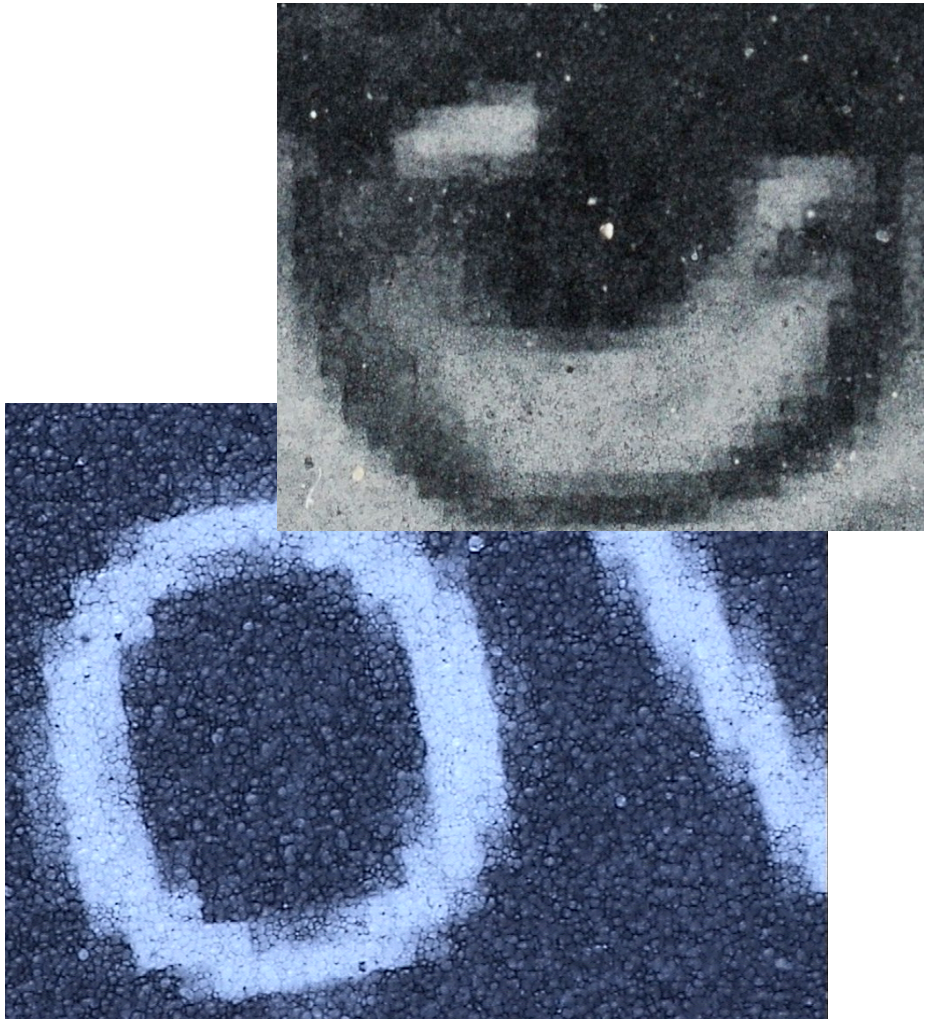
Afișaj subțire flexibil folosind e-ink

Display flexibil Sony organic light-emitting diode (OLED)

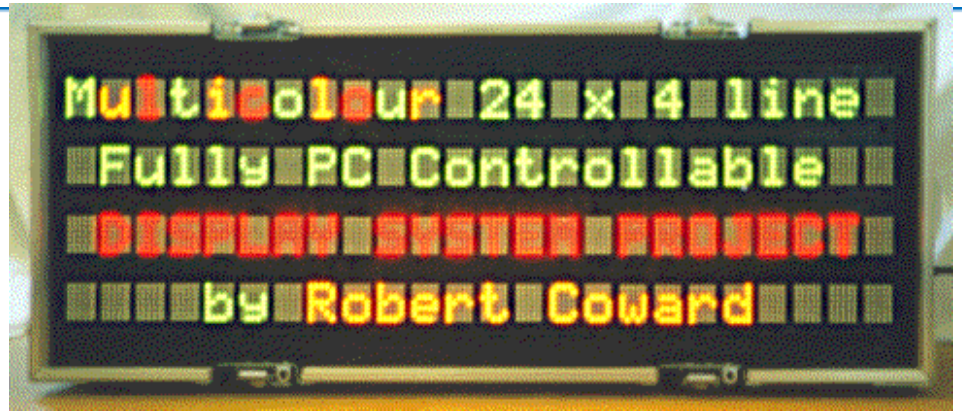
E-Ink

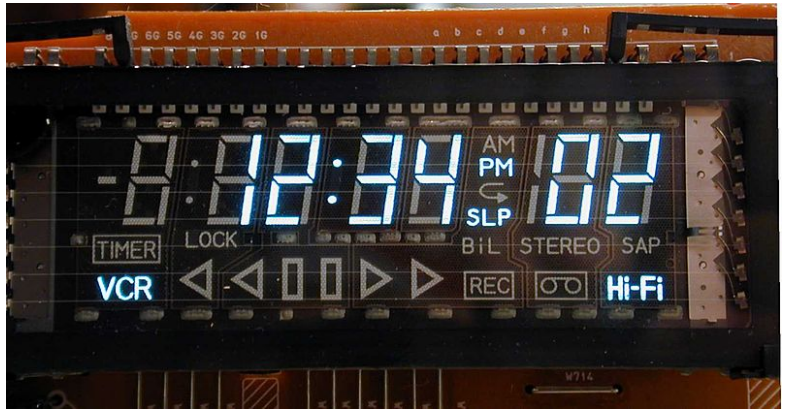


1. upper layer
2. transparent electrode layer
3. transparent micro-capsules
4. positive charged white pigments
5. negative charged black pigments
6. transparent oil
7. electrode pixel layer
8. bottom supporting layer
9. light
10. white
11. black

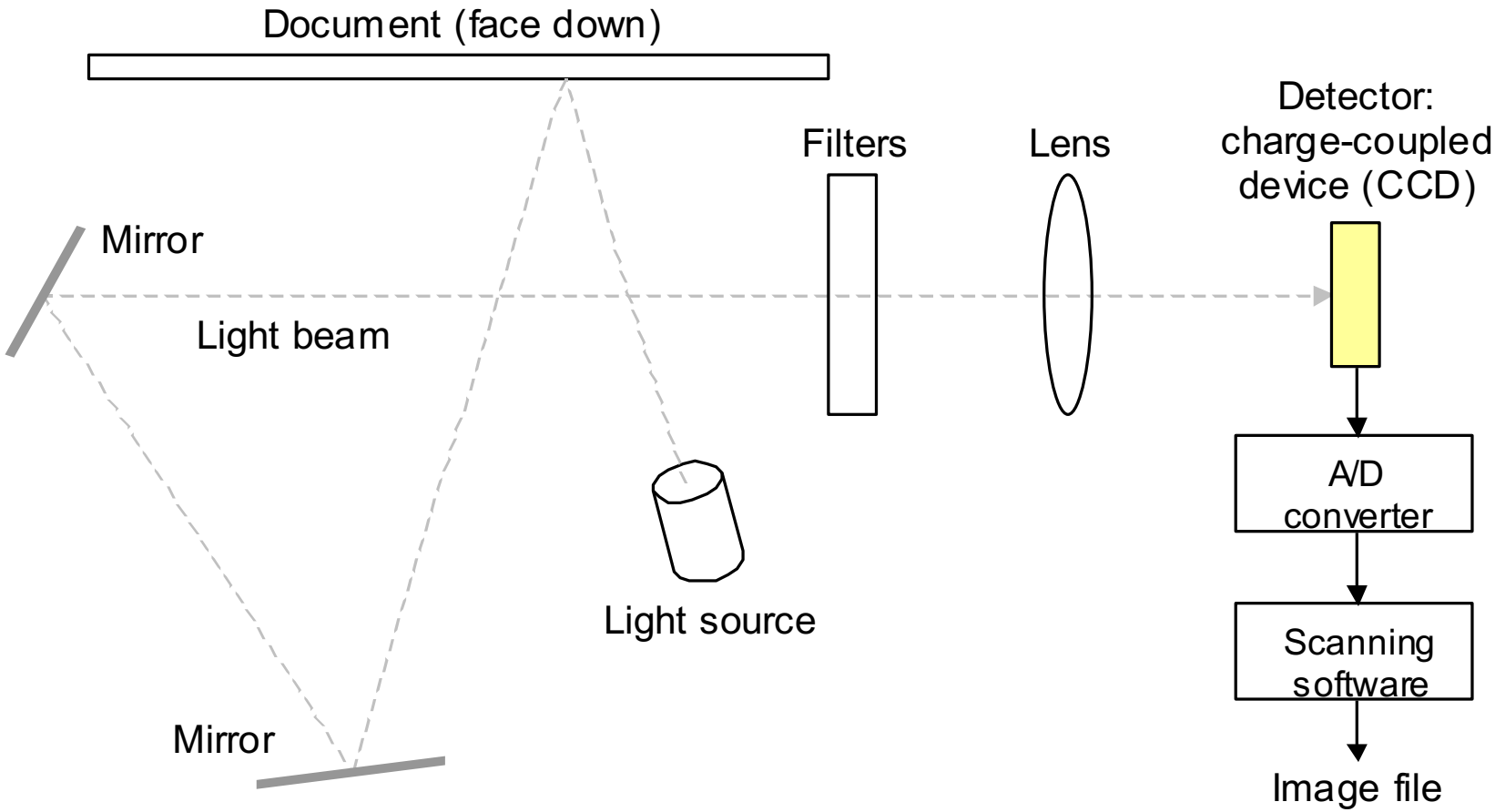


Alte tehnologii pentru display-uri



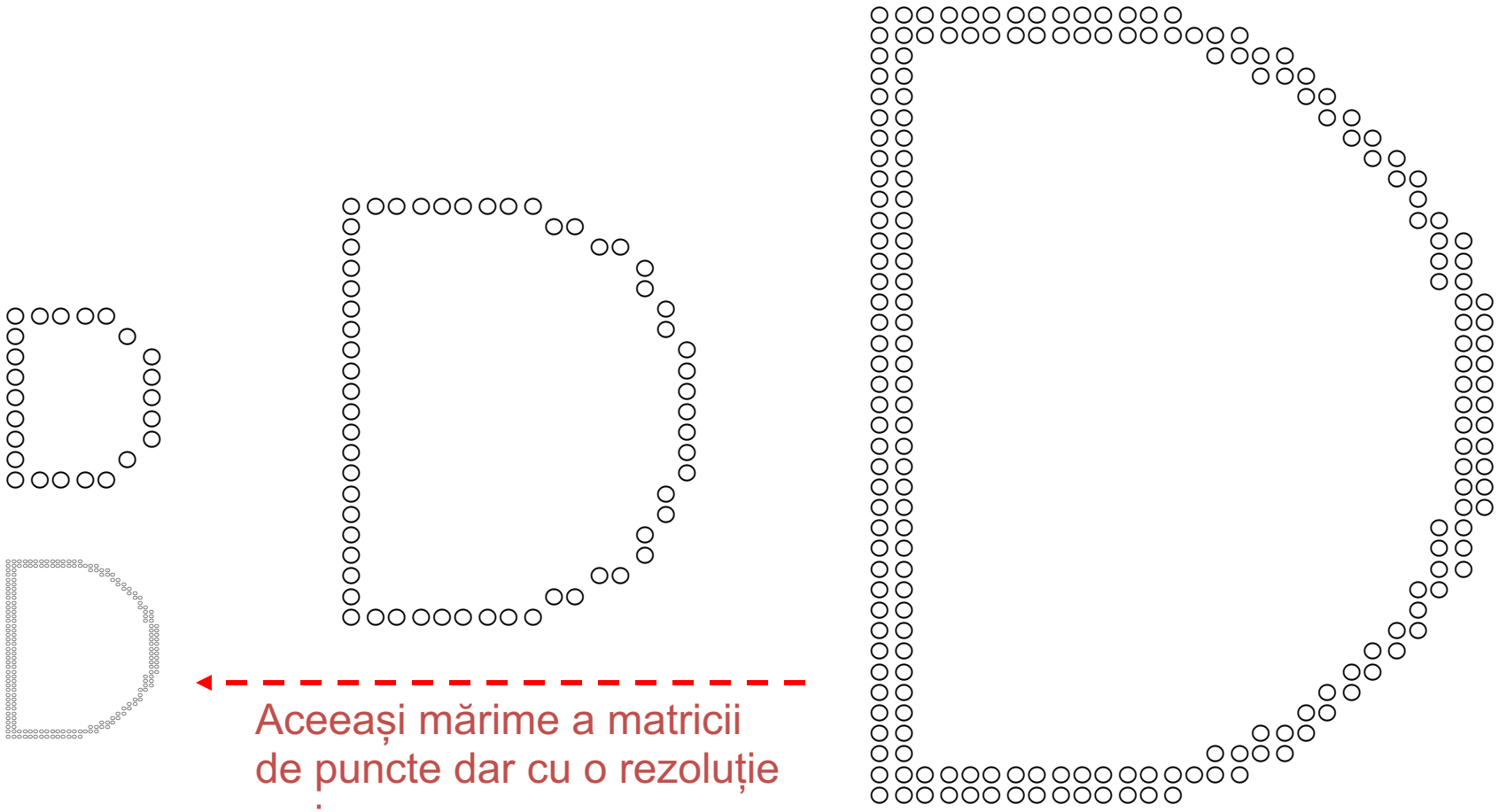


Dispozitive de tipărire



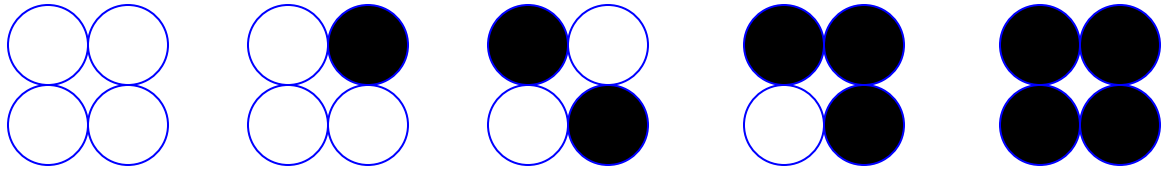
Mecanismul de scanare al imaginii într-un copiator modern.

