



**UNIVERSITATEA  
TEHNICĂ**  
DIN CLUJ-NAPOCA

**Facultatea de Construcții de Mașini  
Departamentul Ingineria Proiectării și Robotică**

# **Sisteme Flexible de Fabricatie**

**Sl. Dr. Ing. Florin COVACIU**

**2020**

## Introducere

- Sistemele flexibile de fabricație sunt utilizate adesea în producția de masă din mai multe motive.
- Un sistem de fabricație flexibil este un bun de fabricație care se poate adapta cerințelor în schimbare. Acest sistem permite producerea a mai mult de o piesă sau o familie de piese, după cum este necesar. În industria de fabricație există mai multe sisteme de fabricație flexibile, fiecare având diverse aplicații.



## Introducere

- Globalizarea, cerintele pietei si un stil de viata modern au creat o provocare extraordinara pentru industriile de productie.
- In scenariul de afaceri actual, competitivitatea oricarei industrii producatoare este determinata de capacitatea sa de a reacționa rapid la schimbarile de pe piata si de a produce produse de înalta calitate la costuri reduse.
- La mijlocul anilor 60, concurenta pe piata a devenit mai intensa.
- In perioada anilor 1960-1970 principala preocupare a fost costul, iar mai tarziu calitatea devine prioritara. Pe masura ce piata a devenit din ce in ce mai complexa, viteza de livrare a devenit ceva ce isi doreste clientul.



## Scurt istoric

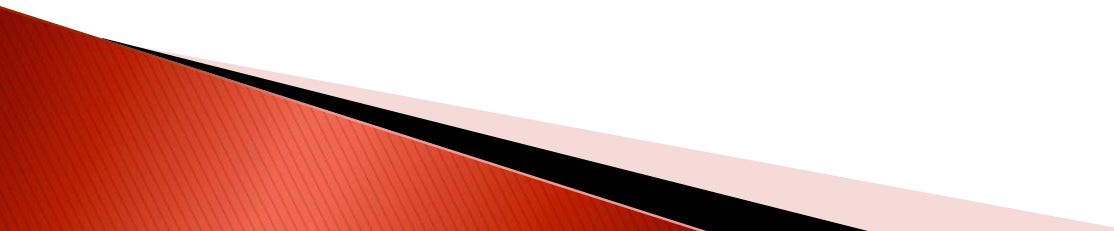
- După al doilea război mondial a apărut o nouă eră în fabricarea de materiale noi și tehnici de producție care au crescut calitatea și producția.
- Primul sistem flexibil de fabricație a fost patentat de către Theo Williamson în anul 1965 constând dintr-un echipament controlat numeric. Exemple de echipamente controlate numeric sunt strungurile și frezele care sunt numite diferite tipuri de sisteme flexibile de fabricație.
- Schimbarea producției în timp.
- În anii 60 - Eficiența în producție.
- În anii 70 - Eficiența în calitate.
- În anii 80 - Eficiența în calitate și flexibilitate.
- Datorită:
  - Creșterii competiției internaționale;
  - Nevoii de reducere a ciclului de timp în producție;
  - Presiunii pentru reducerea costului de producție.



## Sistem flexibil de fabricatie (SFF)

- Ca o prima definitie SFF este o tehnologie de fabricatie.
- Ca o a doua definitie SFF este o filozofie. Cuvantul cheie este **sistem**. SFF include o vedere a sistemului de fabricatie. Cuvantul cheie pentru producatorul de astazi este **agilitate**. Un producator agil este cel mai rapid de pe piata, opereaza cu cel mai mic cost total si are cea mai mare capacitate de a **incanta** clientii. FSS este o modalitate prin care producatorii sunt capabili sa atinga aceasta agilitate.

## Componente de baza ale unui SFF

- Statii de lucru
  - Sistem automat de manipulare si depozitare a materialelor
  - Sistem de control computerizat
  - Resurse umane
- 

## Statii de lucru

- Stațiile de lucru care se gasesc de obicei intr-un SFF sunt urmatoarele:
  - Statii de incarcare/descarcare;
  - Statii de prelucrare;
  - Alte statii de procesare;
  - Statie de asamblare;
  - Alte statii si echipamente.

## Sistem de manipulare a materialelor

- Funcții:
  - Mișcarea independentă a punctelor de lucru între stații;
  - Gestionează o varietate de configurații ale pieselor de lucru;
  - Acces convenabil pentru încărcarea și descărcarea pieselor de lucru;
  - Compatibil cu controlul calculatorului.

## Echipamente de manipulare a materialelor

- Sistemul de manipulare primară stabilește aspectul de bază al unui SFF și este responsabil pentru mutarea pieselor de lucru între stațiile din sistem (Conveyor).
- Sistemul de manipulare secundară constă din dispozitive de transfer, schimbarea automată a paletelor și mecanisme similare amplasate la stațiile de lucru dintr-un SFF.

Funcția sistemului de manipulare secundară este de a transfera lucrările de la sistemul primar la mașina-unelte sau de la altă stație de procesare și de a poziționa piesele cu suficientă precizie și repetabilitate pentru a efectua procesul sau operația de asamblare.



## Control prin controler logic programabil (PLC)

- SFF include un sistem computerizat distribuit care este interfatat la:
  - Statii de lucru;
  - Sistem de manipulare a materialelor;
  - Alte componente hardware.
- Un sistem informatic tipic al unui SFF este format dintr-un calculator central și PLC-uri.
  - PLC-urile controlează utilajele individuale si alte componente;
  - Calculatorul central coordoneaza activitatile componentelor pentru a realiza o functionare generala lina a sistemului.

## Control prin controler logic programabil (PLC)

### PLC - Prezentare generală a sistemului

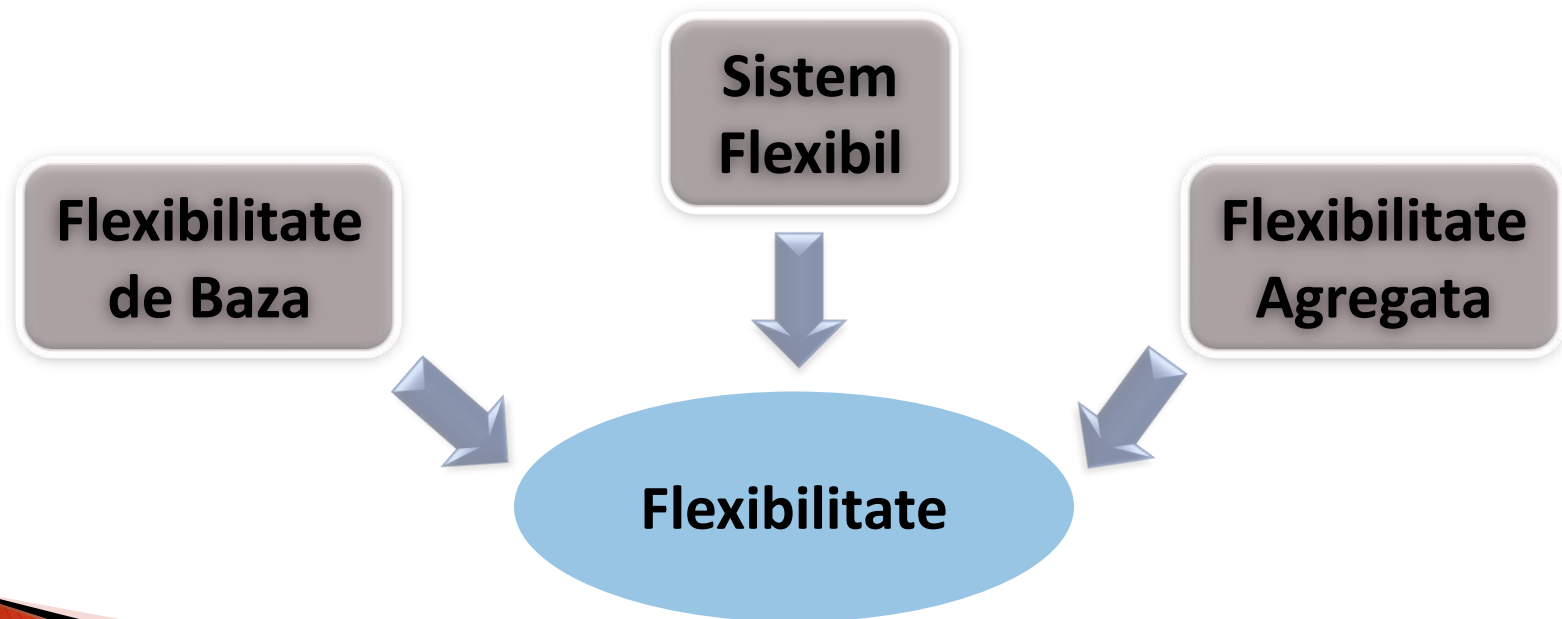


## Resurse umane

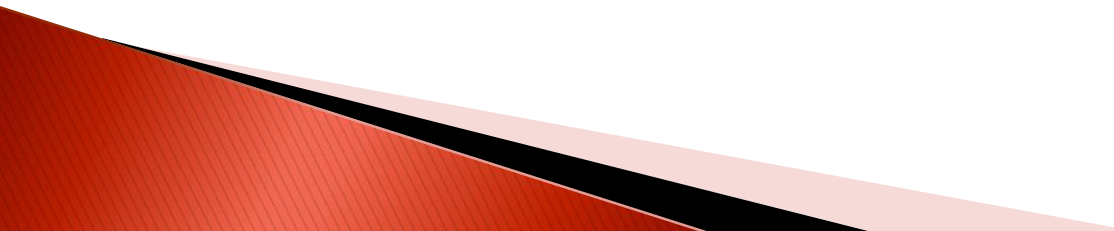
- Pentru a gestiona operațiunile din SFF sunt necesari oameni. Funcțiile care sunt îndeplinite de obicei de către oameni includ:
  - Incărcarea pieselor de lucru brute in sistem;
  - Descărcarea pieselor (sau ansamblurilor) terminate din sistem;
  - Schimbarea si setarea sculelor;
  - Intreținerea si repararea echipamentelor;
  - Programarea in codul G pentru a crea piese din semifabricat;
  - Programarea și operarea sistemului informatic.

## Flexibilitate

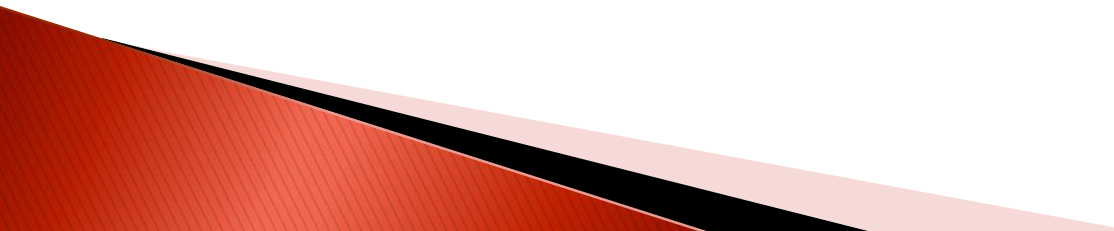
Pentru un sistem mixt de fabricatie flexibilitatea este un atribut care permite sa faca fata unui anumit nivel de variatii ale partii sau a stilului de produs, fara a avea intreruperi in productie din cauza schimbarilor intre module.



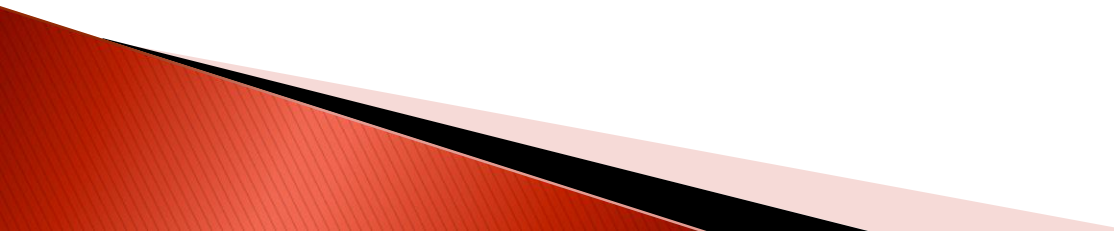
## Flexibilitate de baza

- Flexibilitatea masinii – usurinta cu care o masina poate face diverse operatii.
  - Flexibilitatea manipularii materialelor – un procedeu a usurintei cu care tipuri de piese diferite pot fi transportate si pozionate corect la diferite masini-unelte dintr-un sistem.
  - Flexibilitatea operatiei – un procedeu a usurintei cu care secvente de operare alternative pot fi utilizate pentru procesarea a unui tip de piesa.
- 

## Flexibilitatile sistemului

- Flexibilitatea volumului – capacitatea sistemului de a opera profitabil la diferite volume ale tipurilor de piese existente.
  - Flexibilitate de rutare – o masura a cailor alternative pe care o piesa le poate urmări eficient printr-un sistem pentru un plan de process dat.
  - Flexibilitatea procesului – o masura a volumului setului de tipuri de piese pe care un sistem le poate produce fara a suporta nici o configuratie.
  - Flexibilitatea produsului - volumul setului de tipuri de piese care pot fi fabricate intr-un sistem cu configurare minora.
- 

## Flexibilitati agregate

- Flexibilitatea programului – capacitatea unui sistem de a rula pe perioade rezonabil de lungi, fara interventie externa.
  - Flexibilitatea productiei – volumul setului de tipuri de piese pe care un sistem le poate produce fără investiții majore în echipamente de capital.
  - Flexibilitatea pietei – capacitatea unui sistem de a se adapta eficient la schimbarea condițiilor pieței.
- 

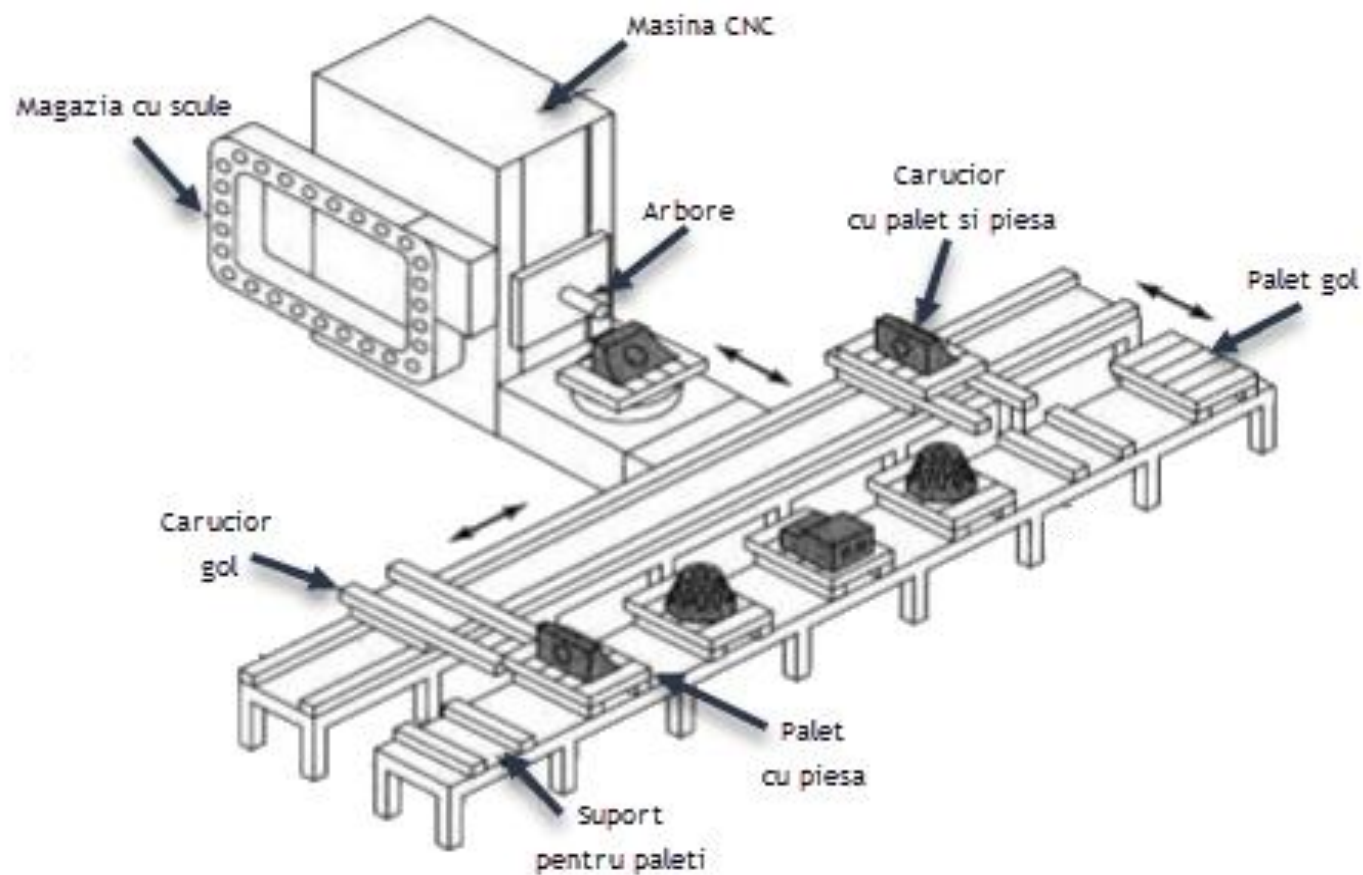
## Tipuri de SFF

### 1. Celula cu o singura masina

- Contine o masina (adesea un centru de prelucrare CNC) conectata la un sistem de stocare a pieselor, care poate incarca si descarca piese catre si din sistemul de stocare.
- Este proiectata pentru a funcționa în mod serie, mod flexibil sau o combinație a celor doua. Cand se afla in modul serie, sistemul prelucrează piese intr-un singur stil în dimensiuni de serie specifice înainte de trecerea fizică și programată la următoare schimbare de serie.
- In modul flexibil, sistemul satisface trei dintre cele patru teste pentru flexibilitate – exceptie fiind recuperarea erorilor, intrucat, daca centrul de prelucrare CNC se strica, sistemul se opreste.



## Celula cu o singura masina CNC

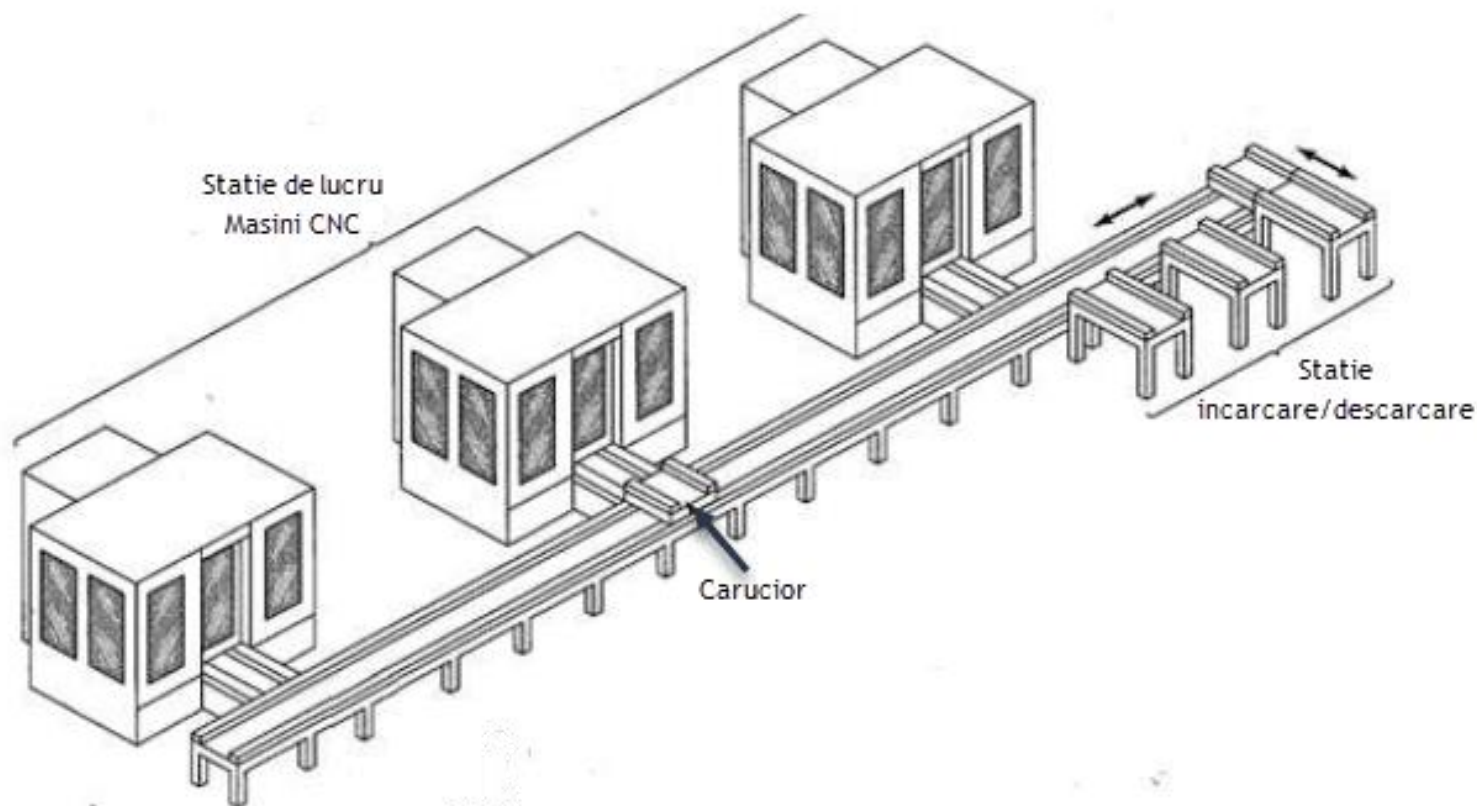


## Tipuri de SFF

### 2. Celula flexibila de fabricatie

- Contine doua sau trei statii de lucru (adesea masini CNC, freze sau strunguri), plus un sistem de manipulare al pieselor. Aceasta configurare poate functiona in mod flexibil sau in mod serie, dupa cum este necesar si se poate adapta cu usurinta la programul de productie si evolutie si la volume de productie crescute.
- Deoarece exista mai mult de o masina, recuperarea erorilor este posibila prin redirectionarea pieselor intentionate ale masinii esuate pentru procesare catre celelalte doua masini din sistem. Mai pot fi introduse noi modele de piese cu o usurinta relativa in configurare.

## Celula flexibila de fabricatie

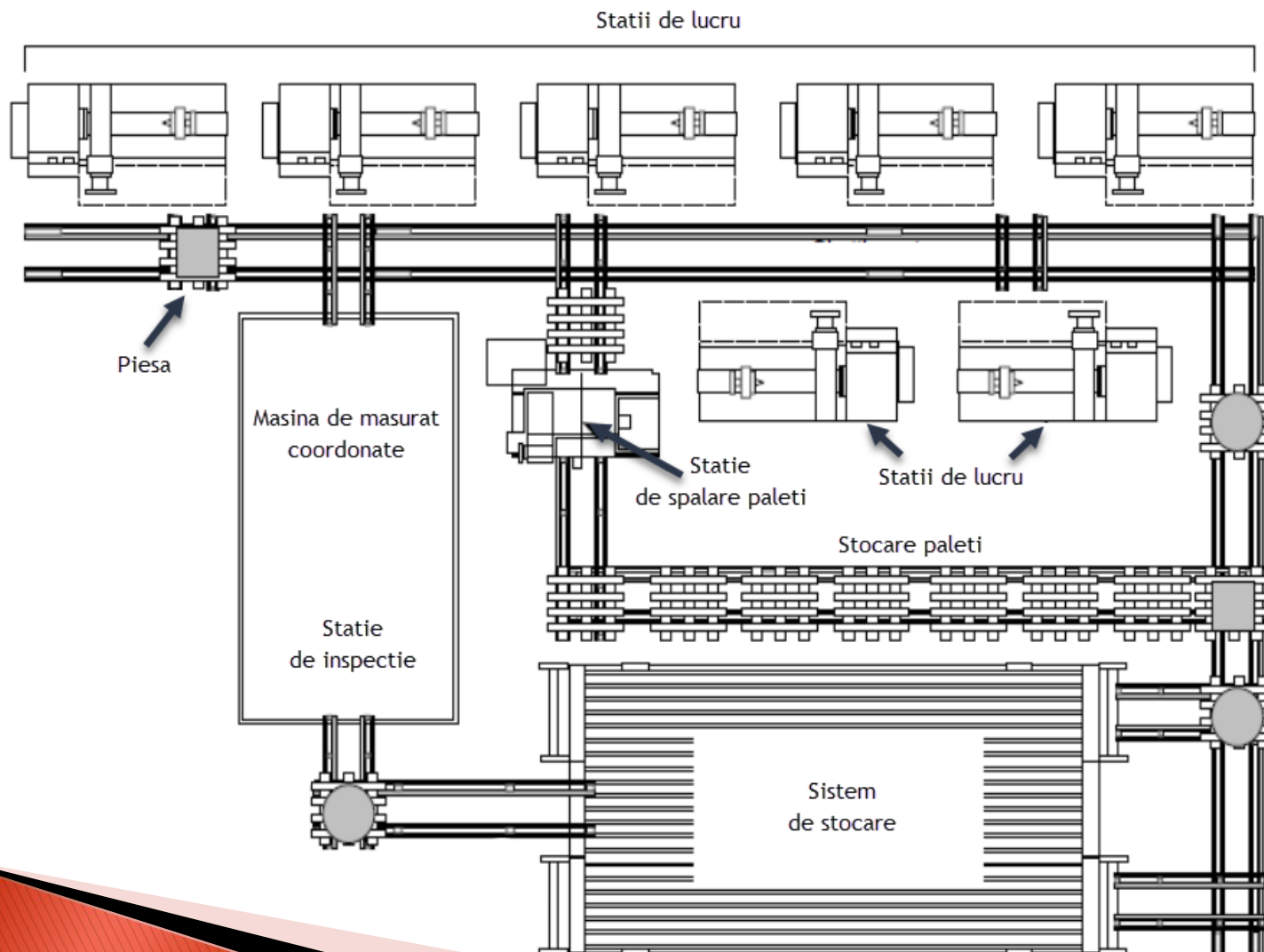


## Tipuri de SFF

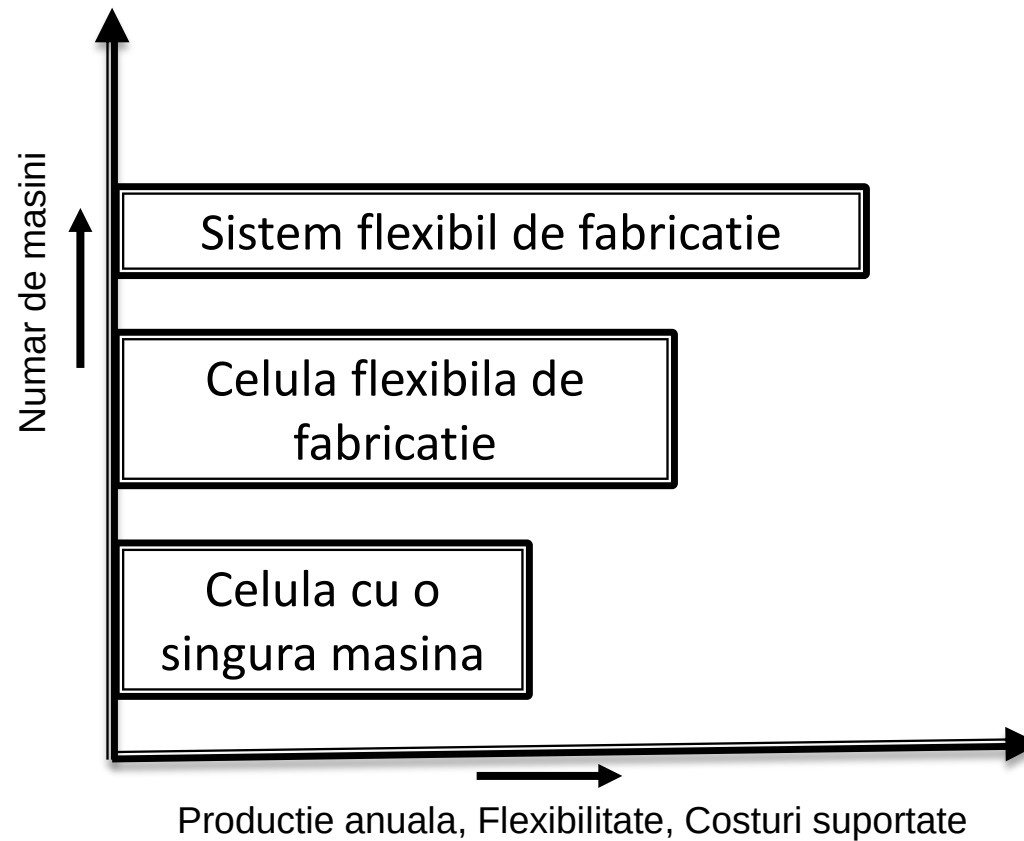
### 3. Sistem flexibil de fabricație

- Este format din patru sau mai multe stații de procesare conectate mecanic de un sistem comun de manipulare a pieselor de un sistem comun de manipulare a pieselor și electronic de către un sistem informatic distribuit. Acest sistem este mai mare decât celula flexibilă de fabricație, nu numai în numărul de stații de lucru pe care le poate conține, dar și în numărul de stații de susținere din sistem. Calculatorul folosit în control este mai sofisticat, include funcții care nu se regăsesc în celula de fabricație flexibilă, cum ar fi monitorizarea sculelor.

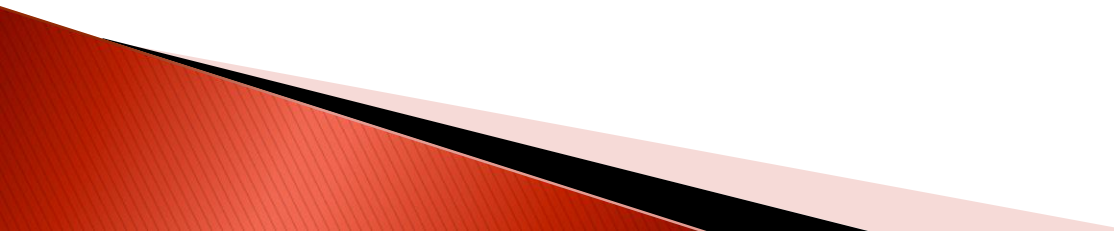
## Sistem flexibil de fabricatie



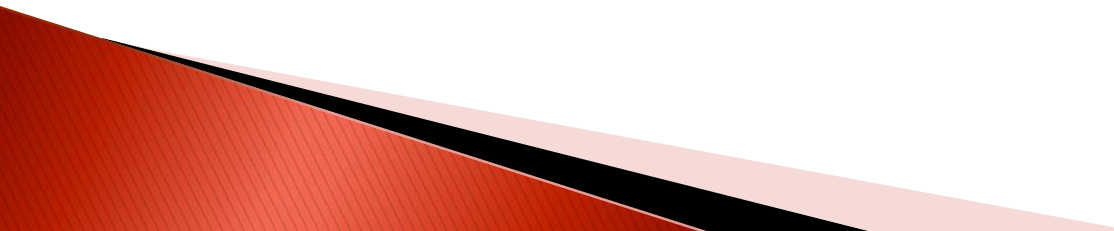
## Tipuri de SFF



## Tipuri de SFF

- SFF secvential;
  - SFF aleatoriu;
  - SFF dedicat;
  - SFF proiectat.
- 

## Aspecte ale unui SFF

- Progresiv sau tip linie;
  - Tip bucla;
  - Tip dispunere dreptunghiulara;
  - Tip scara;
  - Tip camp deschis;
  - Tip centrat pe robot.
- 

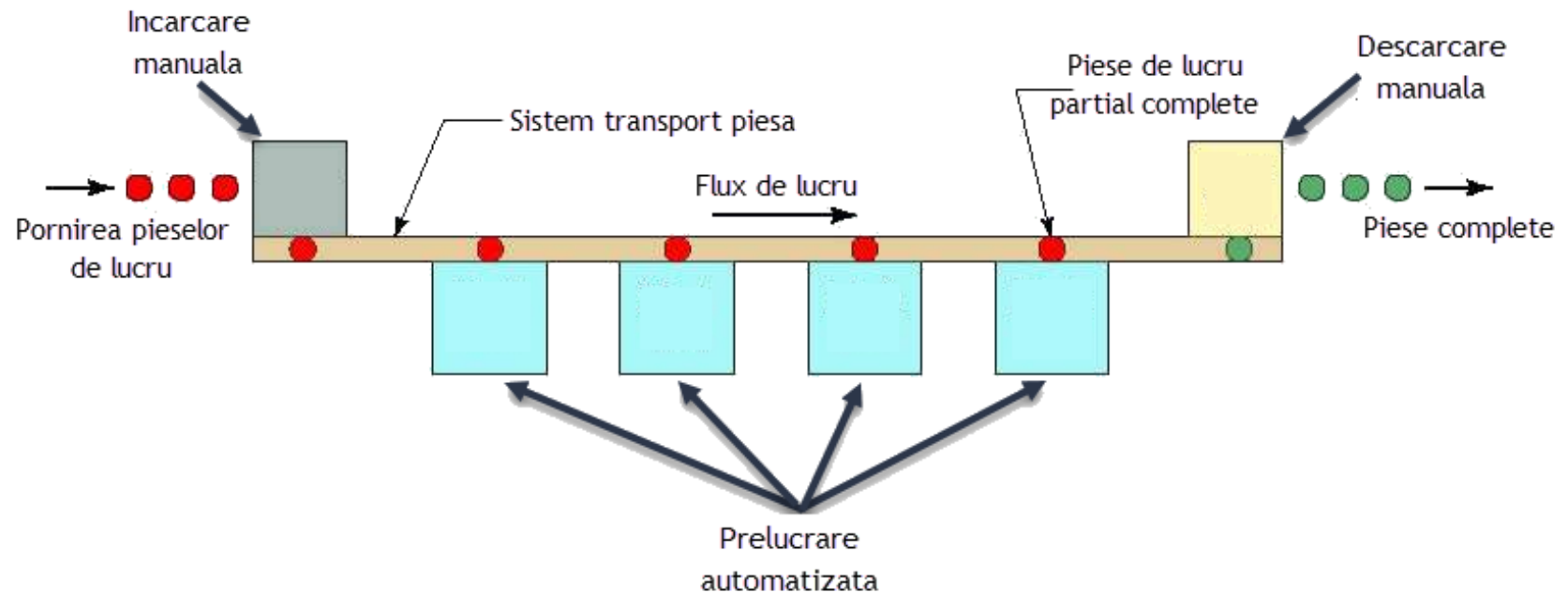


## Progresiv sau tip linie

- Mașinile și sistemul de manipulare sunt aranjate în linie dreaptă. Deplasarea pieselor se face de la o stație de lucru la alta stație într-o secvență bine definită, într-o singură direcție și fără un flux de întoarcere.
  
- Echipamentele de manipulare a materialelor utilizate sunt următoarele:
  - Sisteme de transfer în linie;
  - Sistem transportor (conveyor);
  - Sistem de transport ghidat pe sine.

# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

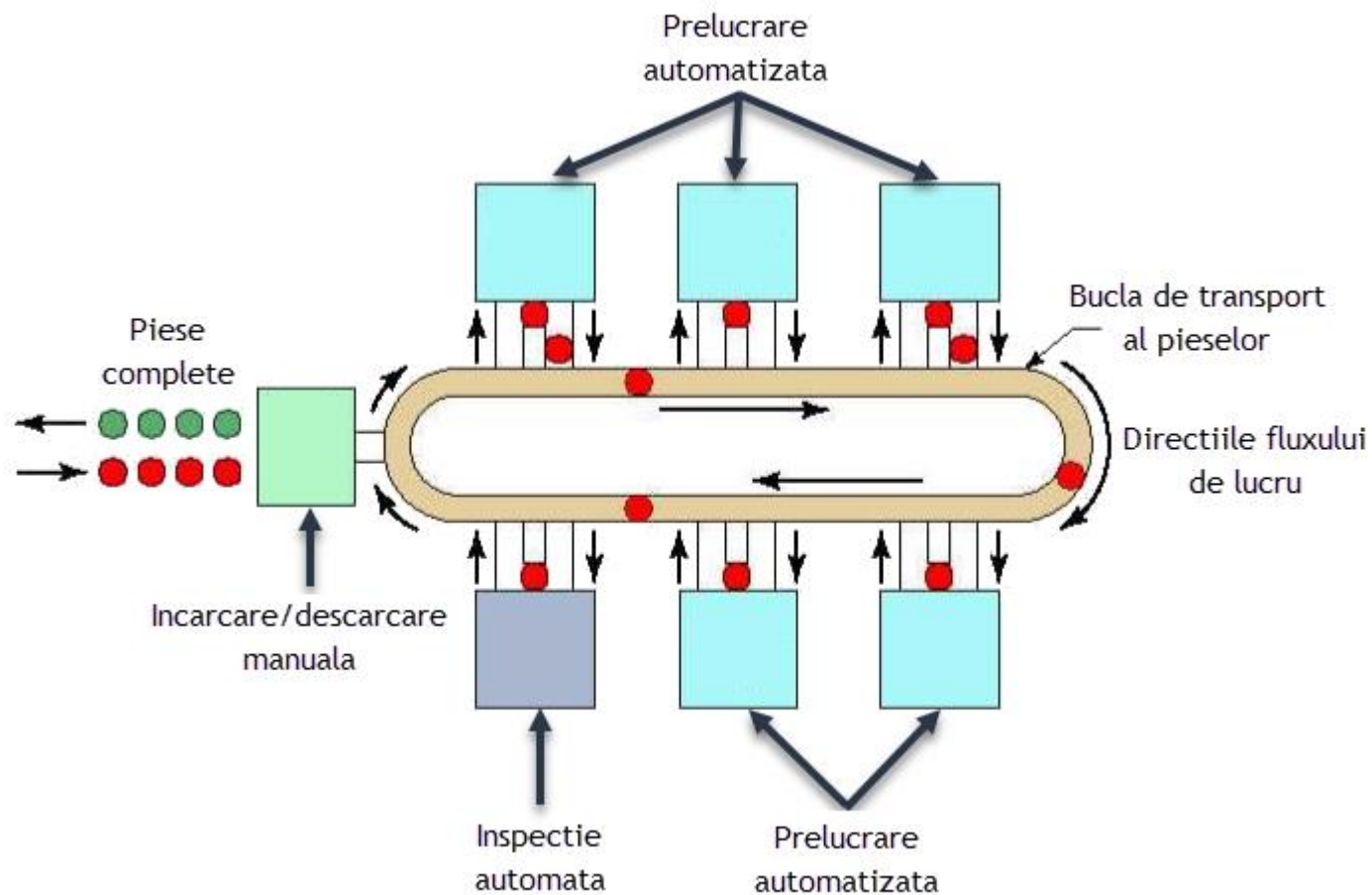
## Progresiv sau tip linie



## Tip bucla

- Posturile de lucru sunt organizate într-o buclă care este deservită de un sistem de manipulare a pieselor în buclă. Fiecare stație are echipamente de manipulare secundare, astfel încât piesa să poată fi adusă și transferată - de la capul de lucru al stației la buclă de manipulare a materialelor.
- Stațiile de încărcare/descărcare sunt de obicei localizate la un capăt al buclei. Acest aranjament permite revenirea paletelor în poziția de pornire într-un aranjament liniar.

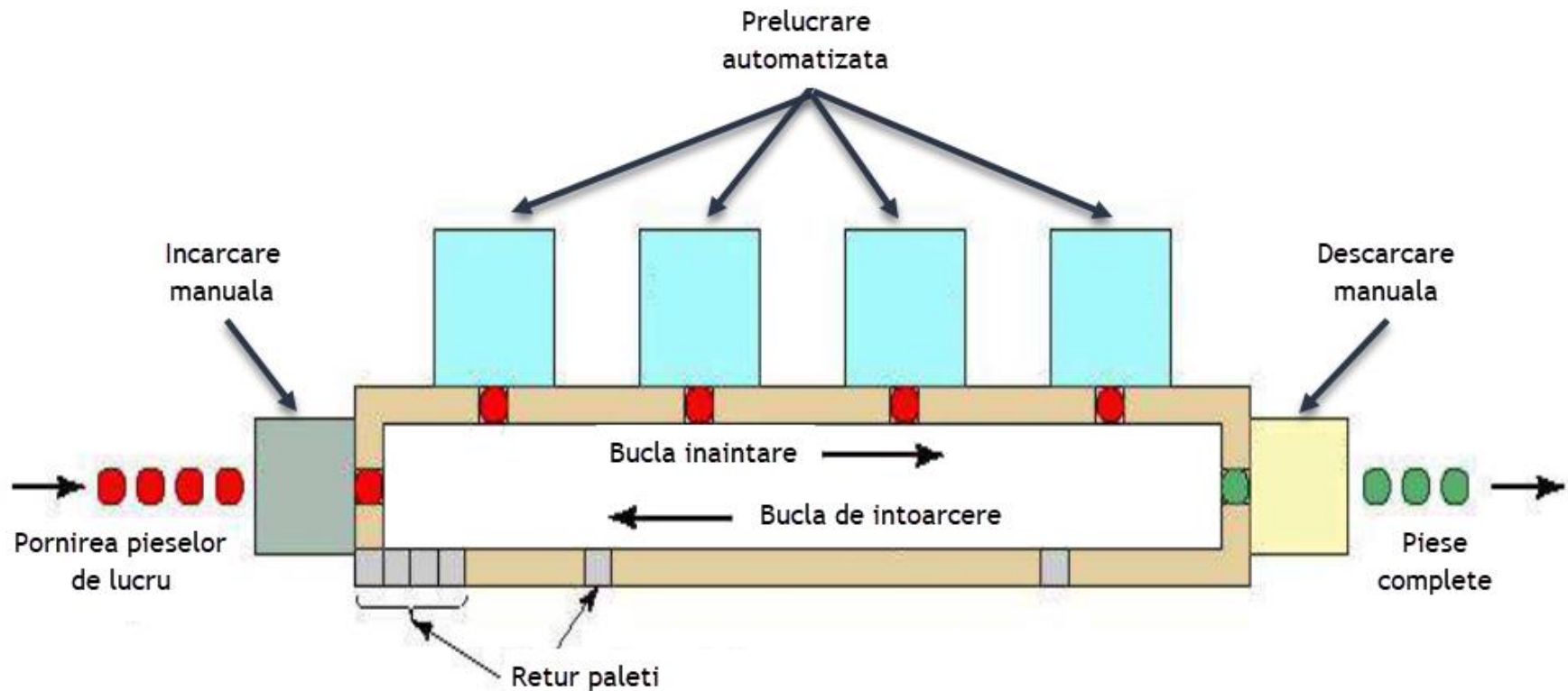
## Tip bucla



## Tip dispunere dreptunghiulara

- Dispunerea dreptunghiulară permite recircularea paletelor inapoi la prima statie in secvență dupa descarcare la statia finala.