Formarea caracterelor într-o imprimantă matriceală



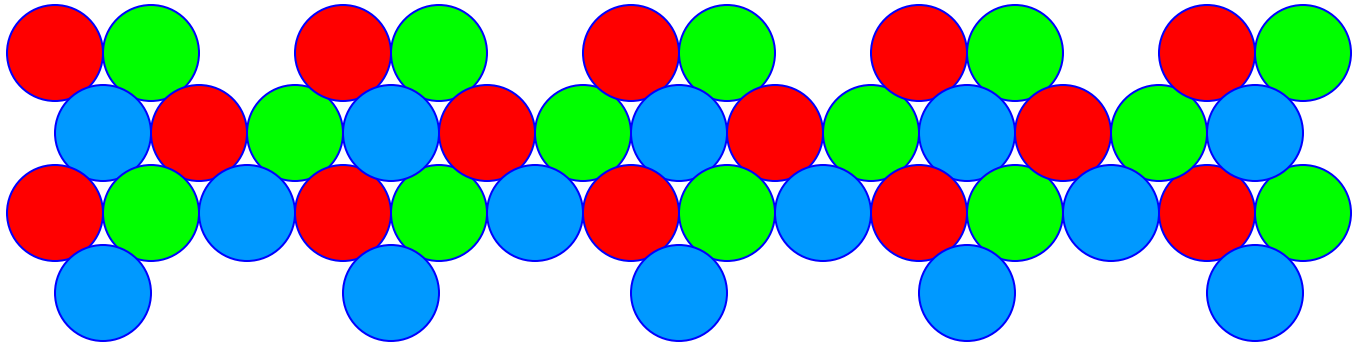
Formarea literei ”D” folosind matrici de puncte de dimesiuni diferite

Simularea diferitelor niveluri de instensitate prin difuzia punctelor

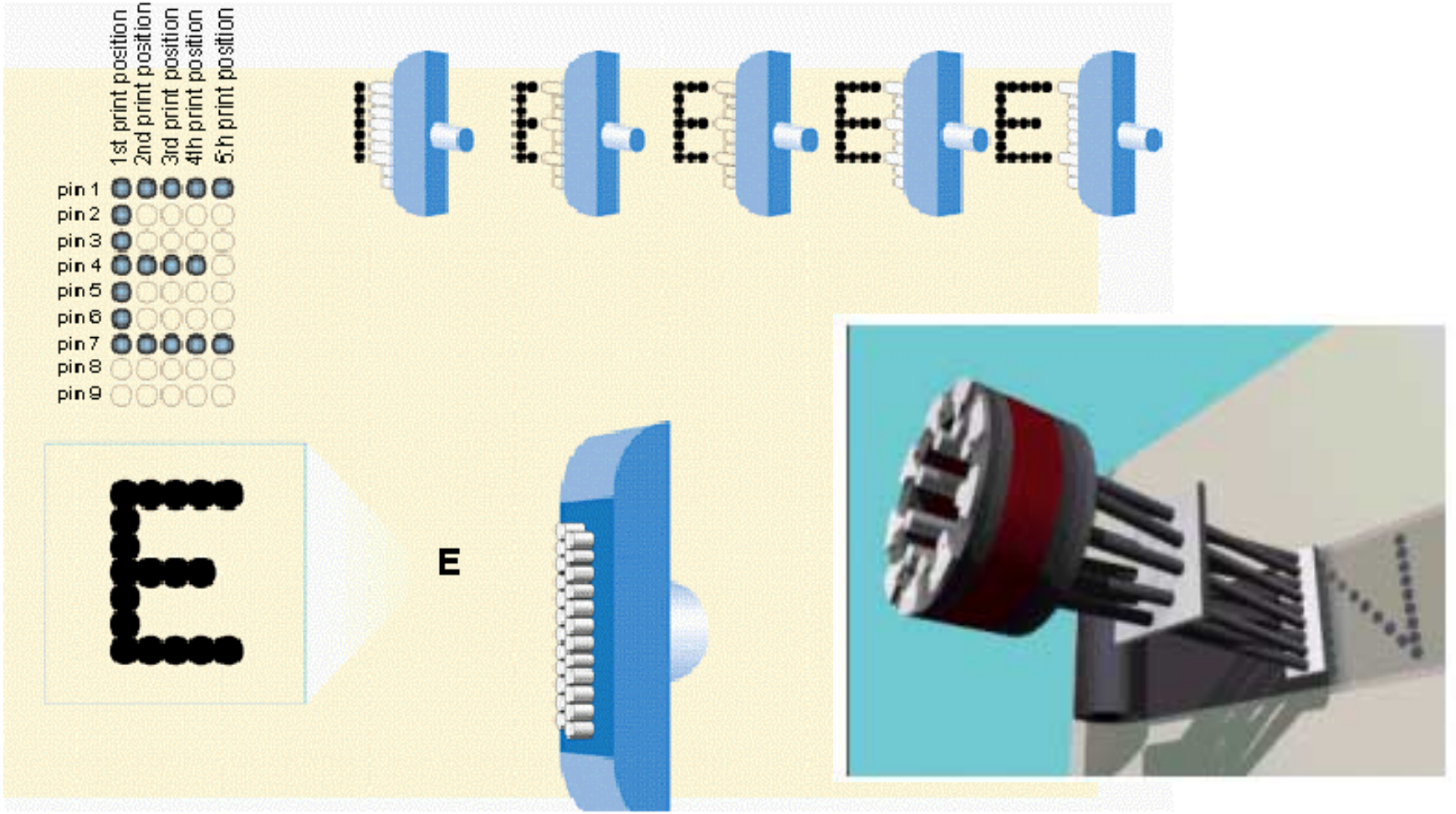


Formarea a cinci neveluri de gri pe un dispozitiv care suportă doar alb și negru (e.g., imprimantă ink-jet sau laser)

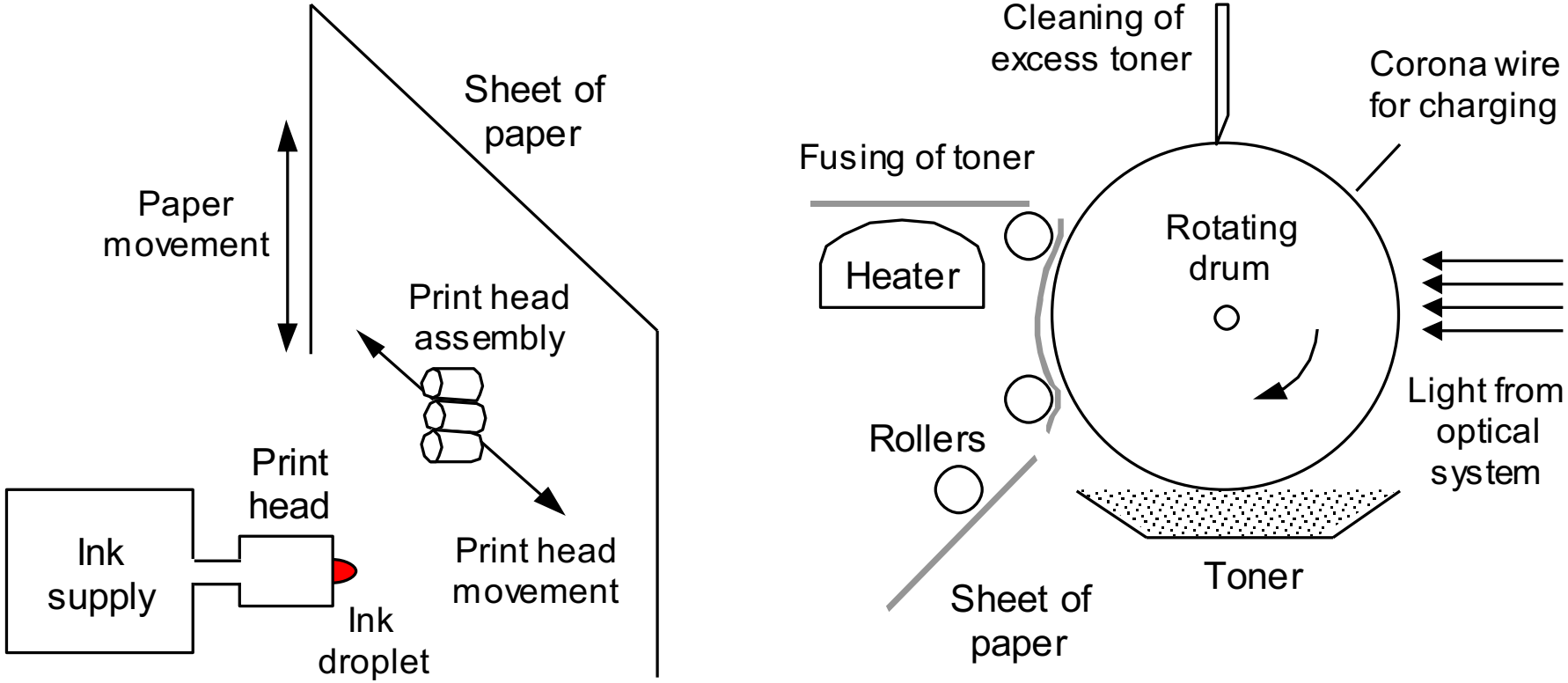
Folosirea acelorași modele în RGB pentru producerea a $5 \times 5 \times 5 = 125$ culori diferite



Mecanism de imprimare pentru o imprimantă dot-matrix



Unitate de tipărire

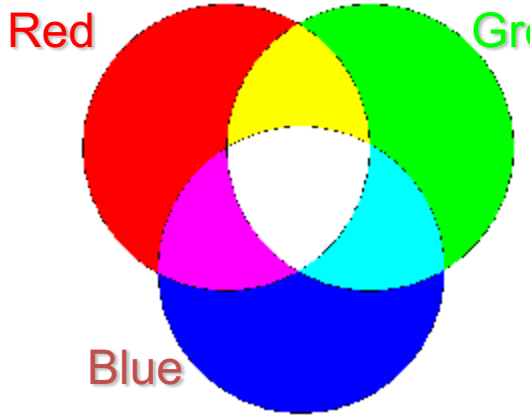


(a) Ink jet printing

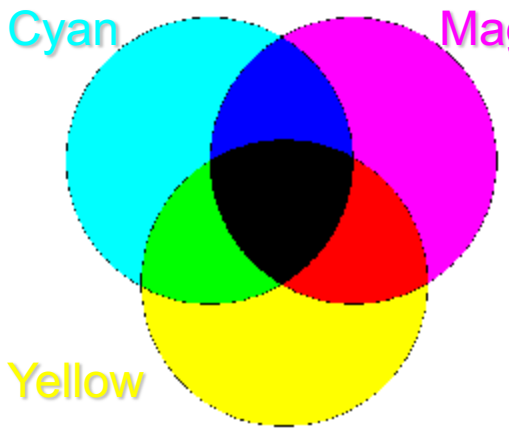
(b) Laser printing

Imprimante Ink-jet și Laser.

Cum funcționează imprimantele color



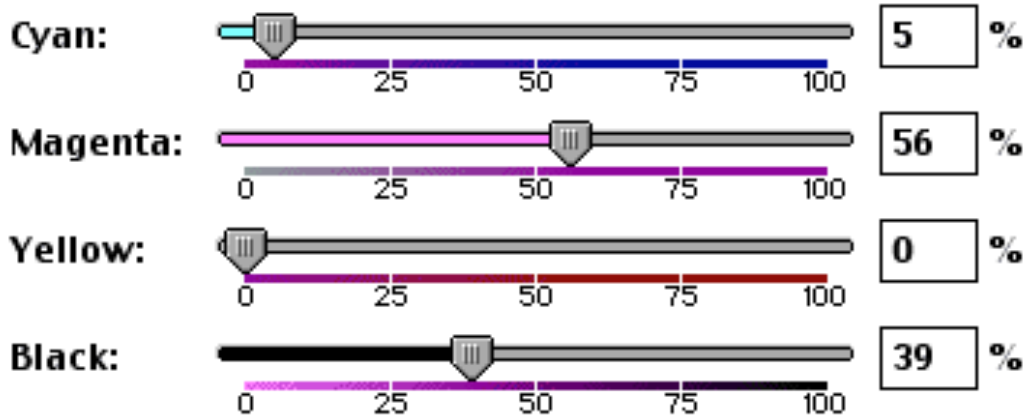
Schema RGB pentru un monitor este aditivă: diferite cantități de culoare primară sunt adăugate pentru a forma culoarea dorită.



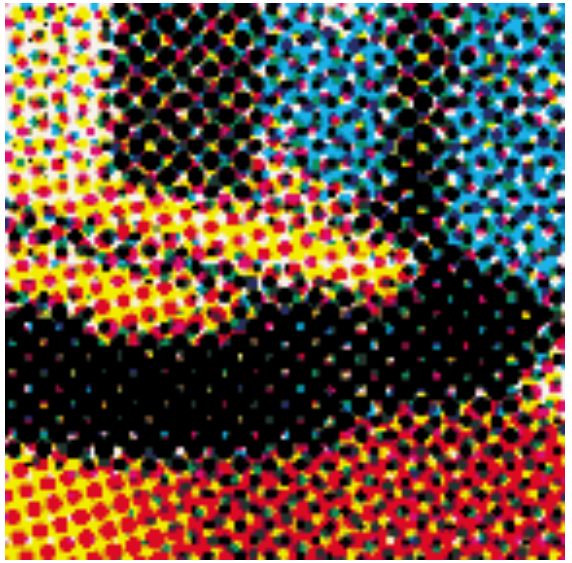
Absența lui VERDE

Schema de culori CMY pentru imprimate este substractivă: diferite cantități de culoare primară sunt substrase din alb pentru a forma culoarea dorită
Pentru a produce o nuanță de negru mai bună este folosită schema CMYK (K = black)

Procesul de imprimare CMYK



Iluzia unei culori pure folosind puncte CMYK



Roți de culoare

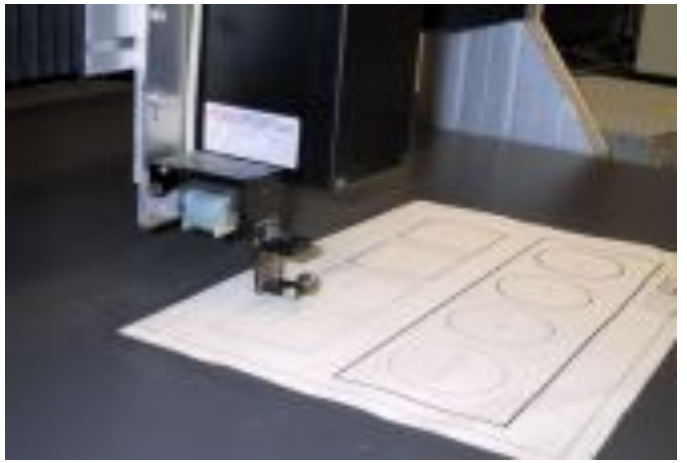
Paleta de culoare pentru artiști, folosită pentru amestecarea vopselurilor

Paleta de culori substractivă pentru imprimare (CMYK)

Paletă de culori aditivă, folosită în display-uri

Culorile primare apar la centru și la distanțe egale de-a lungul perimetrului paletei. Culorile secundare sunt la mijloc între culorile primare. Culorile terțiare sunt între ulorile primare și secundare.