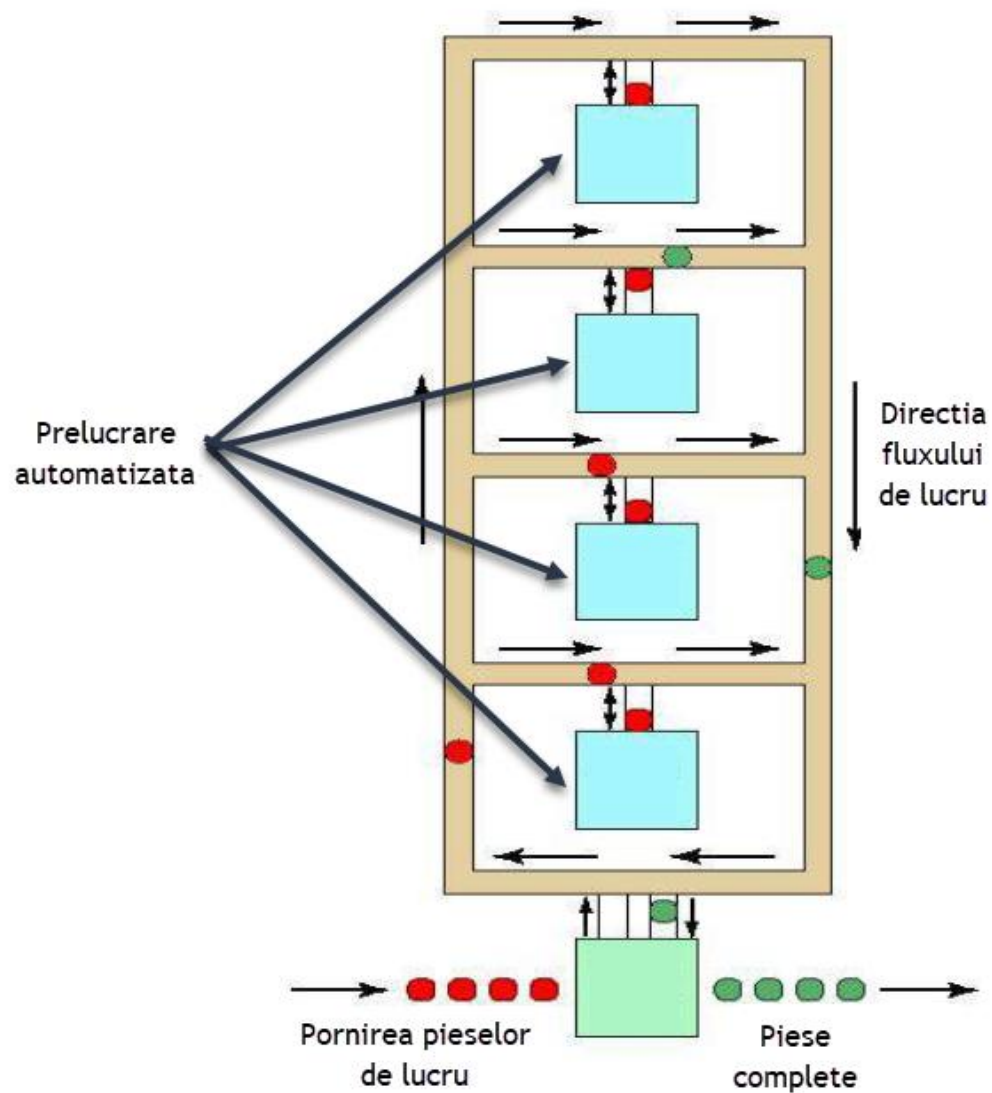
## Tip dispunere dreptunghiulara



## Tip scara

- Aceasta constă dintr-o buclă cu trepte pe care sunt amplasate stațiile de lucru. Treptele cresc numărul de moduri posibile de a trece de la o mașină la alta. Scările cresc numărul de moduri posibile trecând de la o mașină la alta și evita nevoia unui sistem de manipulare a materialelor secundare. Acest lucru reduce distanța medie de deplasare și minimizează congestia în sistemul de manipulare, reducând astfel timpul de transport între stații.

## Tip scara



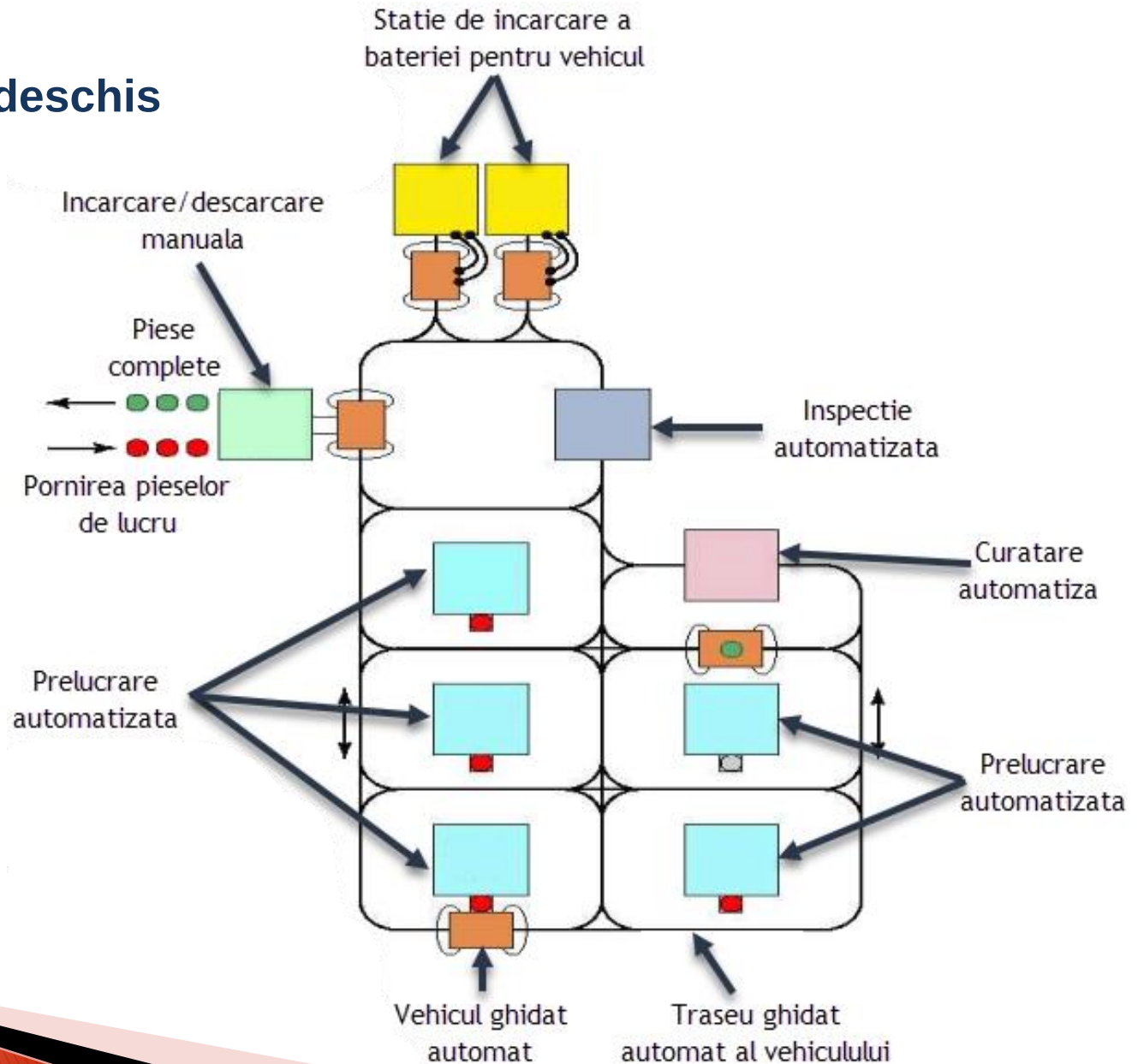


## Tip camp deschis

- Consta in mai multe bucle si scari, si poate include, de asemenea, linii secundare. Acest aspect este utilizat în general pentru a prelucra o familie numeroasă de piese, deși numărul diferitelor tipuri de mașini poate fi limitat, iar piesele sunt, de obicei, direcționate către stații de lucru diferite - în funcție de care stație devine disponibilă mai întâi.

# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Tip camp deschis

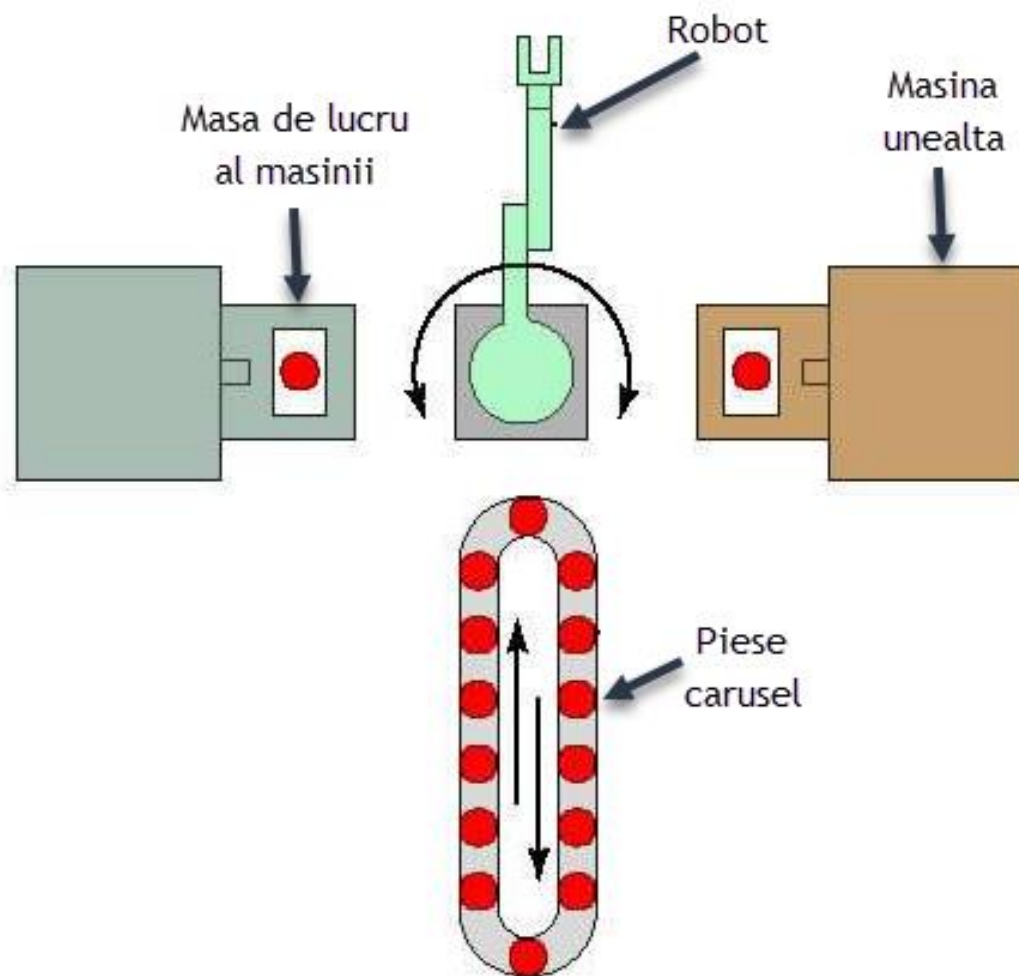


## Tip centrat pe robot

- Constă în mai multe bucle și scări, și poate include, de asemenea, linii secundare. Acest aspect este utilizat în general pentru a prelucra o familie numeroasă de piese, deși numărul diferitelor tipuri de mașini poate fi limitat, iar piesele sunt, de obicei, direcționate către stații de lucru diferite - în funcție de care stație devine disponibilă mai întâi.

# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Tip centrat pe robot



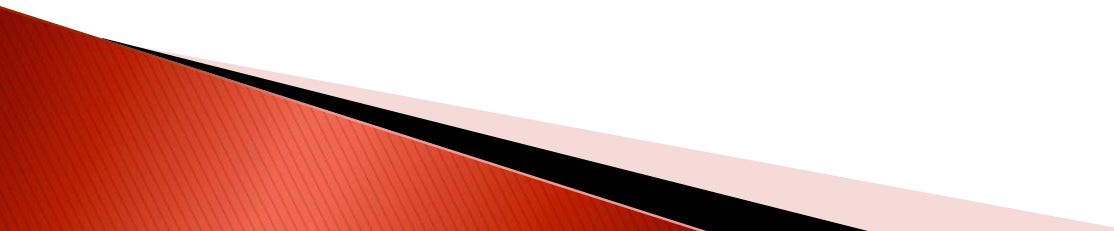


# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

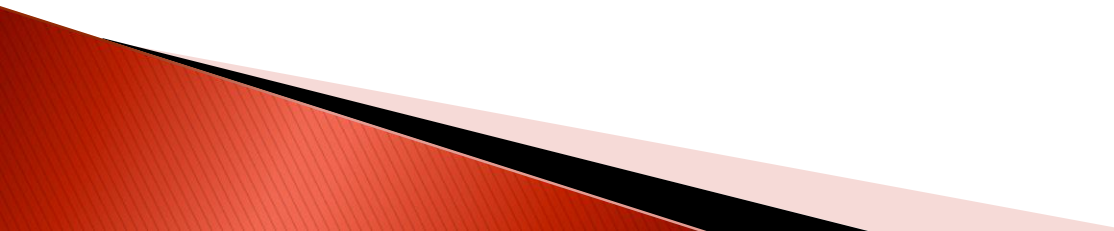
## Film de prezentare



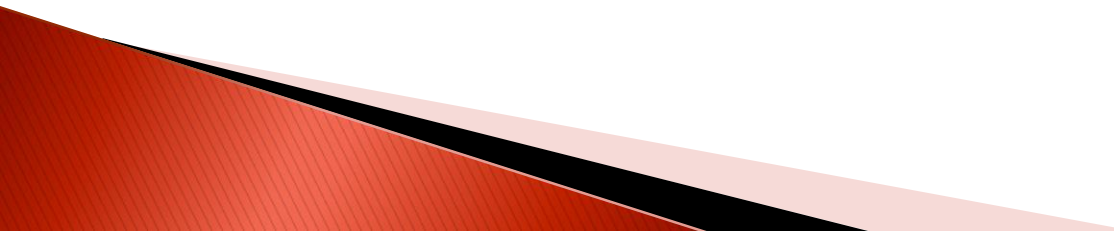
## Avantajele unui sistem flexibil de fabricatie

- Reducerea timpului pentru finalizarea produsului;
  - Utilizarea mai buna a muncitorilor;
  - Servirea unor varietati de furnizori simultan;
  - Imbunatatirea rutarii produsului;
  - Imbunătățirea calitatii produselor;
  - Pentru a reduce timpul de configurare si al cozii;
  - Producerea unor varietati de articole sub un singur acoperis;
  - Produce intr-un timp mai scurt mai multe produse;
  - Imbunatatirea eficientei;
  - Adaptabilitate la operatiile CAD/CAM;
  - Costuri de productie reduce;
  - Cost mai mic pentru unitatea produsa.
- 

## Dezavantajele unui sistem flexibil de fabricatie

- Capacitate limitata de adaptare la modificarile produsului sau mixului de produse;
  - Activitate de pre-planificare substantiala;
  - Costa pentru implementare;
  - Cerinta de munca calificata;
  - Sistem complicat;
  - Probleme tehnologice de pozitionare exacta a componentelor si sincronizare precisa necesare procesarii unei componente.
- 

## Aplicații ale unui sistem flexibil de fabricație

- Asamblare;
  - Imbinare prin sudura;
  - Prelucrarea taierii metalelor;
  - Formarea de metale;
  - Tratament de suprafață;
  - Inspectia;
  - Testarea.
- 





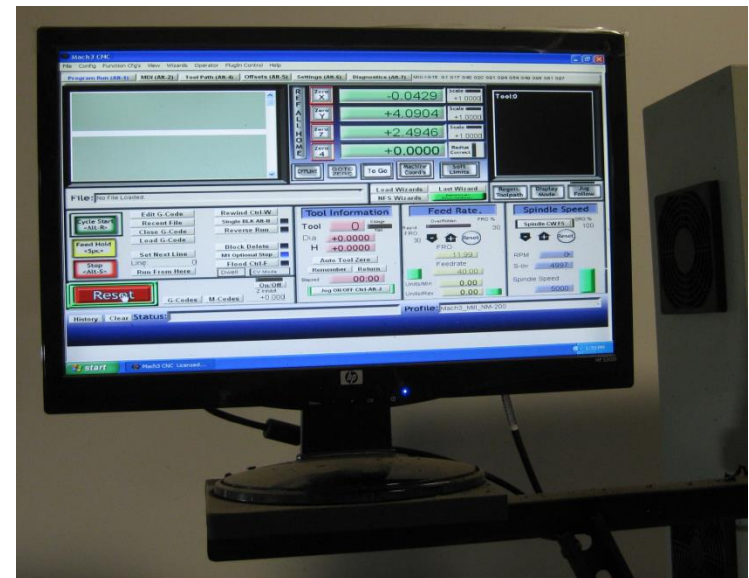
## George Boole (1815–1864)



| A | B | A • B |
|---|---|-------|
| 0 | 0 | 0     |
| 0 | 1 | 0     |
| 1 | 0 | 0     |
| 1 | 1 | 1     |

| A | B | A + B |
|---|---|-------|
| 0 | 0 | 0     |
| 0 | 1 | 1     |
| 1 | 0 | 1     |
| 1 | 1 | 1     |





**UNIVERSITATEA  
TEHNICĂ**  
DIN CLUJ-NAPOCA

Facultatea de Construcții de Mașini  
Departamentul Ingineria Proiectării și Robotică

# Elemente de Algebra Booleana utilizate in SFF

George Boole (1815–1864)



| A | B | $A \cdot B$ |
|---|---|-------------|
| 0 | 0 | 0           |
| 0 | 1 | 0           |
| 1 | 0 | 0           |
| 1 | 1 | 1           |

| A | B | $A + B$ |
|---|---|---------|
| 0 | 0 | 0       |
| 0 | 1 | 1       |
| 1 | 0 | 1       |
| 1 | 1 | 1       |

Sl. Dr. Ing. Florin COVACIU

## Algebra si logica Booleana – Sistemul binar

- Algebra si logica booleana este bazata pe o informatie digitala, iar unitatea fundamentala pentru reprezentarea informatiei este discreta (nu poate lua orice valoare).
- Se foloseste reprezentarea binara a informatiei, unitatea de baza a informatiei este bit-ul si poate avea 2 valori (0 sau 1).
- Conventional se atribuie valoarea “1” prezentei potentialului electric si “0” absentei acestuia.
- O secventa de 8 biti se numeste byte.
- Un byte reprezinta  $2^8 = 256$  putem reprezenta 256 valori: 256 caractere, numerele de la 1 ÷ 256.
  - 1 kilobyte =  $2^{10}$  byte = 1024 byte

## Algebra si logica Booleana – Sistemul binar

- Sistemul binar (baza 2) este alcatuit din 2 cifre (0, 1).

Ex. calcul:

$$10101 = 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3 + 1 \times 2^4$$

$$10101_{(2)} = 21_{(10)}$$

| Zecimal | Binar | Suma argumentilor lui 2                                     |
|---------|-------|-------------------------------------------------------------|
| 1       | 1     | 1                                                           |
| 2       | 10    | $1 \times 2^0 + 0 \times 2^1$                               |
| 3       | 11    | $1 \times 2^0 + 1 \times 2^1$                               |
| 4       | 100   | $1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2$                |
| 5       | 101   | $1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2$                |
| 6       | 110   | $1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2$                |
| 7       | 111   | $1 \times 2^0 + 1 \times 2^1 + 1 \times 2^2$                |
| 8       | 1000  | $1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 0 \times 2^3$ |
| 9       | 1001  | $1 \times 2^0 + 0 \times 2^1 + 0 \times 2^2 + 1 \times 2^3$ |
| 10      | 1010  | $1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 0 \times 2^3$ |

## Aritmetica binara

- Deși **sistemul binar** este cel mai utilizat pentru stocarea și prelucrarea datelor, transmiterea informației se face greu din cauza numărului mare de cifre necesare pentru a exprima chiar și valori relative mici, iar din acest motiv se folosesc în mod curent sisteme de numeratie cu baze mai mari, dar care permit trecerea imediată în binar, condiție a oricărui sistem la care baza este o putere naturală a lui 2.
- Cele mai utilizate sisteme, din acest punct de vedere, sunt: **sistemul octal** (cu baza 8) și **sistemul hexazecimal** - pe scurt "hexa"- (cu baza 16). Pentru că  $8 = 2^3$  și  $16 = 2^4$ , orice cifră octală se poate reprezenta printr-o combinație de 3 cifre binare și orice cifră hexa, printr-o combinație de 4 cifre binare.

## Aritmetica binara

- Sistemul octal utilizează cifrele 0,1,2,3,4,5,6,7. Pentru a reprezenta cele 16 cifre ale sistemului hexazecimal sunt necesare 16 semne grafice distincte: primele 10 dintre acestea coincid cu cifrele sistemului zecimal, iar pentru restul de 6 se folosesc primele litere ale alfabetului latin (A,B,C,D,E,F)

| Binar | Sistemul |      |         |
|-------|----------|------|---------|
|       | Octal    | Hexa | Zecimal |
| 0000  | 0        | 0    | 0       |
| 0001  | 1        | 1    | 1       |
| 0010  | 2        | 2    | 2       |
| 0011  | 3        | 3    | 3       |
| 0100  | 4        | 4    | 4       |
| 0101  | 5        | 5    | 5       |
| 0110  | 6        | 6    | 6       |
| 0111  | 7        | 7    | 7       |
| 1000  | 10       | 8    | 8       |
| 1001  | 11       | 9    | 9       |
| 1010  | 12       | A    | 10      |
| 1011  | 13       | B    | 11      |
| 1100  | 14       | C    | 12      |
| 1101  | 15       | D    | 13      |
| 1110  | 16       | E    | 14      |
| 1111  | 17       | F    | 15      |

## Algebra si logica Booleana

- Și-a căpătat numele după matematicianul englez George **Boole**, care a scris în 1854:  
„**O investigare a legilor gândirii**” („An investigation of the Laws of Thought”)

Aceasta lucrare este importanta prin descrierea in limbaj natural, a funcționarii unui circuit logic (bloc funcțional).

Limbajul natural are, spre deosebire de cel matematic, un anumit grad de ambiguitate.

Enumerari de forma:

„**Marcajul poate fi alb si rosu sau galben**”, nu au o întotdeauna bine stabilita starea initiala a operatorilor fundamentali. (Obs.: **NU, SI, SAU**)

In limbajul natural, unii operatorii fundamentali au particularitatea ca pot depinde de context.



## Algebra Booleana

### Exemplu (1/2)

Se considera următoarea descriere verbală a unui bloc de comandă (DESCHIDEREA UNEI USI):

- usa este deschisă atâta timp cât există în imediata sa proximitate (aproximativ 50 cm de-o parte și de alta a usii) o persoană și rămâne deschisă pentru un interval, relativ scurt, de timp  $t$  chiar și după plecarea oricărei persoane din zona de proximitate.

Se dorește o scriere concisă exactă, în termenii algebrei Booleene, a descrierii funcționării usii făcută în termenii limbajului natural.

Fiecare element relevant din descrierea în limbaj natural va fi notat printr-o variabilă care poate lua doar două valori 0 și 1.

Închiderea și deschiderea usii are loc în urma unei acțiuni. Usa este deschisă, se notează prin variabilă  $U$ .

- câtă vreme usa este deschisă,  $U = 1$ ;
- când usa este închisă  $U = 0$ .

Există o persoană în zona de proximitate. Determinarea unei persoane în zona de proximitate se poate realiza printr-un senzor cu traductor inductiv, fotoelectric etc.

Prezența unei persoane în zona de proximitate se notează prin variabilă  $P$ . Ori de câte ori este o persoană în apropierea usii:  $P = 1$ , altfel  $P = 0$ .



## Algebra booleana

### Exemplu (2/2)

Temporizatorul este activ pentru durata prestabilită  $t$ . Se notează prin  $T$ .

Atunci când temporizatorul este activ  $T = 1$ , altfel  $T = 0$ .

Temporizatorul este activat de îndată ce este sesizată o persoană în zona de proximitate. Activarea temporizatorului se mai numește și startare temporizator.

Prezența unei persoane în zona de proximitate face ca usa să se deschidă, pe de-o parte și activează temporizatorul, pe de-alta parte. Apariția unei persoane, în zona de proximitate, pe durata temporizării are ca efect numai reactivarea temporizării, usa fiind deja deschisă.

Se poate deduce că formalismul descrierii inițiale poate fi:

$$U = \text{SAU}(P, T).$$

Sintetic funcționarea usii poate fi exprimată, utilizând expresia anterioară, cu **blocul functional SAU**.

***Ușa se deschide dacă: este sesizată în apropiere o persoană, sau este activ temporizatorul!***

## Algebra Booleana

Printre aplicațiile algebrei Booleene (algebra logica) se numără: logica matematică, logica digitală, programarea calculatoarelor, statistica, etc.

Cel care a descoperit aplicabilitatea algebrei booleene în electronică este **Shannon Claude Elwood**, care a demonstrat în teza de absolvire a masteratului că orice problemă de logică se poate rezolva utilizând **elemente bistabile**;

O funcție logică (funcție booleană) este de tipul:  $f: \{0,1\} \rightarrow \{0,1\}$

Operatorii logici de bază sunt:

| Operator:               | ȘI                         |          | SAU                       |        | NU        |        |
|-------------------------|----------------------------|----------|---------------------------|--------|-----------|--------|
| Notatie                 | *                          | $\wedge$ | +                         | $\vee$ | $\bar{x}$ | $\neg$ |
| Denumire<br>alternativă | Produs logic<br>Conjuncție |          | Sumă logică<br>Disjuncție |        |           |        |

## Proprietatile algebrei booleene

❖ Reflexivitatea

$$a = a$$

## Proprietatile algebrei booleene

❖ Simetria

$$a = b \rightarrow b = a$$

## Proprietatile algebrei booleene

### ❖ Idempotența

$$✓ \quad a + a = a$$

$$0 + 0 = 0$$

$$1 + 1 = 1$$

$$✓ \quad a \cdot a = a$$

$$0 \cdot 0 = 0$$

$$1 \cdot 1 = 1$$

## Proprietatile algebrei booleene

### ❖ Comutativitatea

$$a + b = b + a$$

$$a \cdot b = b \cdot a$$

## Proprietatile algebrei booleene

### ❖ Asociativitatea

$$(a + b) + c = a + (b + c) = a + b + c$$

$$(a \cdot b) \cdot c = a \cdot (b \cdot c) = a \cdot b \cdot c$$

## Proprietatile algebrei booleene

### ❖ Distributivitatea

$$a \cdot (b + c) = a \cdot b + a \cdot c$$



## Proprietatile algebrei booleene

❖ Tranzitivitatea

$$\left. \begin{array}{l} a = c \\ b = c \end{array} \right\} \rightarrow a = b$$

## Proprietatile algebrei booleene

❖ Alte relatii

$$=$$
$$a = a$$

$$a + 0 = a$$

$$a + 1 = 1$$

$$a \cdot 0 = 0$$

$$a \cdot 1 = a$$

$$a + \bar{a} = 1$$

$$a \cdot \bar{a} = 0$$

## Proprietatile algebrei booleene

### ❖ Teoremele lui de Morgan

$$1. \overline{a + b} = \bar{a} \cdot \bar{b}$$

$$2. \overline{a + b + c} = \bar{a} \cdot \bar{b} \cdot \bar{c}$$

$$\overline{a \cdot b} = \bar{a} + \bar{b}$$

$$\overline{a \cdot b \cdot c} = \bar{a} + \bar{b} + \bar{c}$$

## Proprietatile algebrei booleene

❖ Teoreme de absorbție (utile pentru simplificarea expresiilor logice):

$$1. a + a \cdot b = a$$

$$2. a \cdot (\underline{a} + b) = a$$

$$3. a + \underline{a} \cdot b = a + b$$

$$4. a + (\underline{a} \cdot \underline{b}) = a \cdot b$$

$$5. a \cdot b + \underline{a} \cdot c + b \cdot c = a \cdot b + \underline{a} \cdot c$$

$$6. (a + b) \cdot (\underline{a} + c) \cdot (b + c) = (a + b) \cdot (\underline{a} + c)$$

## Reprezentarea funcțiilor logice

❖ Prin tabele de adevar

### Exemplu

Fie următorul exemplu de funcție booleană:

| x | y | z | $f(x, y, z)$ |
|---|---|---|--------------|
| 0 | 0 | 0 | 1            |
| 0 | 0 | 1 | 0            |
| 0 | 1 | 0 | 1            |
| 0 | 1 | 1 | 1            |
| 1 | 0 | 0 | 1            |
| 1 | 0 | 1 | 0            |
| 1 | 1 | 0 | 0            |
| 1 | 1 | 1 | 0            |

Observăm că  $f(x, y, z) = 1$  dacă și numai dacă:

$x = 0$  și  $y = 0$  și  $z = 0$  sau

$x = 0$  și  $y = 1$  și  $z = 0$  sau

$x = 0$  și  $y = 1$  și  $z = 1$  sau

$x = 1$  și  $y = 0$  și  $z = 0$

## Reprezentarea funcțiilor logice

❖ Prin tabele de adevar

❖ Prin forma algebrica

- Forma normală disjunctivă (FND) (sumă/disjuncție de produse/conjuncții)

- Forma normală conjunctivă (FNC) (produse/conjuncții de suma/disjuncții)

### Exemplu

Fie următorul exemplu de funcție booleană:

| x | y | z | f(x, y, z) |
|---|---|---|------------|
| 0 | 0 | 0 | 1          |
| 0 | 0 | 1 | 0          |
| 0 | 1 | 0 | 1          |
| 0 | 1 | 1 | 1          |
| 1 | 0 | 0 | 1          |
| 1 | 0 | 1 | 0          |
| 1 | 1 | 0 | 0          |
| 1 | 1 | 1 | 0          |

Observăm că  $f(x, y, z) = 1$  dacă și numai dacă:

$x = 0$  și  $y = 0$  și  $z = 0$  sau

$x = 0$  și  $y = 1$  și  $z = 0$  sau

$x = 0$  și  $y = 1$  și  $z = 1$  sau

$x = 1$  și  $y = 0$  și  $z = 0$

Din această descriere obținem ușor forma normală disjunctivă (FND):

$$f(x, y, z) = \bar{x}\bar{y}\bar{z} + \bar{x}y\bar{z} + \bar{x}yz + x\bar{y}\bar{z}$$

## Reprezentarea funcțiilor logice

### ❖ Prin $\Sigma$ - notație

- Pentru valorile din tabelul de adevăr care duc la valoarea de 1 logic la ieșire, se calculează numărul în baza 10 rezultat din transformarea șirului corespunzător de 0 și 1 din intrare.

De exemplu pentru funcția de mai sus, reprezentările variantelor pentru care funcția are valoarea 1 sunt:

- $0\ 0\ 0 \rightarrow 0_{10}$
- $0\ 1\ 0 \rightarrow 2_{10}$
- $0\ 1\ 1 \rightarrow 3_{10}$
- $1\ 0\ 0 \rightarrow 4_{10}$

Funcția poate fi scrisă sub forma:  $f(x, y, z) = \Sigma(0, 2, 3, 4)$



## Reprezentarea funcțiilor logice

### ❖ Prin metoda algebrică

- Se încearcă aducerea funcției logice la o formă simplificată aplicând proprietățile algebrei booleene. Metoda nu garantează ajungerea la forma minimă a funcției.





# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

| AB \ CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 1  | 0  | 0  |
| 01      | 1  | 1  | 1  | 1  |
| 11      | 0  | 0  | 1  | 1  |
| 10      | 1  | 0  | 1  | 1  |

$$S = \overline{A}\overline{C}D + \overline{A}B + AC + A\overline{B}\overline{C}D$$

## Minimizarea funcțiilor logice

### ❖ Prin metoda diagramelor Karnaugh

- Se garantează ajungerea la forma minimizată a funcției;
- Se urmeaza pașii:
  - Se completează diagrama Karnaugh după cum urmează:
    - Distribuția valorilor trebuie să respecte codul Gray (valorile a două căsuțe vecine trebuie să difere printr-un singur bit);
    - Valorile posibile sunt: 0, 1, - (nu este dată valoarea);
  - Se realizează grupări formate din valorile de unu, respectând următoarele reguli:
    - nu trebuie să rămână niciun termen cu valoarea 1 negrupat.
    - o grupare trebuie să conțină un număr de valori de unu egal cu o putere a lui 2 (deci o grupare poate să aibă un număr de căsuțe egal cu: 1, 2, 4, 8...);
    - o grupare nu poate avea forma stelară, trebuie să aibă neapărat forma dreptunghiulară;
    - grupările se pot intersecta;
    - căsuțele de pe extremitățile diagramei se consideră vecine;

# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

| AB \ CD | 00 | 01 | 11 | 10 |
|---------|----|----|----|----|
| 00      | 0  | 1  | 0  | 0  |
| 01      | 1  | 1  | 1  | 1  |
| 11      | 0  | 0  | 1  | 1  |
| 10      | 1  | 0  | 1  | 1  |

$$S = \overline{A}\overline{C}D + \overline{A}B + AC + A\overline{B}\overline{C}D$$

❖ Se află forma funcției minimizate, astfel:

- Pentru fiecare grupare, se calculează valoarea astfel:
  - pentru fiecare variabilă a funcției, dacă variabila își schimbă valoarea din 1 în 0 sau invers se pune 1;
  - dacă variabila are valoarea 1 în toate celulele din grupare, se pune numele variabilei;
  - dacă variabila are valoarea 0 în toate celulele din grupare, se pune variabila negată.
  - Se realizează un produs (ȘI logic între valorile obținute pentru fiecare variabilă de-a lungul grupării);
- Între termenii obținuți din fiecare grupare se realizează o sumă (SAU logic)

## Minimizarea funcțiilor logice

**Exemplu: harta Karnaugh pentru doua variabile**

- hartile Karnaugh sunt folosite pentru simplificarea si minimizarea funcțiilor logice booleene.

Se da urmatoarea functie:

$$F = A'B + AB' + AB$$

Completare tabel adevar:

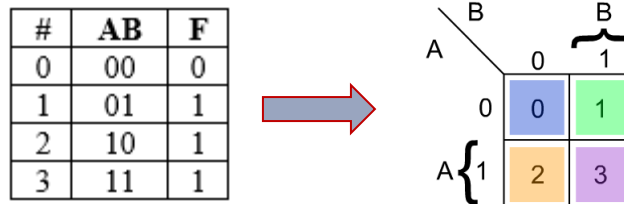
| # | AB | F |
|---|----|---|
| 0 | 00 | 0 |
| 1 | 01 | 1 |
| 2 | 10 | 1 |
| 3 | 11 | 1 |

O harta Karnaugh are  $2^v$  patrate, unde v este nr. de variabile. Pentru 2 variabile harta Karnaugh are 4 patrate.

## Minimizarea funcțiilor logice

Exemplu: harta Karnaugh pentru doua variabile

Completare patrate pentru 2 variabile



Pentru acest exemplu intrucat  $A'B$ ,  $AB'$  si  $AB$  contine 1 harta Karnaugh va avea expresia:  $F(A,B) = \sum(1,2,3)$

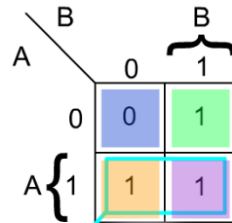
## Minimizarea funcțiilor logice

### Exemplu: harta Karnaugh pentru doua variabile

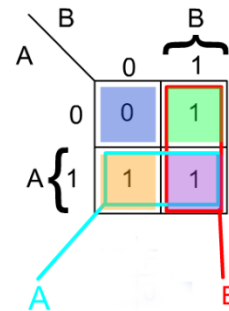
Dupa ce harta Karnaugh a fost completata se poate face minimizarea functiei.

In continuare se va incercui (grupa) termenii 1. Termenii care trebuie incercuiti sunt puterea lui 2 (2 de cate 1).

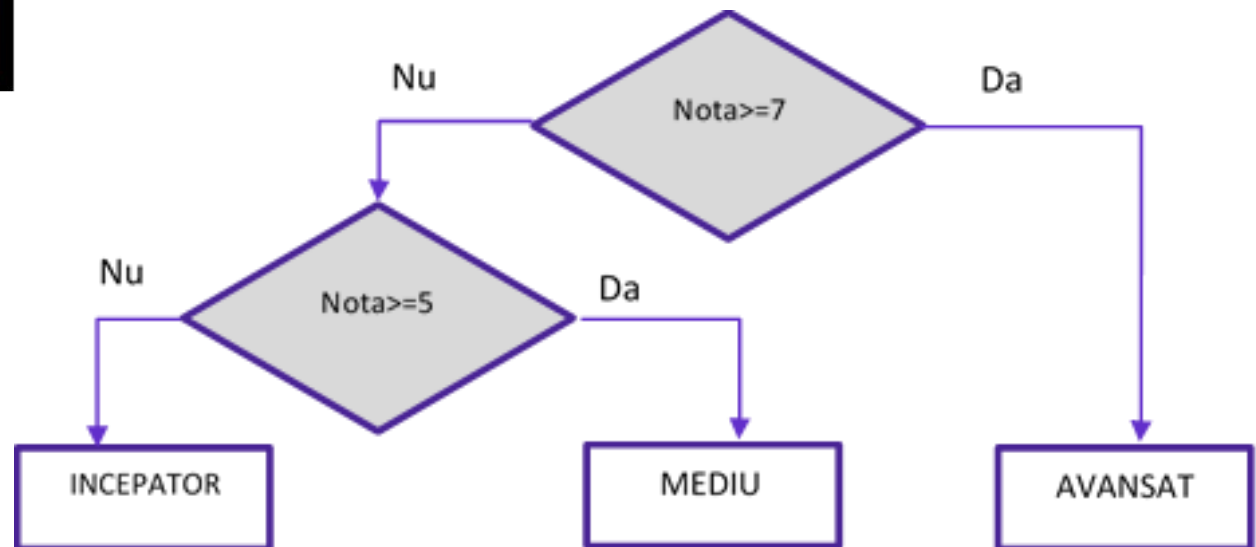
| # | AB | F |
|---|----|---|
| 0 | 00 | 0 |
| 1 | 01 | 1 |
| 2 | 10 | 1 |
| 3 | 11 | 1 |



$$F=A+B$$



## Circuite Bistabile

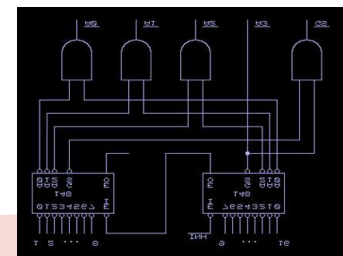




**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

**Facultatea de Construcții de Mașini**  
**Departamentul Ingineria Proiectării și Robotică**

# **Circuitele Logice Folosite in Echipamentele de Comanda**



**Sl. Dr. Ing. Florin COVACIU**



## Componente de decizie in echipamente industriale

Echipamentul de comanda numerica este un sistem cu ajutorul caruia se controleaza masina-unealta CNC, genereaza programul de functionare al acesteia.

Procesul de productie se poate supraveghea de catre un operator uman sau de catre un sistem de calcul prin intermediul unui alt sistem numit DNC (Direct Numerical Control).





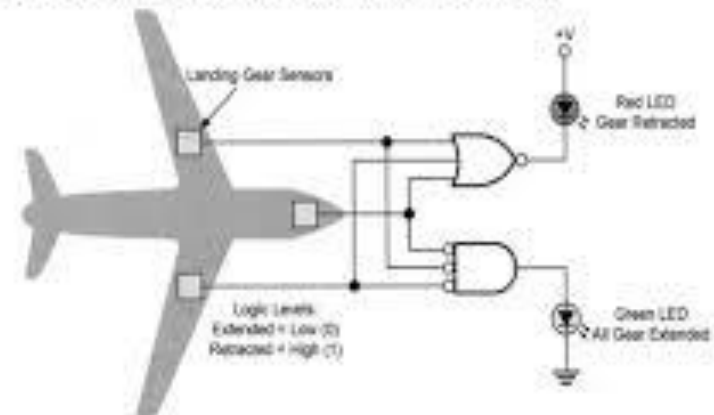
## Circuitele logice

Un circuit logic este materializat de către un circuit electronic fizic, ce implementează una sau mai multe funcții booleene (logice).



### AIRCRAFT LOGIC CIRCUIT

#### • LANDING GEAR WARNING CIRCUIT



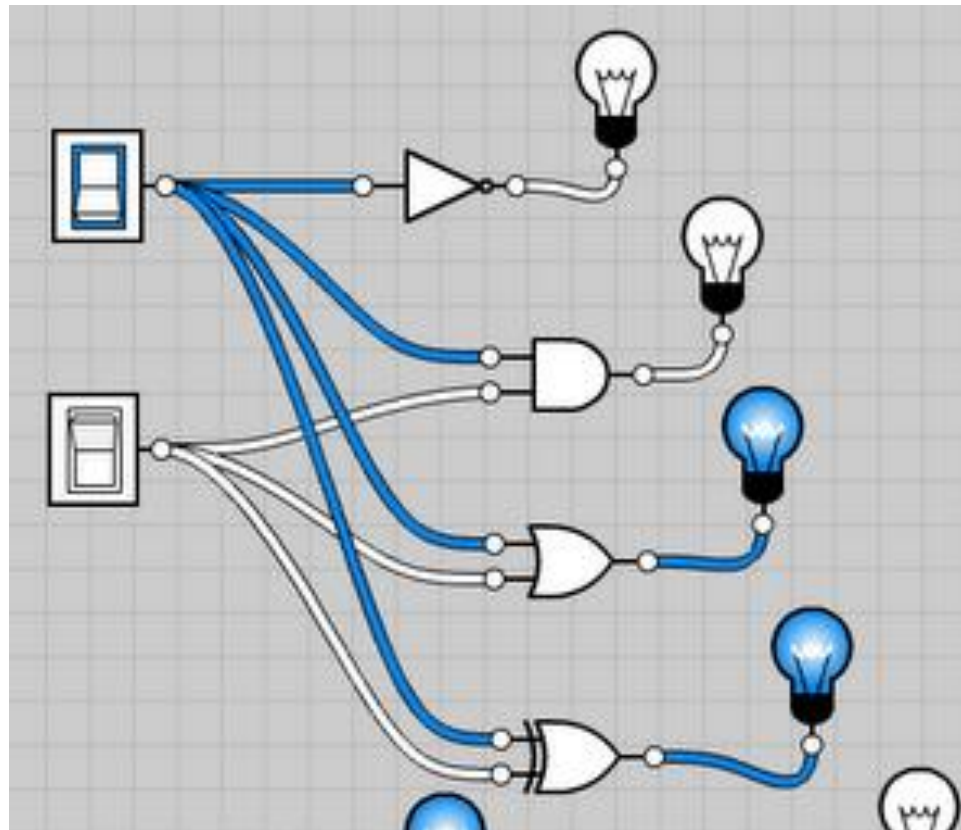
## Circuitele logice

**Componentele fizice** ce formeaza circuitele logice au un comportament definit de doua stări distincte:

- un contact poate fi închis sau deschis ;
- un tranzistor poate fi blocat sau saturat;
- un bec poate fi aprins sau stins;
- un motor poate fi pornit sau oprit;
- etc.

## Circuitele logice

Deoarece efectele enumerate nu au sens în lipsa semnalelor (a mărimilor fizice, tensiune sau curent) este necesară realizarea de corespondențe între valorile logice 0 și 1 și nivelele mărimilor fizice.



## Circuitele logice

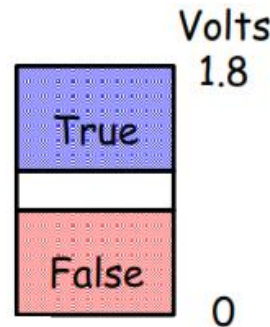
- ❖ la nivel logic, elementar, operațiile circuitelor logice de baza sunt exact operațiile din logica booleana
- ❖ toate operațiile aritmetice, se efectueaza în baza 2 (sistem binar)
- ❖ funcțiile booleene sunt implementate la nivel logic cu ajutorul unor circuite speciale numite: **circuite combinationale**

## Circuitele logice

- ❖ o **tensiune electrica** poate fi utilizata pentru reprezentarea celor doua valori discrete

**1 si 0**

- ❖ tensiunea electrica este utilizata si la reprezentarea valorilor logice 'adevarat' (**true=1**) si 'fals' (**false=0**)



- ❖ In logica directa :
  - **1** este 'adevarat' (true)
  - **0** este 'fals' (false)
- ❖ aceasta interpretare este utilizata atat pentru efectuarea calculelor cat si la **implementarea functiilor** (matematice sau logice) într-o componenta hardware.

## Importanta sistemului binar

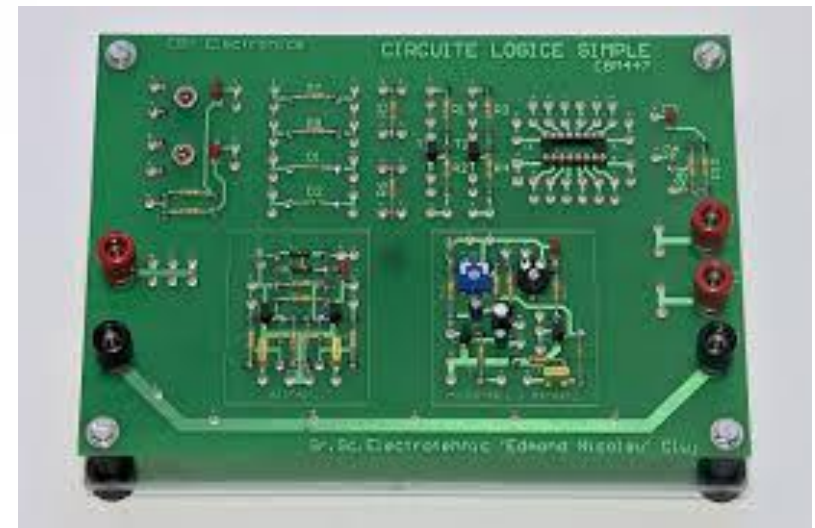
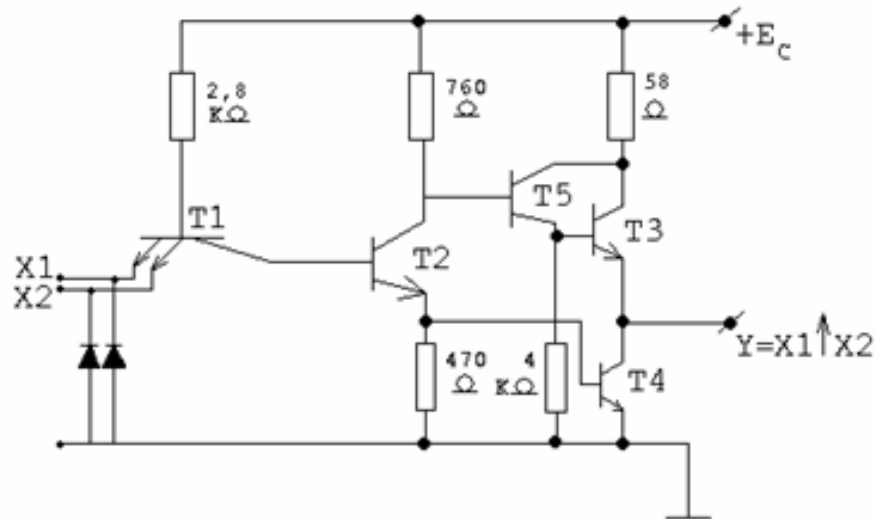
Electronica digitala a aparut datorita existentei sistemului binar.

Usurinta cu care putem reprezenta bitii sub forma fizica (0 si 1) se datoreaza faptului ca orice mediu fizic poate functiona cu doua stari saturate diferite, in concluzie poate fi folosit pentru reprezentarea unui bit.

### În consecinta:

- orice sistem fizic ce este capabil sa reprezinte biti, poate fi folosit pentru a reprezenta valori numerice si are potentialul de a manipula aceste numere;
- **Circuitele electronice** sunt ideale pentru reprezentarea numerelor binare.

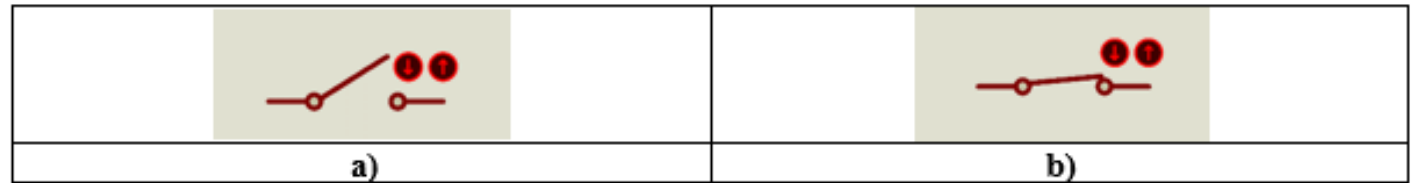
## Circuitele logice



## Circuitele logice

Circuitele logice se pot realiza cu contacte, de comutare cum ar fi:

- butoane de comandă;
- comutatoare;
- întrerupătoare;
- limitatoare;
- contactoare;
- relee.



Contactele sunt de două feluri:

- normal deschis (a), când prin contact nu circulă curent
- normal închis (b), când prin contact circulă curent



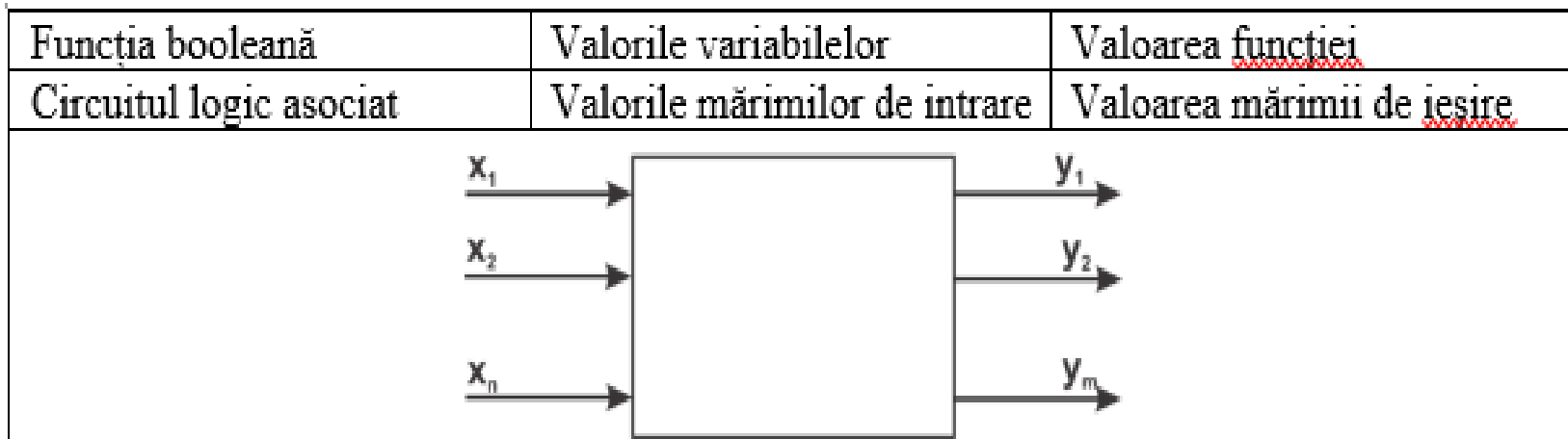
## Circuitele logice

Analogia între funcțiile logice și circuitele logice.

Funcțiile logice se implementează într-un circuit logic.

Circuitele logice pot fi integrate în blocuri logice.

Reprezentarea unei scheme bloc a unui circuit logic cu  $n$  intrări și  $m$  ieșiri.



## Circuitele logice

**Tranzistorii**, atunci când funcționează la limita, se pot afla în două stări:

- în stare blocată (curent de control zero);
- în stare de saturatie (curent de control maxim).

Dacă un **circuit cu tranzistor** este proiectat pentru maximizarea probabilității de funcționare într-una din cele două stări, acesta poate fi folosit ca și reprezentare fizică a unui bit.

Căderea de tensiune măsurată la ieșirea din circuit poate fi folosită pentru reprezentarea unui singur bit și constituie valoarea maximă de ieșire.

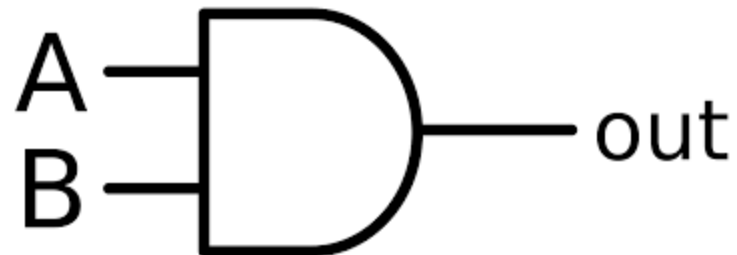
## Porti Logice

O poarta logica, este un circuit special conceput pentru acceptarea si generarea semnalelor de tensiune, ce corespund numerelor binare 0 si 1, fara valori intermediare.

Iesirea fiecărei porti reprezinta un semnal (un singur bit) dintr-un numar binar reprezentat pe mai multi biti.

Mai multe porti logice, conectate impreuna, pot fi folosite pentru stocarea sau manipularea semnalelor binare:

- circuite de memorare
- circuite de calcul



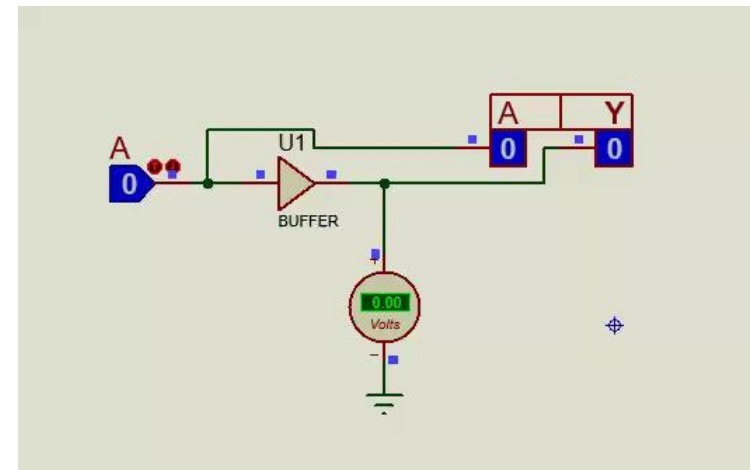
# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Porti Logice

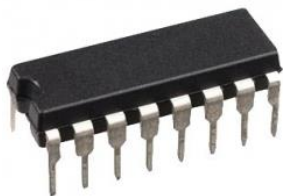
1. **Tampon (Buffer)** - este o poartă logică care are întotdeauna o singură intrare și o ieșire, în care ieșirea întotdeauna coincide cu intrarea;

| Simbol logic(ANSI)                                                                | Tabel de adevăr                                                                                        | Expresie booleană |   |   |   |   |   |         |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---------|
|  | <table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td></tr></table> | A                 | Y | 0 | 0 | 1 | 1 | $Y = A$ |
| A                                                                                 | Y                                                                                                      |                   |   |   |   |   |   |         |
| 0                                                                                 | 0                                                                                                      |                   |   |   |   |   |   |         |
| 1                                                                                 | 1                                                                                                      |                   |   |   |   |   |   |         |

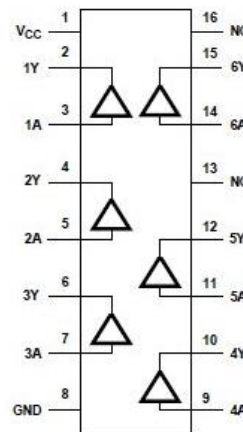
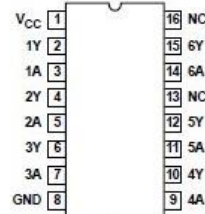
Film simulate



Circuit integrat

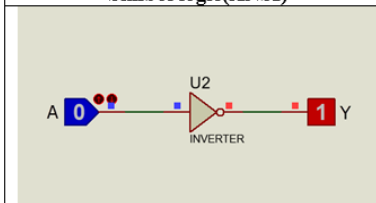


74HC4050

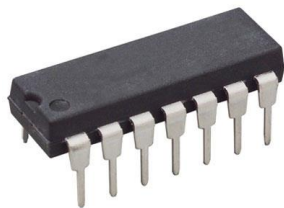


## Porti Logice

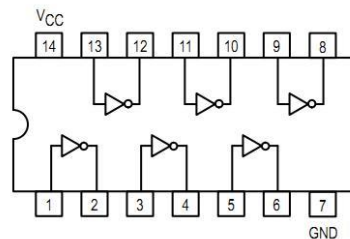
**2. NU (NOT)** - este o poartă logică cu ajutorul caruia se inversează semnalul de intrare, poarta de negare;

| Simbol logic(ANSI)                                                                | Tabel de adevăr                                                                                        | Expresie booleană |   |   |   |   |   |                    |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|--------------------|
|  | <table><tr><th>A</th><th>Y</th></tr><tr><td>0</td><td>1</td></tr><tr><td>1</td><td>0</td></tr></table> | A                 | Y | 0 | 1 | 1 | 0 | $Y = \overline{A}$ |
| A                                                                                 | Y                                                                                                      |                   |   |   |   |   |   |                    |
| 0                                                                                 | 1                                                                                                      |                   |   |   |   |   |   |                    |
| 1                                                                                 | 0                                                                                                      |                   |   |   |   |   |   |                    |

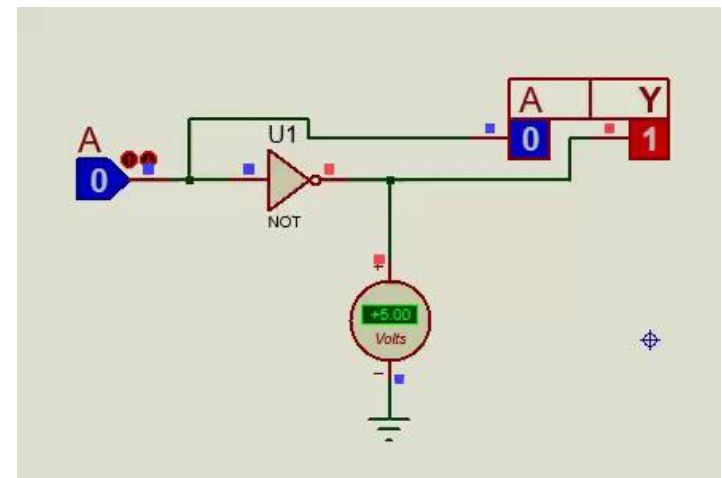
Circuit integrat



Model: 74LS04



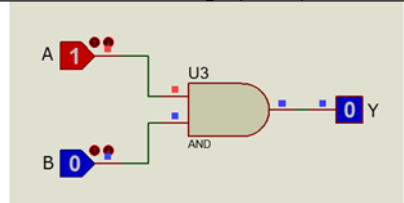
Film simulare



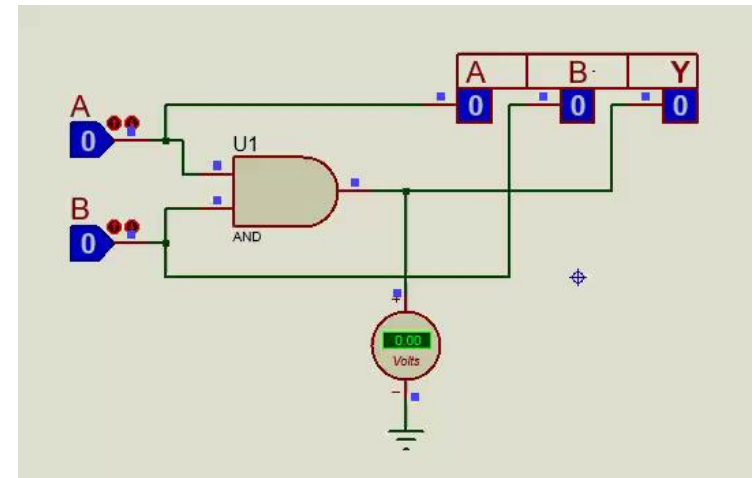
# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Porti Logice

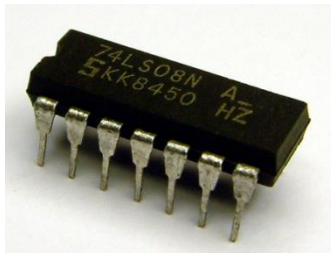
3. **SI (AND)** - este un circuit cu cel puțin două intrări și o singură ieșire, ieșirea având valoarea logică "1", atunci când toate intrările sunt "1" logic.

| Simbol logic(ANSI)                                                                | Tabel de adevăr                                                                                                                                                                                                    | Expresie booleană |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------------|
|  | <table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> | A                 | B | Y | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | $Y = A * B$ |
| A                                                                                 | B                                                                                                                                                                                                                  | Y                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 0                                                                                 | 0                                                                                                                                                                                                                  | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 0                                                                                 | 1                                                                                                                                                                                                                  | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 1                                                                                 | 0                                                                                                                                                                                                                  | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 1                                                                                 | 1                                                                                                                                                                                                                  | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |

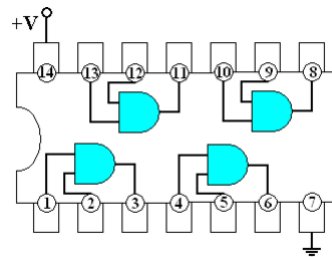
Film simulare



Circuit integrat



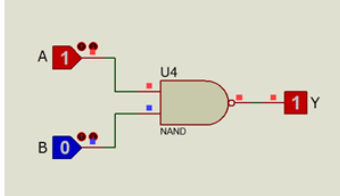
Model: 7408



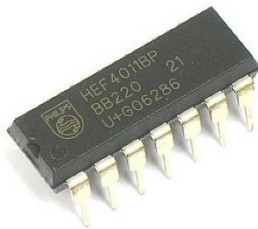
## Porti Logice

4. **Si-NU (NAND)** - este o poartă logică compusa cu urmatoarea functionare:

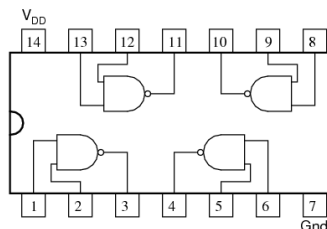
- iesirea sa este 0 logic (fals), daca ambele intrari sunt 1 logic (adevarat) ;
- iesirea sa este 1 logic (adevarat), daca cel putin una din intrari este 0 logic (fals).

| Simbol logic(ANSI)                                                                | Tabel de adevăr                                                                                                                                                                                                    | Expresie booleană |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------------------------|
|  | <table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table> | A                 | B | Y | 0 | 0 | 1 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | $Y = \overline{A * B}$ |
| A                                                                                 | B                                                                                                                                                                                                                  | Y                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |
| 0                                                                                 | 0                                                                                                                                                                                                                  | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |
| 0                                                                                 | 1                                                                                                                                                                                                                  | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |
| 1                                                                                 | 0                                                                                                                                                                                                                  | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |
| 1                                                                                 | 1                                                                                                                                                                                                                  | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |

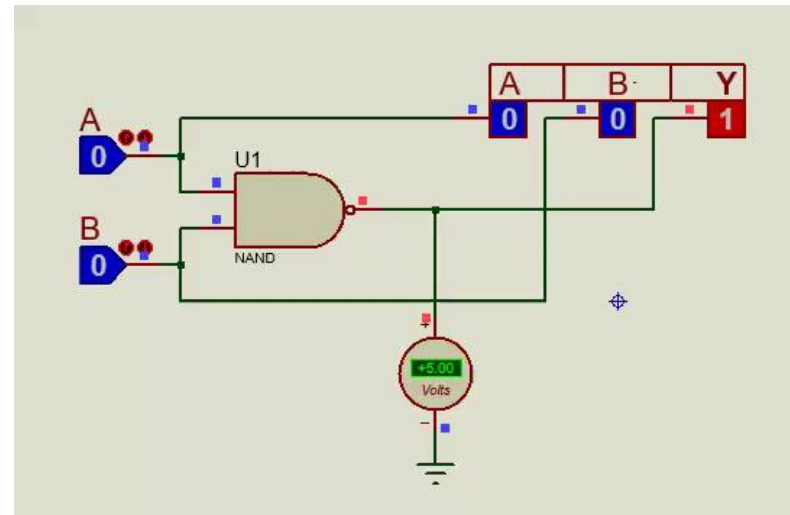
### Circuit integrat



### Model: 4011



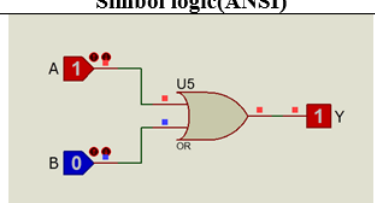
### Film simulate



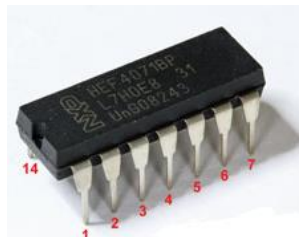
## Porti Logice

5. **SAU (OR)**- aceasta poarta functioneaza astfel:

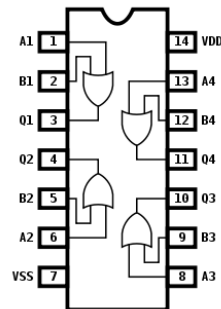
- nivelul de tensiune la iesire corespunde lui 1 logic atunci cand cel putin uneia dintre intrari i se aplica un nivel de tensiune ce corespunde cu 1, adica iesirea este 1 logic daca cel putin una dintre intrari este 1 logic.

| Symbol logic(ANSI)                                                                | Tabel de adevăr                                                                                                                                                                                                    | Expresie booleană |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------------|
|  | <table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> | A                 | B | Y | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | $Y = A + B$ |
| A                                                                                 | B                                                                                                                                                                                                                  | Y                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 0                                                                                 | 0                                                                                                                                                                                                                  | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 0                                                                                 | 1                                                                                                                                                                                                                  | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 1                                                                                 | 0                                                                                                                                                                                                                  | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 1                                                                                 | 1                                                                                                                                                                                                                  | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |

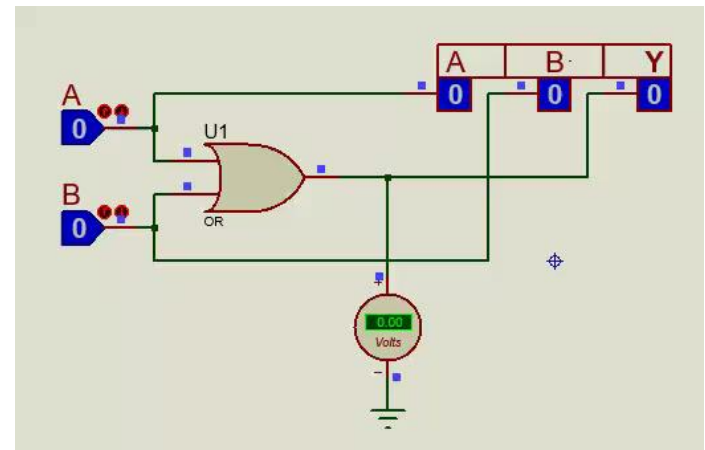
Circuit integrat



Model: 4071



Film simulare

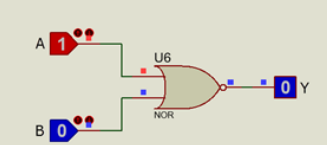




## Porti Logice

6. **SAU-NU (NOR)** – are urmatoare interpretare:

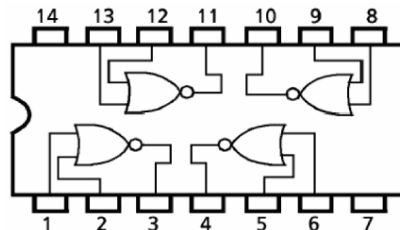
- iesirea sa este falsa (0 logic) daca cel putin una dintre intrari este adevarata (1 logic);
- Iesirea sa este adevarata (1 logic) daca ambele intrari sunt false (0 logic).

| Simbol logic(ANSI)                                                                | Tabel de adevăr                                                                                                                                                                                                                                  | Expresie booleană |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------------------------|
|  | <table><thead><tr><th>A</th><th>B</th><th>Y</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></tbody></table> | A                 | B | Y | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 0 | $Y = \overline{A + B}$ |
| A                                                                                 | B                                                                                                                                                                                                                                                | Y                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |
| 0                                                                                 | 0                                                                                                                                                                                                                                                | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |
| 0                                                                                 | 1                                                                                                                                                                                                                                                | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |
| 1                                                                                 | 0                                                                                                                                                                                                                                                | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |
| 1                                                                                 | 1                                                                                                                                                                                                                                                | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                        |

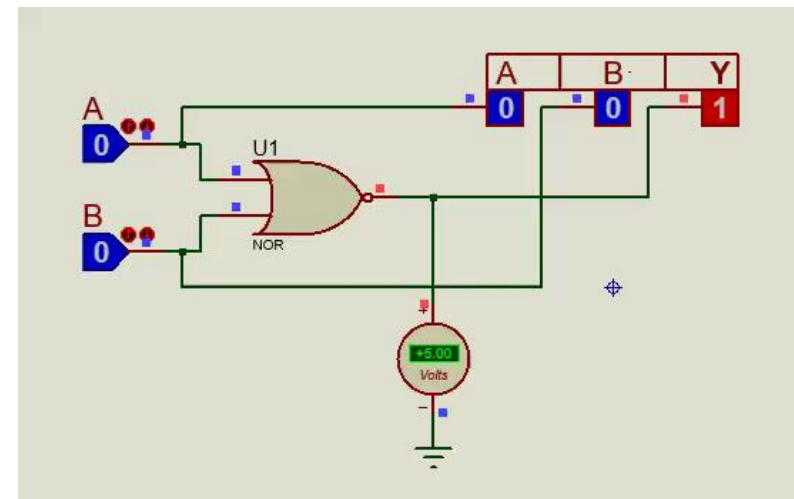
Circuit integrat



Model: 4071



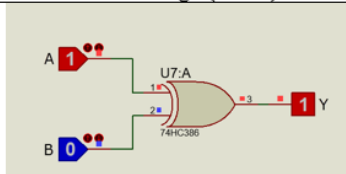
Film simulare



## Porti Logice

### 7. SAU-EXCLUSIV (XOR)

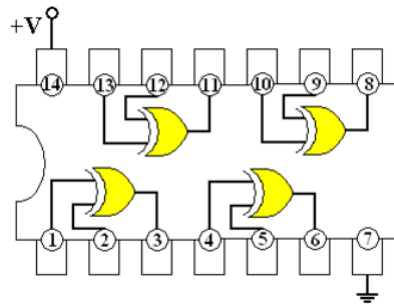
- iesirea este 1 logic, daca intrarile se afla la nivele logice diferite (0-1, sau 1-0)
- iesirea este 0 logic, daca toate intrarile se afla la acelasi nivel logic.

| Simbol logic(ANSI)                                                                | Tabel de adevăr                                                                                                                                                                                                    | Expresie booleană |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------------------|
|  | <table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>0</td></tr></table> | A                 | B | Y | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | $Y = A \oplus B$ |
| A                                                                                 | B                                                                                                                                                                                                                  | Y                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
| 0                                                                                 | 0                                                                                                                                                                                                                  | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
| 0                                                                                 | 1                                                                                                                                                                                                                  | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
| 1                                                                                 | 0                                                                                                                                                                                                                  | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
| 1                                                                                 | 1                                                                                                                                                                                                                  | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |

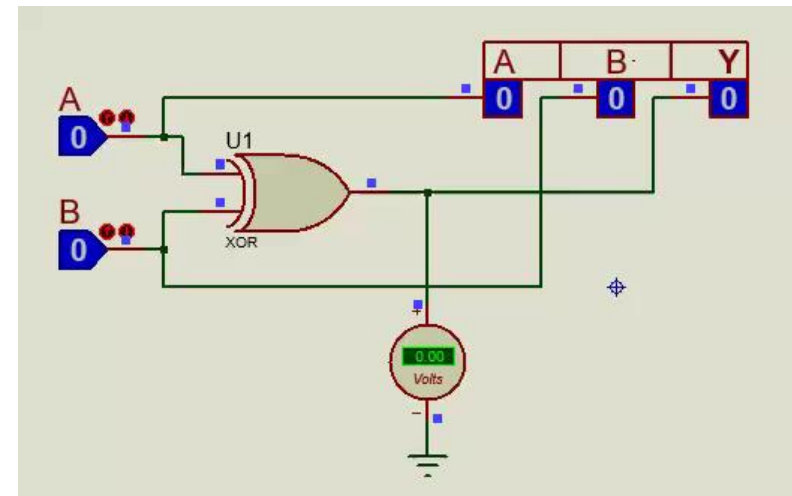
### Circuit integrat



### Model: 7486



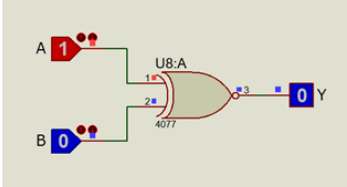
### Film simulate



## Porti Logice

### 8. SAU-EXCLUSIV NEGAT (XNOR) – logica este invers portii SAU-EXCLUSIV

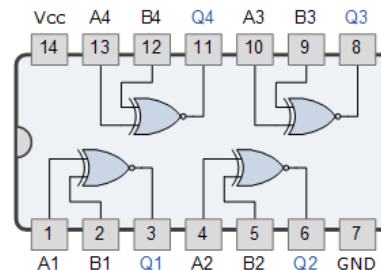
- iesirea este 1 logic, daca intrarile se afla la acelasi nivel logic
- iesirea este 0 logic, daca intrarile se afla la nivele logice diferite.

| Simbol logic(ANSI)                                                                | Tabel de adevăr                                                                                                                                                                                                                                  | Expresie booleană |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                             |
|-----------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-----------------------------|
|  | <table><thead><tr><th>A</th><th>B</th><th>Y</th></tr></thead><tbody><tr><td>0</td><td>0</td><td>1</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></tbody></table> | A                 | B | Y | 0 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | $Y = \overline{A \oplus B}$ |
| A                                                                                 | B                                                                                                                                                                                                                                                | Y                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                             |
| 0                                                                                 | 0                                                                                                                                                                                                                                                | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                             |
| 0                                                                                 | 1                                                                                                                                                                                                                                                | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                             |
| 1                                                                                 | 0                                                                                                                                                                                                                                                | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                             |
| 1                                                                                 | 1                                                                                                                                                                                                                                                | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                             |

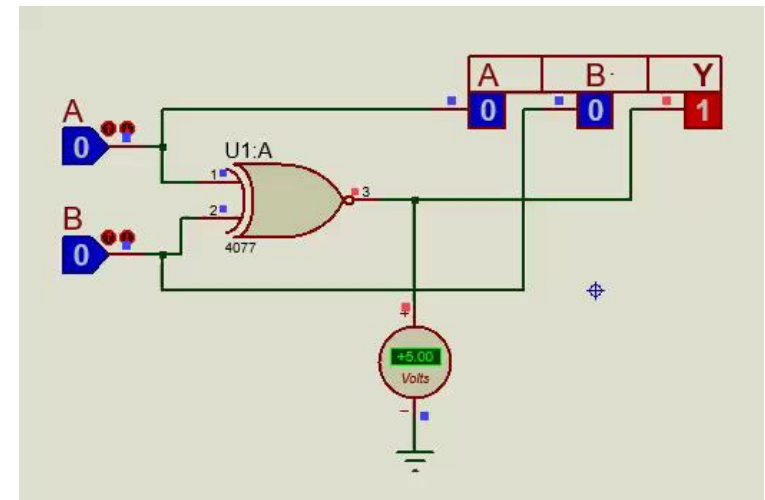
### Circuit integrat



### Model: 74266



### Film simulare



## Constructia portilor logice

Tehnologia circuitelor digitale s-a dezvoltat în paralel cu tehnologia dispozitivelor electronice și a materialelor semiconductoare.

Primele porți și circuite logice au fost realizate cu tuburi electronice, apoi a început era materialelor semiconductoare și odată cu ea dezvoltarea circuitelor digitale integrate (CI), realizate cu diferite tehnologii.

Circuitele care implementează funcții logice binare s-au dovedit a fi potrivite realizării în tehnologie integrată, ajungându-se tehnologic astăzi la integrarea a peste 1 milion de porți elementare pe cip.

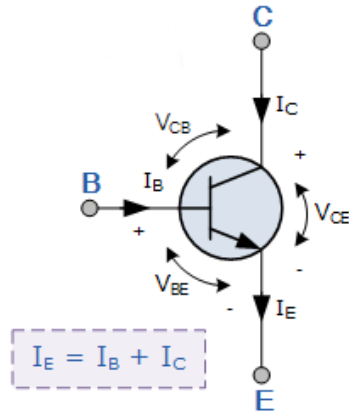
## Construcția portilor logice

Din punct de vedere tehnologic porțile logice pot fi:

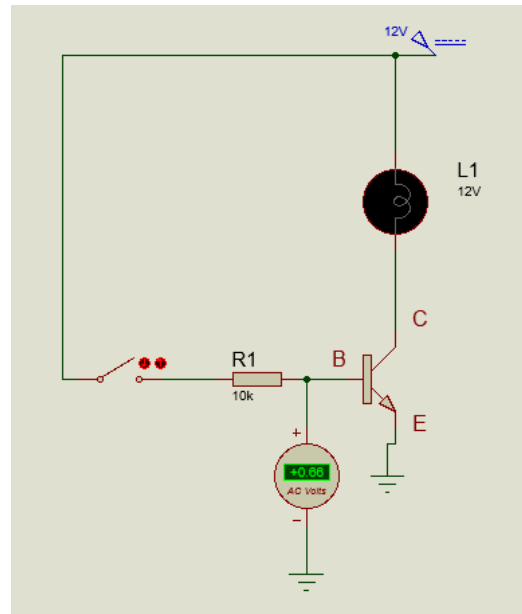
- ✓ porți integrate unipolare care au în construcția lor tranzistoari unipolari (NMOS sau PMOS)
  - ✓ NMOS
  - ✓ PMOS
  
- ✓ porți integrate bipolare care au în construcția lor tranzistoari bipolari (NPN sau PNP)
  - ✓ NPN - Negative-Positive-Negative
  - ✓ PNP
  
- ✓ porți integrate mixte care au în construcția lor atât tranzistoari bipolari cât și unipolari
  - ✓ BiCMOS – Bipolar and Complementary Metal Oxid Semiconductor

## Construcția portilor logice

### Principiul de functionare al unui tranzistor NPN

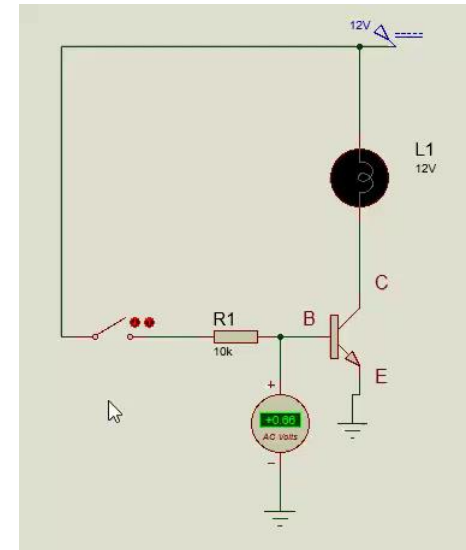


simbol transistor NPN



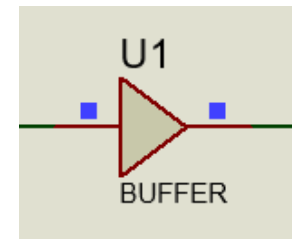
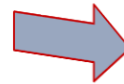
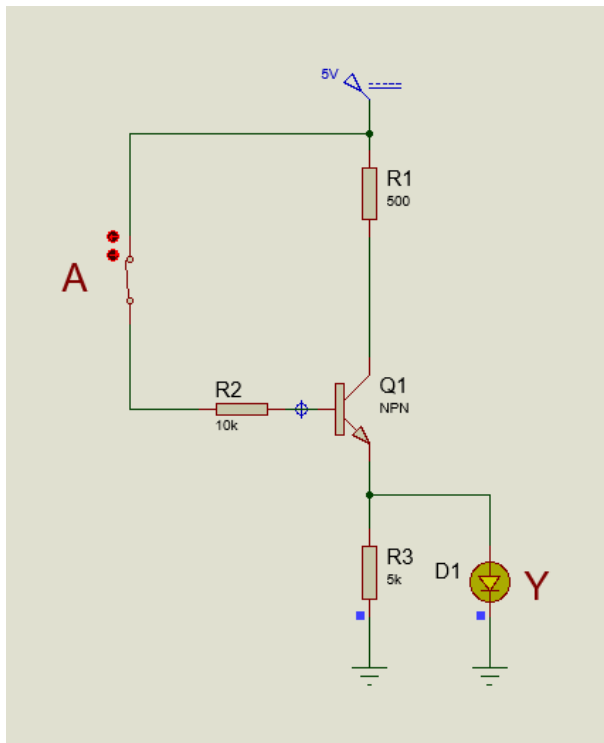
releu

### Film simulare



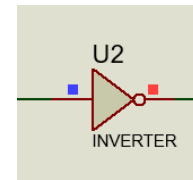
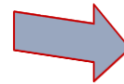
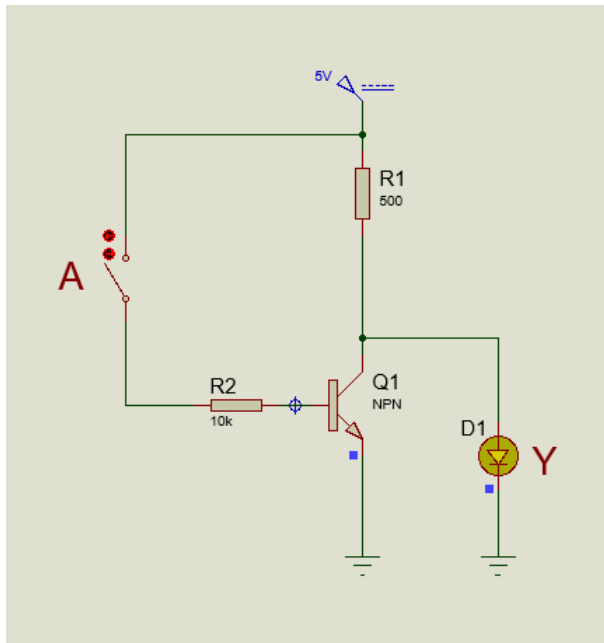
## Construcția portilor logice

### 1. Tampon (Buffer)



## Construcția portilor logice

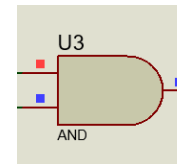
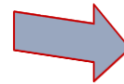
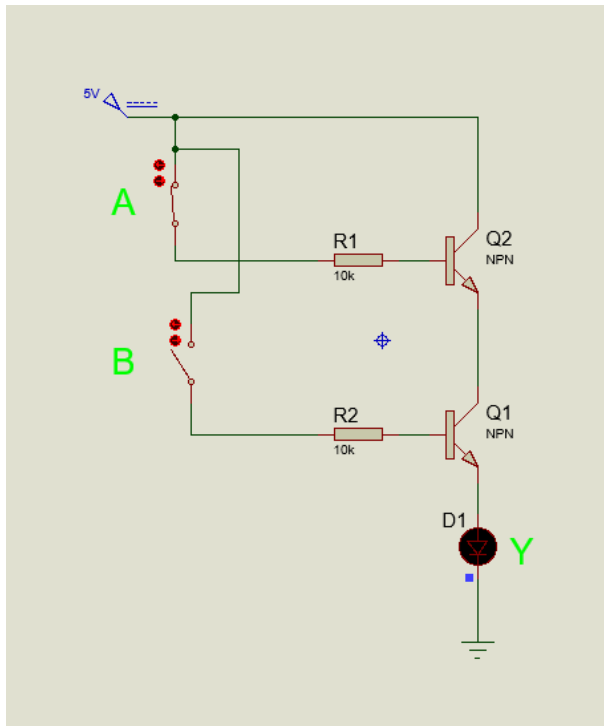
### 2. NOT





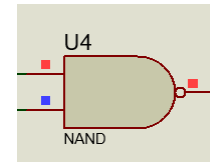
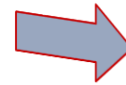
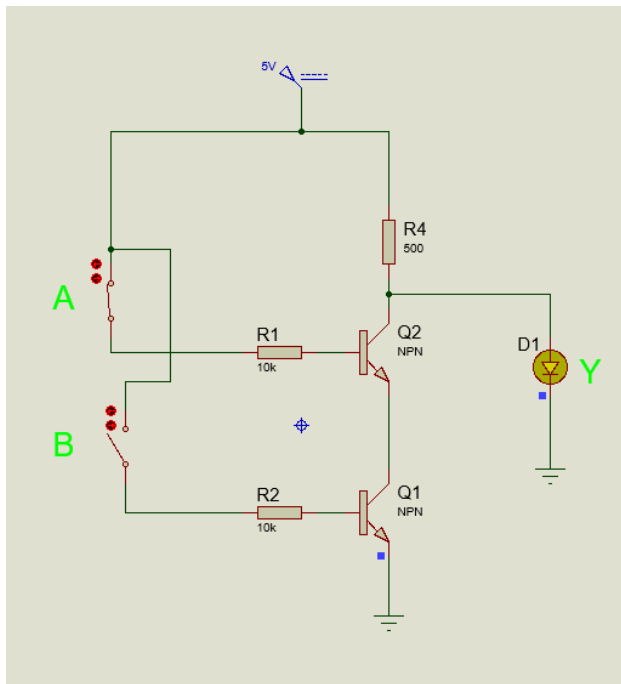
## Construcția portilor logice

### 3. AND



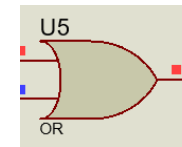
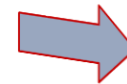
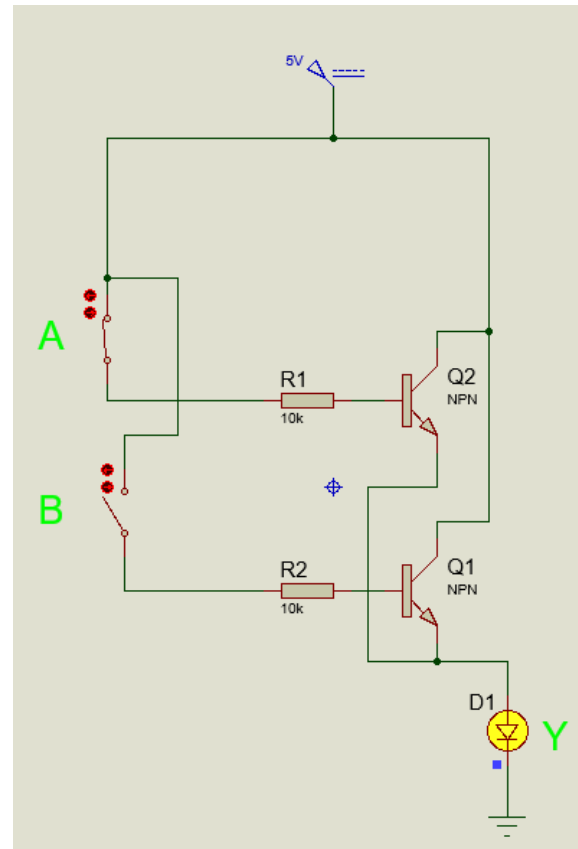
## Construcția portilor logice

### 4. NAND



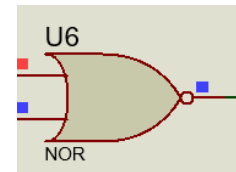
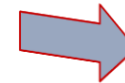
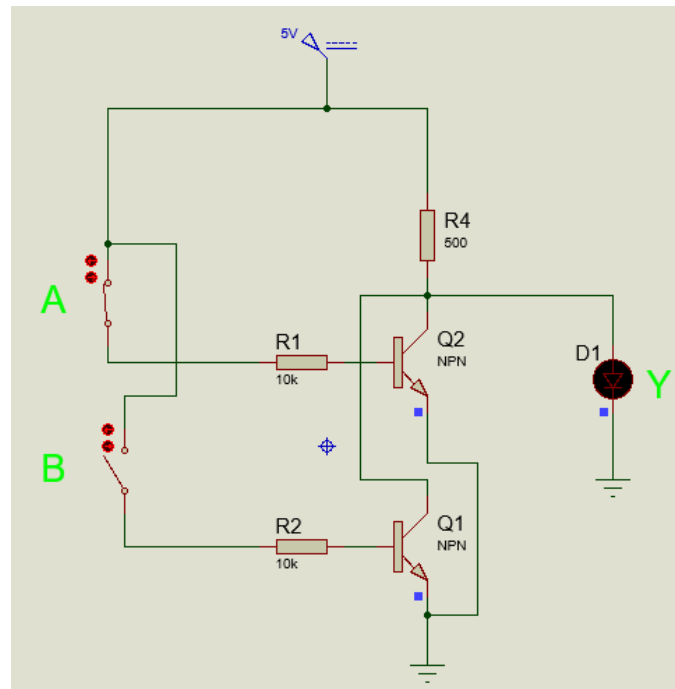
## Construcția portilor logice