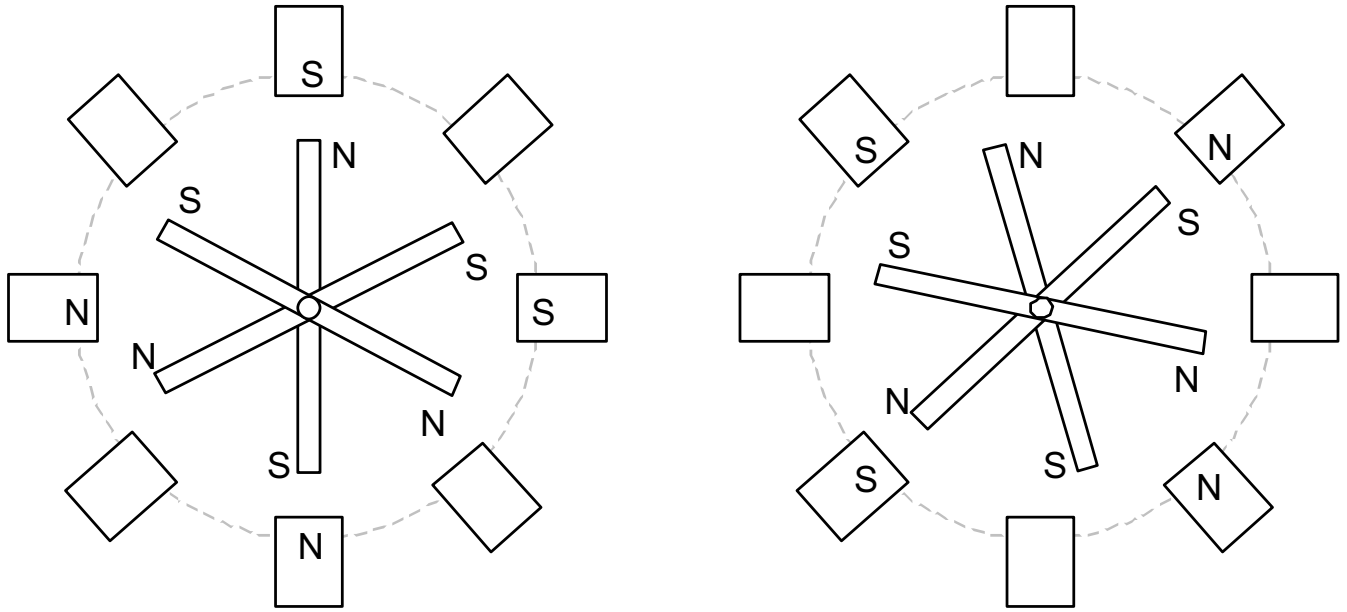
Alte dispozitive de I/O



Senzori și actuatoare

Colectează informații despre mediul înconjurător

- Senzori de lumină (fotocelule)
- Senzori de temperatură (contact și noncontact)
- Senzori de presiune

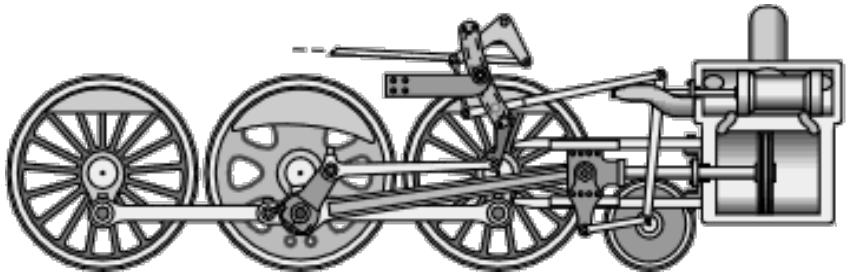


(a) Initial state (a) After rotation

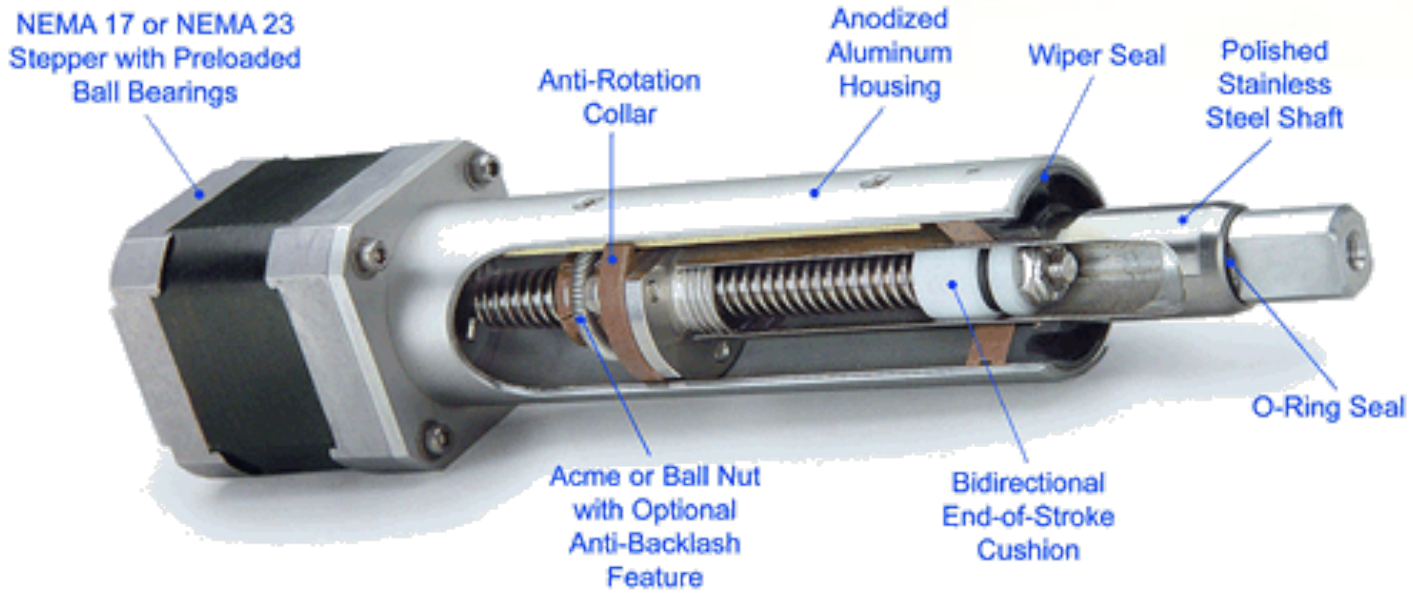
Principiul de operare pentru un motor pas cu pas

Convertirea mișcării circulare în mișcare liniară

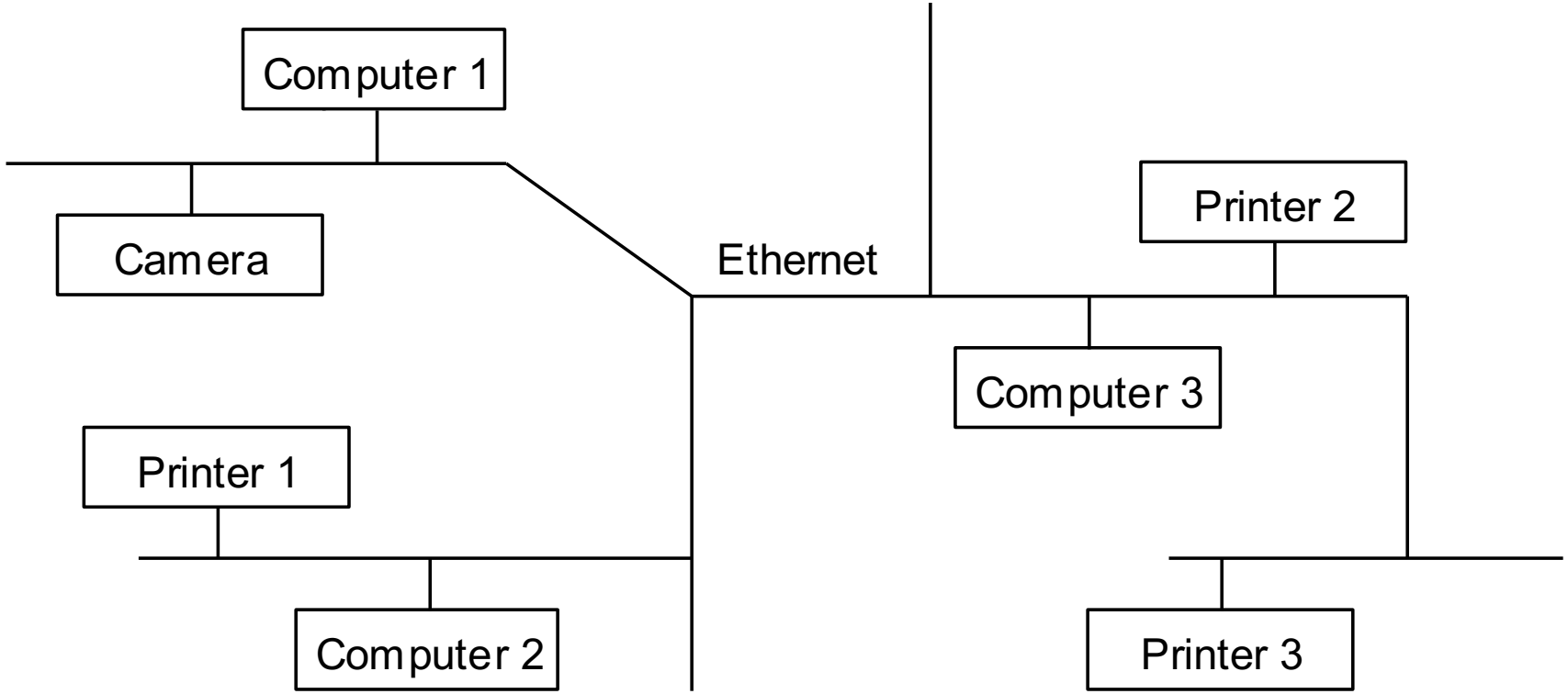
Locomotivă



Șurub

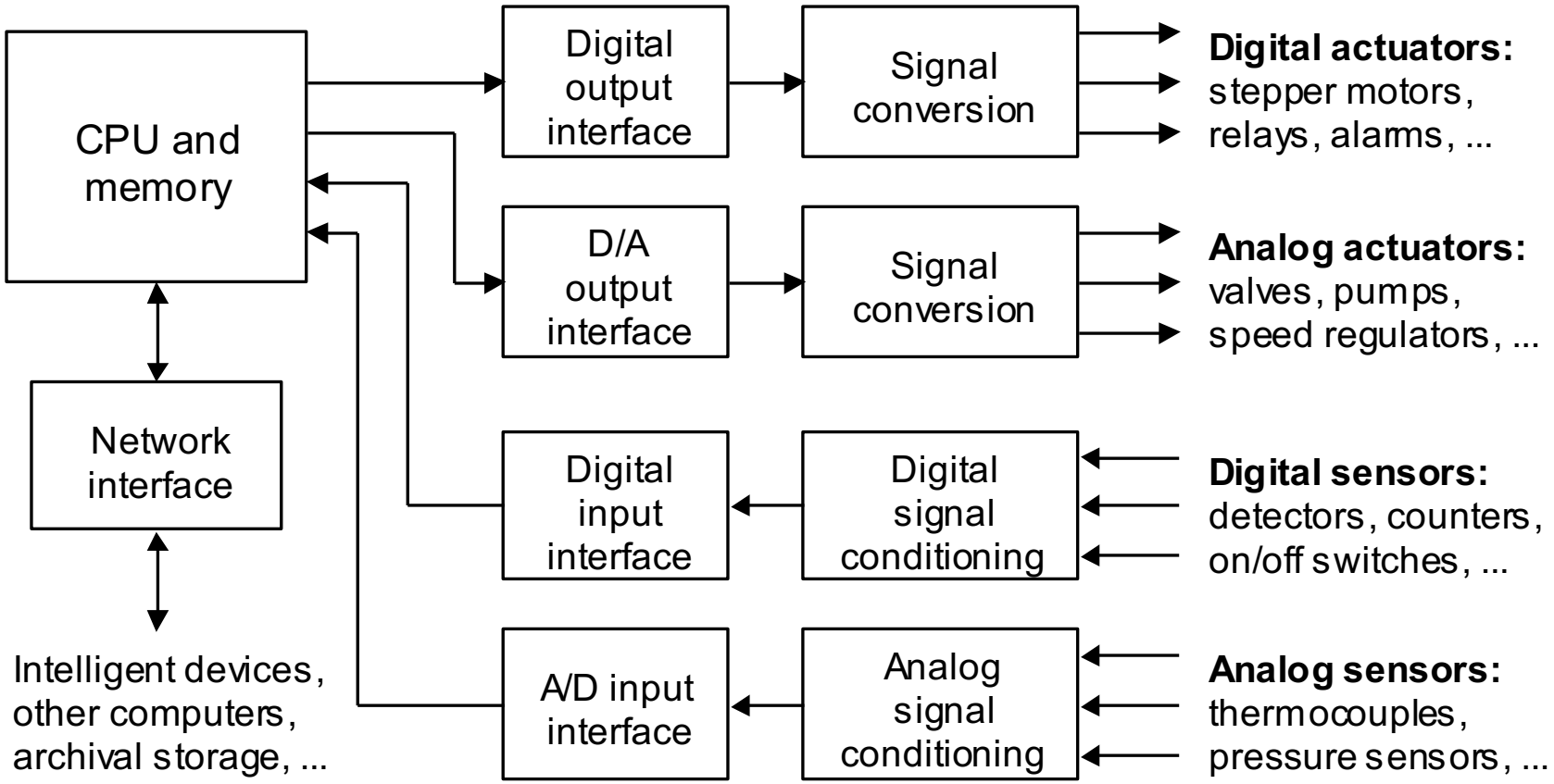


Legarea în rețea a mai multor dispozitive de calcul



Pentru perifericele cu capabilități de rețea, I/O este făcut cu ajutorul transferului de fișiere.