### 5. OR



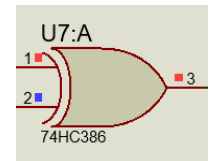
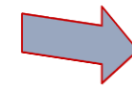
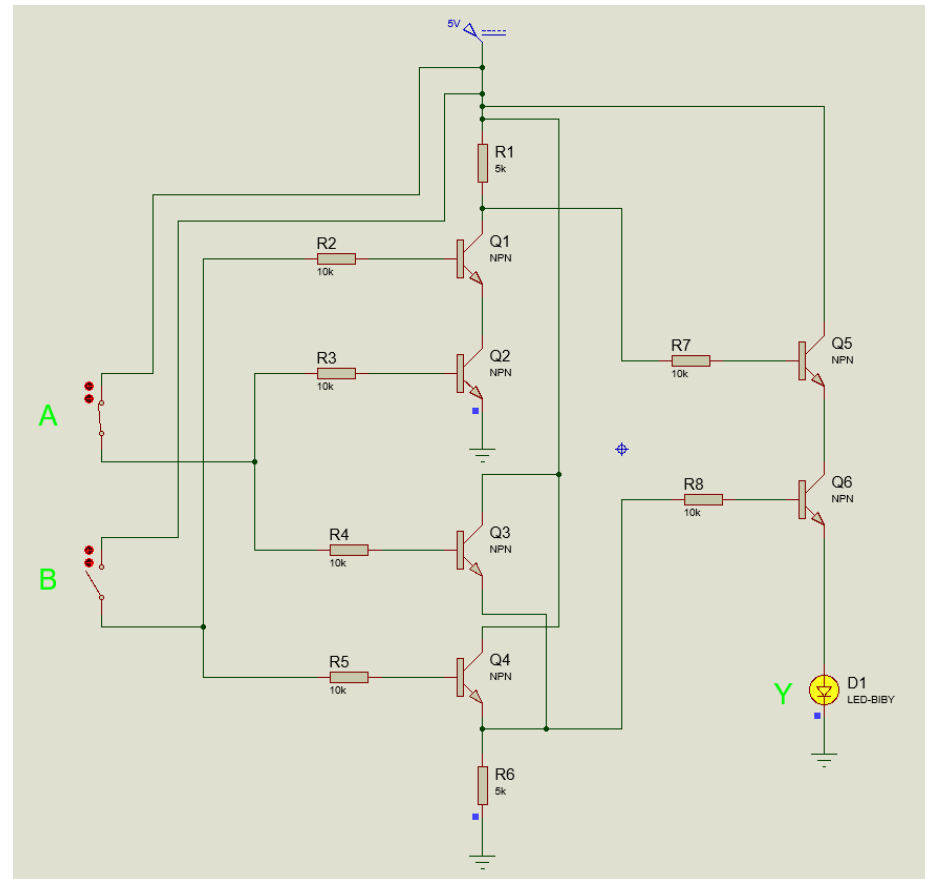
## Construcția portilor logice

### 6. NOR



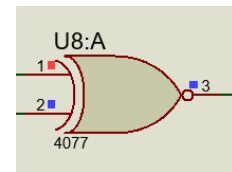
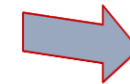
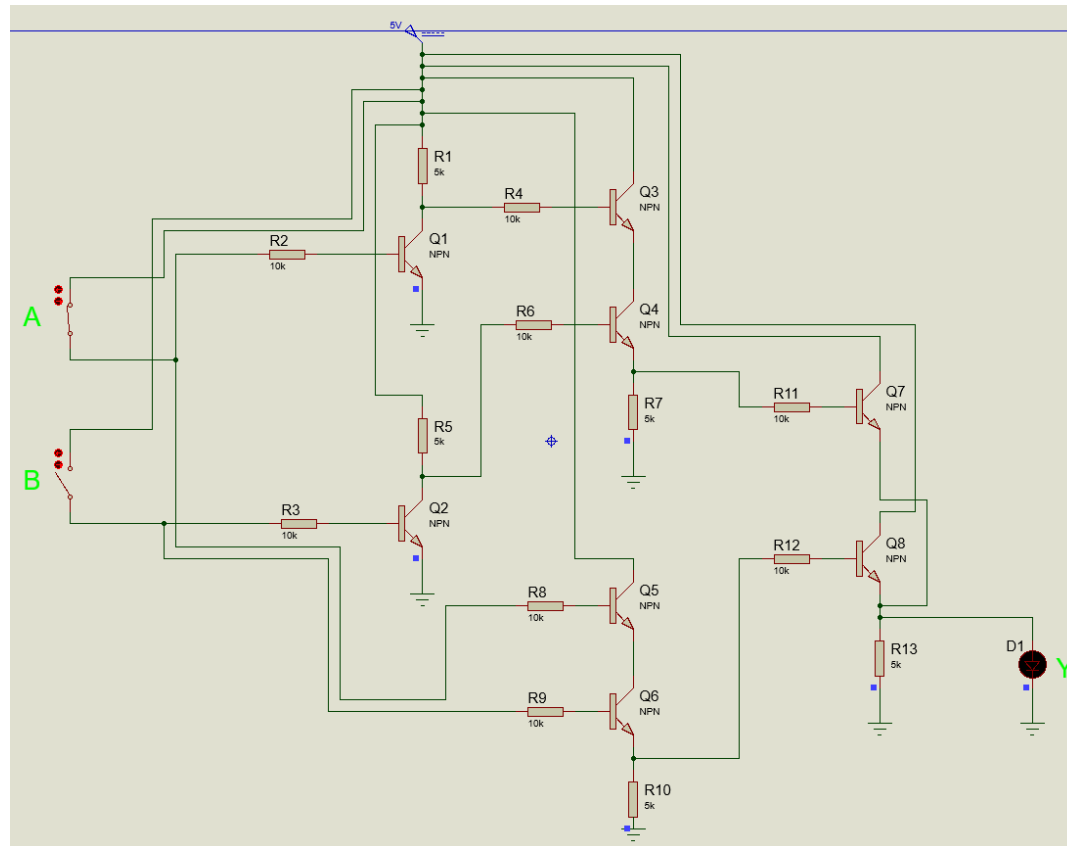
## Construcția portilor logice

### 7. XOR

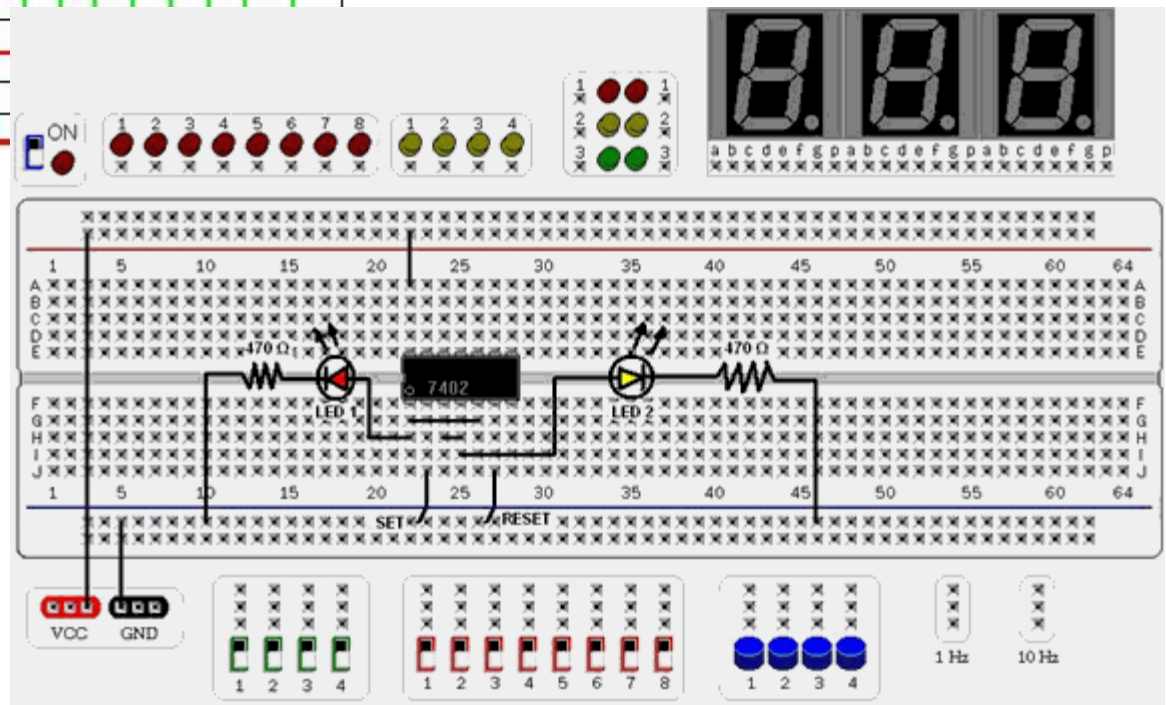
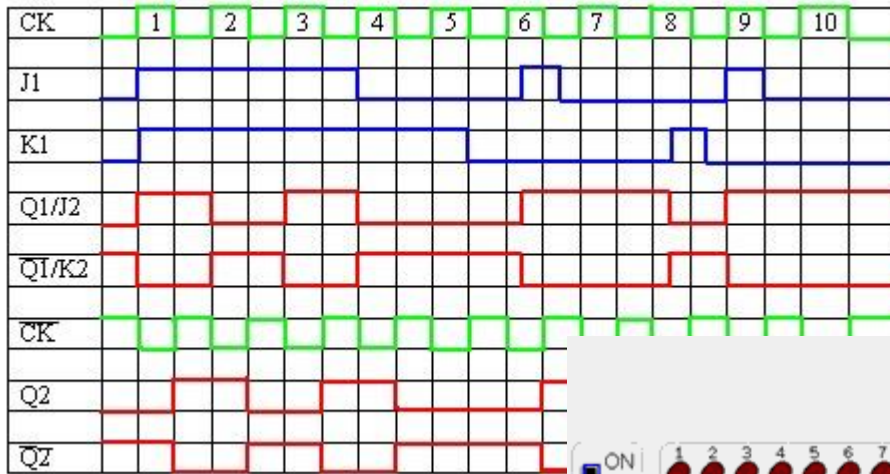


## Construcția portilor logice

### 8. XNOR



# Functii Logice in Algebra Booleana





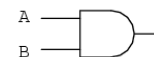
**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

**Facultatea de Construcții de Mașini**  
**Departamentul Ingineria Proiectării și Robotică**

# Functii Logice in Algebra Booleana



| A | B | Output |
|---|---|--------|
| 0 | 0 | 0      |
| 0 | 1 | 0      |
| 1 | 0 | 0      |
| 1 | 1 | 1      |



**Titular curs**  
**Sl. Dr. Ing. Covaciu Florin**

**Anul 2020**



## Funcții logice

- ❖ la nivel logic, elementar, operațiile circuitelor logice de baza sunt exact operațiile din logica booleană
- ❖ toate operațiile aritmetice, se efectuează în baza 2 (sistem binar)
- ❖ funcțiile booleene sunt implementate la nivel logic cu ajutorul unor circuite speciale numite: **circuite combinatoriale**

## Functii logice

Se numeste **functie booleana**, o funcție  $f: B^n \rightarrow B$

Este o functie de  $n$  variabile binare  $y = f(x_1, x_2, x_3),$

unde  $x_i$   
sunt variabile de intrare.

Atât functia cât si variabilele nu pot lua decât doua valori distincte, 0 sau 1.

## Funcții logice

Funcțiile logice pot fi exprimate în două modalități:

- o expresie booleană. finită, dar nu unică

$$f(x,y) = 2x + y$$

$$= x + x + y$$

$$= 2(x + y/2)$$

$$= \dots$$

- tabela de adevăr – unică și finită

| x   | y   | f(x,y) |
|-----|-----|--------|
| 0   | 0   | 0      |
| ... | ... | ...    |
| 2   | 2   | 6      |
| ... | ... | ...    |
| 23  | 41  | 87     |
| ... | ... | ...    |

## Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

# Operatii cu variabile binare

**Operatie:**

AND (produs)  
doua intrari

OR (Suma)  
doua intrari

NOT (complement)  
o intrare

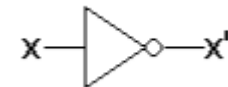
**Expresie:**

$$x \cdot y$$

$x+y$

$$\overline{x}$$

**Poarta:**



***Tabela  
de adeviar***

| x | y | xy |
|---|---|----|
| 0 | 0 | 0  |
| 0 | 1 | 0  |
| 1 | 0 | 0  |
| 1 | 1 | 1  |

| x | y | $x+y$ |
|---|---|-------|
| 0 | 0 | 0     |
| 0 | 1 | 1     |
| 1 | 0 | 1     |
| 1 | 1 | 1     |

| x | NOTx |
|---|------|
| 0 | 1    |
| 1 | 0    |

## Aplicatie cu variabile binare

Consideram un rezervor alimentat de trei robinete  $x$ ,  $y$  și  $z$ , care are scurgerea deschisa.

Ne propunem sa mentinem rezervorul plin cu ajutorul acestor trei robinete.

Rezervorul poate fi mentinut plin daca cel putin doua robinete sunt deschise simultan.

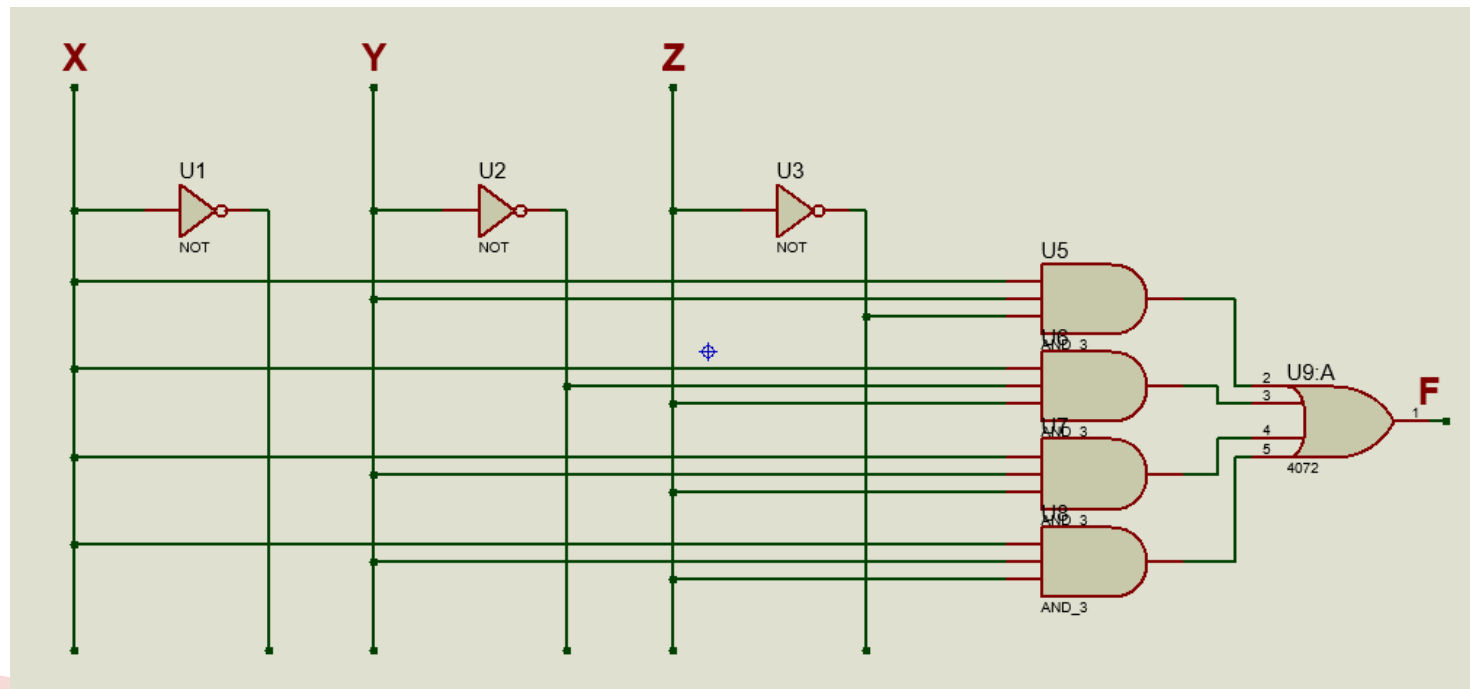
Un robinet deschis are atribuita valoarea logica 1, atunci functia care descrie din punct de vedere logic aceasta aplicatie este urmatoarea:

$$f(x, y, z) = xy\bar{z} + x\bar{y}z + \bar{x}yz + xyz$$

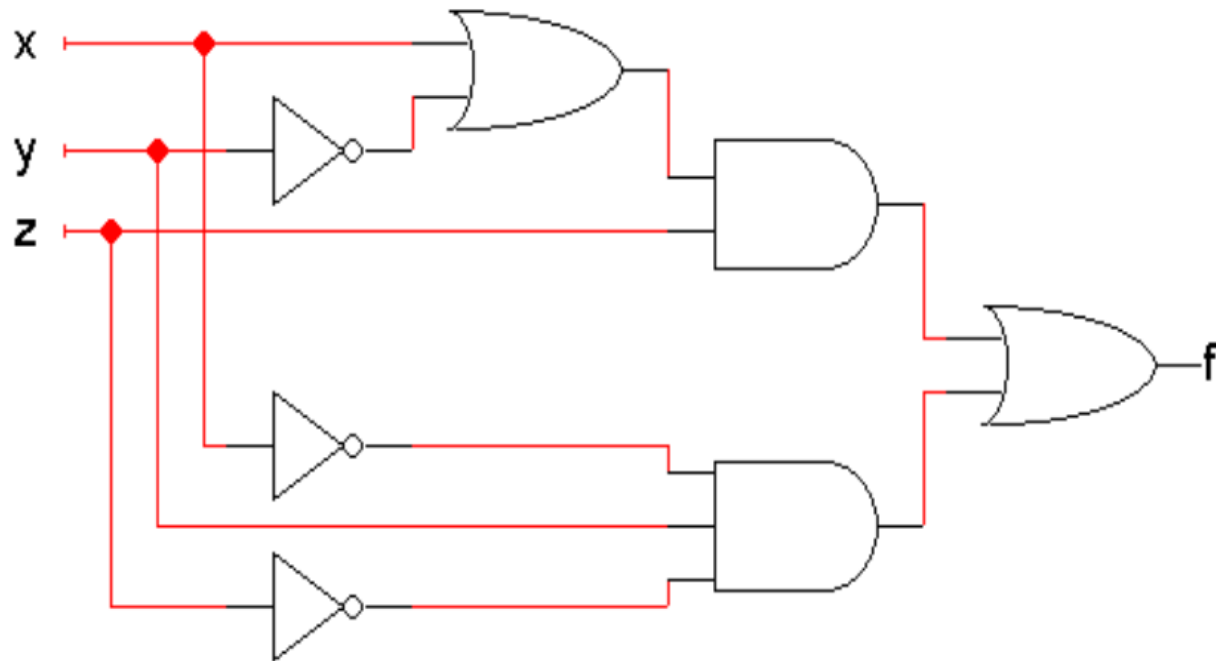
## Aplicatie cu variabile binare

O expresie booleana poate fi convertita intr-un circuit prin combinarea portilor elementare

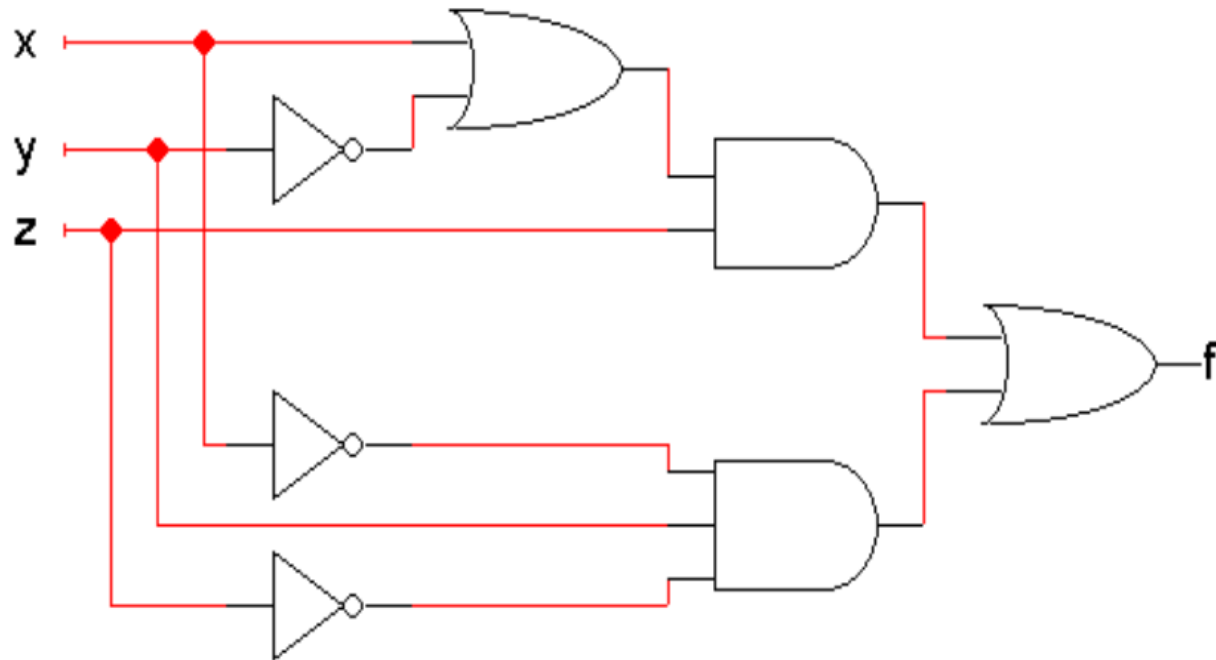
$$f(x, y, z) = xy\bar{z} + x\bar{y}z + \bar{x}yz + xyz$$



## Aplicatie cu variabile binare



## Aplicatie cu variabile binare



$$((x + \bar{y}) \cdot z) + (\bar{x} \cdot y \cdot \bar{z})$$



## Expresii booleene

Cu ajutorul operatiilor de baza se pot forma expresii (functii) booleene

$$f(x, y, z) = (x + \overline{y})z + \overline{x}$$

### **Obs.**

f – numele functie

$$\overline{x} = NOT \ x$$

(x,y,z) – variabile de intrare (1 sau 0)

- notarea complementului

## Expresii booleene

Cu ajutorul operațiilor de baza se pot forma expresii (functii) booleene

$$f(x, y, z) = (x + \overline{y})z + \overline{x}$$

### ***Ordonarea operațiilor:***

NOT urmat de AND (+), apoi OR (·)

Scrierea in paranteze:

$$\overline{x} = NOT \ x$$

$$f(x, y, z) = (((x + \overline{y}))z + \overline{x})$$

## Tabele de adevar

Cu ajutorul tabelii de adevar se vizualizeaza combinatiile valorilor de intrare si rezultatul la iesire.

- O tabela de adevar este finita deoarece prezinta un numar finit de valori (1 sau 0)
- O functie cu  $n$  variabile de intrare permite  $2^n$  combinatii posibile.

Intrarile sunt afisate in ordine binara crescatoare (ex. de la 000 la 111)



$$f(x, y, z) = (x + \bar{y})z + \bar{x}$$

$$f(0, 0, 0) = (0 + 1)0 + 1 = 1$$

$$f(0, 0, 1) = (0 + 1)1 + 1 = 1$$

$$f(0, 1, 0) = (0 + 0)0 + 1 = 1$$

$$f(0, 1, 1) = (0 + 0)1 + 1 = 1$$

$$f(1, 0, 0) = (1 + 1)0 + 0 = 1$$

$$f(1, 0, 1) = (1 + 1)1 + 0 = 1$$

$$f(1, 1, 0) = (1 + 0)0 + 0 = 1$$

$$f(1, 1, 1) = (1 + 0)1 + 0 = 1$$

## Tabele de adevar

$$f(x, y, z) = (x + \bar{y})z + \bar{x}$$

$$f(0, 0, 0) = (0 + 1) 0 + 1 = 1$$

$$f(0, 0, 1) = (0 + 1) 1 + 1 = 1$$

$$f(0, 1, 0) = (0 + 0) 0 + 1 = 1$$

$$f(0, 1, 1) = (0 + 0) 1 + 1 = 1$$

$$f(1, 0, 0) = (1 + 1) 0 + 0 = 1$$

$$f(1, 0, 1) = (1 + 1) 1 + 0 = 1$$

$$f(1, 1, 0) = (1 + 0) 0 + 0 = 1$$

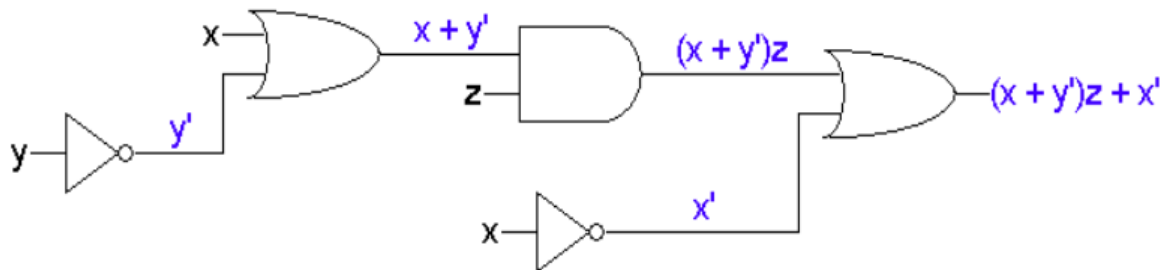
$$f(1, 1, 1) = (1 + 0) 1 + 0 = 1$$

| x | y | z | f |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |

## Expresii si circuite

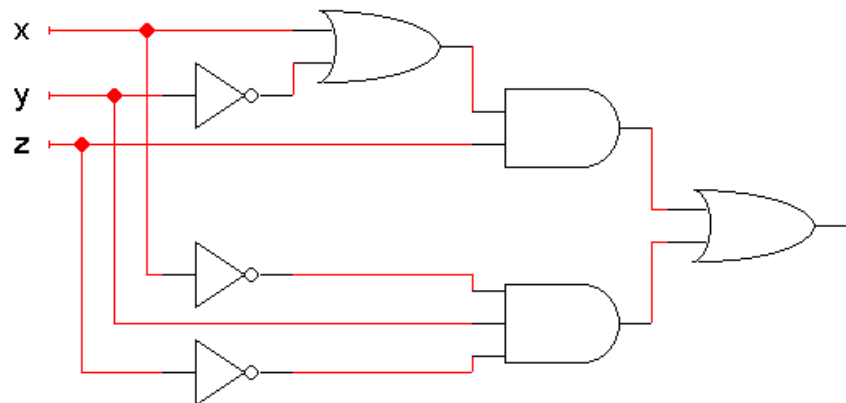
- Cu ajutorul portilor logice expresia booleeana este convertita intr-un circuit
- In circuit este explicita ordinea operatiilor

$$(x + \bar{y})z + \bar{x}$$



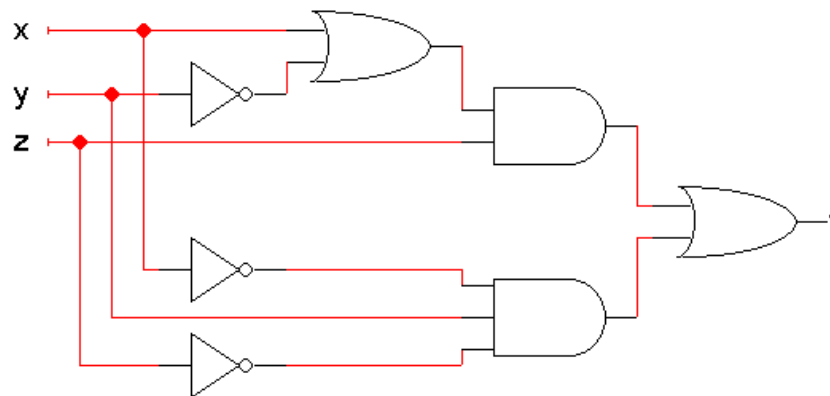
## Analiza circuitului

- O analiza de circuit se face pentru a explica modul de functionare a circuitelor din punct de vedere logic
  - ✓ Un circuit calculeaza o functie care poate fi descrisa ca o expresie booleeana sau prin tabela de adevar
  - ✓ Scopul este de a gasi o expresie echivalenta sau tabela de adevar a circuitului
- 1. Trebuie sa se stabileasca clar marimile de intrare si de iesire pentru circuit



## Ecuatiile algebrice

2. In functie de intrari se scriu expresiile la iesirea fiecărei porti logice
  - Se incepe de la intrari si se termina la iesiri
  - Se prefera efectuări de simplificări algebrice
3. Tabela de adevar se poate obtine direct din circuit
  - Daca se cunosc numarul de intrari si iesiri, se afiseaza toate combinatiile de care sunt posibile in tabela de adevar
    - Un circuit cu  $n$  intrari va avea o tabela de adevar cu  $2^n$  linii
    - In exemplu de mai jos este un circuit cu 3 intrari, tabela de adevar are  $2^3 = 8$  linii



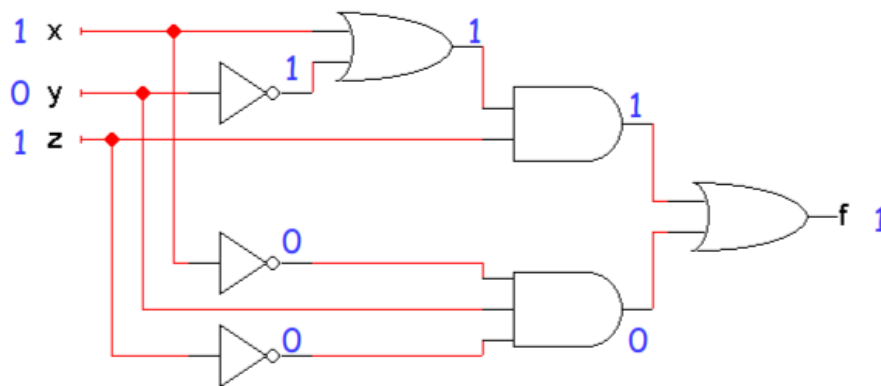
| x | y | z | f |
|---|---|---|---|
| 0 | 0 | 0 |   |
| 0 | 0 | 1 |   |
| 0 | 1 | 0 |   |
| 0 | 1 | 1 |   |
| 1 | 0 | 0 |   |
| 1 | 0 | 1 |   |
| 1 | 1 | 0 |   |
| 1 | 1 | 1 |   |

## Simularea circuitului

- Pentru simularea circuitului se poate folosi un program dedicat, sau poate fi simulat manual pentru a fi găsite stările ieșirilor pentru fiecare combinație posibilă

Se da un exemplu, când  $xyz = 101$ , ieșirile portilor arata astfel:

- Se folosesc tabele de adevar pentru AND, OR si NOT pentru a fi găsite ieșirile portilor
- Ieșirea finală este  $f(1, 0, 1) = 1$

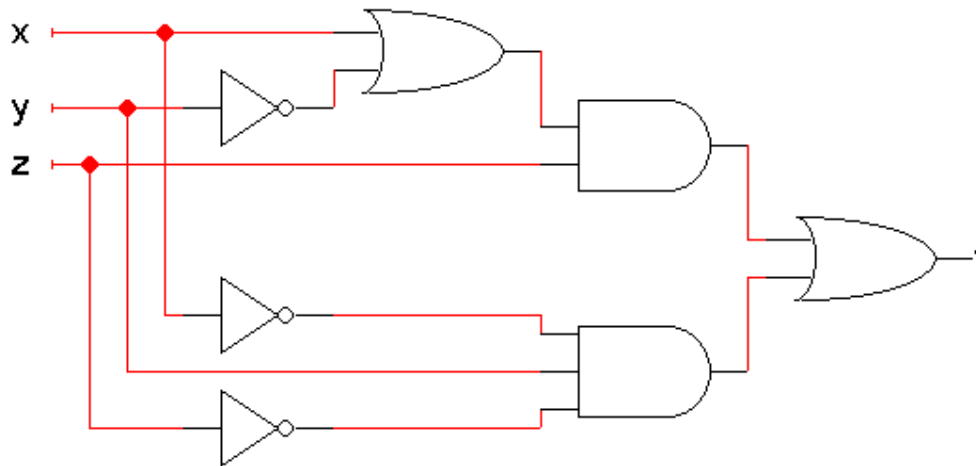


| x | y | z | f |
|---|---|---|---|
| 0 | 0 | 0 |   |
| 0 | 0 | 1 |   |
| 0 | 1 | 0 |   |
| 0 | 1 | 1 |   |
| 1 | 0 | 0 |   |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 |   |
| 1 | 1 | 1 |   |



## Simularea circuitului, finalizarea tabelului de adevar

- In mod identic se va proceda cu toate combinatiile posibile, completandu-se tabela de adevar
- Procesul nu este greu, dar indelungat



| x | y | z | f |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

## Expresii si tabele de adevar

- Tabela de adevar se poate calcula usor daca expresia booleeana exista
- In exemplu folosit s-a gasit ca circuitul calculeaza functia:  $f(x, y, z) = xz + \bar{y}z + \bar{x}y\bar{z}$   
se completeaza tabelul:

| x | y | z | xz | y'z | x'yz' | f |
|---|---|---|----|-----|-------|---|
| 0 | 0 | 0 | 0  | 0   | 0     | 0 |
| 0 | 0 | 1 | 0  | 1   | 0     | 1 |
| 0 | 1 | 0 | 0  | 0   | 1     | 1 |
| 0 | 1 | 1 | 0  | 0   | 0     | 0 |
| 1 | 0 | 0 | 0  | 0   | 0     | 0 |
| 1 | 0 | 1 | 1  | 1   | 0     | 1 |
| 1 | 1 | 0 | 0  | 0   | 0     | 0 |
| 1 | 1 | 1 | 1  | 0   | 0     | 1 |

## Expresii si tabele de adevar

- Iar daca avem tabela de adevar se poate gasi usor expresia booleana
- Tabela de adevar poate fi convertita într-o suma de mintermeni care corespund liniilor din tabela a caror valoare la iesire este **1**

| x | y | z | f |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 |
| 1 | 1 | 1 | 1 |

$$f(x, y, z) = \overline{x}\overline{y}z + \overline{x}y\overline{z} + x\overline{y}z + xyz$$

- Aceasta suma de mintermeni poate fi simplificata cu harta Karnaught

## Simplificarea expresiilor

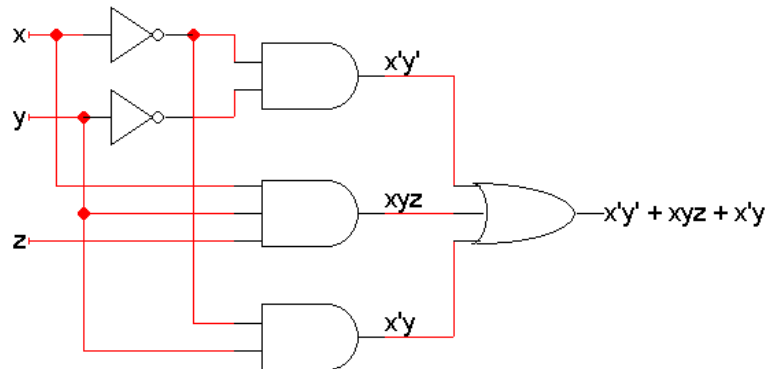
- Se pot face simplificari in expresia booleana:

$$\begin{aligned}\overline{x}\overline{y} + xyz + \overline{x}y \\&= \overline{x}(\overline{y} + y) \\&+ xyz \\&= \overline{x} \cdot 1 + xyz \\&= \overline{x} + xyz \\&= (\overline{x} + x)(\overline{x} \\&+ yz) \\&= 1 \cdot (\overline{x} + yz) \\&= \overline{x} + yz\end{aligned}$$

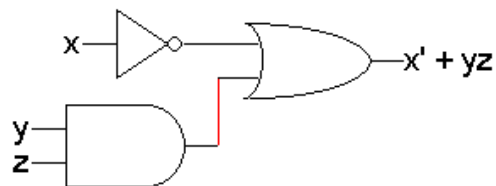
$$\begin{aligned}(\text{Distributivitatea: } \overline{x}\overline{y} + \overline{x}y \\&= \overline{x}(\overline{y} + y)) \\&(\overline{y} + y = 1) \\&(\overline{x} \cdot 1 = \overline{x}) \\&(\text{Distributivitate}) \\&(\overline{x} + x = 1)\end{aligned}$$

## Circuite echivalente

- Exista doua circuite echivalente al expresiei booleene



- Circuitul cu cele mai putine porti este cel mai optim
  - este mai fiabil
  - consuma mai putina energie
  - este mai ieftin



## Complementul unei functii

- Prin complementul unei functii se intelege negarea rezultatului functiei, adica in coloana de iesire al tabelului de adevar 0-urile si 1-urile se interschimba

$$f(x, y, z) = x(\overline{y}z + yz)$$

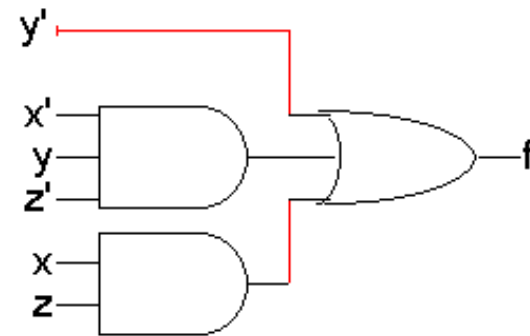
| x | y | z | f(x,y,z) |
|---|---|---|----------|
| 0 | 0 | 0 | 1        |
| 0 | 0 | 1 | 1        |
| 0 | 1 | 0 | 1        |
| 0 | 1 | 1 | 1        |
| 1 | 0 | 0 | 0        |
| 1 | 0 | 1 | 0        |
| 1 | 1 | 0 | 1        |
| 1 | 1 | 1 | 0        |



| x | y | z | f'(x,y,z) |
|---|---|---|-----------|
| 0 | 0 | 0 | 0         |
| 0 | 0 | 1 | 0         |
| 0 | 1 | 0 | 0         |
| 0 | 1 | 1 | 0         |
| 1 | 0 | 0 | 1         |
| 1 | 0 | 1 | 1         |
| 1 | 1 | 0 | 0         |
| 1 | 1 | 1 | 1         |

## Formele standard ale unei expresii

- In mai multe moduri poate fi scrisa o expresie, unele pot fi mai folositoare decat altele
- Notatia  $\Sigma$  sau suma de produse (sum of products SOP) contine:
  - Numai operatii OR la nivelul cel mai apropiat de iesire
  - Este un produs de variabile fiecare termen al sumei
$$f(x, y, z) = \bar{y} + \bar{x}y\bar{z} + xz$$
- Notatia  $\Sigma$  are un avantaj major si anume ca implementarea se face pe un circuit pe doua nivele
  - nivel 0: variabilele si complementele lor
  - nivel 1: porti AND
  - nivel 2: o singura poarta OR



**Obs.:** portile NOT sunt implicate iar variabilele apar de mai multe ori la intrare

## Mintermeni

- Un mintermen este un produs special de variabile în care fiecare variabila apare o singura data
- O funcție cu  $n$  variabile are  $2^n$  mintermeni
- Fiecare mintermen este adevărat pentru o singură combinație de intrări:

| Mintermen                   | Adevărat dacă         | Notatie |
|-----------------------------|-----------------------|---------|
| $\overline{xyz}$            | $x = 0, y = 0, z = 0$ | $m_0$   |
| $\overline{x}yz$            | $x = 0, y = 0, z = 1$ | $m_1$   |
| $\overline{x}y\overline{z}$ | $x = 0, y = 1, z = 0$ | $m_2$   |
| $\overline{x}yz$            | $x = 0, y = 1, z = 1$ | $m_3$   |
| $x\overline{y}\overline{z}$ | $x = 1, y = 0, z = 0$ | $m_4$   |
| $x\overline{y}z$            | $x = 1, y = 0, z = 1$ | $m_5$   |
| $x\overline{y}z$            | $x = 1, y = 1, z = 0$ | $m_6$   |
| $xy\overline{z}$            | $x = 1, y = 1, z = 1$ | $m_7$   |
| $xyz$                       |                       |         |



## Suma de mintermeni

- Ca o suma de mintermeni orice functie poate fi scrisa
- Notatia  $\Sigma$  este unica pentru o functie
- In tabelul de adevar pentru o functie, suma de mintermeni poate fi scrisa luand in considerare doar liniile care au la iesire 1 logic:

| x | y | z | f(x,y,z) | f'(x,y,z) |
|---|---|---|----------|-----------|
| 0 | 0 | 0 | 1        | 0         |
| 0 | 0 | 1 | 1        | 0         |
| 0 | 1 | 0 | 1        | 0         |
| 0 | 1 | 1 | 1        | 0         |
| 1 | 0 | 0 | 0        | 1         |
| 1 | 0 | 1 | 0        | 1         |
| 1 | 1 | 0 | 1        | 0         |
| 1 | 1 | 1 | 0        | 1         |

$$\begin{aligned}f &= \overline{x}\overline{y}\overline{z} + \overline{x}\overline{y}z + \overline{x}y\overline{z} + \overline{x}yz + x\overline{y}\overline{z} \\&= m_0 + m_1 + m_2 + m_3 + m_6 \\&= \Sigma m(0,1,2,3,6)\end{aligned}$$

$$\begin{aligned}\overline{f} &= x\overline{y}\overline{z} + x\overline{y}z + xyz \\&= m_4 + m_5 + m_7 \\&= \Sigma m(4,5,7)\end{aligned}$$

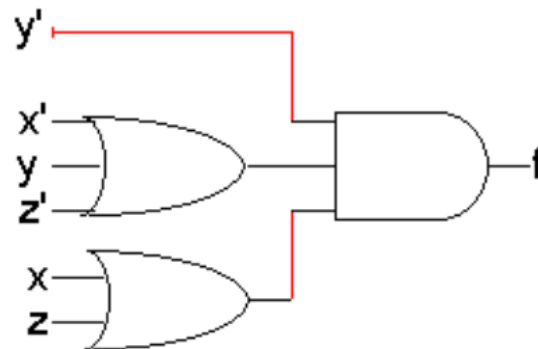
$\overline{f}$  contine mintermenii care nu se gasesc in f

## Produsul de sume – notatia $\Pi$

- Exista si posibilitatea duala, produsul de sume, care contine:
  - Numai operatii AND la nivel final
  - Fiecare termen este o suma de variabile

$$f(x, y, z) = \bar{y}(\bar{x} + y + \bar{z})(x + z)$$

- Implementarea se realizeaza cu un circuit pe doua nivele
  - nivel 0: intrari si complementele lor
  - nivel 1: cu porti OR
  - nivel 2: cu o singura poarta AND



## Maxtermeni

- Un maxtermen este o suma de variabile în care fiecare intrare apare o singura data
- O funcție cu  $n$  variabile are  $2^n$  maxtermeni
- Fiecare maxtermen este fals pentru o singura combinație de intrari:

| Maxtermen                     | Fals daca             | Notatie |
|-------------------------------|-----------------------|---------|
| $x + y + z$                   | $x = 0, y = 0, z = 0$ | $M_0$   |
| $x + y + \bar{z}$             | $x = 0, y = 0, z = 1$ | $M_1$   |
| $x + \bar{y} + z$             | $x = 0, y = 1, z = 0$ | $M_2$   |
| $x + \bar{y} + \bar{z}$       | $x = 0, y = 1, z = 1$ | $M_3$   |
| $\bar{x} + y + z$             | $x = 1, y = 0, z = 0$ | $M_4$   |
| $\bar{x} + y + \bar{z}$       | $x = 1, y = 0, z = 1$ | $M_5$   |
| $\bar{x} + \bar{y} + z$       | $x = 1, y = 1, z = 0$ | $M_6$   |
| $\bar{x} + \bar{y} + \bar{z}$ | $x = 1, y = 1, z = 1$ | $M_7$   |

## Produs de maxtermeni

- Orice funcție poate fi scrisă ca un produs unic de maxtermeni
- Datele tabelului de adevăr al unei funcții, expresia funcției în notația  $\Pi$  se obține luând liniile din tabel care au 0 la ieșire

| x | y | z | f(x,y,z) | f'(x,y,z) |
|---|---|---|----------|-----------|
| 0 | 0 | 0 | 1        | 0         |
| 0 | 0 | 1 | 1        | 0         |
| 0 | 1 | 0 | 1        | 0         |
| 0 | 1 | 1 | 1        | 0         |
| 1 | 0 | 0 | 0        | 1         |
| 1 | 0 | 1 | 0        | 1         |
| 1 | 1 | 0 | 1        | 0         |
| 1 | 1 | 1 | 0        | 1         |

$$f = (\bar{x} + y + z)(\bar{x} + y + \bar{z})(\bar{x} + \bar{y} + \bar{z})$$

$$= M_4 M_5 M_7$$

$$= \Pi M(4,5,7)$$

$$f' = (x + y + z)(x + y + \bar{z})(x + \bar{y} + z)(x + \bar{y} + \bar{z})(\bar{x} + \bar{y} + z)$$

$$= M_0 M_1 M_2 M_3 M_6$$

$$= \Pi M(0,1,2,3,6)$$

$\bar{f}$  conține maxtermeni care nu se găsesc în  $f$

## Reprezentarea funcțiilor logice

În mod curent se folosesc trei metode pentru reprezentarea funcțiilor logice:

### 1. Prin tabela de adevăr

Această reprezentare presupune marcarea, într-un tabel, a corespondenței dintre valorile de adevăr ale variabilelor de intrare și valoarea de adevăr a funcției în fiecare punct al domeniului de definiție.

## Reprezentarea funcțiilor logice

### Operatori logici de bază

| Operator:            | ȘI                         |          | SAU                       |        | NU        |        |
|----------------------|----------------------------|----------|---------------------------|--------|-----------|--------|
| Notăție              | *                          | $\wedge$ | +                         | $\vee$ | $\bar{x}$ | $\neg$ |
| Denumire alternativă | Produs logic<br>Conjuncție |          | Sumă logică<br>Disjuncție |        |           |        |

### SI

| Tabel de adevăr                                                                                                                                                                                                    | Expresie booleană |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------------|
| <table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>0</td></tr><tr><td>1</td><td>0</td><td>0</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> | A                 | B | Y | 0 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 1 | 1 | $Y = A * B$ |
| A                                                                                                                                                                                                                  | B                 | Y |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 0                                                                                                                                                                                                                  | 0                 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 0                                                                                                                                                                                                                  | 1                 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 1                                                                                                                                                                                                                  | 0                 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 1                                                                                                                                                                                                                  | 1                 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |             |

### NU

| Tabel de adevăr                                                                                                                                          | Expresie booleană |   |   |   |   |   |               |
|----------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---------------|
| <table border="1"> <thead> <tr> <th>A</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>1</td></tr> <tr><td>1</td><td>0</td></tr> </tbody> </table> | A                 | Y | 0 | 1 | 1 | 0 | $Y = \bar{A}$ |
| A                                                                                                                                                        | Y                 |   |   |   |   |   |               |
| 0                                                                                                                                                        | 1                 |   |   |   |   |   |               |
| 1                                                                                                                                                        | 0                 |   |   |   |   |   |               |

### SAU

| Tabel de adevăr                                                                                                                                                                                                    | Expresie booleană |   |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|-------------|
| <table><tr><th>A</th><th>B</th><th>Y</th></tr><tr><td>0</td><td>0</td><td>0</td></tr><tr><td>0</td><td>1</td><td>1</td></tr><tr><td>1</td><td>0</td><td>1</td></tr><tr><td>1</td><td>1</td><td>1</td></tr></table> | A                 | B | Y | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | $Y = A + B$ |
| A                                                                                                                                                                                                                  | B                 | Y |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 0                                                                                                                                                                                                                  | 0                 | 0 |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 0                                                                                                                                                                                                                  | 1                 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 1                                                                                                                                                                                                                  | 0                 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |             |
| 1                                                                                                                                                                                                                  | 1                 | 1 |   |   |   |   |   |   |   |   |   |   |   |   |   |             |

## Reprezentarea funcțiilor logice

### Exemple

a. Cu funcția **NU**, o singură intrare

| b | $\sim b$ |
|---|----------|
| 0 | 1        |
| 1 | 0        |

b. Funcția **SI**, cu două intrări

|   | a | b | $a \wedge b$ |
|---|---|---|--------------|
| 0 | 0 | 0 | 0            |
| 1 | 0 | 1 | 0            |
| 2 | 1 | 0 | 0            |
| 3 | 1 | 1 | 1            |

$2^2=4$

## Reprezentarea funcțiilor logice

### Exemple

c. Funcția **SI**, cu trei intrări

|   | a | b | c | $a \wedge b \wedge c$ |
|---|---|---|---|-----------------------|
| 0 | 0 | 0 | 0 | 0                     |
| 1 | 0 | 0 | 1 | 0                     |
| 2 | 0 | 1 | 0 | 0                     |
| 3 | 0 | 1 | 1 | 0                     |
| 4 | 1 | 0 | 0 | 0                     |
| 5 | 1 | 0 | 1 | 0                     |
| 6 | 1 | 1 | 0 | 0                     |
| 7 | 1 | 1 | 1 | 1                     |

$2^3=8$



## Reprezentarea funcțiilor logice

### Exemple

d. Cu combinațiile ale unor două funcții (**SI**, **SAU**), cu trei intrări

$2^3=8$

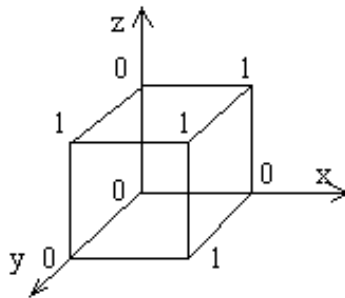
$a \wedge b \vee (b \wedge c)$

|   | a | b | c | $b \wedge c$ | $a \wedge b$ | v |
|---|---|---|---|--------------|--------------|---|
| 0 | 0 | 0 | 0 | 0            | 0            | 0 |
| 1 | 0 | 0 | 1 | 0            | 0            | 0 |
| 2 | 0 | 1 | 0 | 0            | 0            | 0 |
| 3 | 0 | 1 | 1 | 1            | 0            | 1 |
| 4 | 1 | 0 | 0 | 0            | 0            | 0 |
| 5 | 1 | 0 | 1 | 0            | 0            | 0 |
| 6 | 1 | 1 | 0 | 0            | 1            | 1 |
| 7 | 1 | 1 | 1 | 1            | 1            | 1 |

## Reprezentarea funcțiilor logice

### 2. Prin diagrame Karnaugh

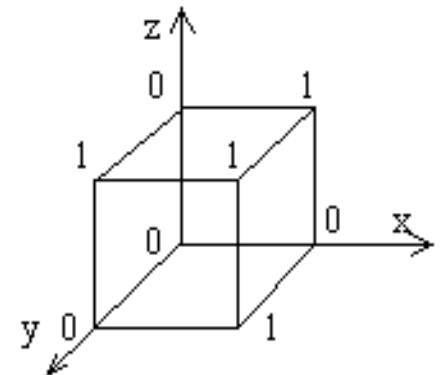
Reprezentarea prin diagrame *Karnaugh* consta în a marca punctele domeniului de definiție într-o diagrama plana si a preciza valoarea functiei în fiecare din aceste puncte.



| $x \backslash yz$ | 00 | 01 | 11 | 10 |
|-------------------|----|----|----|----|
| 0                 | 0  | 0  | 1  | 0  |
| 1                 | 0  | 1  | 1  | 1  |

## Reprezentarea funcțiilor logice

Daca se ia în considerare vârful cubului caracterizat prin coordonatele 000, constatam ca acest vârf este vecin cu vârfurile 001, 010, 100. În diagrama *Karnaugh* constatam ca 000 este vecin doar cu 001 si 100. Pentru ca diagrama *Karnaugh* sa fie echivalenta cu reprezentarea prin cub, ea trebuie sa pastreze acelasi vecinatati, lucru ce devine posibil doar daca ne imaginam latura din stânga a diagramei *Karnaugh* în continuarea celei din dreapta, iar latura de sus în continuarea celei de jos. În acest fel, punctul 000 devine vecin si cu punctul 010.



## Reprezentarea funcțiilor logice

### Exemple:

a1. Diagrama Karnaugh cu două intrări

- se va completa următoarea funcție în tabelul de adevăr:

$$F(A, B) = \sum n(2,3)$$

| A | B | F |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |



|   |   | A | 0 | 1 |
|---|---|---|---|---|
| B | 0 | 0 | 0 | 1 |
|   | 1 | 0 | 0 | 1 |

Rezultat:

$$F = A$$

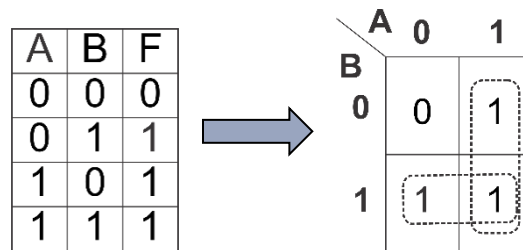
## Reprezentarea funcțiilor logice

### Exemple:

a2. Diagrama Karnaugh cu două intrări

- se va completa următoarea funcție în tabelul de adevăr:

$$F(A, B) = \sum n(1, 2, 3)$$



Rezultat:

$$F = A + B$$

## Reprezentarea funcțiilor logice

### Exemple:

b1. Diagrama Karnaugh cu trei intrări

- se va completa următoarea funcție în tabelul de adevăr:

$$F(A, B, C) = \sum n(0, 1, 2, 3, 4, 6)$$

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 1 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 0 |



|    |    | A |   |
|----|----|---|---|
|    |    | 0 | 1 |
| BC | 00 | 1 | 1 |
|    | 01 | 1 | 1 |
|    | 11 | 1 | 0 |
|    | 10 | 1 | 0 |

Rezultat:

$$F = \bar{A} + \bar{B}$$

## Reprezentarea funcțiilor logice

### Exemple:

b1. Diagrama Karnaugh cu trei intrări

- se va completa următoarea funcție în tabelul de adevăr:

$$F(A, B, C) = \sum n(2, 3, 4, 5, 6, 7)$$

| A | B | C | F |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 |
| 1 | 0 | 1 | 1 |
| 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 |



| A  |  |   |   |
|----|--|---|---|
| BC |  |   |   |
|    |  | 0 | 1 |
| 00 |  | 0 | 1 |
| 01 |  | 1 | 1 |
| 11 |  | 1 | 1 |
| 10 |  | 0 | 1 |

Rezultat:

$$F = A + C$$

## Reprezentarea funcțiilor logice

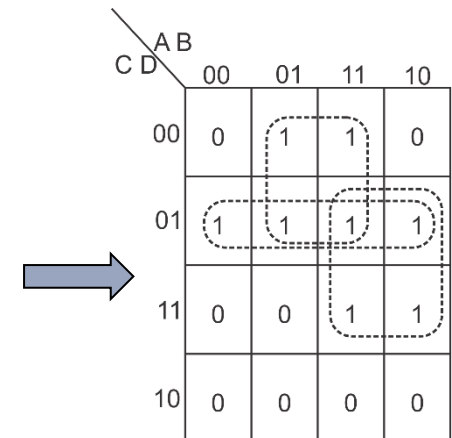
### Exemple:

C. Diagrama Karnaugh cu patru intrări

- se va completa următoarea funcție în tabelul de adevăr:

$$F(A, B, C, D) = \sum n(1, 3, 4, 5, 6, 7, 14, 15)$$

| A | B | C | D | F |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 |
| 0 | 0 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 | 1 |
| 0 | 1 | 1 | 0 | 1 |
| 0 | 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 0 |
| 1 | 0 | 1 | 1 | 0 |
| 1 | 1 | 0 | 0 | 0 |
| 1 | 1 | 0 | 1 | 0 |
| 1 | 1 | 1 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |



Rezultat:

$$F = \bar{C} * D + B * \bar{C} + A * D$$



## Reprezentarea funcțiilor logice

### 3. Reprezentarea prin echivalenți zecimali ai mintermilor și maxtermilor

Reprezentarea prin echivalenți zecimali ai mintermilor consta în indicarea echivalentilor zecimali ai conjuncțiilor pentru care valoarea este 1 sau a echivalentilor zecimali corespunzători valorii 0 pentru maxtermeni.

## Reprezentarea funcțiilor logice

### *Mintermeni*

#### Exemple:

Se da următoarea funcție:

$$F(x, y) = OR(m_1, m_3)$$

- să se reprezinte mintermenii

| x | y | F | $m_0$ | $m_1$ | $m_2$ | $m_3$ |
|---|---|---|-------|-------|-------|-------|
| 0 | 0 | 0 | 1     | 0     | 0     | 0     |
| 0 | 1 | 1 | 0     | 1     | 0     | 0     |
| 1 | 0 | 0 | 0     | 0     | 1     | 0     |
| 1 | 1 | 1 | 0     | 0     | 0     | 1     |

Rezultat:

$$F(x, y) = \bar{x}y + xy \quad - \text{Expresie canonica } \mathbf{SOP}(\text{Sum-of-Products})$$

## Reprezentarea funcțiilor logice

### Maxtermeni

#### Exemple:

Se dă următoarea funcție:

$$F(x, y) = \text{AND}(M_0, M_2)$$

- să se reprezinte maxtermenii

| x | y | F | M <sub>0</sub> | M <sub>1</sub> | M <sub>2</sub> | M <sub>3</sub> |
|---|---|---|----------------|----------------|----------------|----------------|
| 0 | 0 | 0 | 0              | 1              | 1              | 1              |
| 0 | 1 | 1 | 1              | 0              | 1              | 1              |
| 1 | 0 | 0 | 1              | 1              | 0              | 1              |
| 1 | 1 | 1 | 1              | 1              | 1              | 0              |

Rezultat:

$$F(x, y) = (x + y) \cdot (\bar{x} + y) \quad \text{- Expresie canonica } \textbf{POS}(\text{Product-of-Sums})$$

## Reprezentarea funcțiilor logice

### 4. Implementarea funcțiilor logice

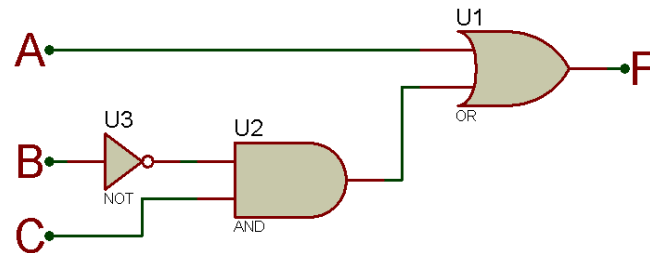
Implementarea unei funcții logice înseamnă realizarea ei cu ajutorul circuitelor (portilor) fundamentale.

## Reprezentarea funcțiilor logice

Exemple de implementare a funcțiilor logice:

a. Se dă următoarea funcție pentru implementare:

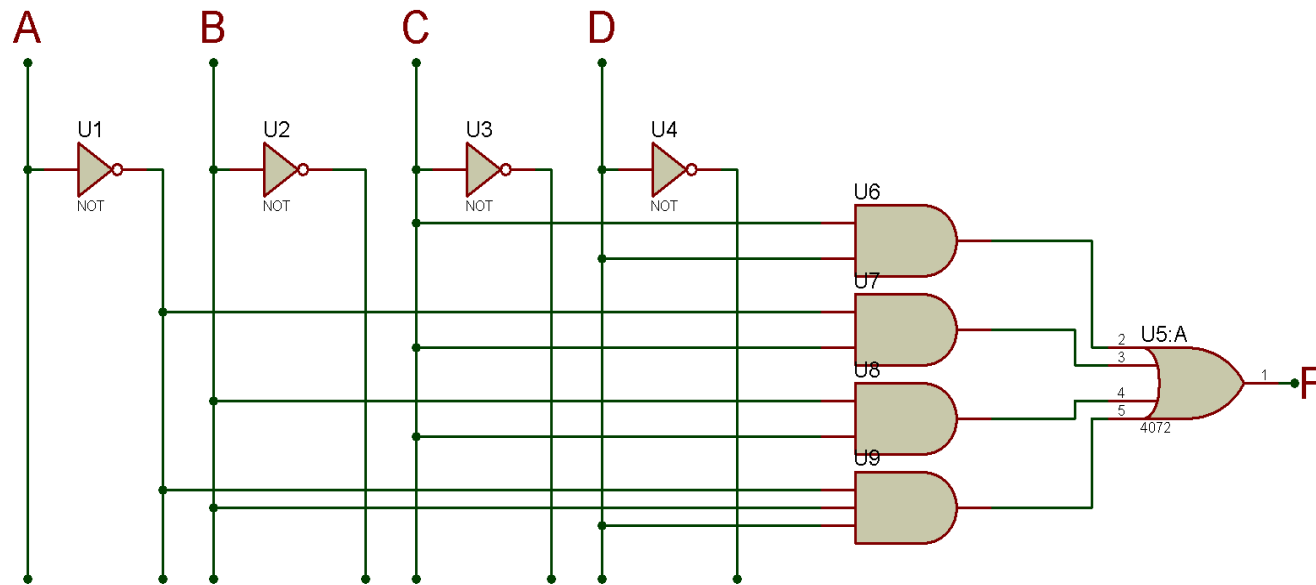
$$F = A + \overline{B}C$$



## Reprezentarea funcțiilor logice

b. Se dă următoarea funcție pentru implementare:

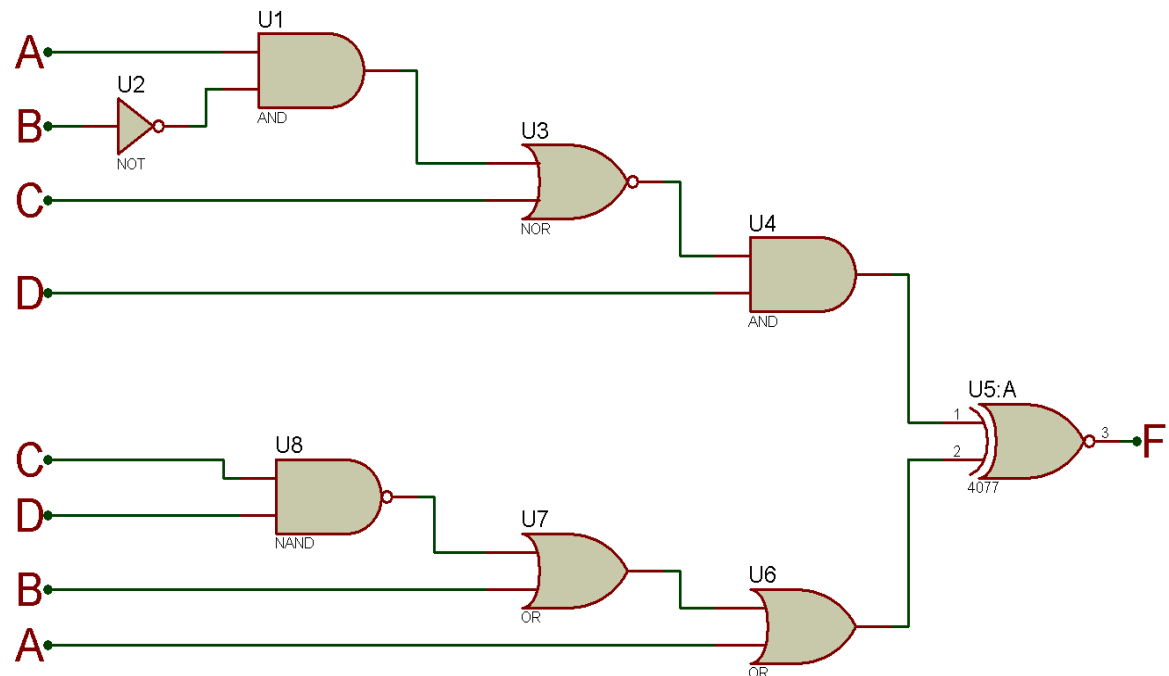
$$F = CD + \bar{A}C + BC + \bar{A}BD$$



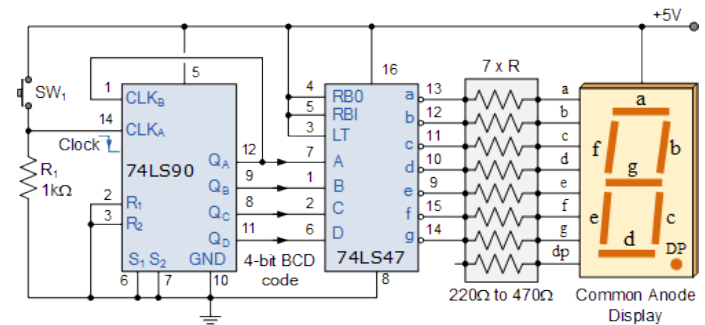
## Reprezentarea funcțiilor logice

C. Se dă următoarea funcție pentru implementare:

$$F = \overline{[(A\overline{B} + C)D] \oplus [(\overline{C}D + B) + A]}$$



# Coduri numerice de programare, conversie si verificare



Sl. Dr. Ing. Florin COVACIU



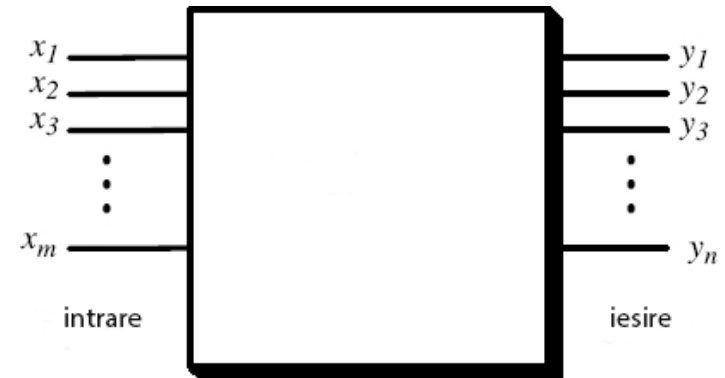
## Ciruite logice combinationale (CLC)

Prin circuit logic combinational (CLC) se intelege un circuit care posedă un număr de  $m$  intrări și are  $n$  ieșiri.

Transferul de la intrare la ieșire este unidirectional, fără reacție.

Algebric modelul este descris de tripleta  $[\mathbf{x}, \mathbf{y}, f]$  în care:

- vectorul de intrare  $\mathbf{x}$  cu  $\dim(\mathbf{x}) = m$ ;
- vectorul de ieșire  $\mathbf{y}$  cu  $\dim(\mathbf{y}) = n$ ;
- $f$  este o aplicație deterministă definită pe  $\mathbf{x}$  cu imagine în  $\mathbf{y}$ .



Componentele vectorilor  $\mathbf{x}$  și  $\mathbf{y}$  sunt semnale binare.

Din această categorie de circuite digitale fac parte:

- memorii;
- circuitele aritmetice;
- etc.

# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Sumatorul

**Sumatorul** este circuitul logic combinational care asigura efectuarea operatiilor aritmetice si logice într-un sistem de calcul.

Pentru a ajunge la conceptul de sumator trebuie definit însa termenul de semisumator.

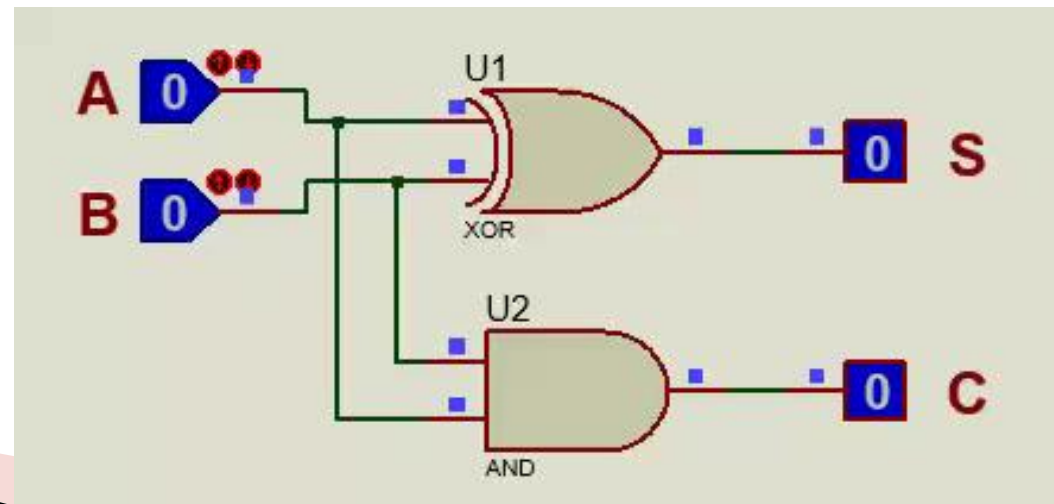
**Semisumatorul** este CLC-ul care serveste la efectuarea sumei a doua numere binare, de câte un bit, fara a tine seama de transportul de la bitul cu semnificatia imediat urmatoare.

| A | B | S | C |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

$$S = A \oplus B$$

$$C = A \cdot B$$

Film simulare



## Sumatorul complet de 1 bit

Exemplu: un sumator complet ce realizeaza suma a trei operanzi de 1 bit, notati:  $A_i, B_i, C_{i-1}$

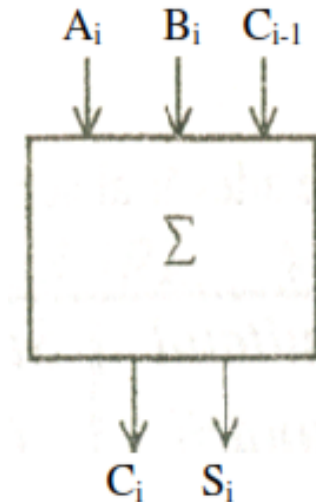
Sumatorul complet are doua iesiri reprezentate de catre semnalele:  $S_i$  si  $C_i$

Formule implementare:

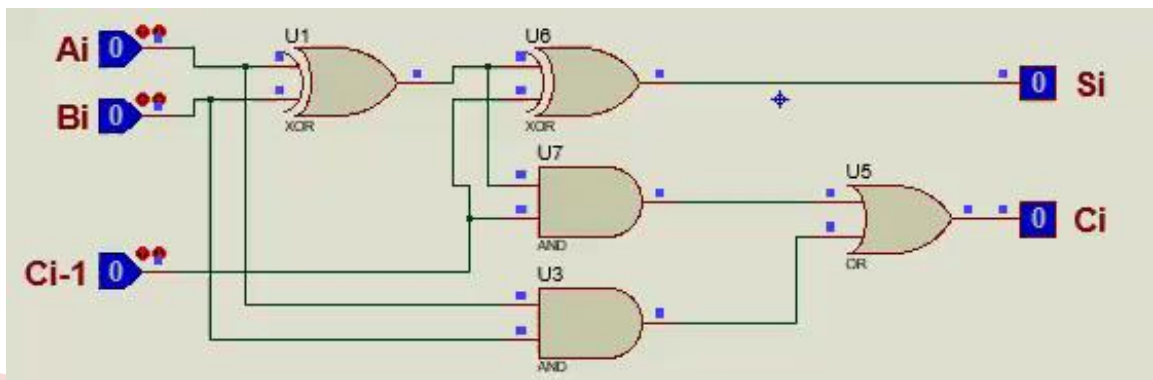
$$S_i = (A \oplus B) \oplus C_{i-1}$$

$$C_i = (A \cdot B) + (C_{i-1} \cdot (A \oplus B))$$

| Intrări |       |           | Suma | Ieșiri |       |
|---------|-------|-----------|------|--------|-------|
| $A_i$   | $B_i$ | $C_{i-1}$ |      | $S_i$  | $C_i$ |
| 0       | 0     | 0         | 00   | 0      | 0     |
| 0       | 0     | 1         | 01   | 1      | 0     |
| 0       | 1     | 0         | 01   | 1      | 0     |
| 0       | 1     | 1         | 10   | 0      | 1     |
| 1       | 0     | 0         | 01   | 1      | 0     |
| 1       | 0     | 1         | 10   | 0      | 1     |
| 1       | 1     | 0         | 10   | 0      | 1     |
| 1       | 1     | 1         | 11   | 1      | 1     |



Film simulare



## Codificatoarele

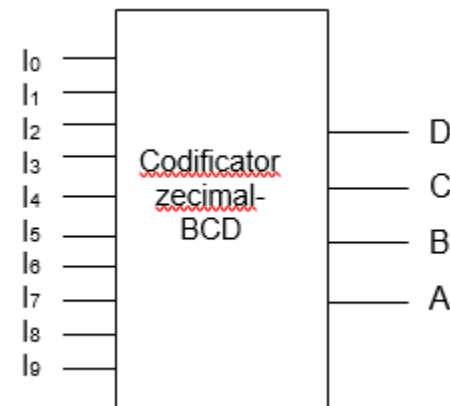
În multe aplicații practice este necesară conversia datelor dintr-un sistem de numerație în altul, cel mai frecvent este cel din sistemul zecimal în BCD și invers (**codificatoare și decodificatoare**).

**Codificatoarele** sunt circuite logice combinaționale la care activarea unei intrări conduce la apariția unui cuvânt de cod la ieșire.

Codificatorul **zecimal – BCD**, este un circuit al caror intrări sunt date în sistemul zecimal, iar ieșirile sunt date în codul BCD.

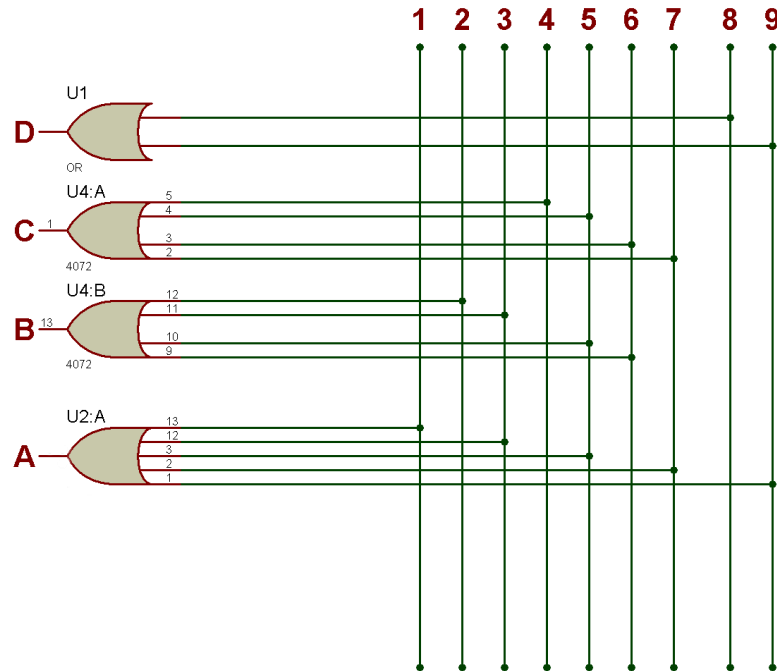
Codificatorul zecimal – BCD  
are 10 intrări, pentru cifrele de la 0 la 9 și  
4 ieșiri pentru cei 4 biți ai codului BCD.

Simbol



## Codificatoare

### Schema cu porți logice a codicatorului zecimal - BCD



Cele patru porți SAU pot să genereze la ieșiri zece combinații corespunzătoare reprezentării în codul BCD a cifrelor de la 0 la 9.

| Decimal | D | C | B | A |
|---------|---|---|---|---|
| 0       | 0 | 0 | 0 | 0 |
| 1       | 0 | 0 | 0 | 1 |
| 2       | 0 | 0 | 1 | 0 |
| 3       | 0 | 0 | 1 | 1 |
| 4       | 0 | 1 | 0 | 0 |
| 5       | 0 | 1 | 0 | 1 |
| 6       | 0 | 1 | 1 | 0 |
| 7       | 0 | 1 | 1 | 1 |
| 8       | 1 | 0 | 0 | 0 |
| 9       | 1 | 0 | 0 | 1 |

În tabelul de adevăr al codului BCD se caută nivelul înalt (1 logic), pe urmă se proiectează schema cu porți logice SAU.

$$D = 8 + 9$$

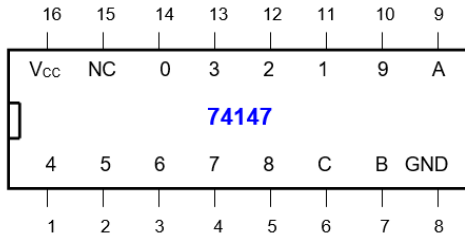
$$C = 4 + 5 + 6 + 7$$

$$B = 2 + 3 + 6 + 7$$

$$A = 1 + 3 + 5 + 7 + 9$$

## Codificatoare

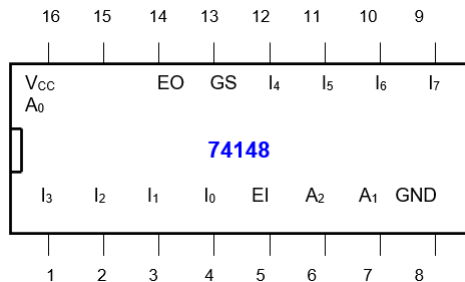
### Exemple de codificatoare integrate



**74147** este un codificator zecimal – BCD prioritar

**0 – 9** – intrări, active în 0

**A, B, C, D** – ieșiri, active în 0 (D este bitul cel mai semnificativ)



**74148** este un codificator prioritar de trei biți

**I<sub>0</sub> – I<sub>7</sub>** – intrări, active în 0

**A<sub>0</sub>, A<sub>1</sub>, A<sub>2</sub>** – ieșiri, active în 1

De regulă, intrările codificatoarelor sunt **active în 0**, deoarece prin activarea unei intrări (apăsarea tastei respective de la claviatura sistemului numeric) aceasta este pusă la masă, deci capătă valoarea 0 logic.

## Decodificatoare

Decodificatorul este un circuit logic combinațional care servește la identificarea unui cod de intrare prin activarea unei singure linii de ieșire, corespunzătoare aceluși cod.

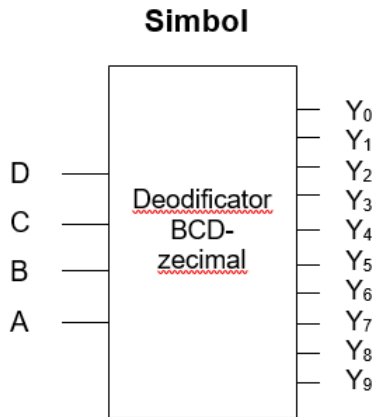
Decodificatoarele sunt necesare în numeroase aplicații cum sunt:

- adresarea memoriilor
- afișarea numerică
- multiplexarea datelor

Din această categorie de circuite sunt prezentate în cele ce urmează două exemple semnificative: decodificatorul BCD – zecimal și decodificatorul BCD – 7 segmente.

## Decodificatoare

### Decodificatorul BCD - zecimal



Decodificatorul BCD – zecimal are 4 intrări, pentru cei 4 biți ai codului BCD și 10 ieșiri pentru cifrele de la 0 la 9.

### Funcționare

La aplicarea unui cod binar pe intrări circuitul va activa o singură linie de ieșire (de exemplu la aplicarea pe intrări a codului 0011 va fi activă doar ieșirea Y<sub>3</sub>).

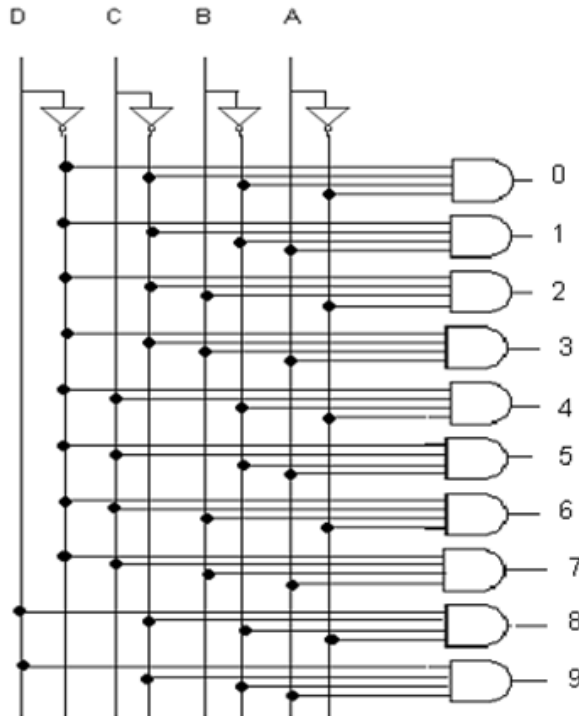
*Pentru cele patru variabile de intrare sunt posibile 16 combinații (corespunzătoare reprezentării în binar a numerelor 0 – 15). Doar 10 dintre ele sunt acceptate, și anume cele corespunzătoare cifrelor 0 – 9.*

*Celelalte combinații reprezintă **stări interzise**.*



## Decodificatoare

### Schema cu porți logice



Tabel adevăr

| D | C | B | A | Y <sub>0</sub> | Y <sub>1</sub> | Y <sub>2</sub> | Y <sub>3</sub> | Y <sub>4</sub> | Y <sub>5</sub> | Y <sub>6</sub> | Y <sub>7</sub> | Y <sub>8</sub> | Y <sub>9</sub> |
|---|---|---|---|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|----------------|
| 0 | 0 | 0 | 0 | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 0 | 0 | 0 | 1 | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 0 | 0 | 1 | 0 | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 0 | 0 | 1 | 1 | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              | 0              |
| 0 | 1 | 0 | 0 | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              | 0              |
| 0 | 1 | 0 | 1 | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              | 0              |
| 0 | 1 | 1 | 0 | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 0              | 0              |
| 0 | 1 | 1 | 1 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              | 0              |
| 1 | 0 | 0 | 0 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              | 0              |
| 1 | 0 | 0 | 1 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 1              |
| 1 | 0 | 1 | 0 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 1 | 0 | 1 | 1 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 1 | 1 | 0 | 0 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 1 | 1 | 0 | 1 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 1 | 1 | 1 | 0 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |
| 1 | 1 | 1 | 1 | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              | 0              |

Stări interzise

Din cele 10 porți ȘI de ieșire doar una va avea semnal și anume poarta corespunzătoare codului binar aplicat la intrare.

De exemplu, la aplicarea pe intrări a codului 0101, doar poarta 5 va avea semnal (1 logic).

Se consideră bitul **D** ca fiind bitul cel mai semnificativ (MSB) și intrările active în 1 logic (logica pozitivă).

## Circuitul de decodificare BCD - 7 segmente

Circuitul de decodificare BCD – 7 segmente este un circuit logic combinational care primește la intrare un semnal numeric, codificat BCD sau în cod binar natural și furnizează la ieșire semnalele necesare afișării cifrelor cu ajutorul a șapte segmente.

Dacă semnalele aplicate la intrarea circuitului sunt codificate în cod BCD, atunci acestea sunt în domeniul  $0000b = 0z$  până la  $1001b = 9z$ .

Pentru anumite circuite este posibil ca semnalul de intrare să fie în cod binar natural în gama  $0000b = 0z$  până la  $1111b = 15z$  și atunci circuitul furnizează la ieșire semnalele necesare afișării cifrelor hexazecimale.

Elementele de afișare cu șapte segmente pot fi cu anod comun și în acest caz ieșirea activă (cea care trebuie să activeze un segment) ia valoarea zero logic (tensiune scăzută în logica pozitivă) sau cu catod comun, caz în care ieșirea activă trebuie să ia valoarea logică unu (tensiune ridicată).

### Reprezentarea numerelor zecimale



0 - a,b,c,d,e,f

1 - b,c

2 - a,b,d,e,g

3 - a,b,c,d,g

4 - b,c,f,g

5 - a,c,d,f,g

6 - a,c,d,e,f,g

7 - a,b,c

8 - a,b,c,d,e,f,g

9 - a,b,c,d,f,g



### Reprezentarea numerelor hexazecimale



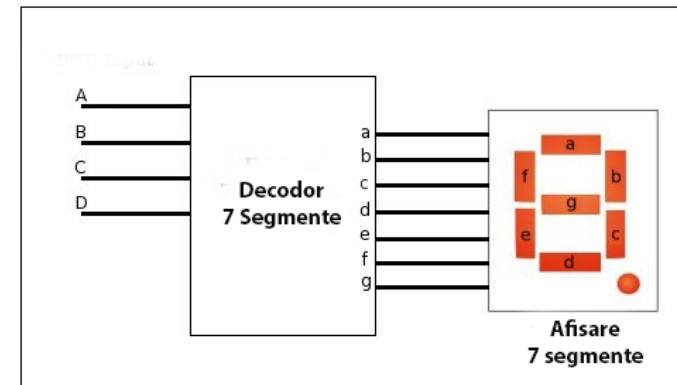
10z=Ah 11z=Bh 12z=Ch 13z=Dh 14z=Eh 15z=Fh

# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Construcție decodor 7 segmente

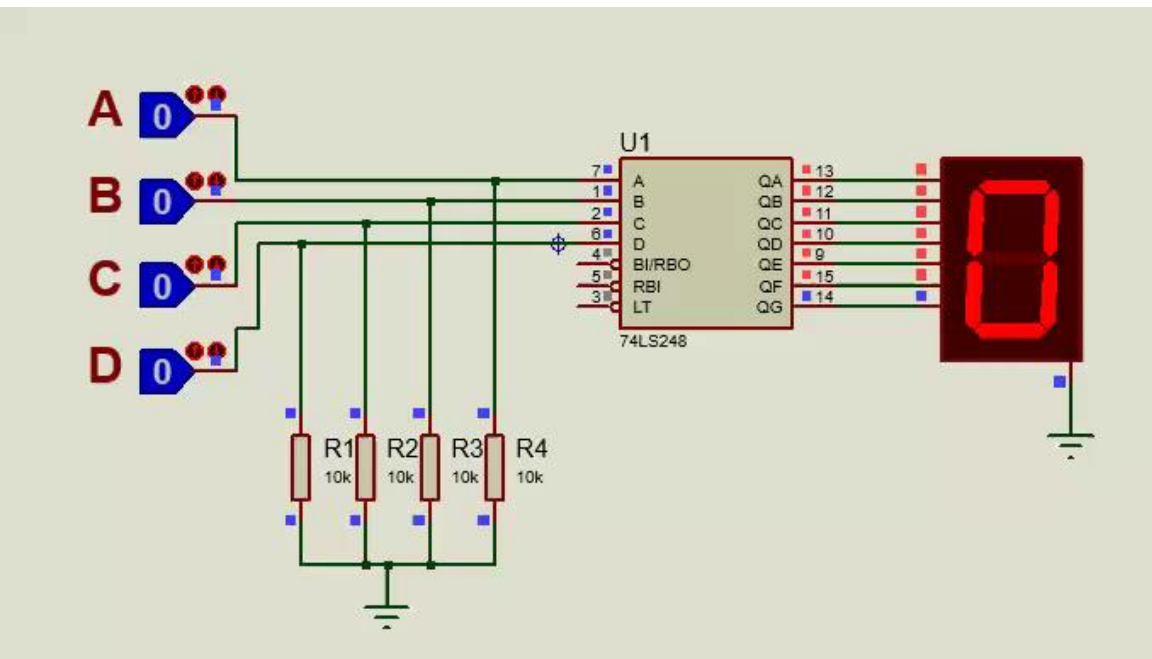
Tabel de adevar pentru un circuit de decodificare binar – 7 segmente  
destinat elementelor de afisare cu catod comun

| intrari BCD |   |   |   | iesiri segment |   |   |   |   |   |   | afisare |
|-------------|---|---|---|----------------|---|---|---|---|---|---|---------|
| D           | C | B | A | a              | b | c | d | e | f | g |         |
| 0           | 0 | 0 | 0 | 1              | 1 | 1 | 1 | 1 | 1 | 0 | 0       |
| 0           | 0 | 0 | 1 | 0              | 1 | 1 | 0 | 0 | 0 | 0 | 1       |
| 0           | 0 | 1 | 0 | 1              | 1 | 0 | 1 | 1 | 0 | 1 | 2       |
| 0           | 0 | 1 | 1 | 1              | 1 | 1 | 1 | 0 | 0 | 1 | 3       |
| 0           | 1 | 0 | 0 | 0              | 1 | 1 | 0 | 0 | 1 | 1 | 4       |
| 0           | 1 | 0 | 1 | 1              | 0 | 1 | 1 | 0 | 1 | 1 | 5       |
| 0           | 1 | 1 | 0 | 0              | 0 | 1 | 1 | 1 | 1 | 1 | 6       |
| 0           | 1 | 1 | 1 | 1              | 1 | 1 | 0 | 0 | 0 | 0 | 7       |
| 1           | 0 | 0 | 0 | 1              | 1 | 1 | 1 | 1 | 1 | 1 | 8       |
| 1           | 0 | 0 | 1 | 1              | 1 | 1 | 0 | 0 | 1 | 1 | 9       |



## Construcție decodor 7 segmente

Film simulare



Tabel adevăr

| intrari BCD |   |   |   | iesiri segment |   |   |   |   |   |   | afisare |
|-------------|---|---|---|----------------|---|---|---|---|---|---|---------|
| D           | C | B | A | a              | b | c | d | e | f | g |         |
| 0           | 0 | 0 | 0 | 1              | 1 | 1 | 1 | 1 | 1 | 0 | 0       |
| 0           | 0 | 0 | 1 | 0              | 1 | 1 | 0 | 0 | 0 | 0 | 1       |
| 0           | 0 | 1 | 0 | 1              | 1 | 0 | 1 | 1 | 0 | 1 | 2       |
| 0           | 0 | 1 | 1 | 1              | 1 | 1 | 1 | 0 | 0 | 1 | 3       |
| 0           | 1 | 0 | 0 | 0              | 1 | 1 | 0 | 0 | 1 | 1 | 4       |
| 0           | 1 | 0 | 1 | 1              | 0 | 1 | 1 | 0 | 1 | 1 | 5       |
| 0           | 1 | 1 | 0 | 0              | 0 | 1 | 1 | 1 | 1 | 1 | 6       |
| 0           | 1 | 1 | 1 | 1              | 1 | 1 | 0 | 0 | 0 | 0 | 7       |
| 1           | 0 | 0 | 0 | 1              | 1 | 1 | 1 | 1 | 1 | 1 | 8       |
| 1           | 0 | 0 | 1 | 1              | 1 | 1 | 0 | 0 | 1 | 1 | 9       |

## Coduri utilizate in tehnica comenzii numerice

O formulare de tipul:

“X-45679”

**este** înțeleasa de catre operatorul uman ca o pozitie la o valoare in baza 10, dar **NU este** înțeleasa de echipamentul numeric.