Input/Output în sistemele de control și embedded



Structura unui sistem de control în buclă închisă.

Programarea Input/Output

Ca orice altceva, I/O-ul este controlat prin instrucțiuni în cod mașină

- Adresarea I/O (memory-mapped) și performanța acestuia
- Scheduled vs demand-based I/O: polling vs interrupts

Cuprins
1. Performanța I/O și benchmarking
2. Adresarea Input/Output
3. I/O planificat: Polling
4. Demand-Based I/O: Interrupts
5. Transfer de date I/O și DMA
6. Îmbunătățirea performanțelor I/O

Performanță și benchmarking pentru I/O

Zidul I/O

O aplicațe de control industrial petrecea 90% din timp pentu operații cu CPU când a fost dezvoltată în anii 80. De atunci, componenta CPU fost actualizată la fiecare 5 ani dar componenta I/O a rămas neschimbată. Presupunând că performanța CPU s-a îmbunătățit de 10x cu fiecare upgrade, determinați fracțiunea de timp petrecută în operații I/O pe întreaga durată de viață a sistemului.

Soluție

Aplicăm legea lui Amdahl pentru 90% din task cu speedup de 10, 100, 1000 și 10000 peste o perioadă de 20 de ani. Pe parcursul acestor upgrade-uri, timpul de funcționare a fost redus de la originalul 1 la 0.1 + 0.9/10 = 0.19, 0.109, 0.1009 și 0.10009, deci fracția de timp petrecut în operații I/O este 52.6, 91.7, 99.1 respectiv 99.9%. Ultimele două upgrade-uri de CPU nu au ajutat foarte mult.

Tipuri de Input/Output Benchmarking

Benchmarking pentru supercomputing I/O

Citirea unei cantități mari de date de intrare
Scrierea multor instanțe pentru checkpointing
Salvarea unui set foarte mic de rezultate
Productivitatea I/O, în MB/s, e importantă

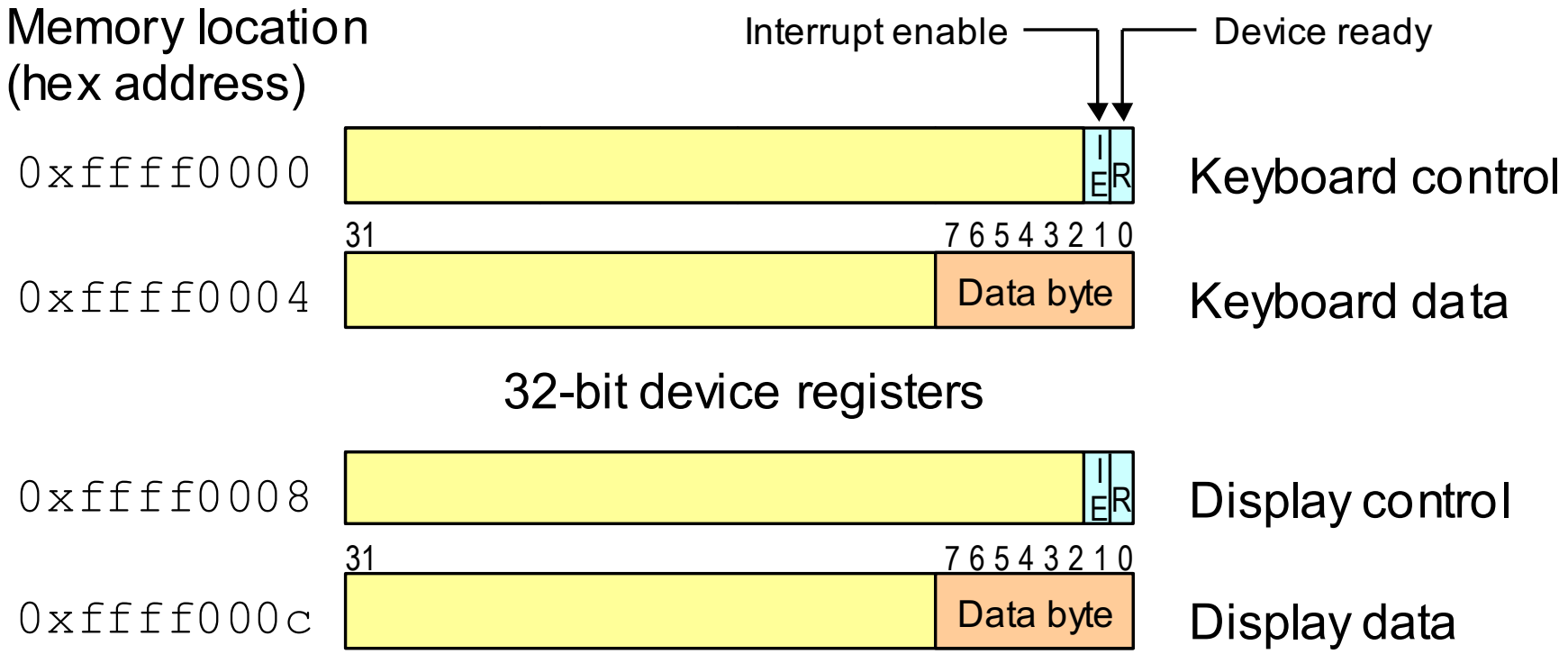
Benchmarking pentru procesarea tranzacțiilor I/O

Bază de date imensă dar fiecare tranzacție este de mici dimensiuni
Câteva (2-10) accesuri la disc per tranzacție
Rata I/O (accese la disc pe secundă) este importantă

Benchmarking pentru file system I/O

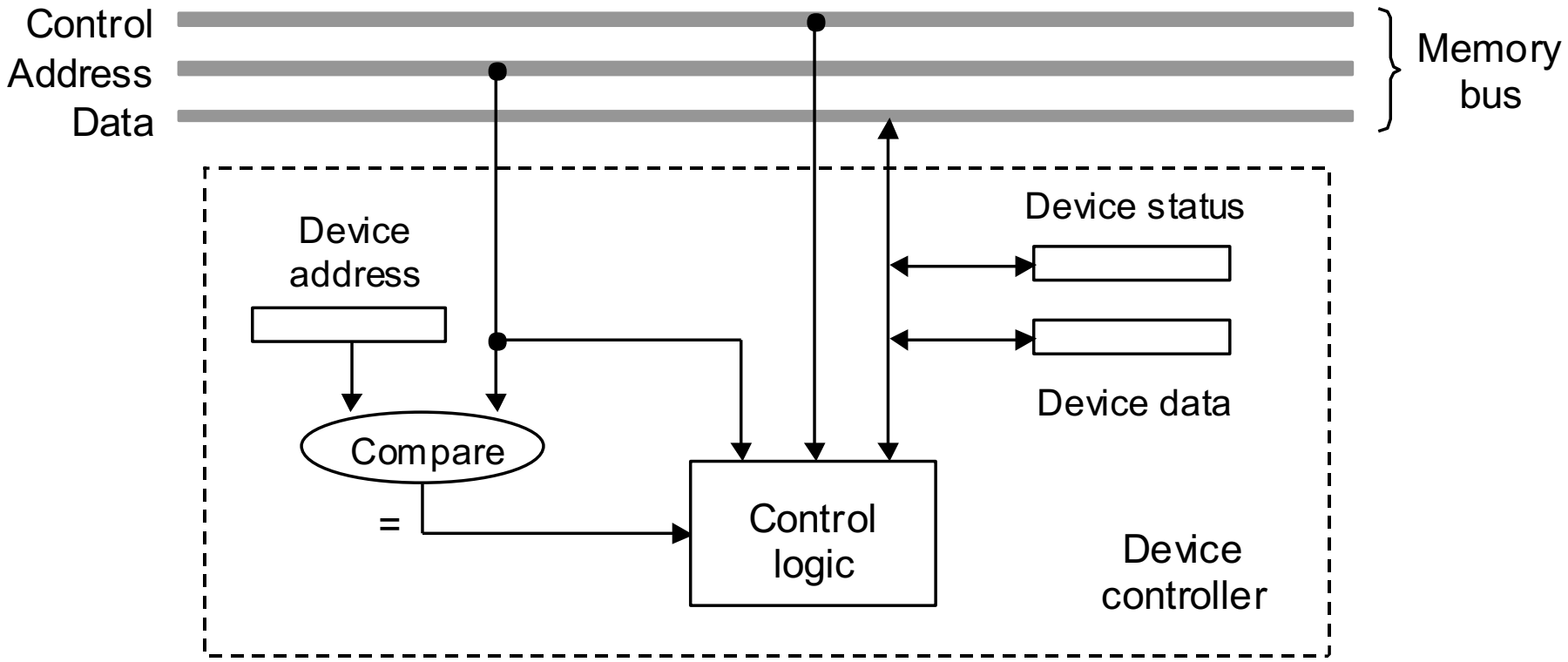
Crearea unui fișier, managementul directoarelor, indexare...
Benchmark-urile sunt de obicei specifice domeniului

Adresarea perifericelor I/O



Registre de control și date pentru tastatură și unitatea de display la procesorul MiniMIPS

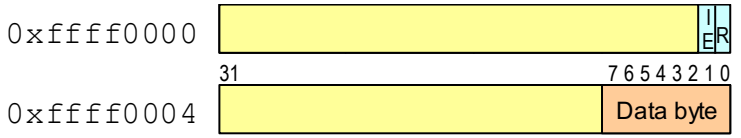
Hardware pentru adresarea I/O



Logica de adresare pentru un controller I/O

Citirea datelor dintr-un periferic I/O

Exemplu de cod pentru procesorul MiniMIPS care așteaptă până când tastatura are de transmis un simbol apoi citește simbolul respectiv în registrul `$v0`.



Soluție

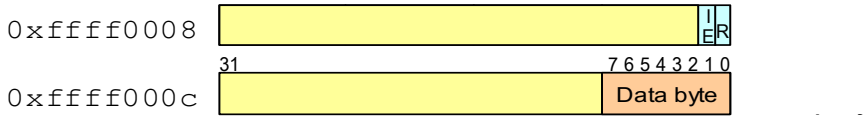
Programul examinează în continuu registrul de control al tastaturii și iese din ciclul de ”busy waiting” atunci când bitul R a fost activat.

```
        lui  $t0,0xffff      # put 0xffff0000 in $t0
idle:   lw   $t1,0($t0)      # get keyboard's control word
        andi $t1,$t1,0x0001  # isolate the LSB (R bit)
        beq  $t1,$zero,idle  # if not ready (R = 0), wait
        lw   $v0,4($t0)      # retrieve data from keyboard
```

Acest tip de citire este potrivit numai dacă procesorul așteaptă un input critic de la utilizator și nu poate continua în absența acestuia.

Scrierea de date într-un periferic I/O

Exemplu de secvență de cod pentru procesorul MiniMIPS pentru scrierea în unitatea de afișare. Programul așteaptă până când unitatea de display este gata să accepte un nou simbol apoi transferă conținutul registrului `$a0`



Soluție

Programul examinează în continuu registrul de control al unității de display și iese din ciclul de busy waiting atunci când bitul R este activat de periferic.

```
        lui   $t0,0xffff      # put 0xffff0000 in $t0
idle:   lw    $t1,8($t0)      # get display's control word
        andi  $t1,$t1,0x0001  # isolate the LSB (R bit)
        beq   $t1,$zero,idle  # if not ready (R = 0), wait
        sw    $a0,12($t0)     # supply data to display unit
```

Acest tip de scriere este potrivit doar dacă ne permitem să avem un procesor dedicat scrierii în unitatea de display.

I/O planificat: Polling

Ce procent din timpul de execuție al unui procesor de 1GHz este petrecut în polling dacă fiecare acces durează 800 cicli de ceas?

Tastatura trebuie să fie interogată cel puțin de 10 ori pe secundă

Unitatea floppy trimite pachete de 4 octeți la o rată de 50 KB/s

Hard discul trimite pachete de 4 octeți la o rată de 3 MB/s

Soluție

Pentru tastatură, împărțim numărul de cicli necesari pentru 10 interogări cu numărul de cicli disponibili într-o secundă:
$(10 \times 800)/10^9 \cong 0.001\%$

Unitatea floppy trebuie interogată de 50K/4 = 12.5K ori pe secundă
$(12.5K \times 800)/10^9 \cong 1\%$

Unitatea hard disk trebuie interogată 3M/4 = 750K ori pe secundă
$(750K \times 800)/10^9 \cong 60\%$

I/O asincron: Întreruperi

Luăm hard disk-ul din exemplul anterior (transferul de pachete de 4B la 3 MB/s atunci când e activ). Presupunem că discul este activ 5% din timp.

Overhead-ul generat de întreruperea CPU-ului și efectuarea transferului este de 1200 cicli de ceas. Ce procent din timpul total de execuție al unui CPU de 1GHz este petrecut în operații cu hard disk-ul?

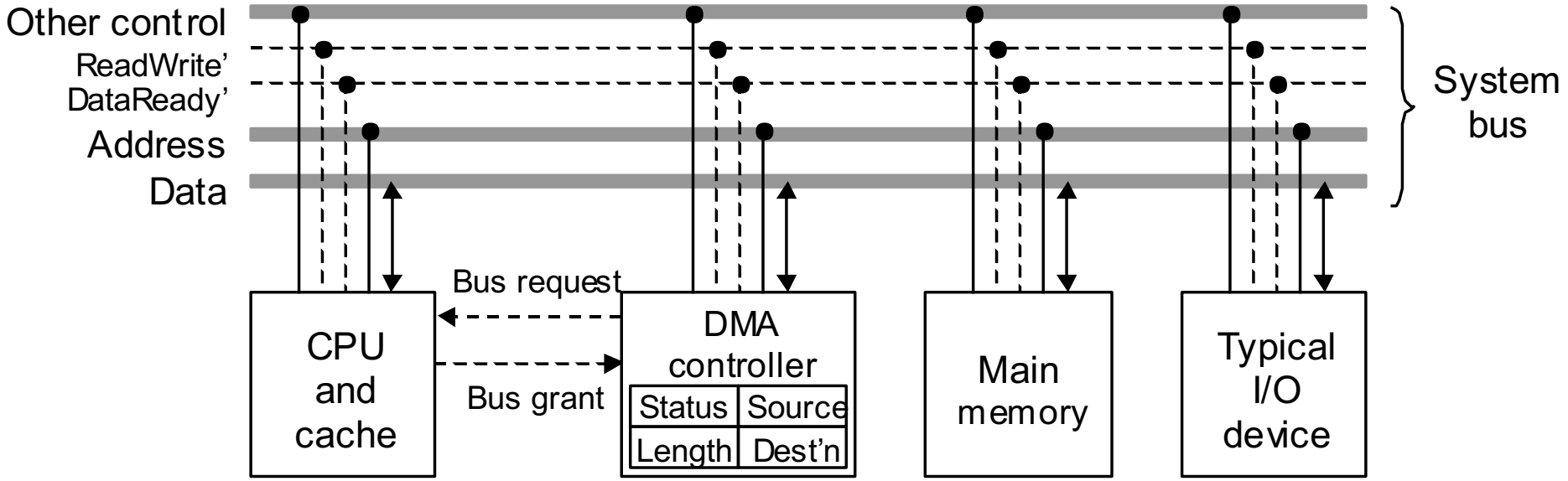
Soluție

Atunci când e activ, hard disk-ul produce 750K întreruperi pe secundă

$0.05 \times (750K \times 1200)/10^9 \cong 4.5\%$ (în comparație, aveam 60% pt polling)

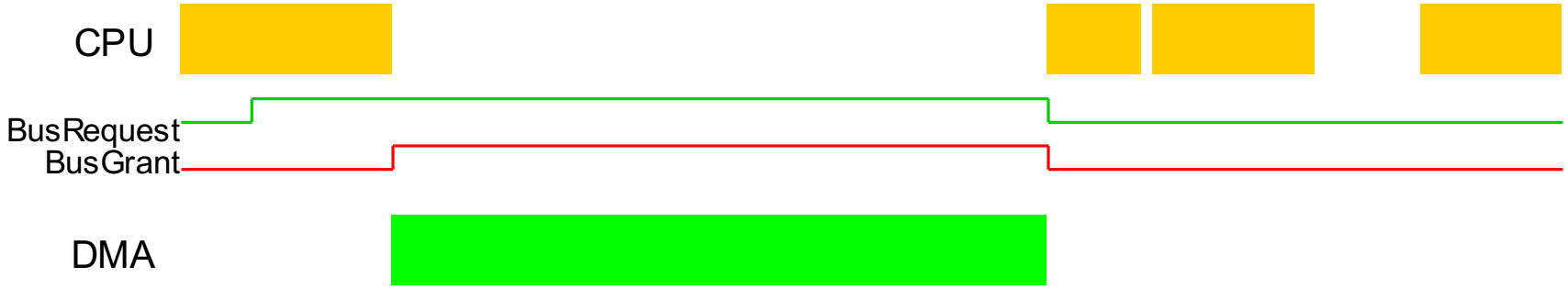
Chiar dacă întreruperea CPU-ului pentru polling generează o penalizare mai mare, din cauză că discul este in general nefolosit, operațiile bazate pe întreruperi duc la performanță mărită.

Transferul de date I/O și DMA

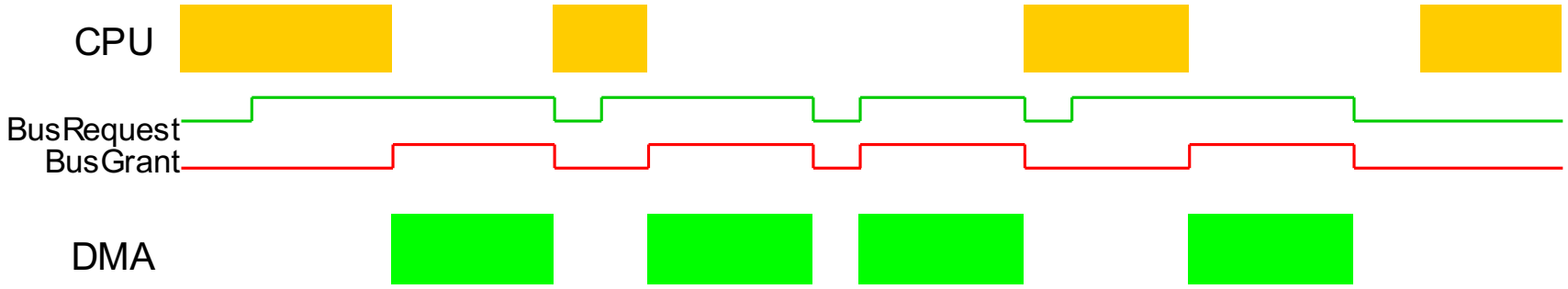


Un controller DMA partajează aceeași magistrală de memorie cu procesorul.

Operații DMA



(a) DMA transfer in one continuous burst



(b) DMA transfer in several shorter bursts

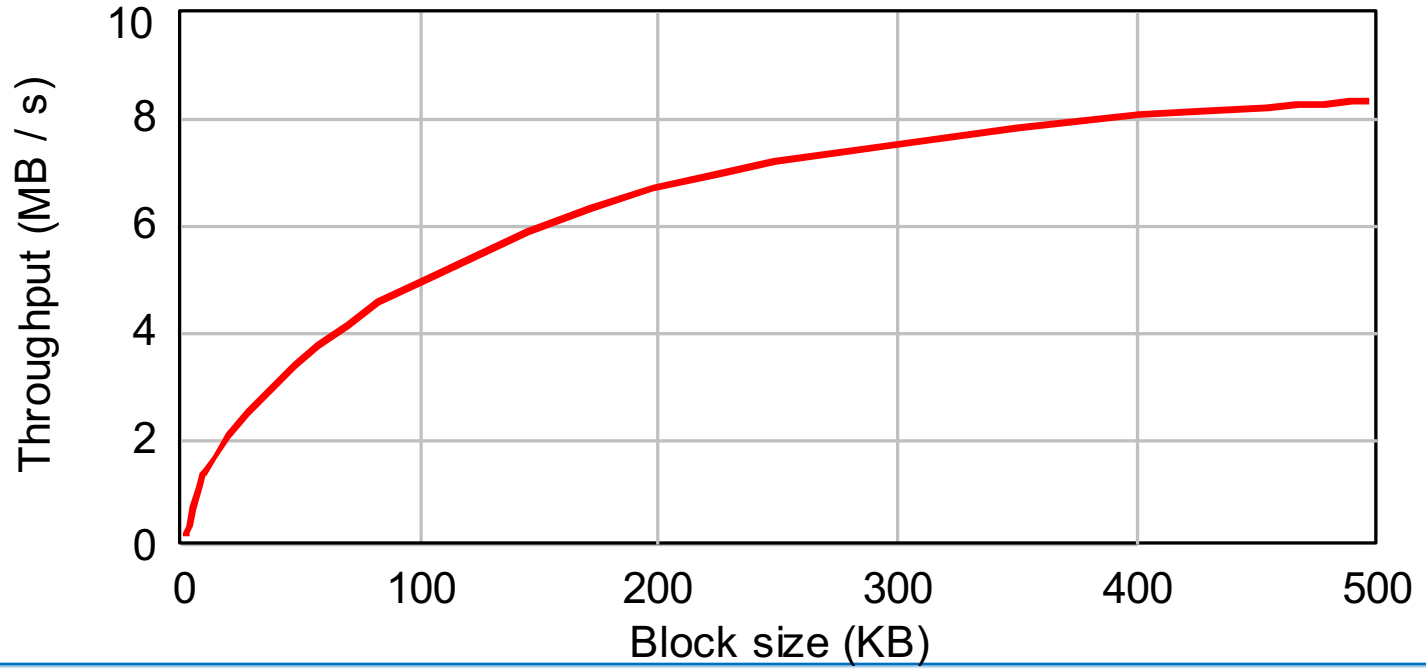
Funcționarea DMA și semnalele asociate de control de pe magistrală.

Îmbunătățirea performanțelor I/O

Lățimea de bandă I/O efectivă pentru o unitate de disc

Avem un hard disk cu sectoare de 512B și o latență medie de acces de date de 10ms la rata maximă de transfer de 10MB/s. Afișați grafic variația lățimii de bandă efective pe măsură ce unitatea de transfer de date (bloc de date) variază în dimensiune de la 1 sector (0.5 KB) la 1024 sectoare (500 KB).

Soluție



Calcularea productivității efective

Lățimea de bandă efectivă pentru I/O cu unitatea de disc

Timp total de acces pentru x octeți = 10 ms + xfer time = $(0.01 + 10^{-7}x)$ s

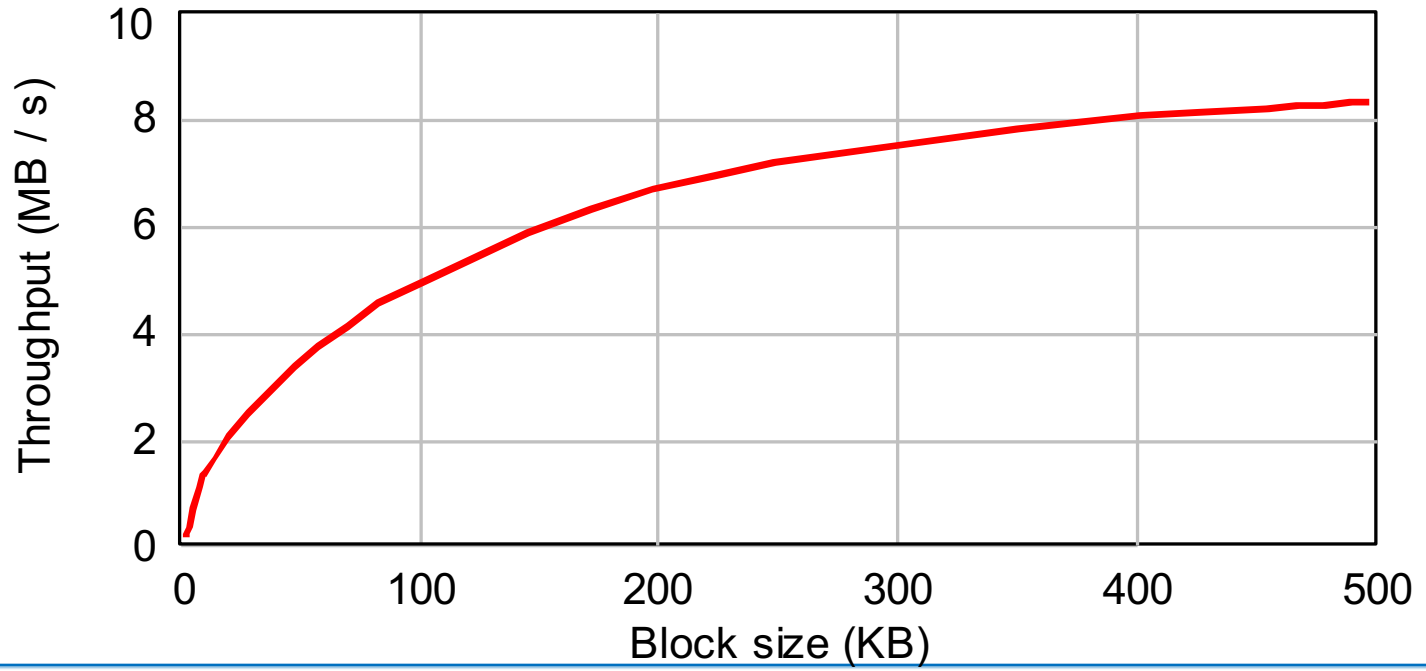
Timp de acces efectiv pentru un octet = $(0.01 + 10^{-7}x)/x$ s/B

Rata efectivă de trasfer = $x/(0.01 + 10^{-7}x)$ B/s

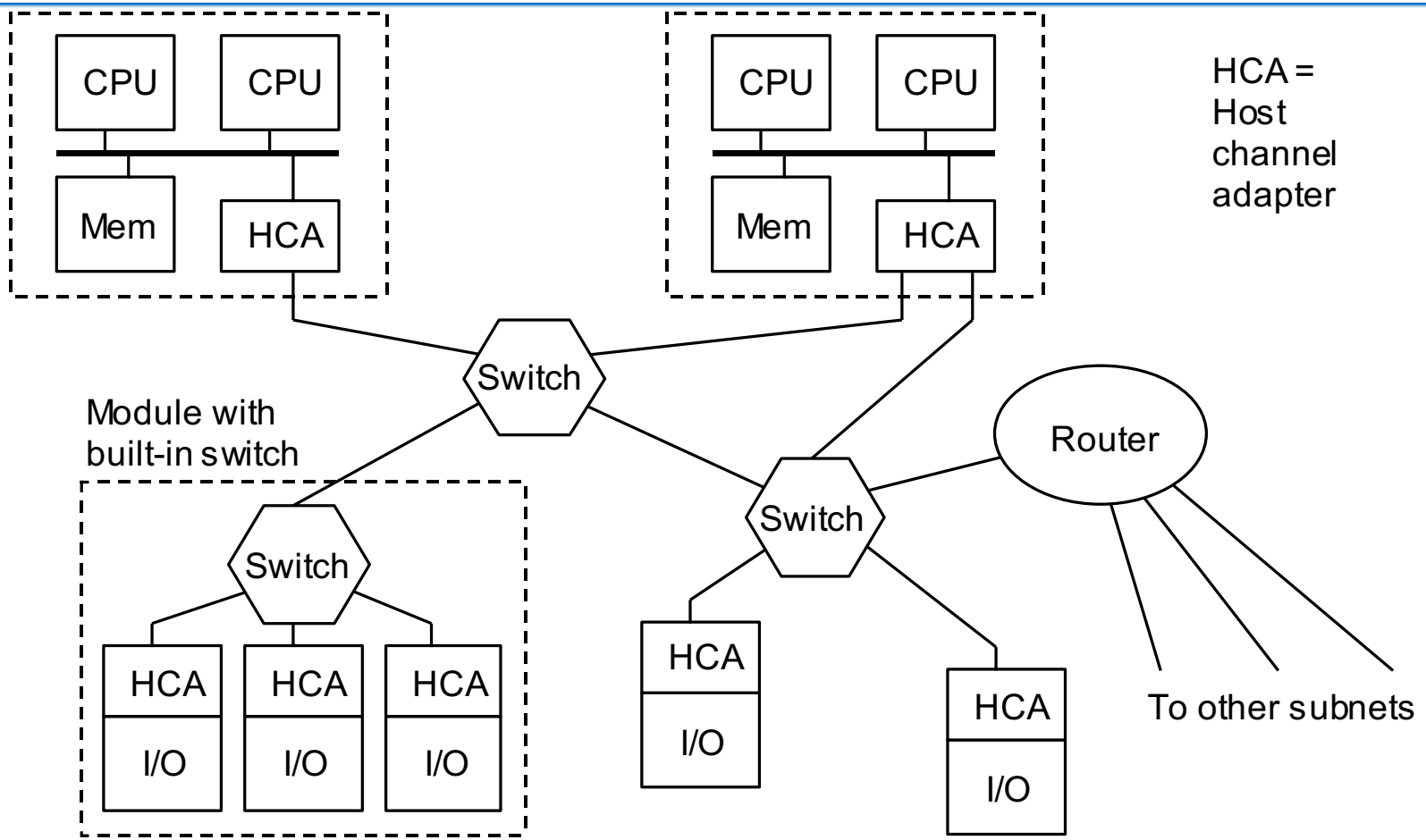
Pentru x = 100 KB: Rata efectivă de transfer = $10^5/(0.01 + 10^{-2}) = 5\times10^6$ B/s