Pentru CNC trebuiesc folosite coduri, pentru care reprezentarea numerelor se face tot prin serii de puteri,

$$N_b = \sum_{i=0}^{r-1} a_i \cdot b^i$$

## Coduri utilizate in tehnica comenzii numerice

$$N_b = \sum_{i=0}^{r-1} a_i \cdot b^i$$

în care:

- $N_b$  - reprezinta numarul în codul “b”;
- $a_i$  - valoarea rangului (ordinului) în pozitia “i”;
- $b$  - baza de numerație sau codul;
- $i$  - ponderea pozitiei (rangului), crescator de la dreapta la stânga.

Exemplu cu , **b = 10**

**“X 45679”**

## Coduri utilizate in tehnica comenzii numerice

$$N_b = \sum_{i=0}^{r-1} a_i \cdot b^i$$

Exemplu cu , **b = 10**

**“X 45679”**

## Coduri utilizate in tehnica comenzii numerice

$$N_b = \sum_{i=0}^{r-1} a_i \cdot b^i$$

Exemplu cu , **b = 10**

**“X 45679”**

Exemplu cu , **b = 2**

**“X 1011001001101111”**



## Coduri utilizate in tehnica comenzii numerice

Reprezentari pentru 1, 2, si 3 biti.

$n$  = nr. de stari (mesaje) posibile,  $m = 2^n$   $m$  = nr. de variabile (biti)

$n=2, m=1$

|    |   |
|----|---|
| s1 | 0 |
| s2 | 1 |

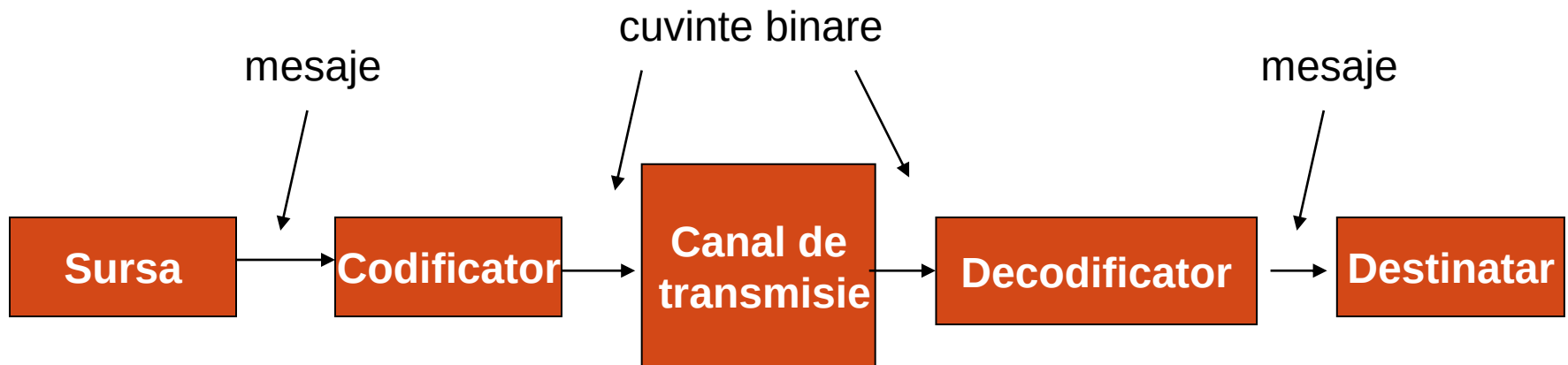
$n=4, m=2$

|    |    |
|----|----|
| s1 | 00 |
| s2 | 01 |
| s3 | 10 |
| s4 | 11 |

$n=8, m=3$

|    |     |
|----|-----|
| s1 | 000 |
| s2 | 001 |
| s3 | 010 |
| s4 | 011 |
| s5 | 100 |
| s6 | 101 |
| s7 | 110 |
| s8 | 111 |

## Codificarea și decodificarea mesajelor în sistemele de transmisie a informației



## Coduri utilizate in tehnica comenzii numerice

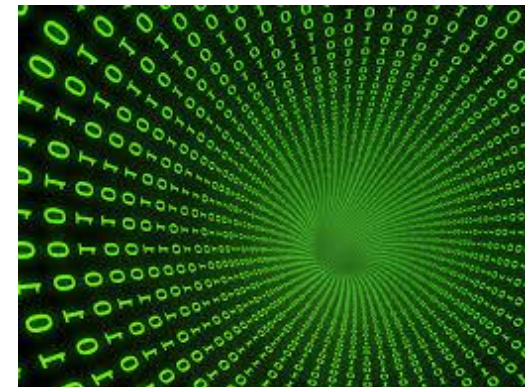
*Codurile pot fi clasificate :*

- Coduri Numerice
- Coduri Alfanumerice

| Decimal | BCD     | Gray    |
|---------|---------|---------|
| 0       | 0 0 0 0 | 0 0 0 0 |
| 1       | 0 0 0 1 | 0 0 0 1 |
| 2       | 0 0 1 0 | 0 0 1 1 |
| 3       | 0 0 1 1 | 0 0 1 0 |
| 4       | 0 1 0 0 | 0 1 1 0 |
| 5       | 0 1 0 1 | 0 1 1 1 |
| 6       | 0 1 1 0 | 0 1 0 1 |
| 7       | 0 1 1 1 | 0 1 0 0 |
| 8       | 1 0 0 0 | 1 1 0 0 |
| 9       | 1 0 0 1 | 1 1 0 1 |

### Alphanumeric Character codes

| Character | 6-bit internal code | ASCII code | 8-bit EBCDIC |
|-----------|---------------------|------------|--------------|
| A         | 010 001             | 100 0001   | 1100 0001    |
| B         | 010 010             | 100 0010   | 1100 0010    |
| C         | 010 011             | 100 0011   | 1100 0011    |
| D         | 010 100             | 100 0100   | 1100 0100    |
| E         | 010 101             | 100 0101   | 1100 0101    |
| F         | 010 110             | 100 0110   | 1100 0110    |
| G         | 010 111             | 100 0111   | 1100 0111    |
| H         | 011 000             | 100 1000   | 1100 1000    |
| I         | 011 001             | 100 1001   | 1100 1001    |
| J         | 100 001             | 100 1010   | 1101 0001    |
| .....     |                     |            |              |



## Coduri utilizate in tehnica comenzii numerice

### *Coduri numerice in CNC*

Pot reprezenta cifrele (0,1,2,3..9) prin cuvinte binare 4-poziționale.

| cifra | Denumirea codului |      |       |         |
|-------|-------------------|------|-------|---------|
|       | Direct            | Gray | Aiken | Exces 3 |
| 0     | 0000              | 0000 | 0000  | 0011    |
| 1     | 0001              | 0001 | 0001  | 0100    |
| 2     | 0010              | 0011 | 0010  | 0101    |
| 3     | 0011              | 0010 | 0011  | 0110    |
| 4     | 0100              | 0110 | 0100  | 0111    |
| 5     | 0101              | 0111 | 1011  | 1000    |
| 6     | 0110              | 0101 | 1100  | 1001    |
| 7     | 0111              | 0100 | 1101  | 1010    |
| 8     | 1000              | 1100 | 1110  | 1011    |
| 9     | 1001              | 1101 | 1111  | 1100    |

## Coduri utilizate in tehnica comenzii numerice

### *Coduri alfanumerice*

Codurile alfanumerice reprezintă prin cuvinte binare:

- cifre;
- literele mari;
- litere mici
- semne de punctuație si grafice.

**Codul ASCII (American Code for Information Interchange) – standard sunt  $2^7=128$  simboluri**

# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Coduri utilizate in tehnica comenzii numerice

### Codul ASCII

|                                                          |                                                               |                                                          |                                                                |                                                          |                                                                  |                                                                         |                                                            |
|----------------------------------------------------------|---------------------------------------------------------------|----------------------------------------------------------|----------------------------------------------------------------|----------------------------------------------------------|------------------------------------------------------------------|-------------------------------------------------------------------------|------------------------------------------------------------|
| <b>NUL</b><br>0 0 0<br>N 0 1000<br>Null                  | <b>SOH</b><br>1 0 1<br>N 1 1001<br>Start of Header            | <b>STX</b><br>2 0 2<br>N 2 1010<br>Start of Text         | <b>ETX</b><br>3 0 3<br>N 3 1011<br>End of Text                 | <b>EOT</b><br>4 0 4<br>N 4 1100<br>End of Transmission   | <b>ENQ</b><br>5 0 5<br>N 5 1101<br>Enquiry                       | <b>ACK</b><br>6 0 6<br>N 6 1110<br>Acknowledgment                       | <b>BEL</b><br>7 0 7<br>N 7 1111<br>Bell                    |
| <b>BS</b><br>8 0 10<br>N 8 10000<br>Backspace            | <b>HT</b><br>9 0 11<br>N 9 10001<br>Horizontal Tab            | <b>LF</b><br>10 0 12<br>N 10 1010<br>Line Feed           | <b>VT</b><br>11 0 13<br>N 11 1011<br>Vertical Tab              | <b>FF</b><br>12 0 14<br>N 12 1100<br>Form Feed           | <b>CR</b><br>13 0 15<br>N 13 1101<br>Carriage Return             | <b>SO</b><br>14 0 16<br>N 14 1110<br>Shift Out                          | <b>SI</b><br>15 0 17<br>N 15 1111<br>Shift In              |
| <b>DLE</b><br>16 0 20<br>N 16 100000<br>Data Link Escape | <b>DC1</b><br>17 0 21<br>N 17 100001<br>XON, Device Control 1 | <b>DC2</b><br>18 0 22<br>N 18 100010<br>Device Control 2 | <b>DC3</b><br>19 0 23<br>N 19 100011<br>XOFF, Device Control 3 | <b>DC4</b><br>20 0 24<br>N 20 101000<br>Device Control 4 | <b>NAK</b><br>21 0 25<br>N 21 101001<br>Negative Acknowledgement | <b>SYN</b><br>22 0 26<br>N 22 10110<br>Synchronous Idle                 | <b>ETB</b><br>23 0 27<br>N 23 10111<br>End of Trans. Block |
| <b>CAN</b><br>24 0 30<br>N 24 110000<br>Cancel           | <b>EM</b><br>25 0 31<br>N 25 110001<br>End of Medium          | <b>SUB</b><br>26 0 32<br>N 26 110010<br>Substitute       | <b>ESC</b><br>27 0 33<br>N 27 110011<br>Escape                 | <b>FS</b><br>28 0 34<br>N 28 111000<br>File Separator    | <b>GS</b><br>29 0 35<br>N 29 111001<br>Group Separator           | <b>RS</b><br>30 0 36<br>N 30 11110<br>Request to Send, Record Separator | <b>US</b><br>31 0 37<br>N 31 11111<br>Unit Separator       |
| <b>SP</b><br>32 0 40<br>N 32 1000000<br>Space            | <b>!</b><br>33 0 41<br>N 33 1000001<br>N 21 1000001           | <b>"</b><br>34 0 42<br>N 34 1000010<br>N 22 1000010      | <b>#</b><br>35 0 43<br>N 35 1000011<br>N 23 1000011            | <b>\$</b><br>36 0 44<br>N 36 1001000<br>N 24 1001000     | <b>%</b><br>37 0 45<br>N 37 1001001<br>N 25 1001001              | <b>&amp;</b><br>38 0 46<br>N 38 1001010<br>N 26 1001010                 | <b>'</b><br>39 0 47<br>N 39 1001011<br>N 27 1001011        |
| <b>(</b><br>40 0 50<br>N 40 1001000<br>N 28 1001000      | <b>)</b><br>41 0 51<br>N 41 1001001<br>N 29 1001001           | <b>*</b><br>42 0 52<br>N 42 1001010<br>N 30 1001010      | <b>+</b><br>43 0 53<br>N 43 1001011<br>N 31 1001011            | <b>,</b><br>44 0 54<br>N 44 1010000<br>N 32 1010000      | <b>-</b><br>45 0 55<br>N 45 1010001<br>N 33 1010001              | <b>.</b><br>46 0 56<br>N 46 1010010<br>N 34 1010010                     | <b>/</b><br>47 0 57<br>N 47 1010011<br>N 35 1010011        |
| <b>0</b><br>48 0 60<br>N 48 1100000<br>N 36 1100000      | <b>1</b><br>49 0 61<br>N 49 1100001<br>N 37 1100001           | <b>2</b><br>50 0 62<br>N 50 1100010<br>N 38 1100010      | <b>3</b><br>51 0 63<br>N 51 1100011<br>N 39 1100011            | <b>4</b><br>52 0 64<br>N 52 1101000<br>N 40 1101000      | <b>5</b><br>53 0 65<br>N 53 1101001<br>N 41 1101001              | <b>6</b><br>54 0 66<br>N 54 1101010<br>N 42 1101010                     | <b>7</b><br>55 0 67<br>N 55 1101011<br>N 43 1101011        |

|                                                           |                                                           |                                                           |                                                           |                                                           |                                                           |                                                           |                                                           |
|-----------------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------|-----------------------------------------------------------|
| <b>8</b><br>56 0 70<br>N 56 1110000<br>N 38 1110000       | <b>9</b><br>57 0 71<br>N 57 1110001<br>N 39 1110001       | <b>:</b><br>58 0 72<br>N 58 1110010<br>N 40 1110010       | <b>;</b><br>59 0 73<br>N 59 1110011<br>N 41 1110011       | <b>&lt;</b><br>60 0 74<br>N 60 1111000<br>N 42 1111000    | <b>=</b><br>61 0 75<br>N 61 1111001<br>N 43 1111001       | <b>&gt;</b><br>62 0 76<br>N 62 1111010<br>N 44 1111010    | <b>?</b><br>63 0 77<br>N 63 1111011<br>N 45 1111011       |
| <b>@</b><br>64 0 100<br>N 64 10000000<br>N 46 10000000    | <b>A</b><br>65 0 101<br>N 65 10000001<br>N 47 10000001    | <b>B</b><br>66 0 102<br>N 66 10000010<br>N 48 10000010    | <b>C</b><br>67 0 103<br>N 67 10000011<br>N 49 10000011    | <b>D</b><br>68 0 104<br>N 68 10000100<br>N 50 10000100    | <b>E</b><br>69 0 105<br>N 69 10000101<br>N 51 10000101    | <b>F</b><br>70 0 106<br>N 70 10000110<br>N 52 10000110    | <b>G</b><br>71 0 107<br>N 71 10000111<br>N 53 10000111    |
| <b>H</b><br>72 0 110<br>N 72 10001000<br>N 54 10001000    | <b>I</b><br>73 0 111<br>N 73 10001001<br>N 55 10001001    | <b>J</b><br>74 0 112<br>N 74 10001010<br>N 56 10001010    | <b>K</b><br>75 0 113<br>N 75 10001011<br>N 57 10001011    | <b>L</b><br>76 0 114<br>N 76 10001100<br>N 58 10001100    | <b>M</b><br>77 0 115<br>N 77 10001101<br>N 59 10001101    | <b>N</b><br>78 0 116<br>N 78 10001110<br>N 60 10001110    | <b>O</b><br>79 0 117<br>N 79 10001111<br>N 61 10001111    |
| <b>P</b><br>80 0 120<br>N 80 10010000<br>N 62 10010000    | <b>Q</b><br>81 0 121<br>N 81 10010001<br>N 63 10010001    | <b>R</b><br>82 0 122<br>N 82 10010010<br>N 64 10010010    | <b>S</b><br>83 0 123<br>N 83 10010011<br>N 65 10010011    | <b>T</b><br>84 0 124<br>N 84 10010100<br>N 66 10010100    | <b>U</b><br>85 0 125<br>N 85 10010101<br>N 67 10010101    | <b>V</b><br>86 0 126<br>N 86 10010110<br>N 68 10010110    | <b>W</b><br>87 0 127<br>N 87 10010111<br>N 69 10010111    |
| <b>X</b><br>88 0 130<br>N 88 10011000<br>N 70 10011000    | <b>Y</b><br>89 0 131<br>N 89 10011001<br>N 71 10011001    | <b>Z</b><br>90 0 132<br>N 90 10011010<br>N 72 10011010    | <b>[</b><br>91 0 133<br>N 91 10011011<br>N 73 10011011    | <b>\</b><br>92 0 134<br>N 92 10011100<br>N 74 10011100    | <b>]</b><br>93 0 135<br>N 93 10011101<br>N 75 10011101    | <b>^</b><br>94 0 136<br>N 94 10011110<br>N 76 10011110    | <b>_</b><br>95 0 137<br>N 95 10011111<br>N 77 10011111    |
| <b>`</b><br>96 0 140<br>N 96 11000000<br>N 80 11000000    | <b>a</b><br>97 0 141<br>N 97 11000001<br>N 81 11000001    | <b>b</b><br>98 0 142<br>N 98 11000010<br>N 82 11000010    | <b>c</b><br>99 0 143<br>N 99 11000011<br>N 83 11000011    | <b>d</b><br>100 0 144<br>N 100 11000100<br>N 84 11000100  | <b>e</b><br>101 0 145<br>N 101 11000101<br>N 85 11000101  | <b>f</b><br>102 0 146<br>N 102 11000110<br>N 86 11000110  | <b>g</b><br>103 0 147<br>N 103 11000111<br>N 87 11000111  |
| <b>h</b><br>104 0 150<br>N 104 11001000<br>N 88 11001000  | <b>i</b><br>105 0 151<br>N 105 11001001<br>N 89 11001001  | <b>j</b><br>106 0 152<br>N 106 11001010<br>N 90 11001010  | <b>k</b><br>107 0 153<br>N 107 11001011<br>N 91 11001011  | <b>l</b><br>108 0 154<br>N 108 11001100<br>N 92 11001100  | <b>m</b><br>109 0 155<br>N 109 11001101<br>N 93 11001101  | <b>n</b><br>110 0 156<br>N 110 11001110<br>N 94 11001110  | <b>o</b><br>111 0 157<br>N 111 11001111<br>N 95 11001111  |
| <b>p</b><br>112 0 160<br>N 112 11001000<br>N 96 11001000  | <b>q</b><br>113 0 161<br>N 113 11001001<br>N 97 11001001  | <b>r</b><br>114 0 162<br>N 114 11001010<br>N 98 11001010  | <b>s</b><br>115 0 163<br>N 115 11001011<br>N 99 11001011  | <b>t</b><br>116 0 164<br>N 116 11001100<br>N 100 11001100 | <b>u</b><br>117 0 165<br>N 117 11001101<br>N 101 11001101 | <b>v</b><br>118 0 166<br>N 118 11001110<br>N 102 11001110 | <b>w</b><br>119 0 167<br>N 119 11001111<br>N 103 11001111 |
| <b>x</b><br>120 0 170<br>N 120 11001000<br>N 104 11001000 | <b>y</b><br>121 0 171<br>N 121 11001001<br>N 105 11001001 | <b>z</b><br>122 0 172<br>N 122 11001010<br>N 106 11001010 | <b>{</b><br>123 0 173<br>N 123 11001011<br>N 107 11001011 | <b> </b><br>124 0 174<br>N 124 11001100<br>N 108 11001100 | <b>}</b><br>125 0 175<br>N 125 11001101<br>N 109 11001101 | <b>~</b><br>126 0 176<br>N 126 11001110<br>N 110 11001110 | <b>DEL</b><br>127 0 177<br>N 127 11111111<br>Delete       |

## Coduri utilizate in tehnica comenzii numerice

### *Codul ASCII extins*

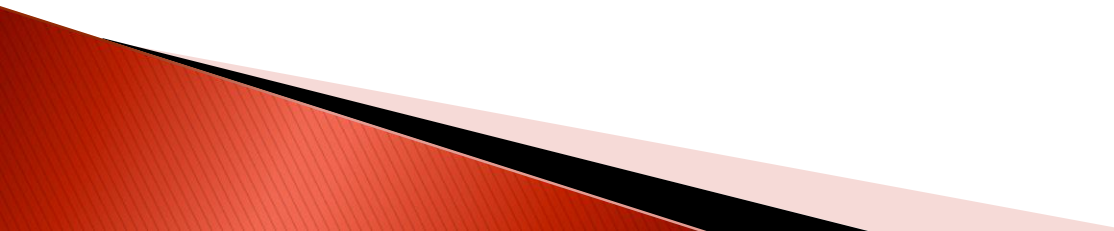
Pentru a reprezenta caracterele deacritice, literele chirilice se folosește codul ASCII extins

Este reprezentat pe 8 biți

$2^8 = 256$  simboluri

Extinderea codurilor alfanumerice a condus la folosirea octetului pentru măsurarea cantității de informație

1 octet =  $2^3 = 8$  biți = 1 byte



## Coduri utilizate in tehnica comenzii numerice

### *Codul Aiken*

Codul Aiken este similar codului natural BCD, dar “ponderea” sau “valoarea” este distribuita diferit. Ponderea naturala este (2 la puterea) 8-4-2-1, iar al codului **Aiken** distributia este: 2-4-2-1 (2 la puterea). Motivul pentru această codificare este de a realiza o simetrie între anumite numere:

- 4 cu 5;
- 3 cu 6;
- 2 cu 7;
- 1 cu 8;
- 0 cu 9.

Exemplu:

3 (0011) cu 6 (1100) – in aceasta combinatie bitul “0” devine “1”.

| Decimal | Aiken | BCD  |
|---------|-------|------|
|         | 2421  | 8421 |
| 0       | 0000  | 0000 |
| 1       | 0001  | 0001 |
| 2       | 0010  | 0010 |
| 3       | 0011  | 0011 |
| 4       | 0100  | 0100 |
| 5       | 1011  | 0101 |
| 6       | 1100  | 0110 |
| 7       | 1101  | 0111 |
| 8       | 1110  | 1000 |
| 9       | 1111  | 1001 |



## Coduri utilizate in tehnica comenzii numerice

### ***Codul Excess 3***

Codul Excess 3 este obtinut prin adaugarea a cator 3 unitati pentru fiecare combinatie a codului natural BCD, fara modificarea valorii.

La fel ca si codul Aiken codul Excess 3 indeplineste aceeasi caracteristica de simetrie:

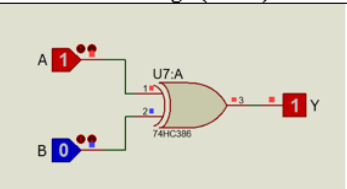
- 4 cu 5;
- 3 cu 6;
- 2 cu 7;
- 1 cu 8;
- 0 cu 9;

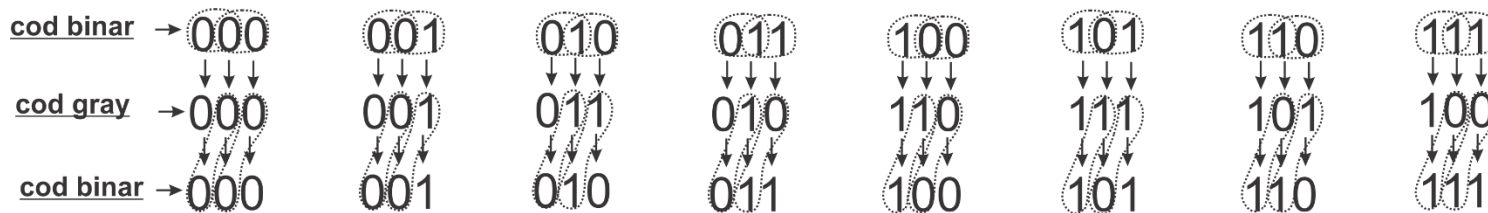
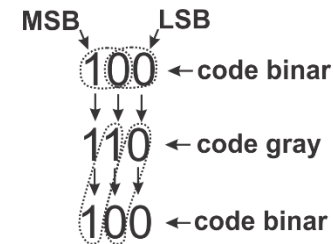
| Decimal | BCD  | Excess 3 |
|---------|------|----------|
|         | 8421 |          |
| 0       | 0000 | 0011     |
| 1       | 0001 | 0100     |
| 2       | 0010 | 0101     |
| 3       | 0011 | 0110     |
| 4       | 0100 | 0111     |
| 5       | 0101 | 1000     |
| 6       | 0110 | 1001     |
| 7       | 0111 | 1010     |
| 8       | 1000 | 1011     |
| 9       | 1001 | 1100     |

## Codul Gray

Codul Gray este un cod numeric reflectat, care are proprietatea că 2 numere adiacente diferă prin valoarea unui singur bit.

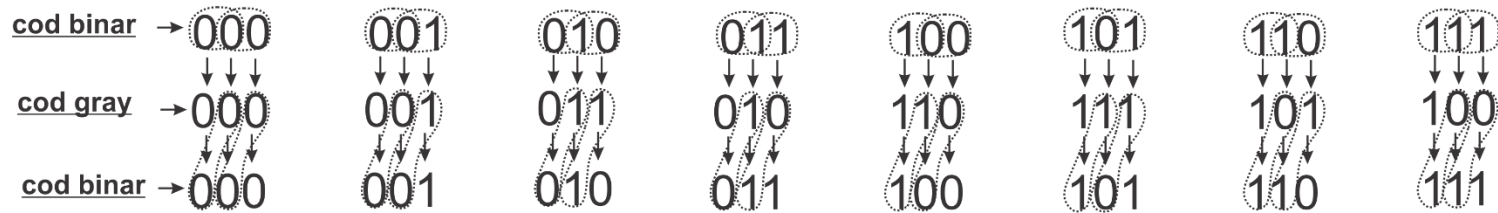
### XOR

| Simbol logic(ANSI)                                                                | Tabel de adevăr                                                                                                                                                                                                                                                        | Expresie booleană |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------|---|---|---|---|---|---|---|---|---|---|---|---|---|---|------------------|
|  | <table border="1"> <thead> <tr> <th>A</th><th>B</th><th>Y</th></tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>0</td></tr> </tbody> </table> | A                 | B | Y | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 0 | $Y = A \oplus B$ |
| A                                                                                 | B                                                                                                                                                                                                                                                                      | Y                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
| 0                                                                                 | 0                                                                                                                                                                                                                                                                      | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
| 0                                                                                 | 1                                                                                                                                                                                                                                                                      | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
| 1                                                                                 | 0                                                                                                                                                                                                                                                                      | 1                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |
| 1                                                                                 | 1                                                                                                                                                                                                                                                                      | 0                 |   |   |   |   |   |   |   |   |   |   |   |   |   |   |                  |

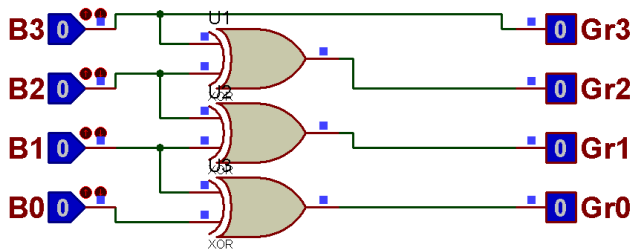


## Convertoare de cod

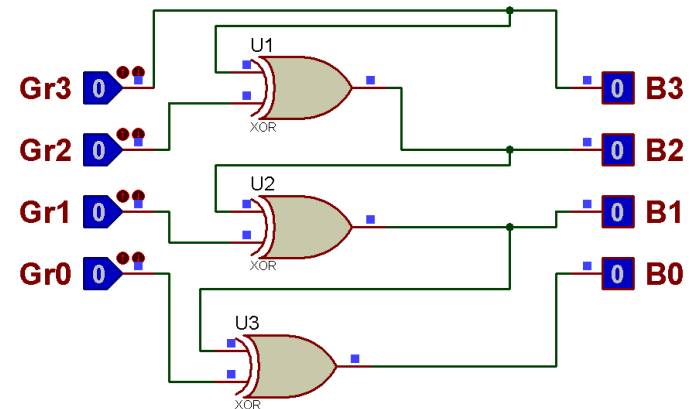
**Convertoarele de cod** sunt circuite logice combinaționale care realizează conversia numerelor binare dintr-un cod în alt cod.



### Conversie Binar - Gray

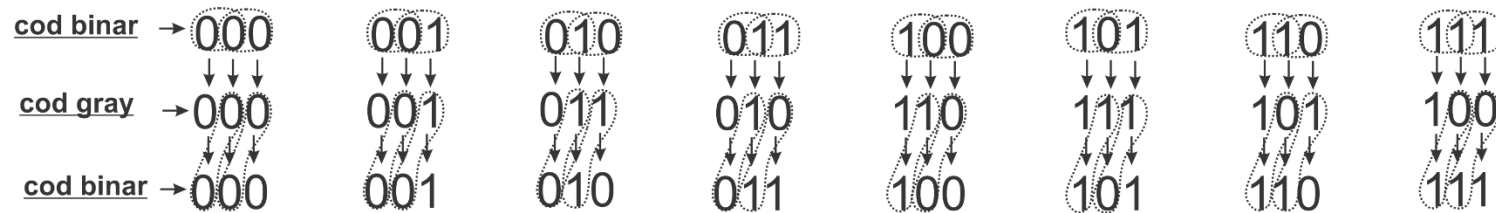


### Conversie Gray - Binar



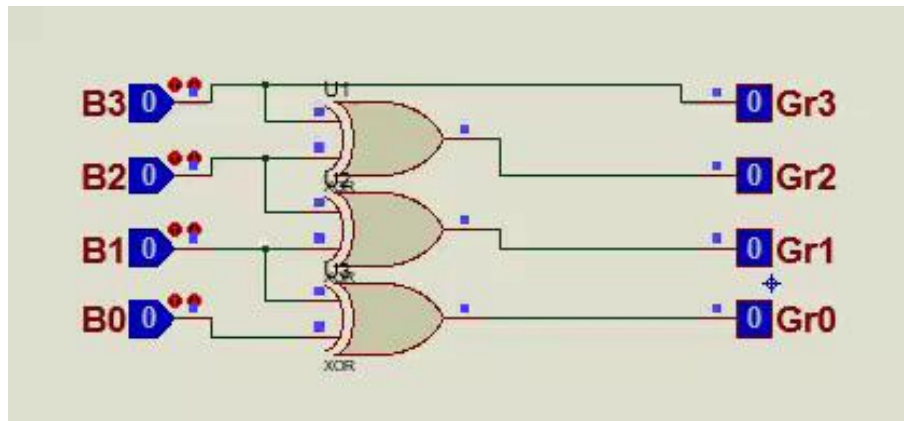
## Codul Gray

### Verificare



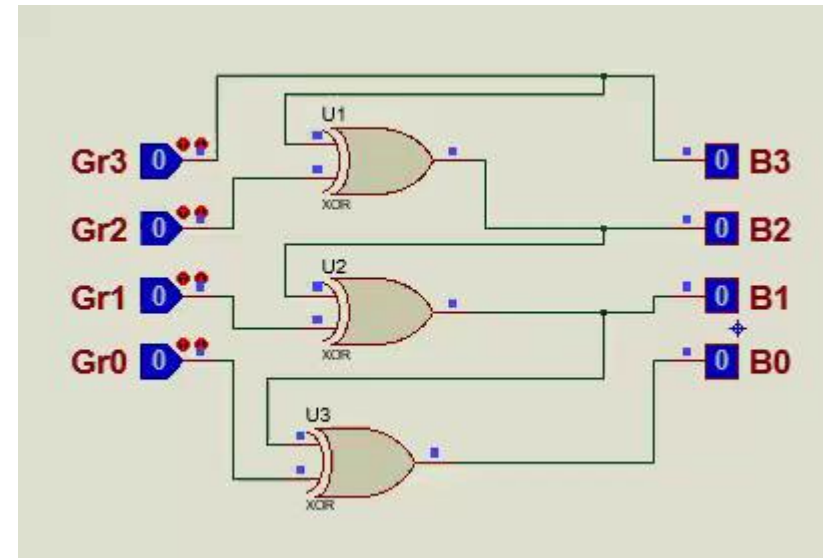
Film simulare

Conversie Binar - Gray



Film simulare

Conversie Gray - Binar



## Traductoare de pozitie

- Traductoare liniare de pozitie;



- Traductoare circulare de pozitie



# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Traductoare de pozitie – masurarea directa a pozitiei cu encoder liniar

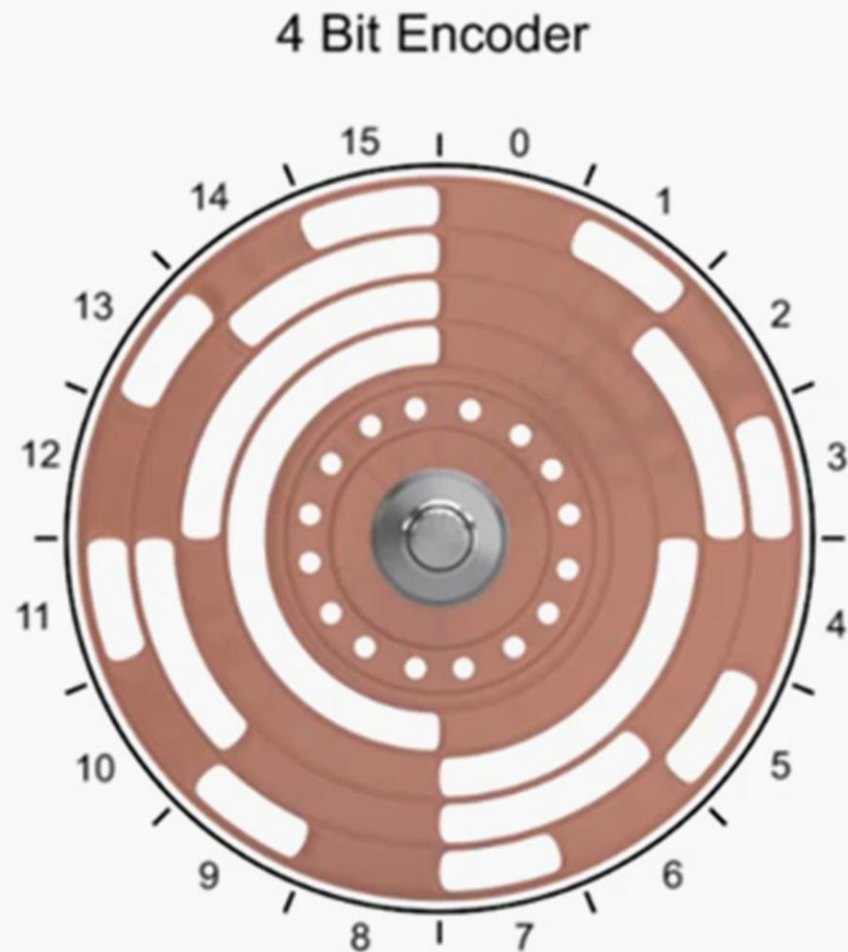
Film prezentare



## Traductoare de pozitie

- Exemplu: traductoare circulare de pozitie, masurarea directa a pozitiei (masurare in sistem absolut)

Film prezentare





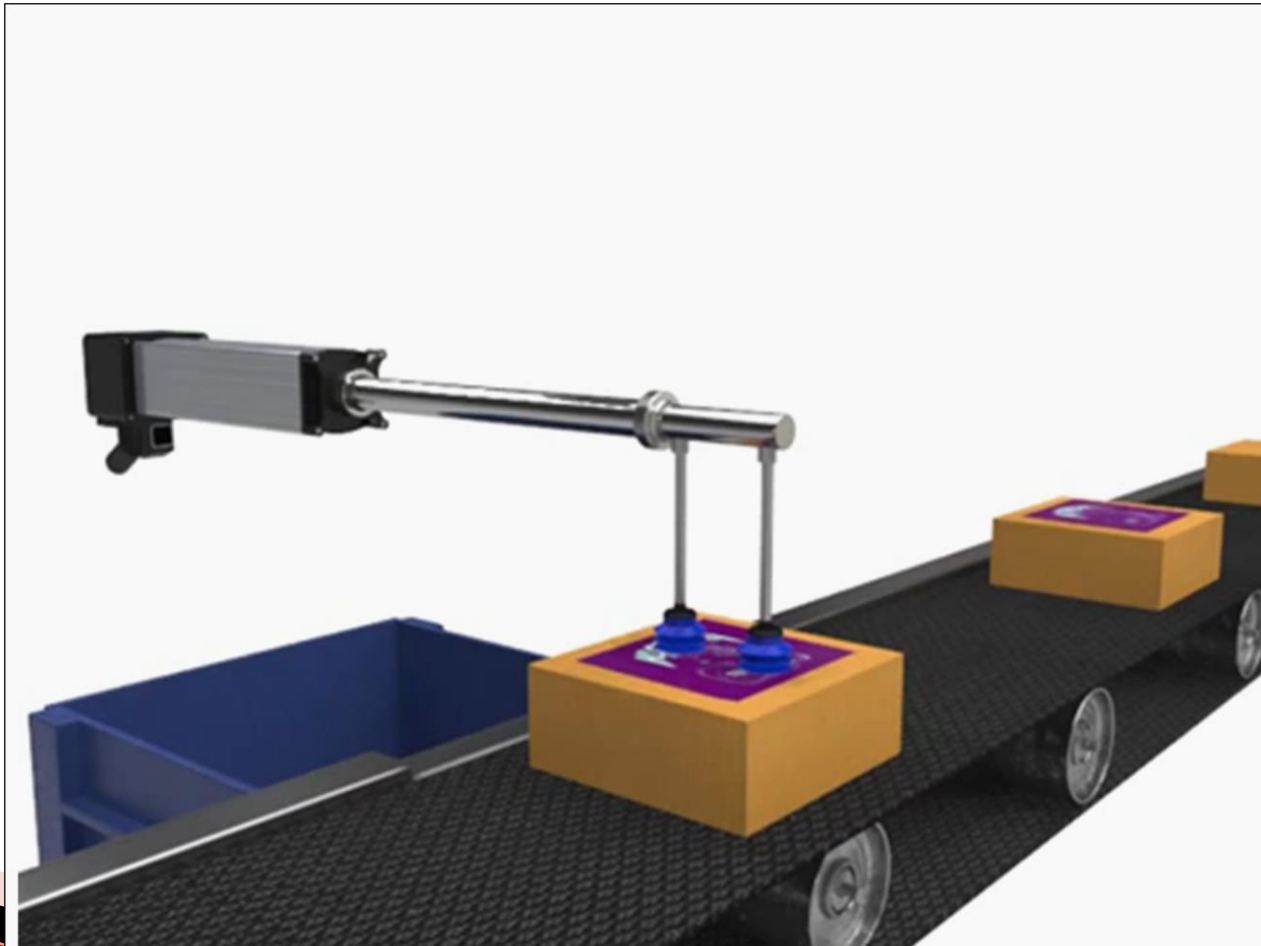
# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Aplicatie cod Gray

Identificarea unei pozitii pe baza semnalelor primite de la un encoder rotativ.

Daca este folosit BCD, vor exista unele erori la marginea regiunii, pot exista mai multe tranzitii in cod dintr-o data deoarece bitii nu pot fi cititi in acelasi timp si prin urmare citirea se va face incorect.

Film prezentare





## Circuite Logice Secventiale

Circuitele logice secventiale (CLS) sunt circuite logice la care starea iesirilor depinde de starile anterioare ale circuitului. Din acest motiv se spune ca circuitele logice secventiale sunt circuite cu memorie.

## Circuite basculante bistabile (CBB) sincrone

Circuite basculante bistabile sincrone au la intrare un semnalul de tact (ceas/clock).

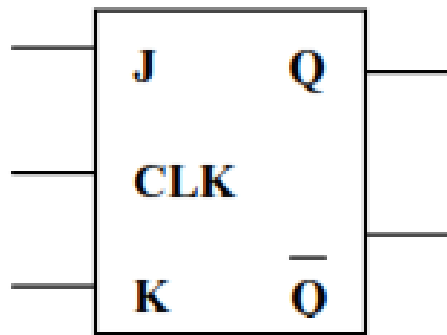
Schimbarea starii iesirii circuitului basculant bistabil se face în functie de semnalul de tact, care este semnalul de comanda.

Comanda se poate face pe palier sau pe front.

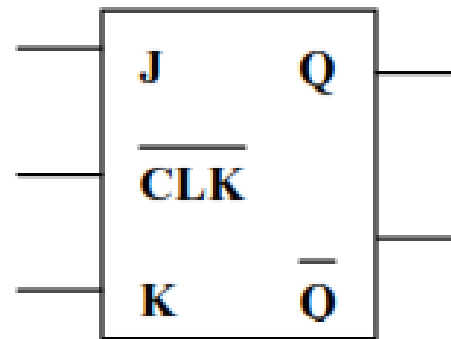
Principala diferenta între circuitele latch si circuitele flip-flop este metoda folosita pentru a determina modificari de stare la iesire :

## Circuite basculante bistabile (CBB) sincrone

În cazul comenzii pe palier este important ca semnalele aplicate pe intrările circuitului basculant bistabil să fie stabile în momentul începerii palierului. În caz contrar se pot produce mai multe tranzii pe un singur palier.



Activ pe palier HIGH



Activ pe palier LOW

## Circuite basculante bistabile (CBB) sincrone

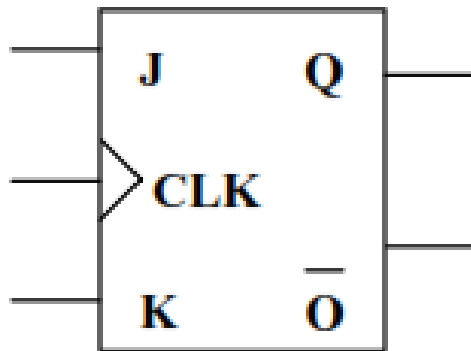
Principala diferență între circuitele latch și circuitele flip-flop este metoda folosită pentru a determina modificări de stare la ieșire :

- circuitele latch sunt active pe palier;
- ieșirile circuitului comută la nivel de tensiune (palier) și nu la o tranziție de semnal ;
- circuitele flip-flop sunt active pe front;
- ieșirile circuitului comută la tranziția semnalului, fie din 1 în 0 (front negativ) fie din 0 în 1 (front pozitiv).

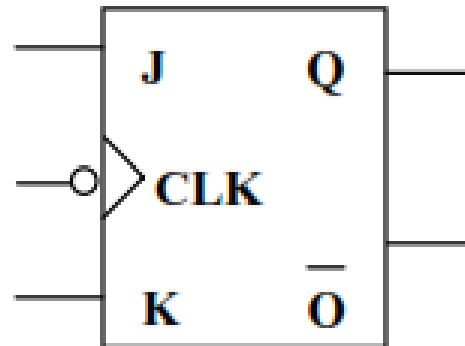
## Circuite basculante bistabile (CBB) sincrone

În simbolul logic, circuitele active pe front sunt reprezentate cu un triunghi mic, în interiorul blocului, la intrarea de clock.

Prezența sau absența unui cerculeț în afara blocului, la intrarea de clock, indică tipul frontului activ (front negativ, respectiv front pozitiv).