Latență medie = 10 ms

Throughput maxim = 10 MB/s



Input/Output distribuit



Exemplu de configurație pentru I/O distribuit Infiniband

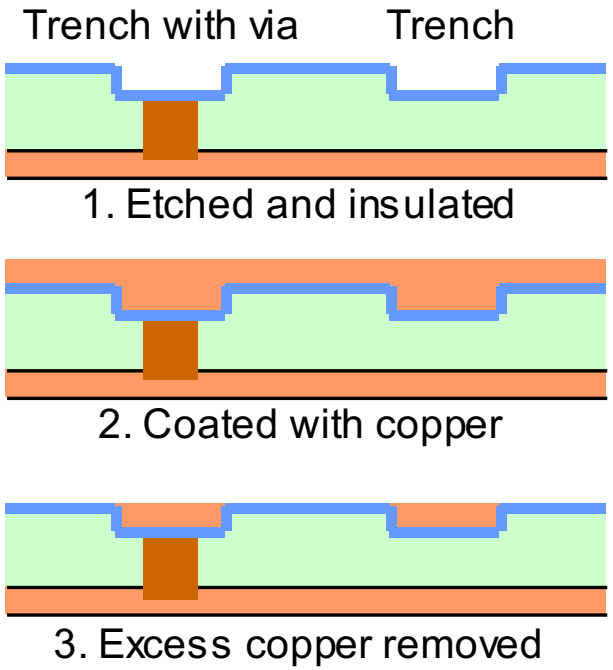
Magistrale, legături și interfațare

Magistralele de date partajate sunt des întâlnire în calculatoarele moderne:

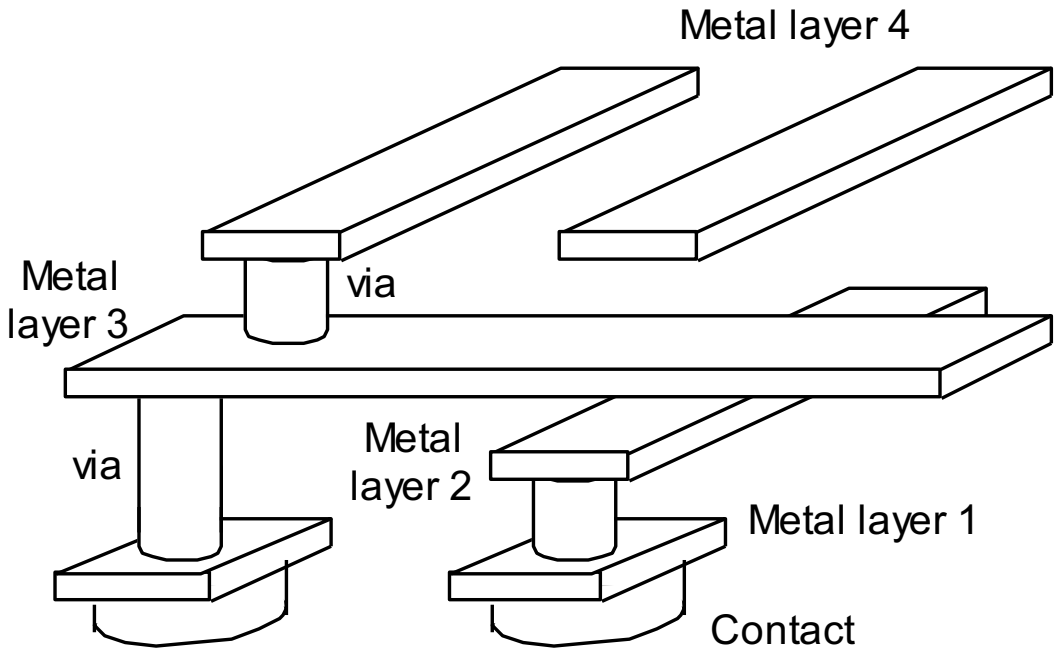
- Mai puține fire și conexiuni, flexibilitate și extendabilitate
- Trebuie să implementeze mecanisme de arbitrare și sincronizare

Cuprins
1. Legături Intra- și Inter-sistem
2. Legături între sisteme
3. Protocoale de comunicație pe magistrală
4. Arbitrare și performanță
5. Interfațare
6. Standarde pentru interfațare

Legături inter și intra-sistem



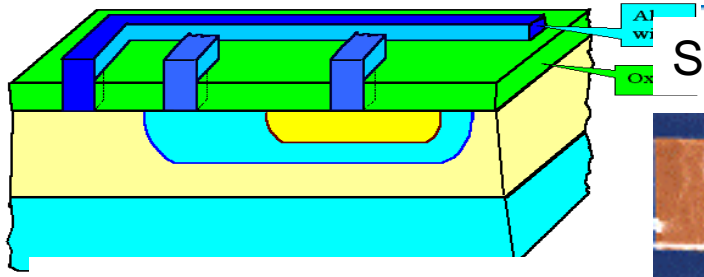
(a) Cross section of layers



(b) 3D view of wires on multiple metal layers

Conectivitatea în interiorul microprocesoarelor și a circuitelor imprimate (PCB) este realizată prin straturi multiple de conductori.

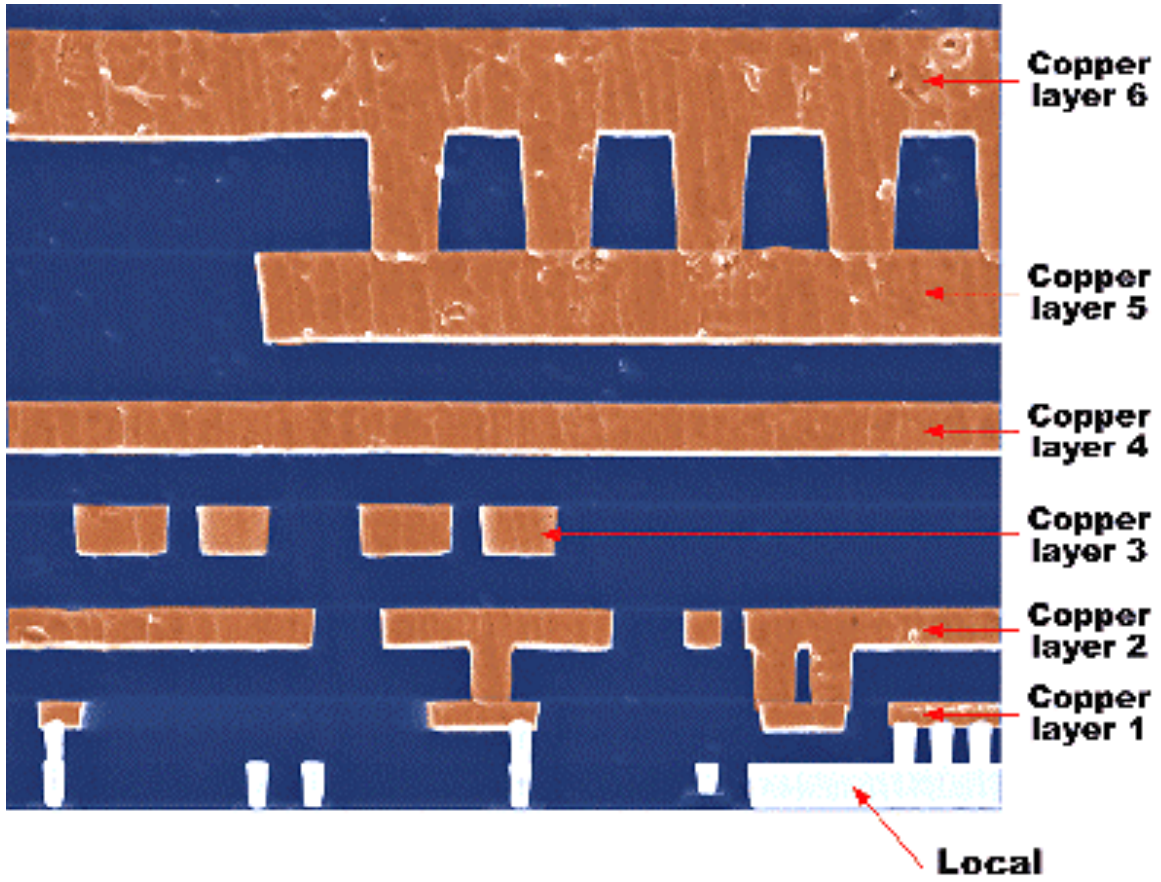
Straturi multiple metalizate pe un microchip sau un PCB



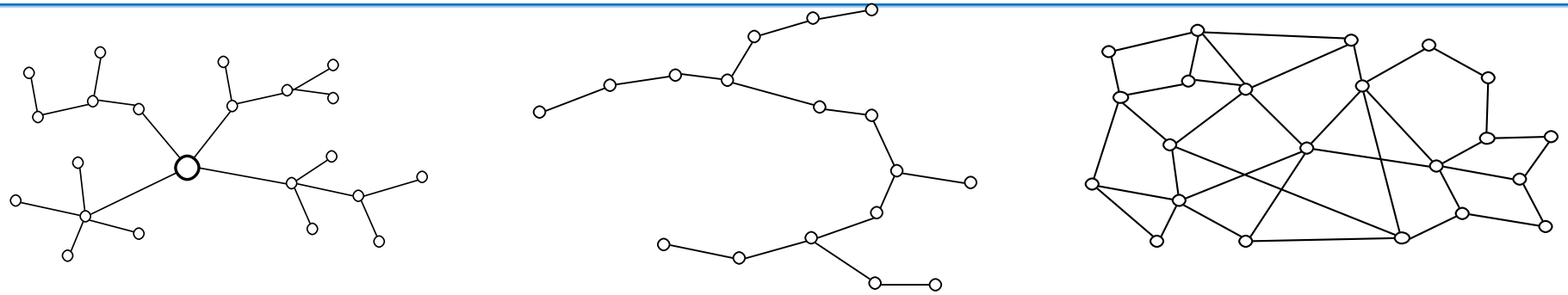
Elementele active și conexiunile dintre ele

-Chipurile moderne au în jur de 8-9 straturi de conductor metalizat
-Straturile superioare conțin de obicei liniile mai lungi sau conexiunile pentru alimentare

Secțiune transversală prin straturile conductoare

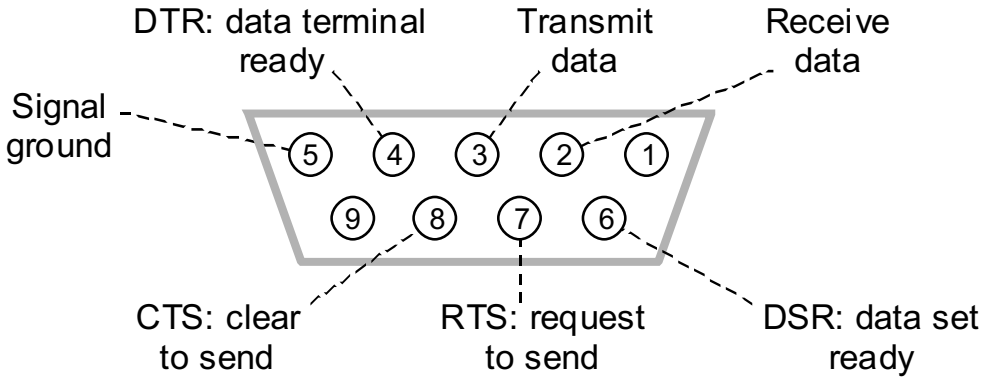


Legături între sisteme



(a) RS-232 (b) Ethernet (c) ATM

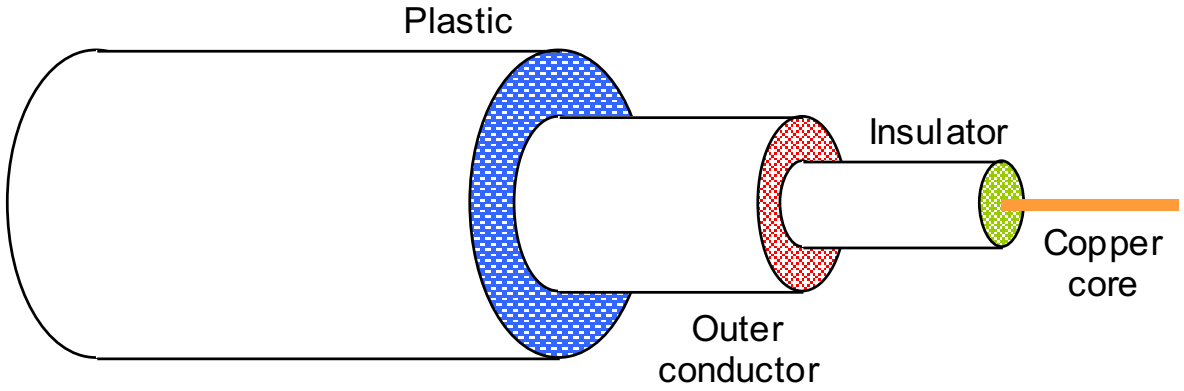
Exemple de scheme de conexiune pentru diferite protocoale



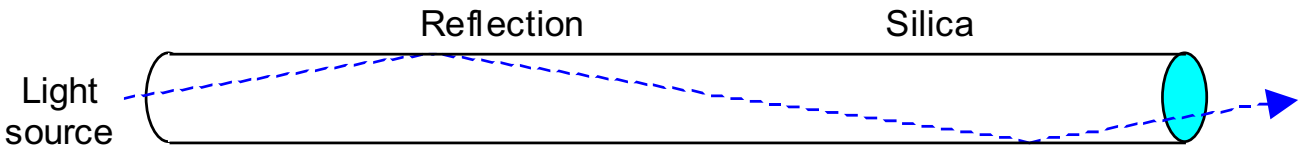
Interfața serială RS-232. Conectorul de 9 pini și semnalele standard.

Medii de comunicație a datelor

Twisted pair



Coaxial cable



Optical fiber

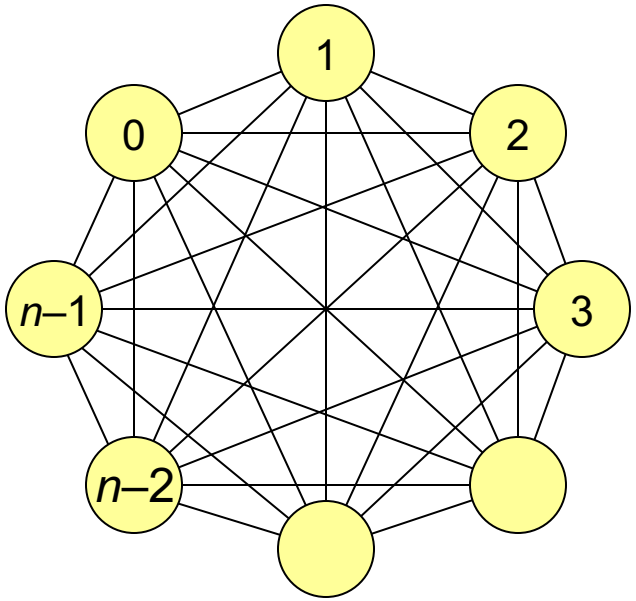
Cele mai folosite medii de comunicație de date pentru transmisia prin cablu.

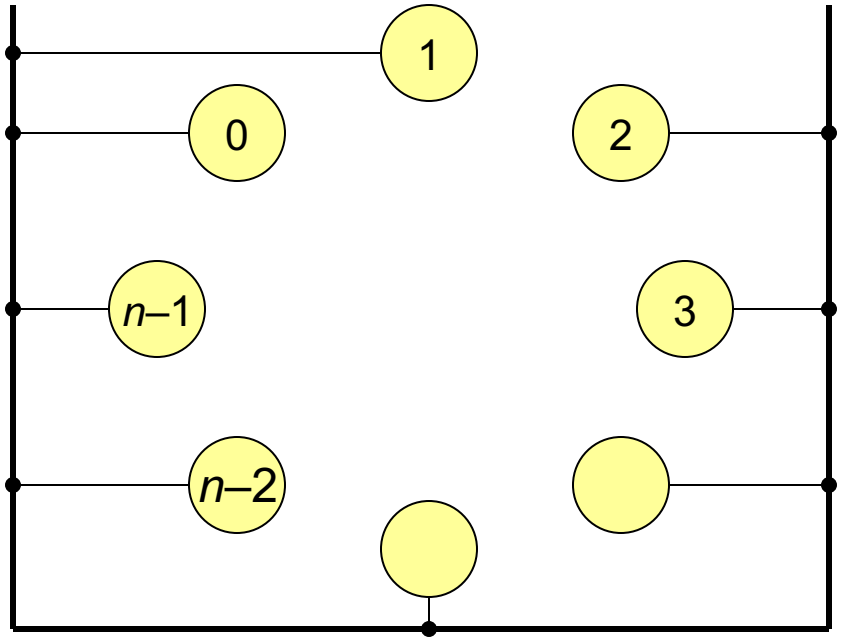
Comparație între legăturile dintre sisteme

Performanțele celor trei protocoale de conectare prezentate

Proprietăți	RS-232	Ethernet	ATM
Lungime maximă segment (m)	~10m	~100m	~1000m
Lungime maximă rețea(m)	~10m	~100m	Nelimitat
Bit rate (Mb/s)	Up to 0.02	10/100/1000	155-2500
Unitate de transmisie (B)	1	~100	53
Latență maximă (ms)	< 1	10-100	100
Domeniul tipic de aplicație	Input/Output	LAN	Backbone
Complexitatea și costul unității	Mic	Mic	Mare

Magistralele de date

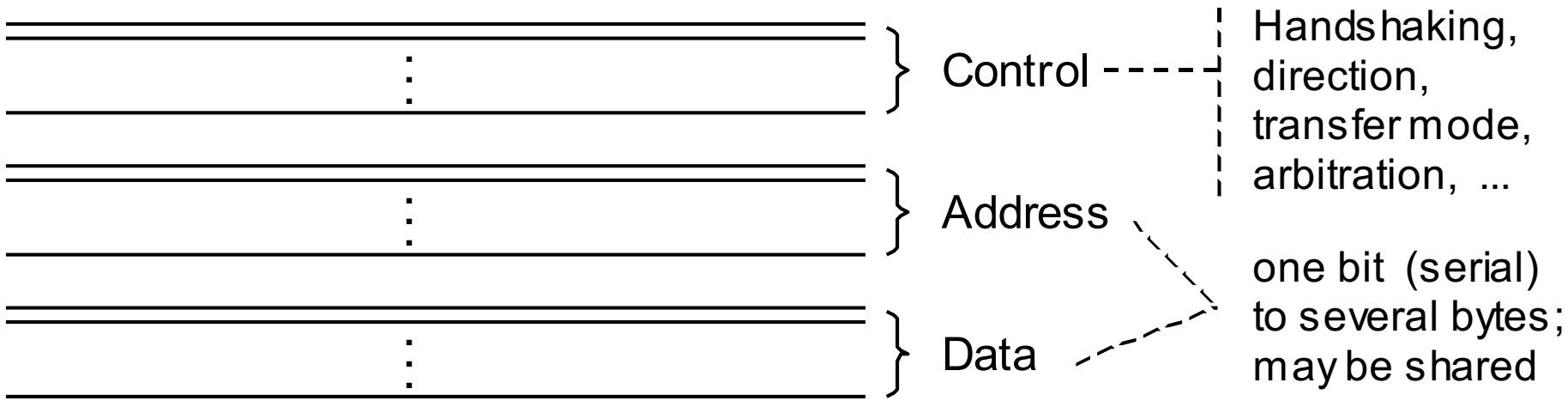




Conexiunile punct la punct dintre n unități necesită $n(n-1)$ canale, sau $n(n-1)/2$ legături bidirecționale; adică, $O(n^2)$ legături

Conectivitatea pe bus necesită doar un port de intrare și unul de ieșire pentru fiecare unitate, adică $O(n)$ conexiuni în total.

Tipuri de magistrale și semnale de date

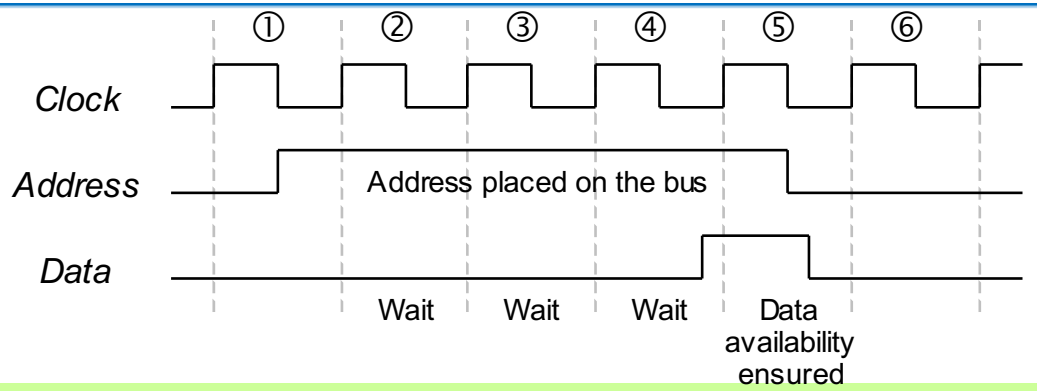


Cele trei tipuri de linii care se găsesc într-o magistrală.

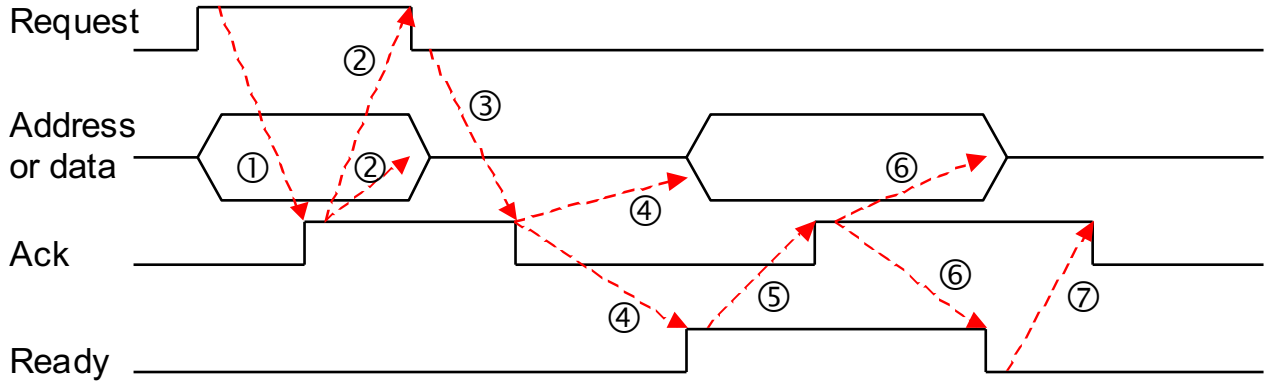
Un sistem ce calcul tipic poate să conțină în jur de 10 magistrale diferite de date:

1. Magistrale Legacy: PC bus, ISA, RS-232, port paralel
2. Magistrale Standard: PCI, SCSI, USB, Ethernet
3. Magistrale Proprietare: acces de mare viteză sau dedicat

Protocoale de comunicație pe magistrală

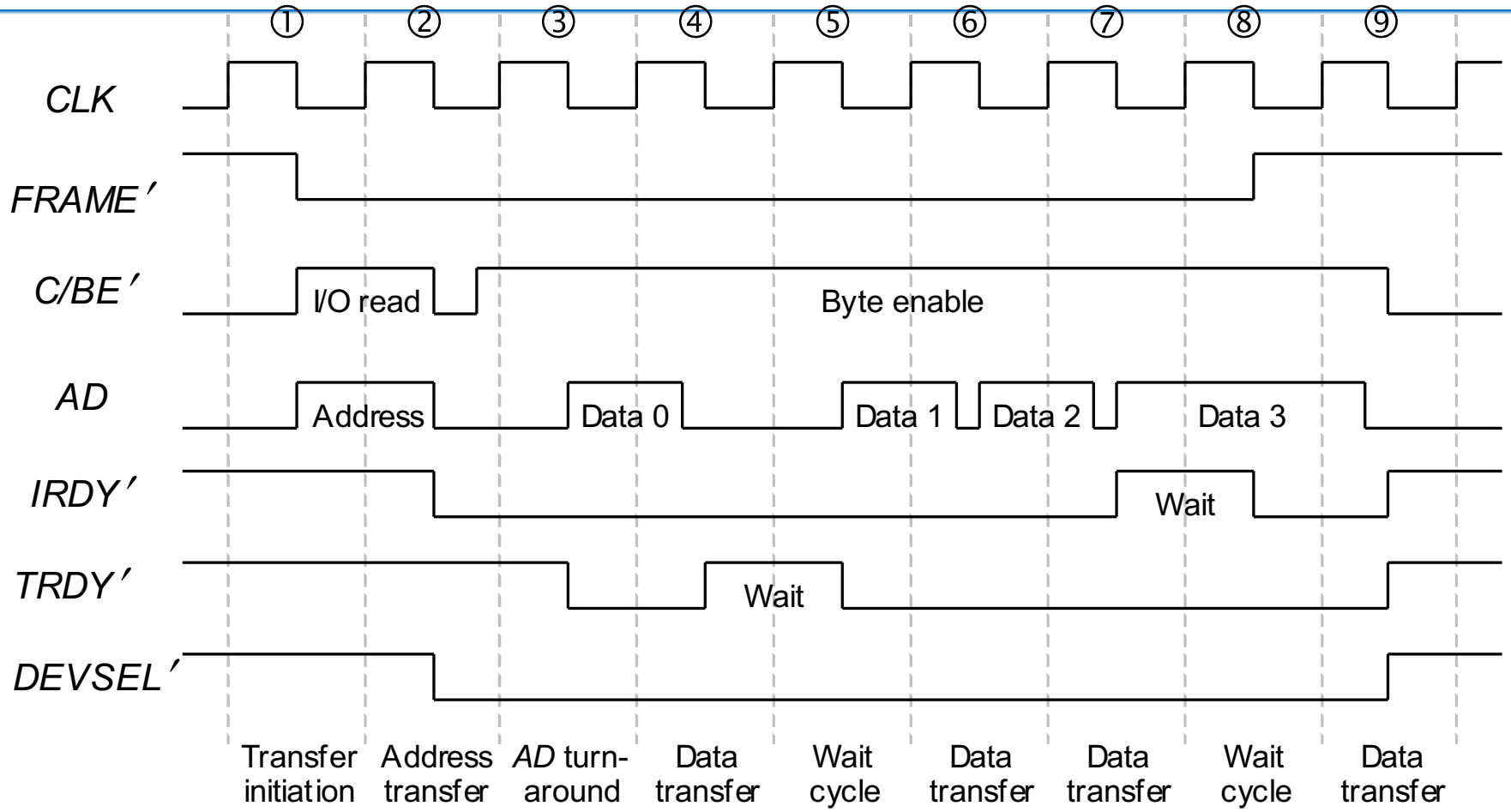


Magistrală sincronă cu dispozitive de latență fixă.