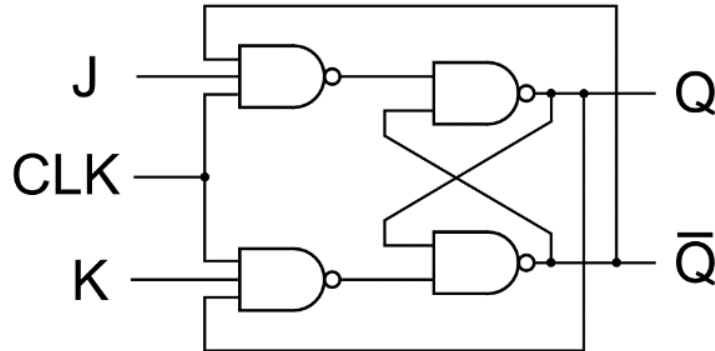
Activ pe front pozitiv  
(crescator)



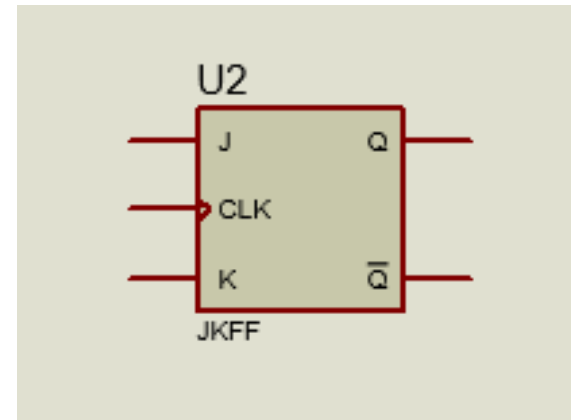
Activ pe front negativ  
(descrescator)

## JK Flip Flop (JKFF)

Construcție



Circuit

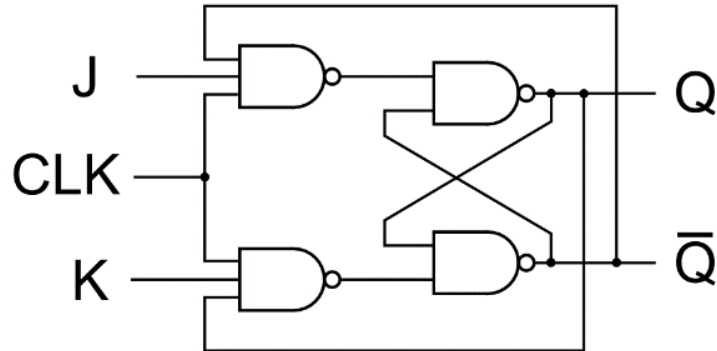


Tabel adevar

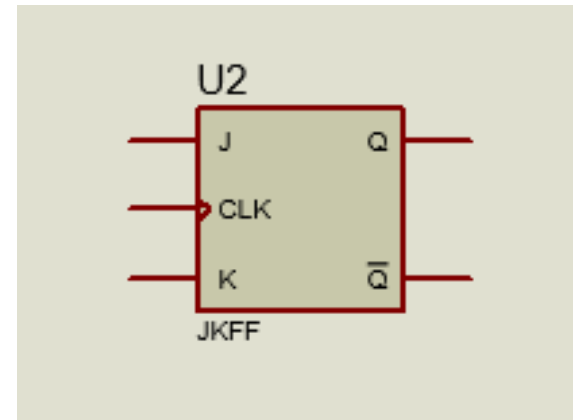
| J | K | CLK        | $Q$       | $\bar{Q}$ | Comenzi  |
|---|---|------------|-----------|-----------|----------|
| 0 | 0 | $\uparrow$ | $Q$       | $\bar{Q}$ | închis   |
| 1 | 0 | $\uparrow$ | 1         | 0         | SET      |
| 0 | 1 | $\uparrow$ | 0         | 1         | RESET    |
| 1 | 1 | $\uparrow$ | $\bar{Q}$ | $Q$       | Comutare |

## JK Flip Flop (JKFF)

Construcție



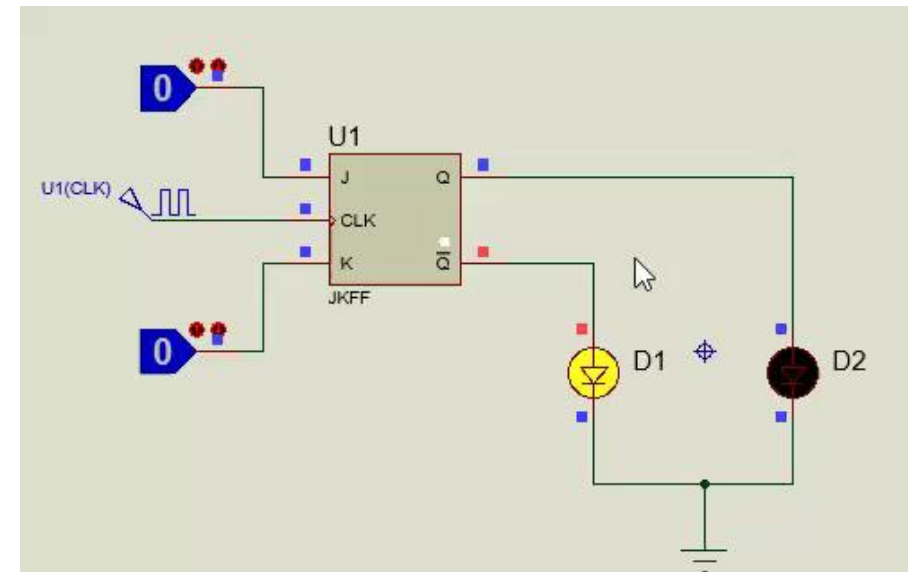
Circuit



Tabel adevăr

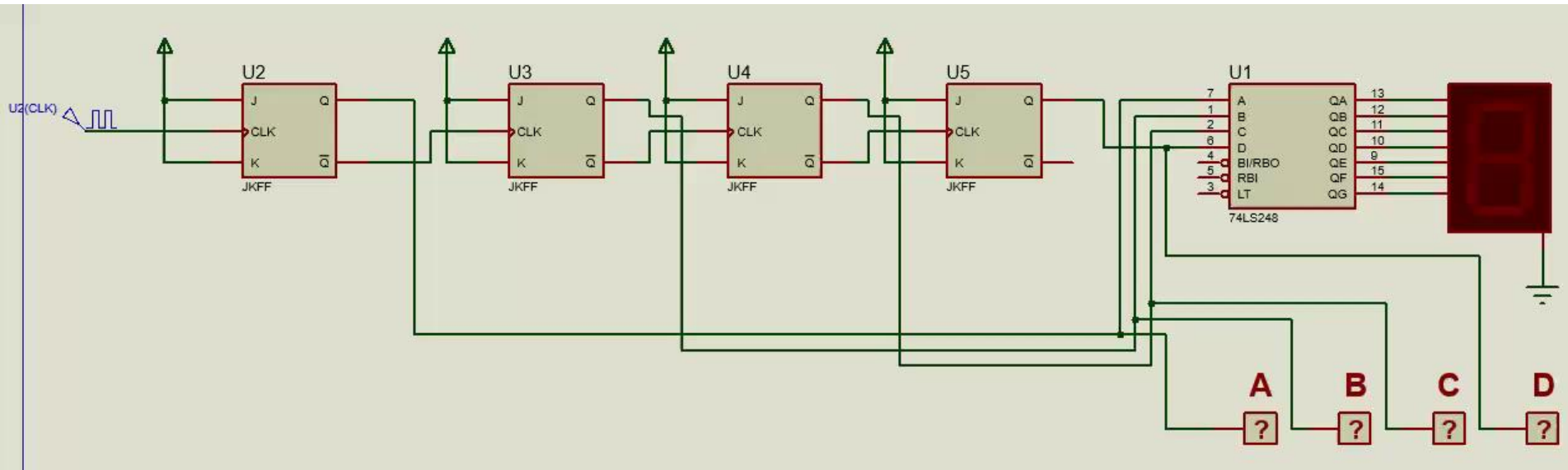
| J | K | CLK | Q         | $\bar{Q}$ | Comenzi  |
|---|---|-----|-----------|-----------|----------|
| 0 | 0 | ↑   | Q         | $\bar{Q}$ | închis   |
| 1 | 0 | ↑   | 1         | 0         | SET      |
| 0 | 1 | ↑   | 0         | 1         | RESET    |
| 1 | 1 | ↑   | $\bar{Q}$ | Q         | Comutare |

Film simulare



## Construcție counter pe 7 segmente (JKFF)

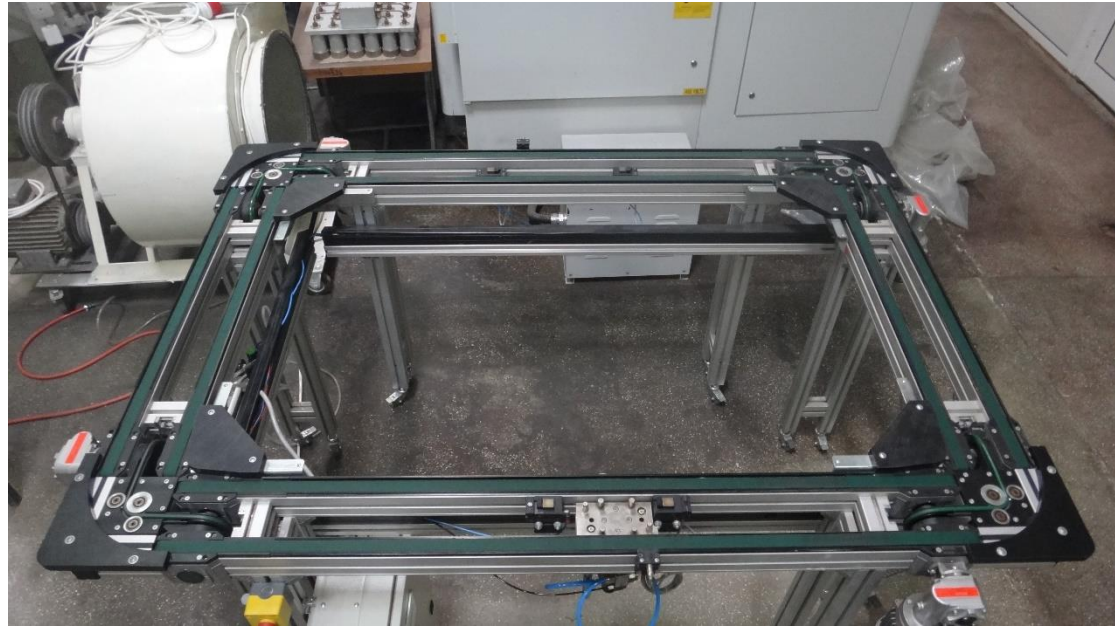
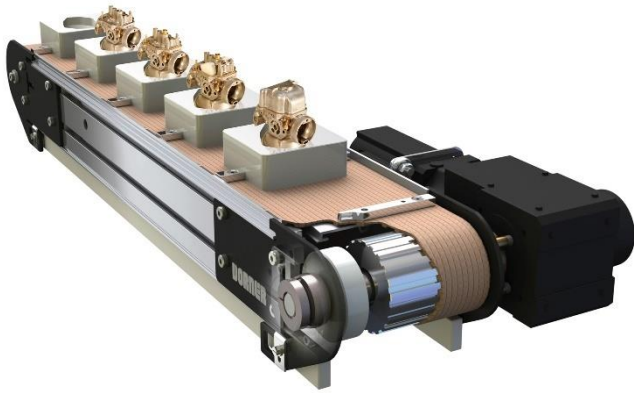
Film simulare





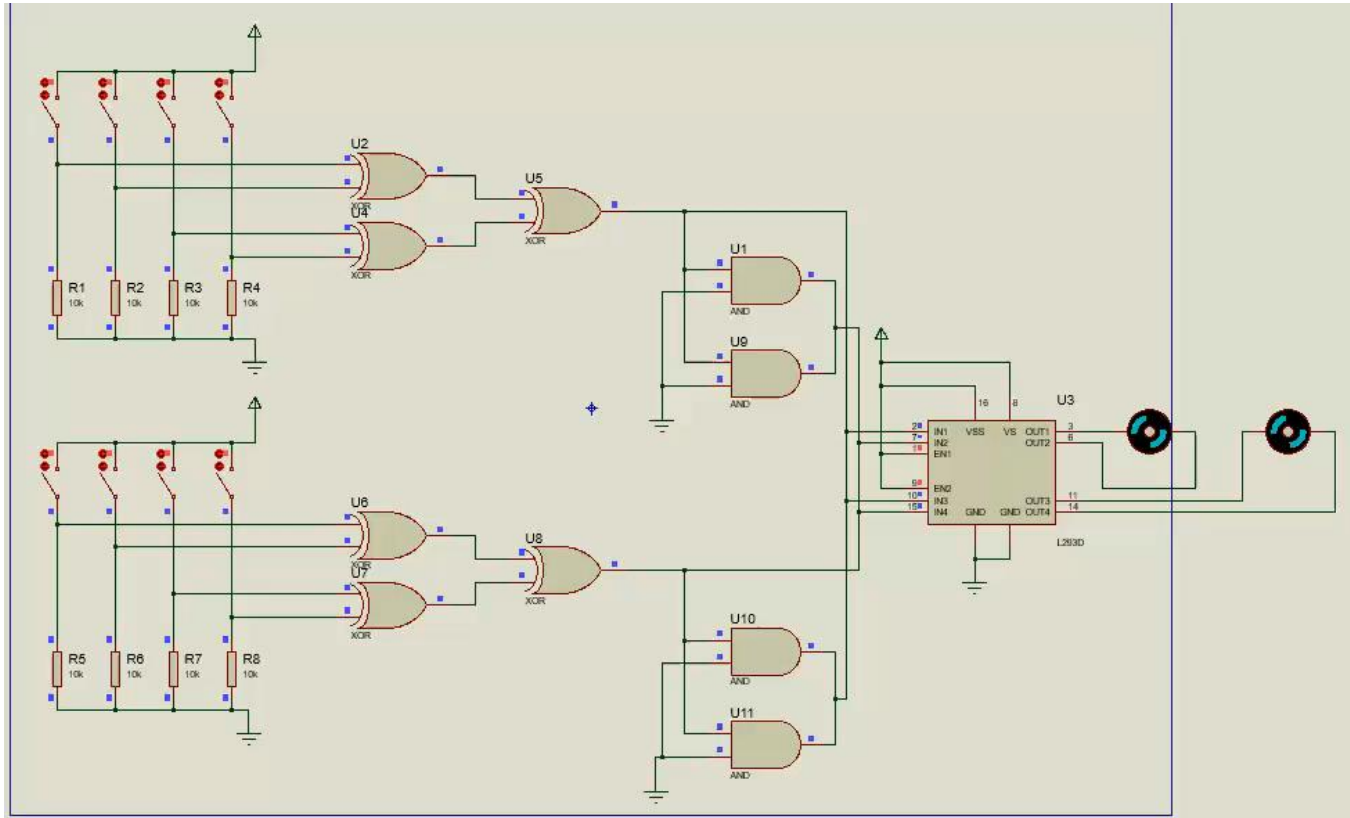
# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Comanda unui conveior cu ajutorul portilor logice



## Comanda unui conveior cu ajutorul portilor logice

Film simulare





**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

**Facultatea de Construcții de Mașini**  
**Departamentul Ingineria Proiectării și Robotică**

# **ROBOTII IN CADRUL SISTEMELOR FLEXIBILE DE FABRICATIE**



**Sl. Dr. Ing. Florin COVACIU**

**2020**

## Introducere în robotica

Cuvantul “**robot**” – de origine slava

- “rabota” – a muncii
- “robotnik” – muncitor sau sclav

Oficial, cuvantul robot este utilizat pentru prima data de catre scriitorul ceh Karel Capek – 1920

“Robotii universali ai lui Rossum”

drama stiintifico-fantastica

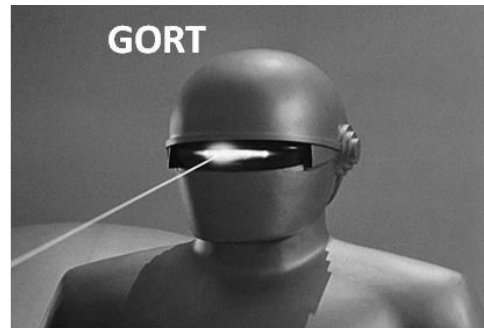
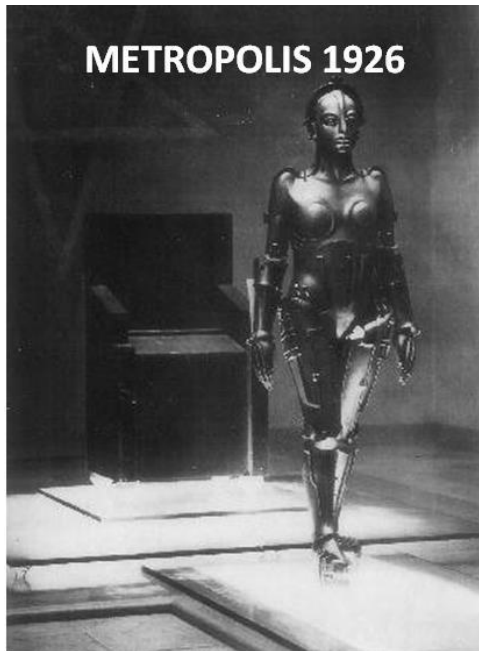
Ulterior, cuvântul ROBOT, a fost preluat si popularizat, în povestiri stiintifico-fantastice de catre scriitorul canadian de origine rusa Isaac ASIMOV



# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

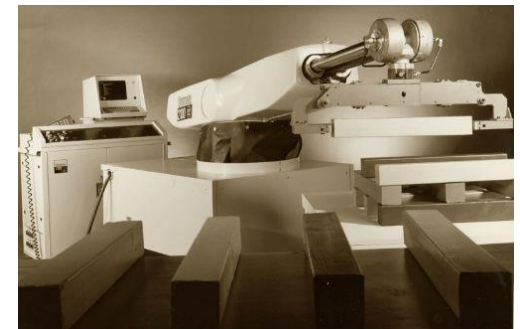
## Introducere in robotica

### Robotii din cinematografie



## Momente cheie in evolutia robotilor

- **1938** este proiectata prima masina programabila de vopsire prin pulverizare;
- **1951**, Raymond Goertz, Franta, proiecteaza primul manipulator articulat, denumit ***teleoperate***;
- **1953**, Seiko Corporation utilizeaza roboti în miniatura pentru asamblare de ceasuri;
- **1954**, George Devol inventeaza termenul *Universal Automation*, *dupa ce a proiectat primul robot programabil*;
- **1958**, Joseph F. Engelberger si George C. Devol fondeaza *Unimation*, *prima firma comerciala de productie a robotilor*. Realizeaza primul lor robot, numit “*Unimate*”;
- **1960**, *Condec Corporation cumpara Unimation si începe producerea robotilor **Unimate***;
- **1962**, General Motors cumpara robotul industrial Unimate si îl instaleaza pe linia de productie, pentru turnarea sub presiune.



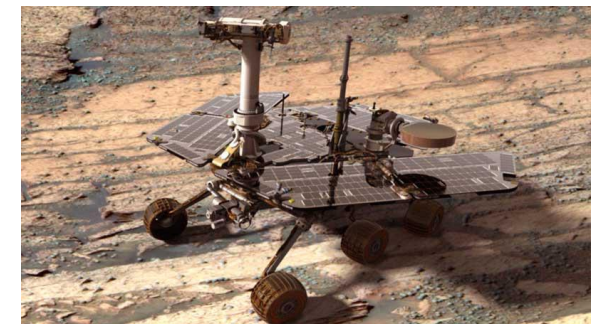
Robotul Unimate

## Momente cheie in evolutia robotilor

- **1965**, Carnegie Mellon University înființează Institutul de Robotica;
- **1966**, aterizează pe luna naveta spațială robotizată “Surveyor”, trimisă de NASA;
- **1968**, la Stanford Research Institute, California, este construit primul sistem robotizat, Shakey, cu vedere artificială;
- **1987-1993**, Honda produce roboții umanoizi din seria E (E1 – E6);
- **1993-1996**, Honda produce roboții umanoizi din seria P (P1, P2, P3);
- **1997**, naveta spațială Pathfinder (NASA) aterizează pe Marte, având la bord robotul mobil Sojourner, pentru explorarea planetei;
- **1999**, Sony scoate pe piață prima versiune a robotului de companie AIBO;
- **2004**, NASA, roboții Spirit și Opportunity ajung pe Marte;



AIBO



Opportunity



## Probleme etice in robotica

**Isaac ASIMOV** a propus "*cele trei legi fundamentale*", legi ce descriu regulile pe care trebuie sa le respecte robotii, astfel încât sa nu creeze prejudicii oamenilor:

- *Un robot nu are voie sa pricinuiasca ceva rau omului sau sa permita, prin neinterventia lui, sa i se întâmple ceva unei fiinte umane.*
- *Un robot trebuie sa execute ordinele (comenzile) primite de la o fiinta umana, exceptând cazurile când acestea ar contraveni primei legi.*
- *Un robot trebuie sa-si conserve propria existenta, atât timp cât aceasta nu contravine primelor doua legi.*

Profesorul **Gerald NORMAN**, de la *Oregon Institute of Technology*, a sugerat ca poate fi adoptata si a patra lege a roboticii:

- *Un robot poate lua locul unui om. Dar nu poate lasa acea persoana fara slujba (Industry 4.0).*



## Robotica

Stiinta care se ocupa cu studiul robotilor este numita “**robotica**”.

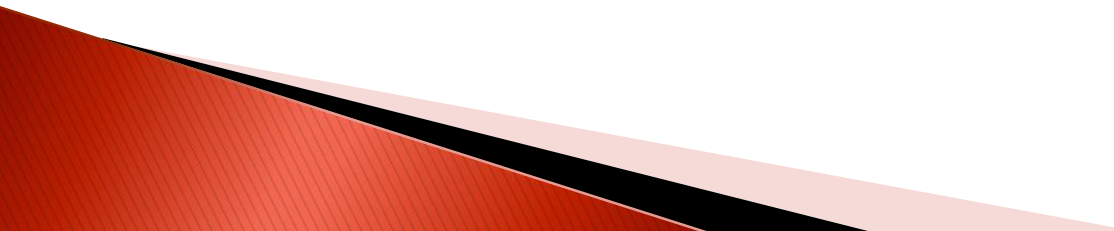
**Robotica** este disciplina care implica:

- a) proiectarea, fabricarea, programarea si controlul robotilor;
- b) utilizarea robotilor pentru efectuarea unor activitati;
- c) studiul modelelor si metodelor de control si simulare;
- d) senzori si sisteme de actionare;

## Robotica

**Robotii** au devenit sisteme indispensabile functionarii societatii umane.

Robotii sunt întâlniți atât în procesele de producție, cât și în alte domenii de interes precum:

- divertisment;
  - transporturi;
  - comunicații;
  - medicină;
  - agricultură;
  - etc. pentru imbunatatirea performanțelor.
- 

## Robotica

Pentru sistemele flexibile de fabricatie, atat din punct de vedere ingineresc cat si managerial sunt folositi **Robotii industriali**.

Marea majoritate a structurilor robotice folosite în industrie au bază fixă, operând ca brațe robotice (manipulatoare) într-un mediu bine delimitat, ca răspuns la necesitatea omului de automatizare a proceselor de producție, mai ales, a celor repetitive, sau periculoase.

## Robotica: cercetare multi-disciplinara

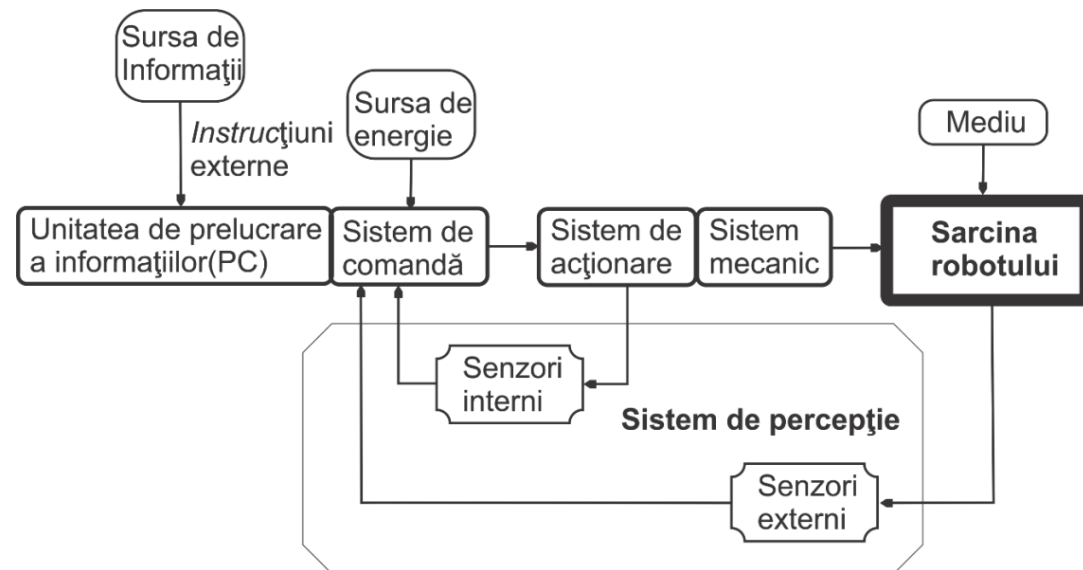
*Construcția unui robot cere cunoștințe din domenii foarte diferite. Pentru a îndeplini chiar o misiune foarte simplă, este nevoie de sisteme complicate, care acoperă multe discipline.*

## Structura generala a unui sistem robotic

Sistemul de comanda, sistemul de actionare si sistemul de perceptie constituie componentele de conducere a sistemul mecanic (sistemul condus).

Prin sistemul de comanda a unui robot se determina flexibilitatea si eficienta acestuia, considerand:

- ✓ sursa de informatie – ce defineste modul de operare, algoritmi de conducere si prelucrarea informatiilor;
- ✓ sursa de energie;
- ✓ spatiul de operare – este definit direct de parametrii structurii mecanice.



## Clasificarea robotilor

- **Dupa generatia tehnologica:**
  - Roboti care reproduc o secventa a unor instructiuni înregistrate (vopsire, sudare prin puncte, robotii "pick-and-place"), care utilizeaza limitatori de cursa.
  - Roboti controlati prin senzori, control în bucla închisa cu cicluri fixe;
  - Roboti adaptivi, ce se pot reprograma automat, pe baza informatiilor de la senzori.
  - Roboti inteligenti (smart), sunt robotii ce utilizeaza tehnici de inteligenta artificiala pentru luarea deciziilor si rezolvarea problemelor.

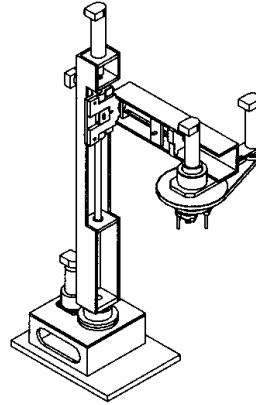
## Clasificarea robotilor

- Dupa nivelul de inteligenta:
  - Sisteme cu comanda manuala, teleoperare manuala;
  - Roboti cu program fix de executie a task-urilor;
  - Roboti cu program flexibil, de executie a task-urilor;
  - Roboti comandati pe baza unui cod sursa;
  - Roboti comandati prin invatarea miscarilor pe care le are de executat;
  - Roboti inteligenti, care interactioneaza cu mediul, reactionand la modificarile acestuia.

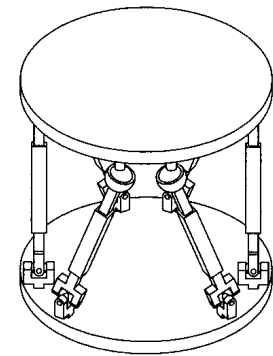
## Clasificarea robotilor

- După structura cinematică

- Roboți seriali



- Roboți paraleli





## Clasificarea robotilor

- Dupa mobilitate

- Roboti mobili

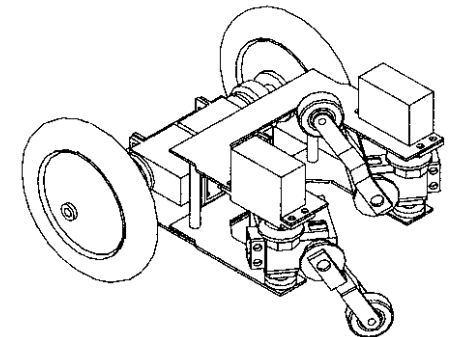
- Roboti mobili cu roti sau senile



- Roboti pasitori (dintre acestia, o categorie aparte o constituie robotii umanoizi)



- Roboti mobili cu locomotie hibrida etc.



## Clasificarea robotilor

- Dupa domeniul de aplicare:
  - Roboti industriali;
  - Roboti pentru servicii;
  - Roboti de explorare;
  - Roboti sociali.

## Clasificarea robotilor

- Dupa structura mecanica

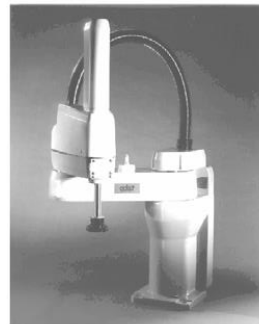
- Robot tip brat articulat



- Robot tip pistol (cilindric)



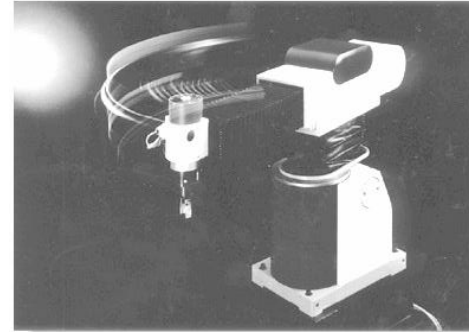
- Robot tip SCARA



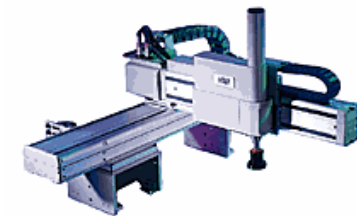
## Clasificarea robotilor

- Dupa structura mecanica

- Robot tip turela



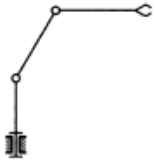
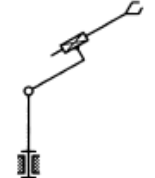
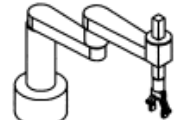
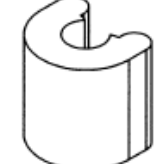
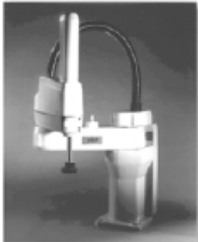
- Robot cartezian

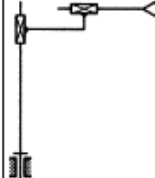
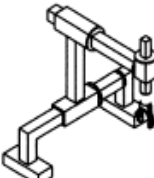
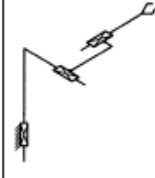
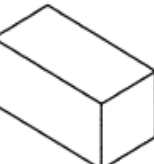
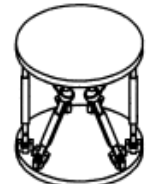
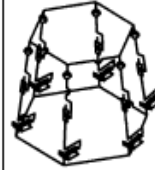


- Robot paralel



## Clasificarea robotilor

| Soluție constructivă                                                              | Schemă structurală                                                                 | Spațiu de lucru                                                                    | Exemplu                                                                            |
|-----------------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
|   |   |   |   |
|   |   |   |   |
|  |  |  |  |

| Soluție constructivă                                                                | Schemă structurală                                                                   | Spațiu de lucru                                                                      | Exemplu                                                                              |
|-------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------|
|   |   |   |   |
|   |   |   |   |
|  |  |  |  |

## Clasificarea robotilor

- **Clasificarea robotilor pentru servicii**

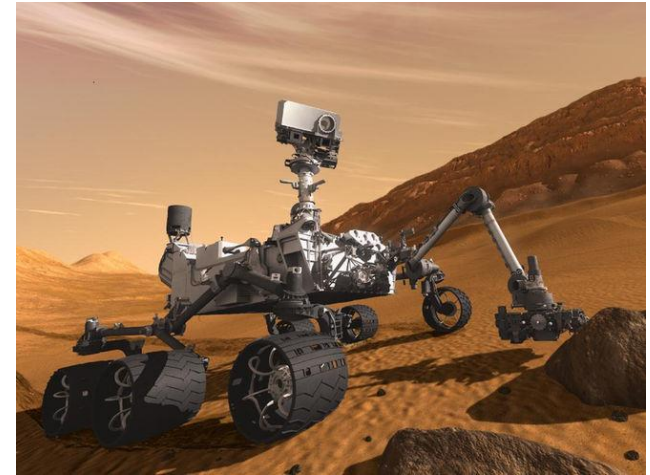
Robotul pentru servicii este un robot autonom sau semi-autonom, care efectueaza servicii utile omului sau echipamentelor, excluzând operatiile industriale.

- *Roboti ce deserveasc oamenii (roboti casnici; roboti pentru: securitate personala, divertisment, chirurgie, asistarea persoanelor handicapate, etc.)*
- *Roboti ce deserveasc echipamente (roboti pentru: alimentarea autoturismelor cu combustibil, spalarea geamurilor avioanelor; inspectia, curatarea si întretinerea unor echipamente, etc.)*
- *Alti roboti pentru servicii, ce efectueaza o activitate autonoma (robot mobil pentru servicii de tip „curier” în spitale; roboti pentru curatenie în aeroporturi, gari, etc.; roboti pentru taierea gazonului; roboti de paza; etc.)*



## Clasificarea robotilor

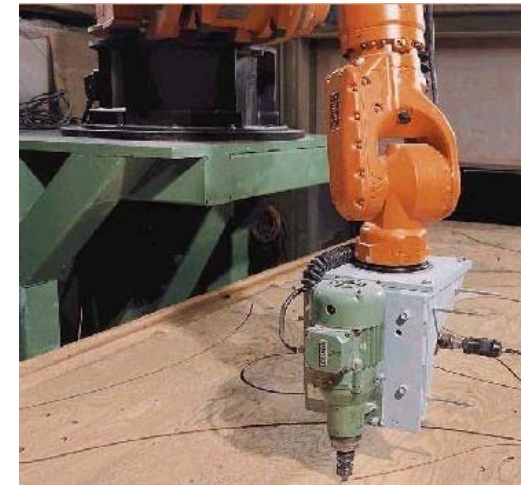
- Clasificarea robotilor de explorare
    - Roboti comandati prin unde radio
      - *Roboti subacvatici*
      - *Roboti militari*
      - *Roboti spatiali*
      - *Roboti pentru medii nucleare etc.*
- teme master-slave (teleroboti)



# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Aplicatii industriale

- Alimentare cu piese a M-U
- Montaj, autoturisme
- Debavurare piese injectate
- Decupare prin aschiere





## Aplicatii industriale

- Asamblare
- Debavurare
- Decupare
- Manipulare
- Masurare-control
- Paletizare
- Sudare:
  - prin puncte
  - cu arc electric
  - cu gaz
  - cu laser
- Taiere cu plasma
- Vopsire, etc.



## Aplicatii medicale



The Chir Team at INRIA Sophia Antipolis integrates vision and robotic algorithms around the Da Vinci (TM Intuitive Surgical, Inc) system in collaboration with Professor Carpentier's team (Hopital Broussais) in heart surgery.



# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

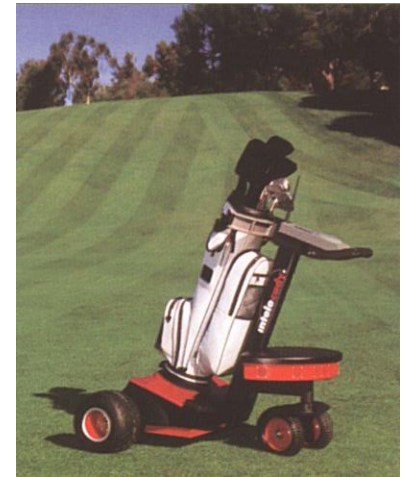
## Aplicatii ale robotilor pentru servicii

- Roboți pentru domeniul forestier
- Roboți pentru alimentare cu combustibil
- Ghidare în aeroporturi
- Roboți pentru construcții
- Roboți pentru curățenie
- Proteze
- Teleteze
- Asistentă subacvatică



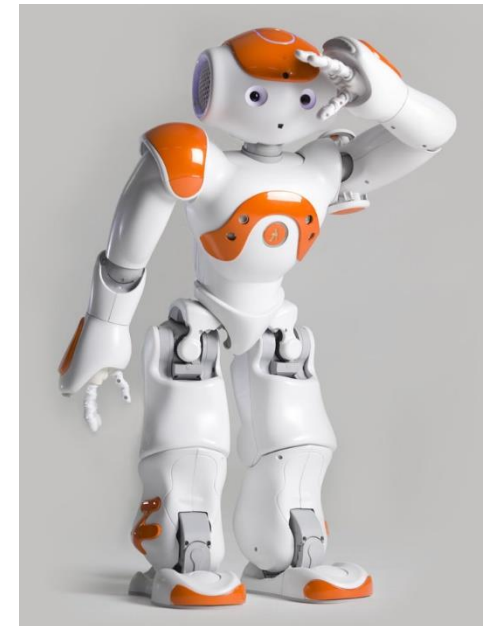
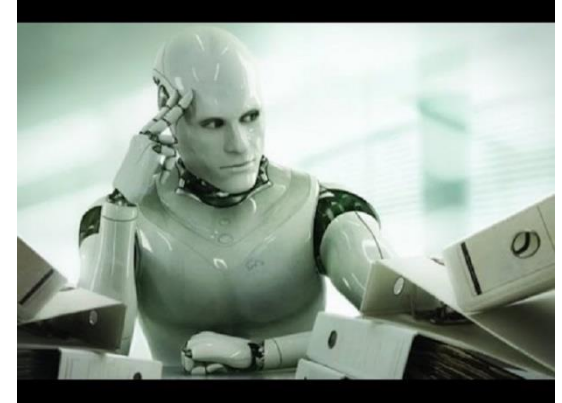
## Aplicatii ale robotilor pentru servicii

- Roboti pentru divertisment
- Sport
- Roboti pentru hotel/restaurant
- Roboti pentru marketing



## Aplicatii ale robotilor sociali

- Roboți pentru divertisment





# **Programarea robotului industrial**





**UNIVERSITATEA TEHNICĂ**  
DIN CLUJ-NAPOCA

**Facultatea de Construcții de Mașini**  
**Departamentul Ingineria Proiectării și Robotică**

# **PROGRAMAREA ROBOTILOR INDUSTRIALI**



**Sl. Dr. Ing. Florin COVACIU**

**2019**

## Implementarea robotilor industriali

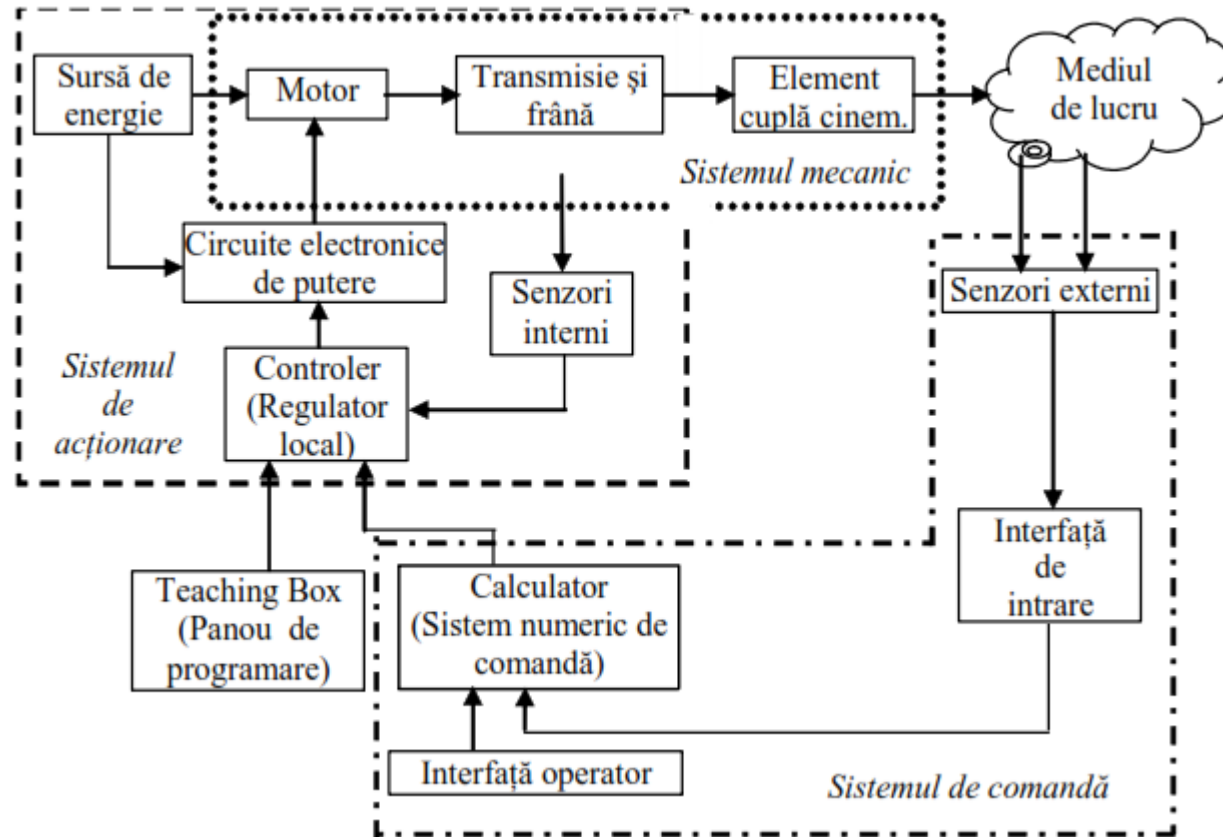
Implementarea sistemelor flexibile de fabricatie (SFF-FMS) presupune utilizarea robotilor industriali, ca produse mecatronice, ce combina componente mecanice, electronice si informatice.

Robotii industriali sunt utilizati pentru a executa în principal operatii de manipulare, deplasare si transport, cu viteza, pozitionare si forte controlate. Despre robotii industriali se poate spune ca:

- **poseda mai multe grade de libertate** (2-6), încât sa poata executa operatii complexe;
- **sunt relativ autonomi**, dotati cu o memorie reprogramabila capabila sa stocheze date pentru executarea operatiilor, pe baza capacitatii de a utiliza functii logice cu diferite grade de complexitate, ceea ce confera un anumit nivel de “inteligenta” sistemului robotizat.