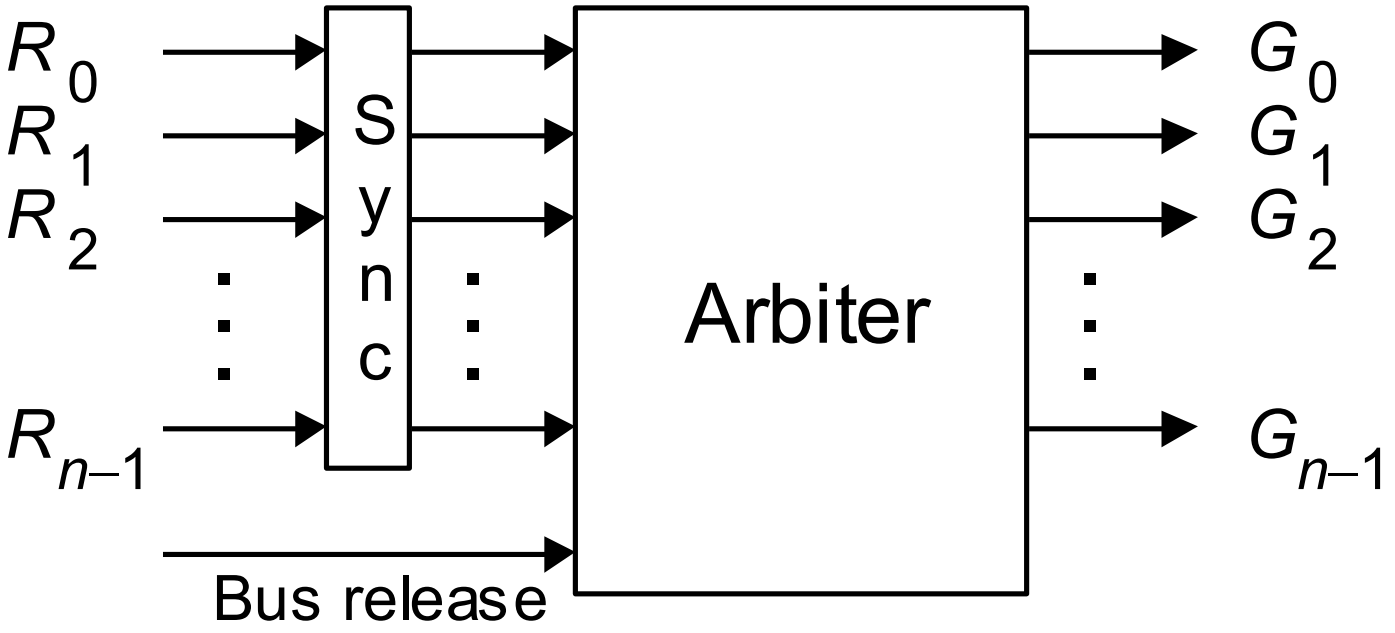
Handshaking pe o magistrală asincronă pentru o operație de date (e.g., o citire din memorie).

Exemplu de comunicație pe bus



Operație I/O read pe un bus PCI.

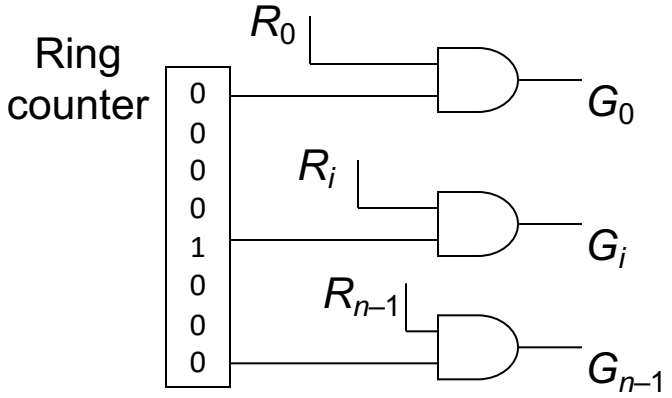
Arbitrarea accesului la magistrala de date



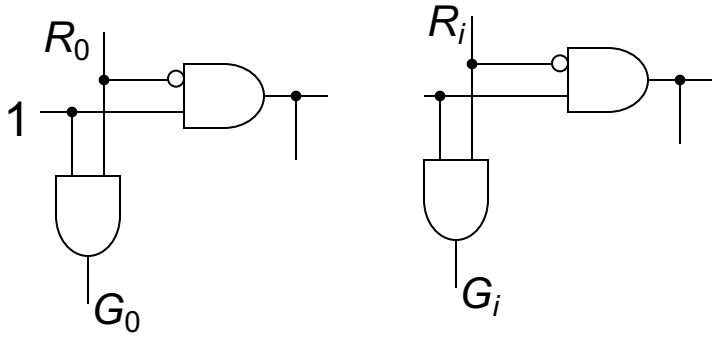
Structura generalizată a unui arbitru centralizat de magistrală.

Implementări simple de arbitri

Round robin



Fixed-priority



Starvation avoidance

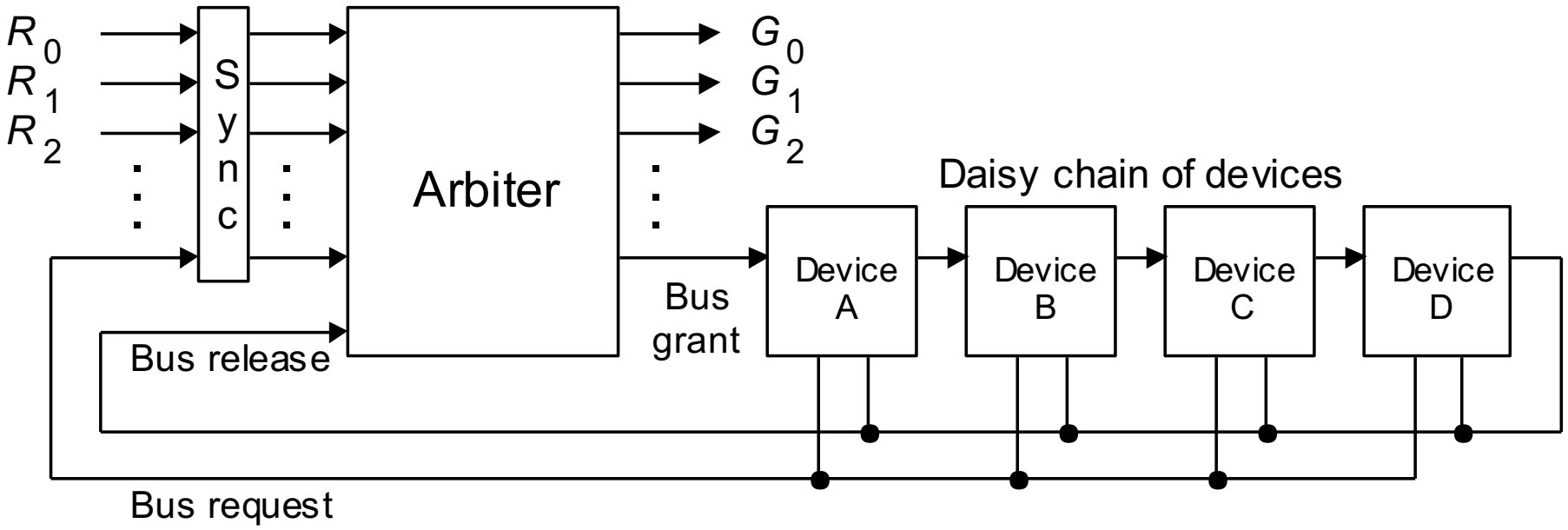
Dacă avem priorități fixe, unitățile cu prioritate mică pot să nu aibă niciodată acces la bus (pot să ”moară de foame”)

Prioritatea trebuie combinată cu un mecanism de asigurare a serviciilor

Rotating priority

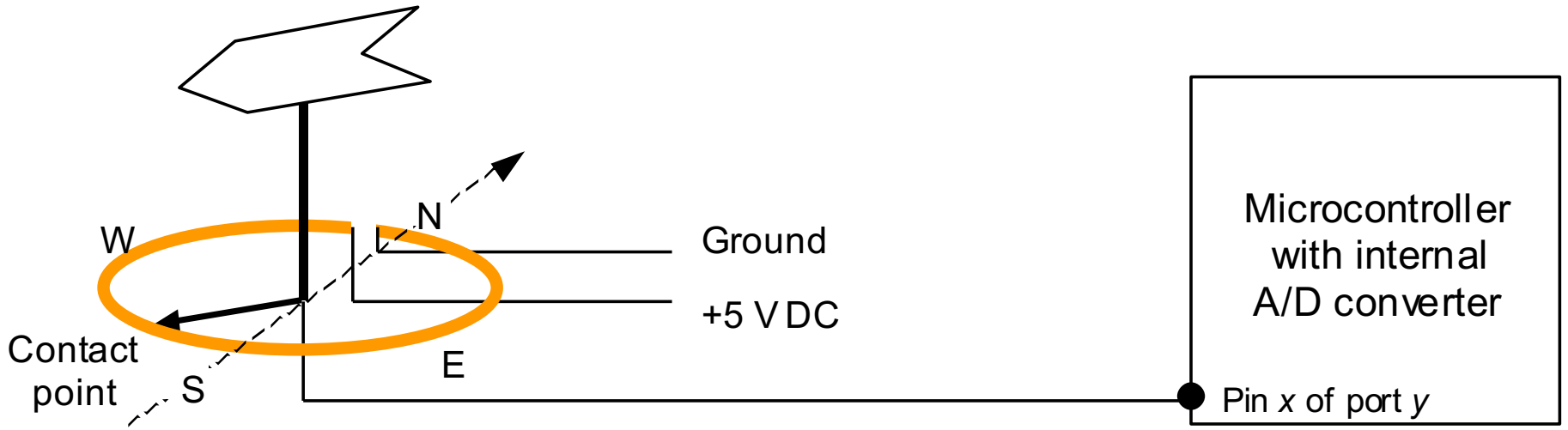
Idee: Ordonăm circular unitățile și permitem ca prioritatea cea mai mare să se ”rotească” între unități (combinăm un numărător circular cu un circuit de acordare a priorității)

Daisy Chaining



Daisy chaining permite unui arbitru de bus central să acceseze un număr mare de dispozitive care folosesc o resursă partajată.

Elementele de bază ale interfațării



Giruetă care dă o tensiune de ieșire de 0-5V în funcție de direcția vântului.

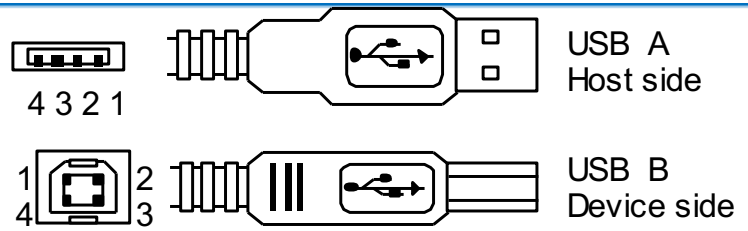
Standarde de interfațare

Sumar a patru dintre cele mai folosite standarde de interfațare

Attributes ↓ Name →	PCI	SCSI	FireWire	USB
Type of bus	Backplane	Parallel I/O	Serial I/O	Serial I/O
Standard designation	PCI	ANSI X3.131	IEEE 1394	USB 2.0
Typical application domain	System	Fast I/O	Fast I/O	Low-cost I/O
Bus width (data bits)	32-64	8-32	2	1
Peak bandwidth (MB/s)	133-512	5-40	12.5-50	0.2-15
Maximum number of devices	1024*	7-31#	63	127$
Maximum span (m)	< 1	3-25	4.5-72$	5-30$
Arbitration method	Centralized	Self-select	Distributed	Daisy chain
Transceiver complexity or cost	High	Medium	Medium	Low

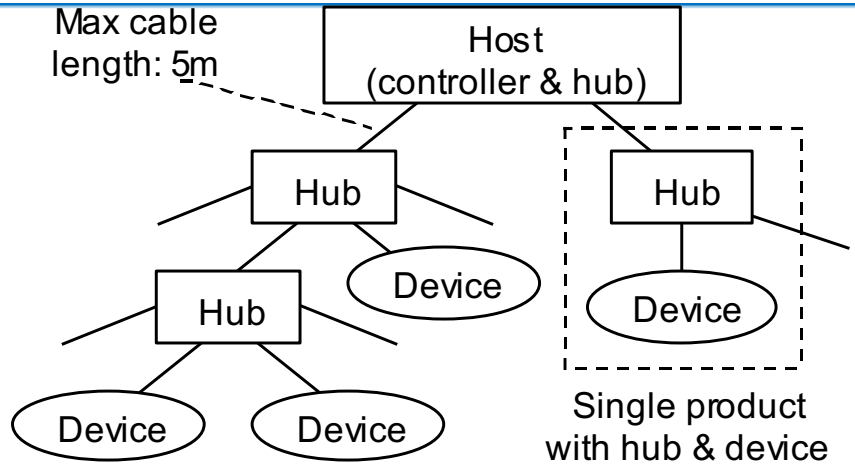
Notes: * 32 per bus segment; # One less than bus width; $ With hubs (repeaters)

Conectori standard

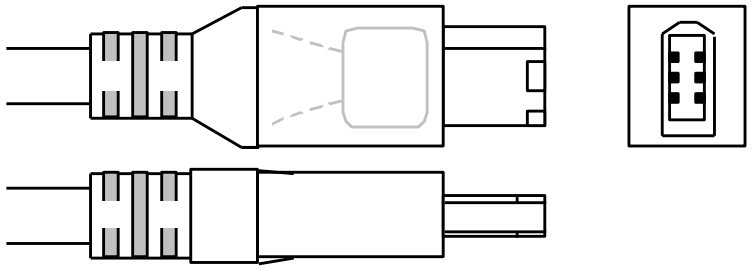


Pin 1: +5V DC
Pin 4: Ground

Pin 2: Data –
Pin 3: Data +



Conectori USB și structura de conectivitate.



Pin 1: 8-40V DC, 1.5 A
Pin 2: Ground
Pin 3: Twisted pair B –
Pin 4: Twisted pair B +
Pin 5: Twisted pair A –
Pin 6: Twisted pair A +
Shell: Outer shield

Conector IEEE 1394 (FireWire). Același conector este folosit la ambele capete.

Acknowledgements

- These slides contain material developed and copyright by:
  - Arvind (MIT)
  - Krste Asanovic (MIT/UCB)
  - Joel Emer (Intel/MIT)
  - James Hoe (CMU)
  - John Kubiatowicz (UCB)
  - David Patterson (UCB)

- MIT material derived from course 6.823
- UCB material derived from course CS252