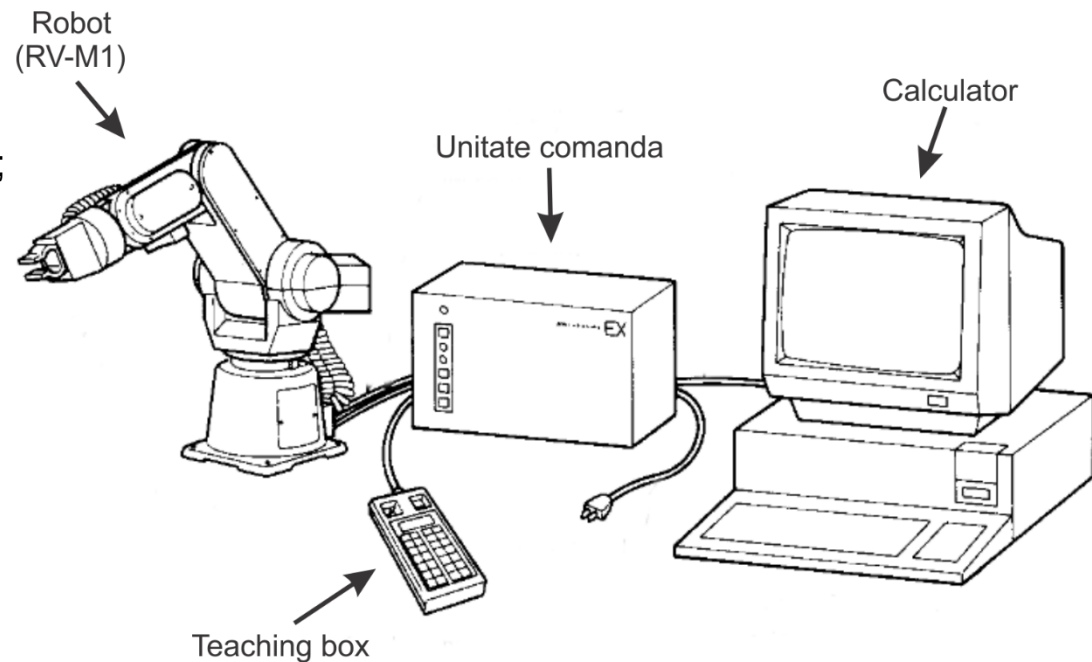
## Schema bloc a unei structuri robotice



## Sistem robotic model MITSUBISHI RV-M1

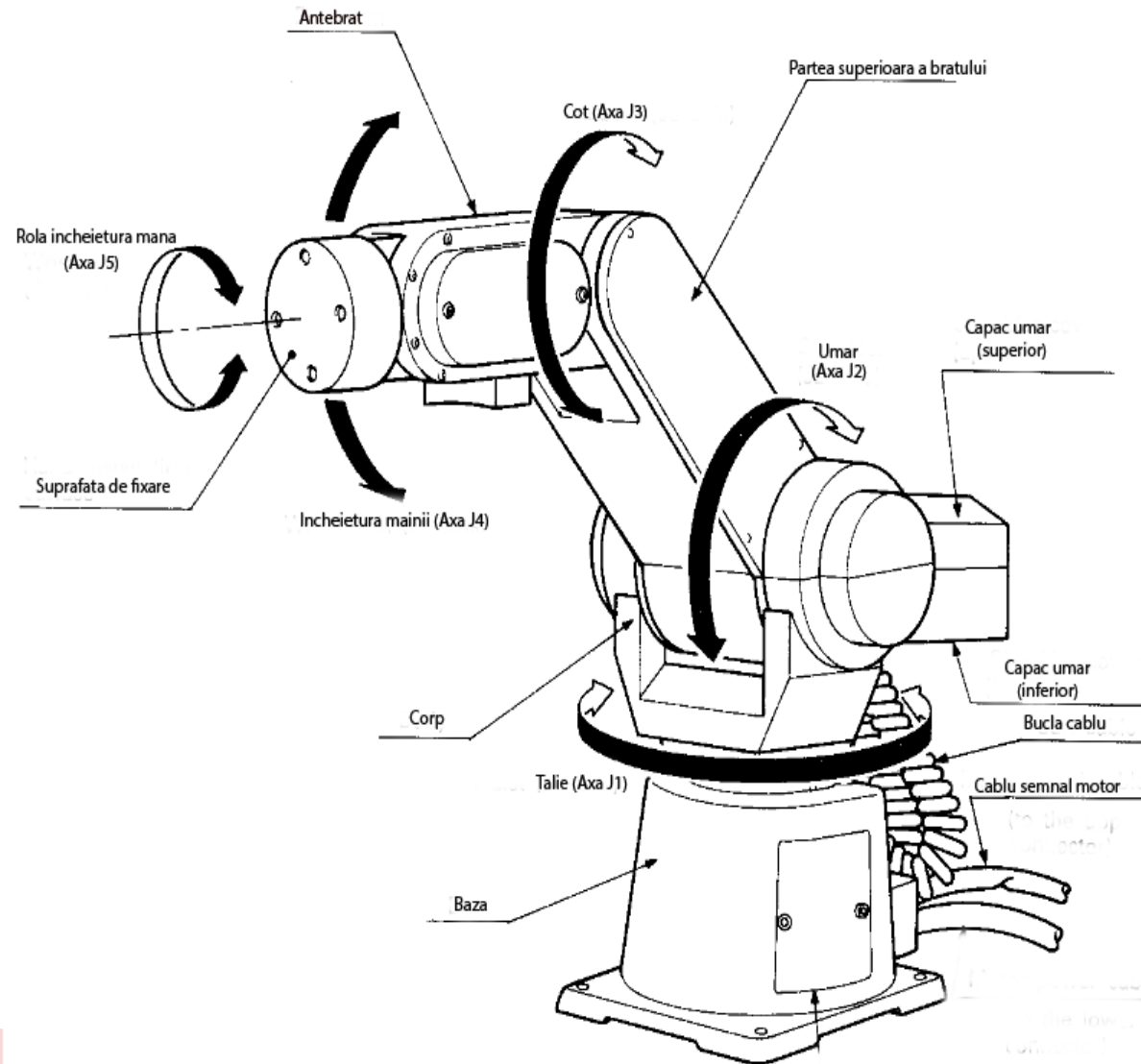
Este un sistem cu 5 DOF ce poate fi comandat pentru executarea unor operatii de manipulare, ce se compune din:

- structura mecanica;
- UC - unitate comanda (controler);
- Teaching box – unitatea de teleoperare;
- sistemul de calcul.



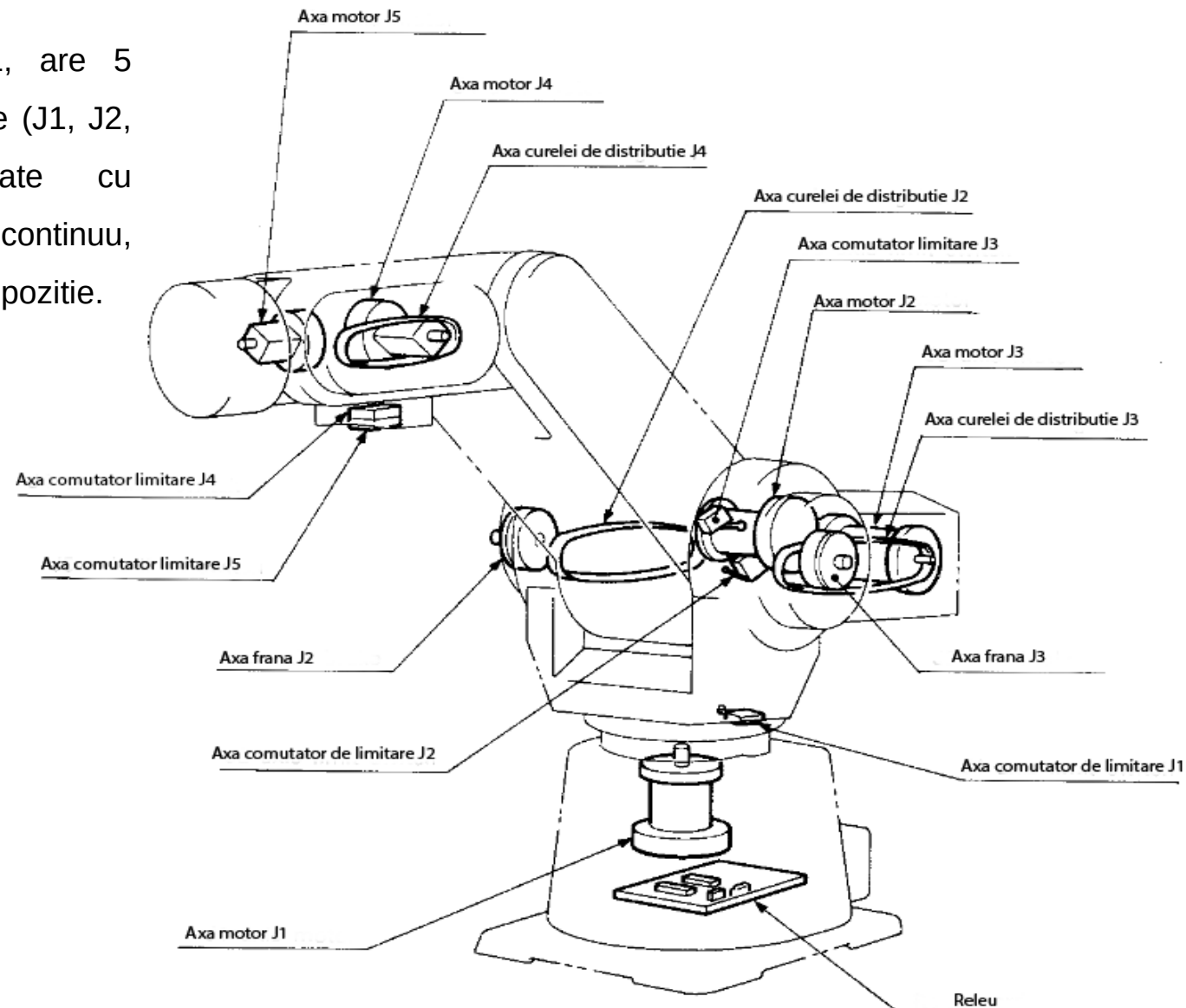
## Sistem robotic model MITSUBISHI RV-M1

Robotul industrial RV-M1, are 5 cuple cinematice de rotație (J1, J2, J3, J4, J5), actionate cu servomotoare de curent continuu, echipate cu traductoare de pozitie.

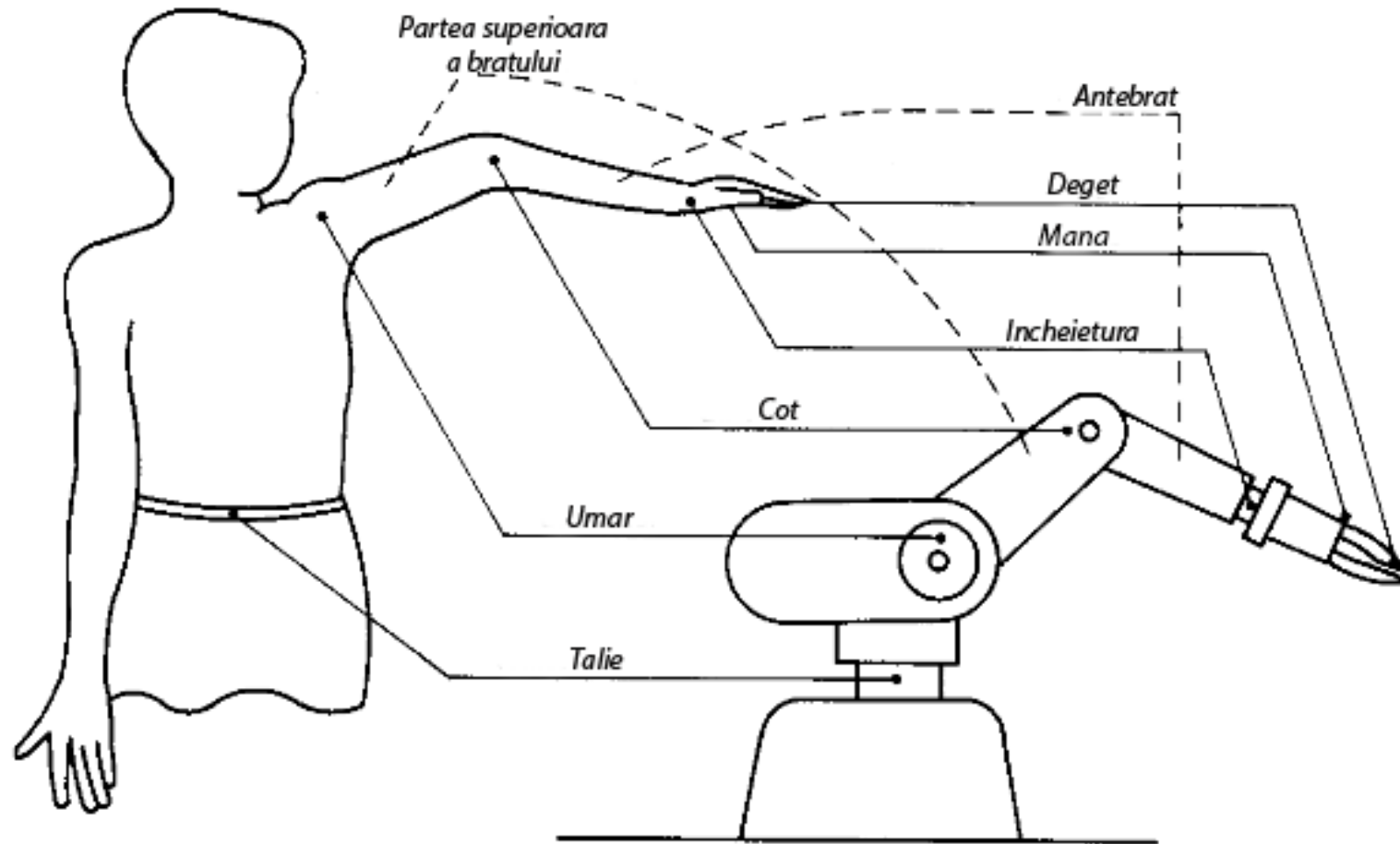


## Sistem robotic model MITSUBISHI RV-M1

Robotul industrial RV-M1, are 5 cuple cinematice de rotație (J1, J2, J3, J4, J5), actionate cu servomotoare de curent continuu, echipate cu traductoare de pozitie.



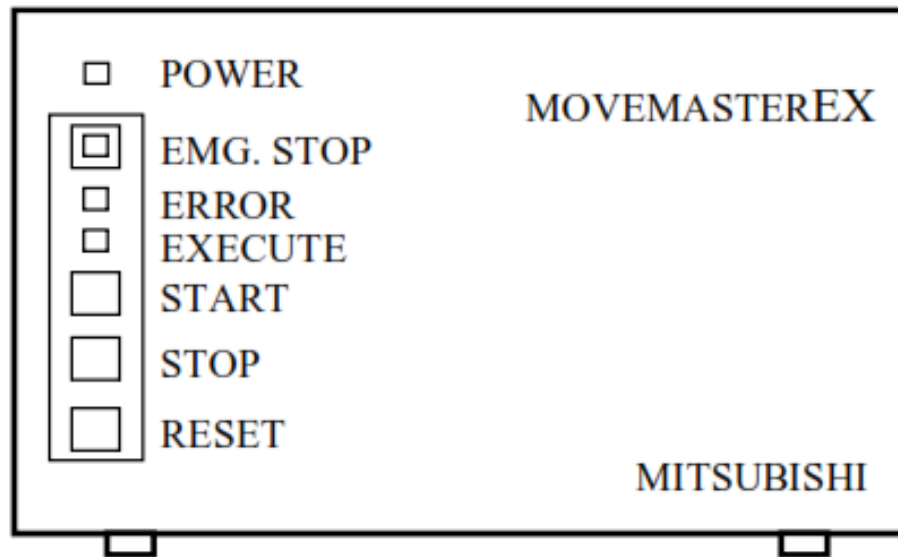
## Analogia dintre structura mecanica si corpul uman



## Unitatea de comanda (UC)

Cu UC se realizeaza alimentarea si comanda robotului, utilizand ca terminale sistemul de calcul si Teaching box-ul, la care se pot adauga orice echipament extern.

UC contine 4 placi electronice care asigura alimentarea si transferul de date dintre UC si terminale prin cabluri specializate, separate pentru alimentare si transfer de date.



POWER – alimentare unitate comanda;  
EMG. STOP – stop de urgenta;  
ERROR – semnalizare erori;  
EXECUTE – comanda data este executata;  
START – executare program;  
STOP – opreste executia programului;  
RESET – resetare program.

## Teaching box – unitatea de teleoperare

Teaching box se utilizeaza pentru comanda manuala a sistemului robotic si pentru memorarea pozitiilor prin care trece punctul caracteristic al robotului si se poate realiza:

- sincronizarea robotului (deplasarea axelor în originea lor mecanica);
- memorarea pozitiilor în care este situat robotul;
- rularea programului în regim pas cu pas (*Step by Step*).

**Obs:** Dupa fiecare punere sub tensiune a sistemului robotic, robotul trebuie adus în pozitia de origine mecanica (NEST) pentru sincronizare axelor de rotatie.

## Teaching box – unitatea de teleoperare

Comutatorul **ON/OFF**, activeaza teaching box-ul. Butonul pentru oprirea de avarie (*Emergency Stop*), actioneaza similar cu cel de pe UC.

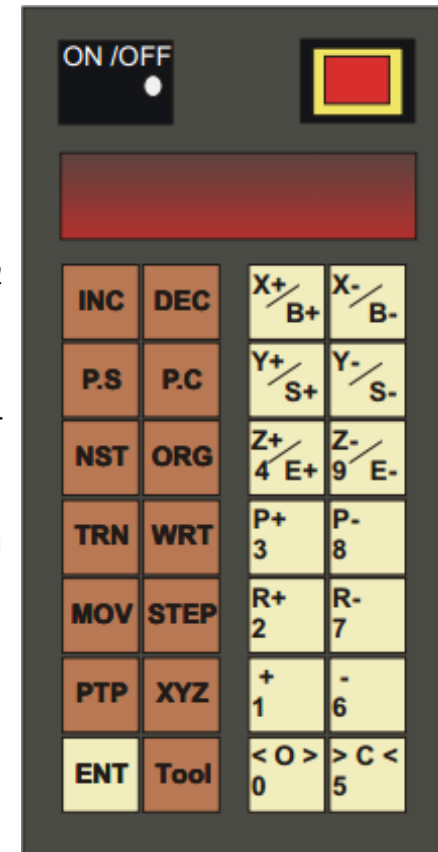
Miscările coordonate de teaching box, pot avea loc in doua sisteme de coordonate:

- cel al articulatiilor (PTP);
- Cartezian (X, Y, Z).

Blocurile de taste de pe teaching box, pot avea mai multe functii si determina doua moduri de reactie:

- actiune imediata atat timp cat tasta este apasata, matricea de taste din partea dreapta;
- Programare de functii, cu confirmate prin tasta ENT, matricea de taste din partea stanga.

Teaching box





## Teaching box – unitatea de teleoperare

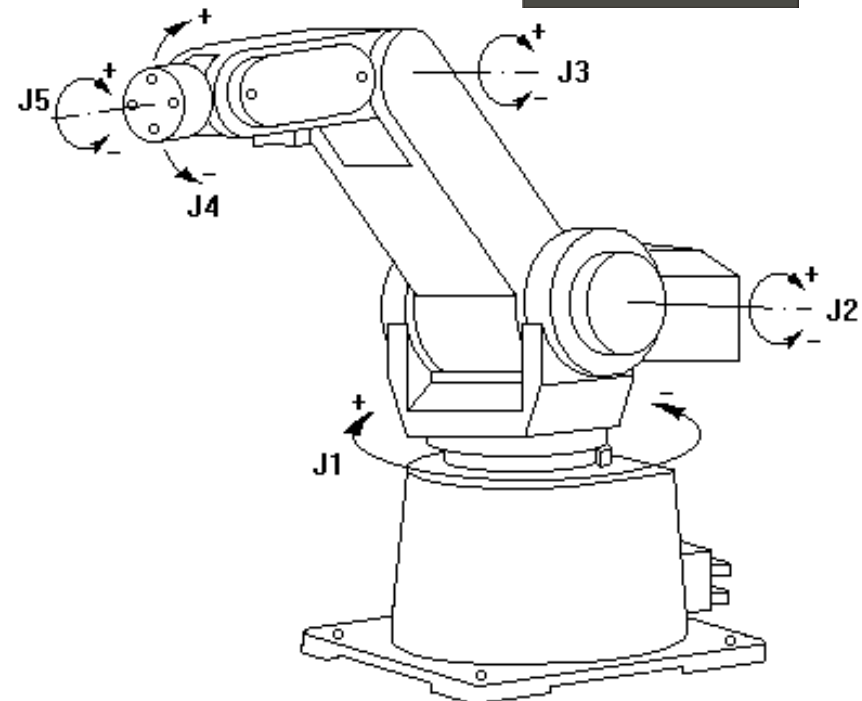
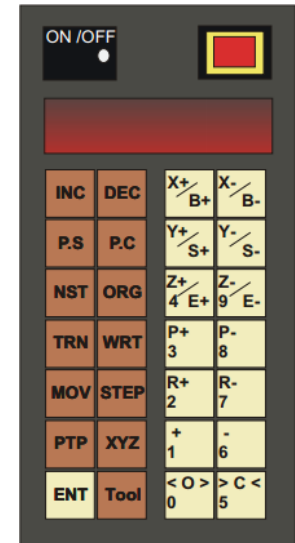
### Actiune imediata

Implicit sistemul functioneaza in sistemul de coordonate al articulatiilor.

Apasarea tastelor (B+, B-, S+, S-, E+, E-, P+, P-, R+, R-) se poate face doar independent.

Robotul va efectua o miscare de rotatie la nivelul cuplelor cinematice (J1, J2, J3, J4, sau J5) în sensul specificat de semnul "+" sau "-".

### Teaching box



## Teaching box – unitatea de teleoperare

### XYZ

Miscare paralela cu axele sistemului de coordonate cartezian si invalideaza regimurile PTP si TOOL.

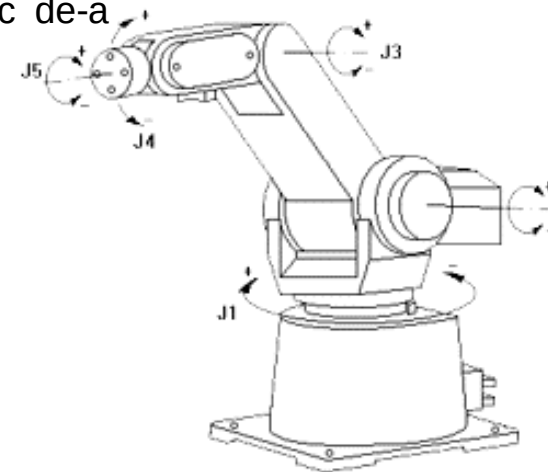
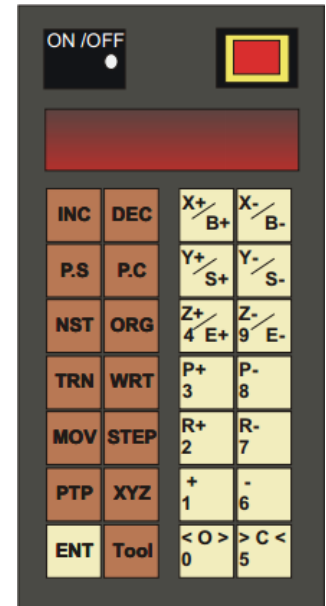
Vârful sculei (punctul caracteristic - TCP) se va putea misca paralel cu axele sistemului cartezian dupa directiile X+, X -, Y+, Y-, Z+, Z-, impunand miscarea simultana a mai multor cuple cinematice ale robotului (J1, J2, J3, J4, J5), utilizand modelul DKP.

Originea sistemului cartezian este fixata la baza robotului, in raport cu aceasta sunt calculate coordonatele pozitiilor in care trebuie plasat TCP.

### TOOL

Invalideaza regimurile PTP si XYZ, permitand miscarea bratului robotic de-a lungul axei efectorului final. Sunt active tastele: Z+ si Z-.

### Teaching box



## Descriere functii de pe Teaching box – PTP (sistem al articulatiilor)

### PTP

**B+/ B** - pereche de taste ce asigura rotirea in sens orar respectiv antiorar in jurul axei bazei;

### PTP

**S+/ S** - pereche de taste ce asigura miscarea bratului robotic de la umar (shoulder) pe verticala;

### PTP

**E+/ E** - pereche de taste ce asigura miscarea bratului robotic de la cot (elbow) pe verticala;

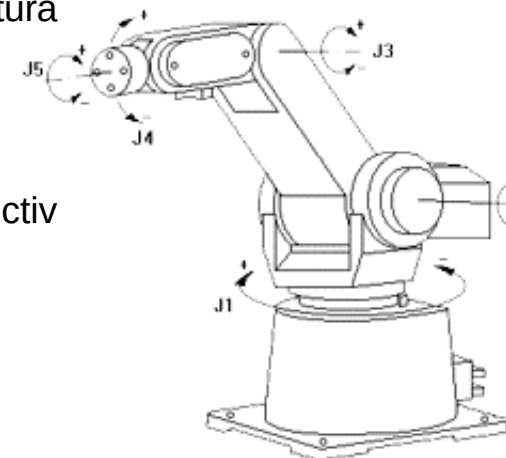
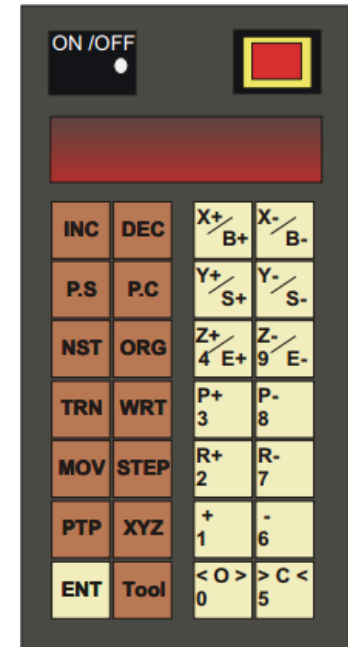
### PTP

**P+/ P** - pereche de taste ce asigura miscarea bratului robotic de la incheietura mainii (pitch/wrist) pe verticala;

### PTP

**R+/ R** - pereche de taste ce asigura rotirea incheierurii in sens orar, respectiv antiorar.

### Teaching box



## Descriere functii de pe Teaching box – XYZ (sistem cartezian)

XYZ

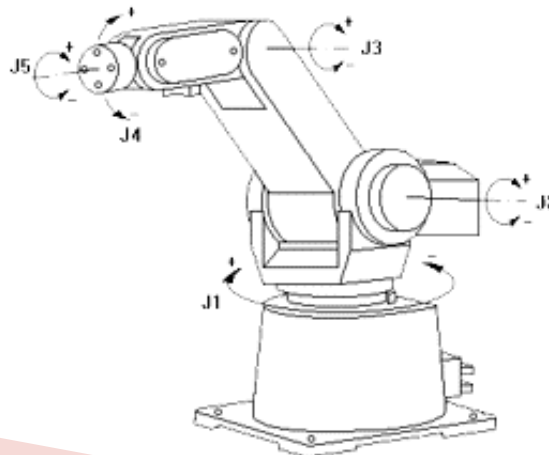
**X+ / X -** pereche de taste ce asigura miscarea robotului in jurul bazei pe orizontala

XYZ

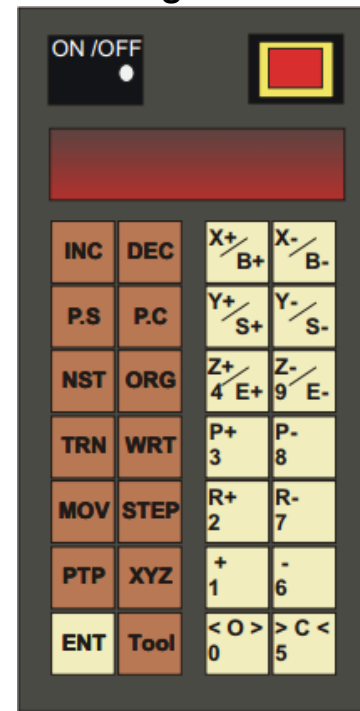
**Y+ / Y -** pereche de taste ce asigura miscarea bratului robotic de la talie, cot, umar si incheietura pe verticala;

XYZ

**Z+ / Z -** pereche de taste ce asigura miscarea bratului robotic pe axa Z.



### Teaching box



# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Descriere functii de pe Teaching box

### TOOL

**Z+/ Z-** permite miscarea bratului robotic de-a lungul axei efectorului final pe verticala;

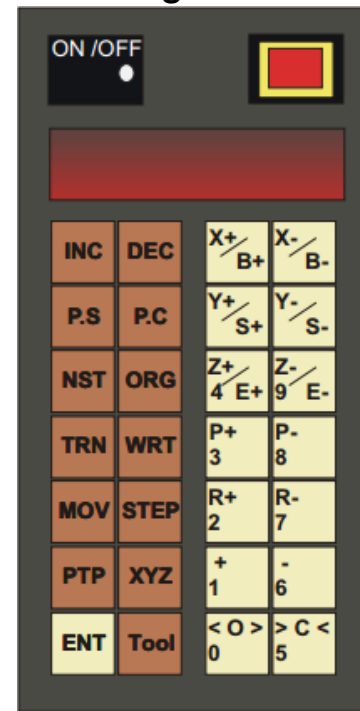
### efector final

**O** - deschiderea griper-ului;

### efector final

**C** - inchiderea griper-ului;

### Teaching box



# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Descriere functii de pe Teaching box

### INC+ENT

Comanda pentru deplasarea robotului într-o pozitie memorata, având numar de ordine mai mare si totodata cel mai apropiat de cel al pozitiei curente.

### DEC+ENT

Comanda pentru deplasarea robotului într-o pozitie memorata, având numar de ordine mai mic si totodata cel mai apropiat de cel al pozitiei curente.

### PS+numar+ENT

Memoreaza coordonatele punctului curent din spatiu, unde se afla punctul caracteristic al robotului, atasându-i un numar de ordine. Daca pentru doua pozitii diferite se ataseaza acelasi numar, ultima definita ramâne memorata.

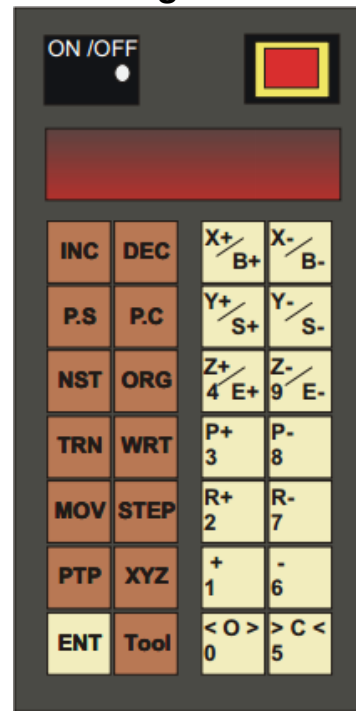
### PC+numar+ENT

Elimina din memorie datele pozitiei cu numarul specificat.

### NST+ENT

Comanda pentru deplasarea robotului în pozitia de origine mecanica în vederea sincronizarii cuplelor cinematice.

### Teaching box



# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Descriere functii de pe Teaching box

### ORG+ENT

Comanda pentru deplasarea robotului în pozitia de referinta a sistemului de coordonate cartezian.

### TRN+ENT

Comanda pentru transferul continutului EPROM-ului utilizat (programul de comanda si datele pozitiilor memorate) în memoria RAM a unitatii de comanda.

### WRT+ENT

Comanda pentru înscrierea programului de comanda si a datelor pozitiilor memorate (aflate în memoria unitatii de comanda) în EPROM.

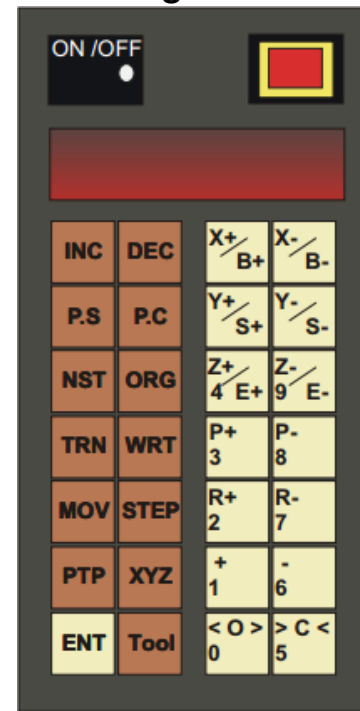
### MOV+numar+ENT

Comanda pentru deplasarea robotului, respectiv a punctului caracteristic într-o pozitie specificata. Miscarea are loc doar daca pozitia specificata a fost memorata anterior.

### STEP+numar+ENT

Comanda pentru executia programului în regim pas cu pas începând cu linia indicata prin "numar".

### Teaching box



# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Descriere functii de pe Teaching box

### PTP + ENT

Selectarea sistemului de articulatii.

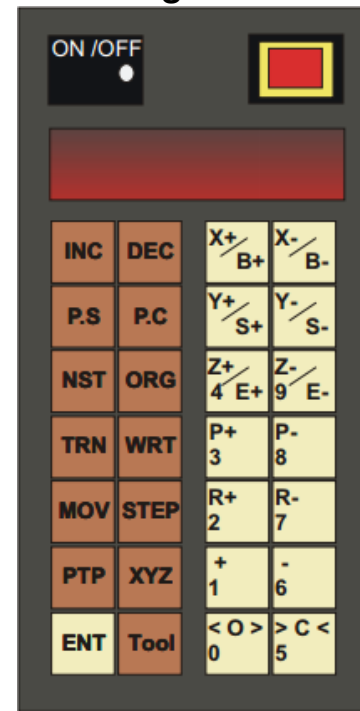
### XYZ + ENT

Selectarea sistemului cartezian.

### Tool + ENT

Selectarea operatiei miscarii bratului robotic de-a lungul axei efectorului final.

### Teaching box





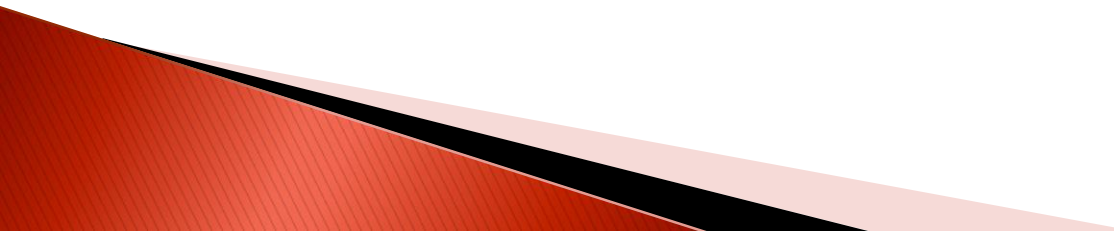
## Comanda PTP – deplasare pe traiectorie

**Comanda** robotilor industriali presupune executarea de operatii, a caror caracteristici si succesiune au fost stabilite sub forma unui program.

**Programul** este compus dintr-o succesiune de macroinstructiuni ce formeaza linii de program, transmise succesiv comenzii catre elementele de actionare ale robotului, prin intermediul unitatii de comanda.

**Macroinstructiunile** determina pozitionari, mod de deplasare, temporizari, salturi la subrutine, comenzi pentru setarea efectorului final.

**Deplasarea** robotului poate fi programata: punct cu punct (PTP), sau pe o traiectorie continua.



## Comanda PTP – deplasare pe traiectorie

În cazul **comenzii de tip punct cu punct**, se deplasează TCP-ul prin salt la punctul de coordonate programat.

- activitatea tuturor articulațiilor are același moment de start și același moment de stop;
- traiectoria rezultată depinde de poziționarea punctelor și structura geometrică a robotului.

În cazul comenzii de deplasare pe o traiectorie (path control), între anumite puncte esențiale se pot genera traiectorii de mișcare prin interpolare:

- traiectorii liniare;
- circulare;
- eliptice;
- curbe de tip spline.

Indiferent de tipul de comandă, se poate seta numărul de puncte intermediare pentru deplasare (rafinarea mișcării) și trebuie verificată condiția de includere a traiectoriei a TCP-ului nu doar în spațiul operational ci și în spațiul de lucru.

## Principalele macroinstrucțiuni utilizate la realizarea programelor

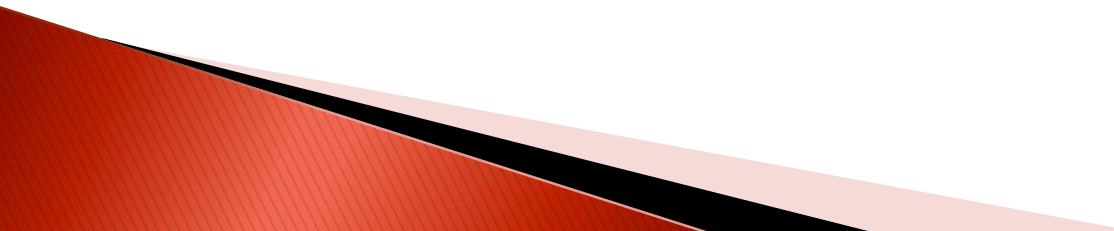
### **NT** (Nest)

PS(point set) Determina deplasarea robotului în poziția de origine mecanică în vederea sincronizării axelor.

**DP** (Decrement Position) - Este comanda pentru deplasarea punctului caracteristic al robotului într-o poziție memorată având numărul de ordine mai mic și în același timp cel mai apropiat de cel al poziției curente.

**IP** (Increment Position) - Reprezintă comanda pentru mișcarea robotului într-o poziție memorată, având numărul de ordine mai mare și în același timp cel mai apropiat de cel al poziției curente.

**GT** (Go To) Determină un salt necondiționat la linia din program specificată.



## Principalele macroinstrucțiuni utilizate la realizarea programelor

**MO** (Move) - Determina deplasarea punctului caracteristic în poziția specificată, prin interpolare articulată. Se poate specifica și starea gripper-ului (**C**-închis / **O**-deschis) indiferent de starea acestuia în momentul memorării poziției.

**MS** (Move Straight) – Miscare la o anumită poziție prin interpolare circulară.

**MC** (Move Continuous) - Miscare prin puncte intermediare continuu.

**GO** (Gripper Open) - Determina deschiderea gripper-ului.

**GC** (Gripper Close) - Determina închiderea gripper-ului.

**TI** (Timer) Introduce o anumită temporizare în timpul rularii programului.

# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

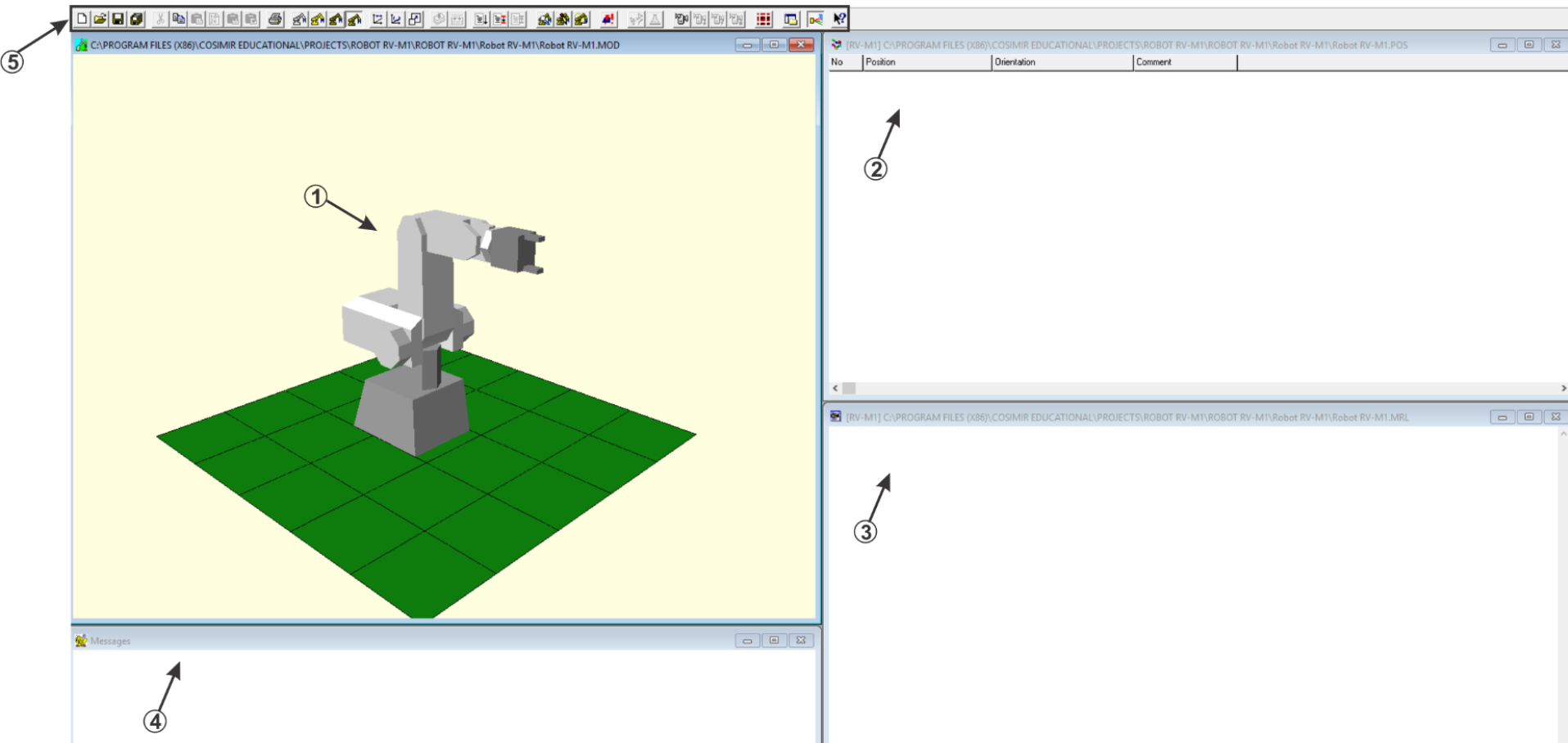
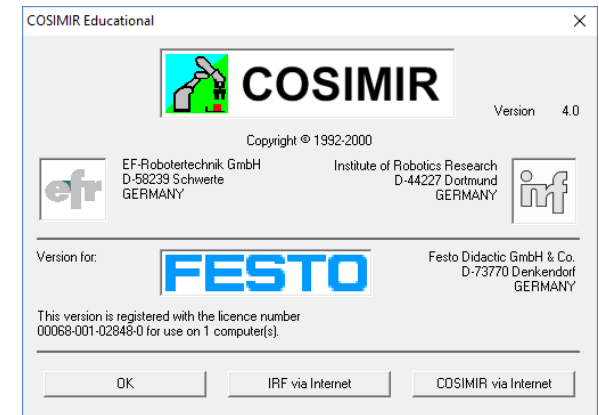
## Film demonstrativ robot RV-M1



# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Mediul virtual COSIMIR

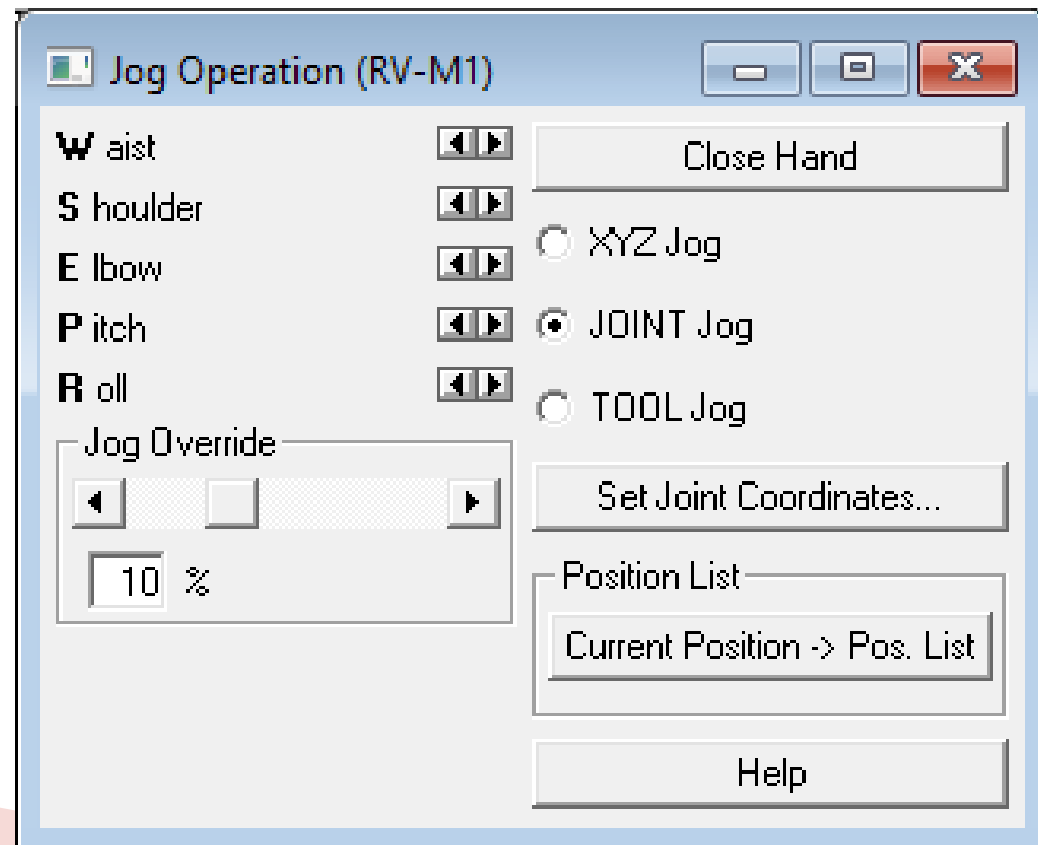
1. Robot RV-M1 Movemaster EX;
2. Vizualizare puncte setate (PS);
3. Editare cod sursa;
4. Afisare Mesaje si Erori.
5. Bara de lucru



## Mediul virtual COSIMIR

### Fereastra de teleoperare

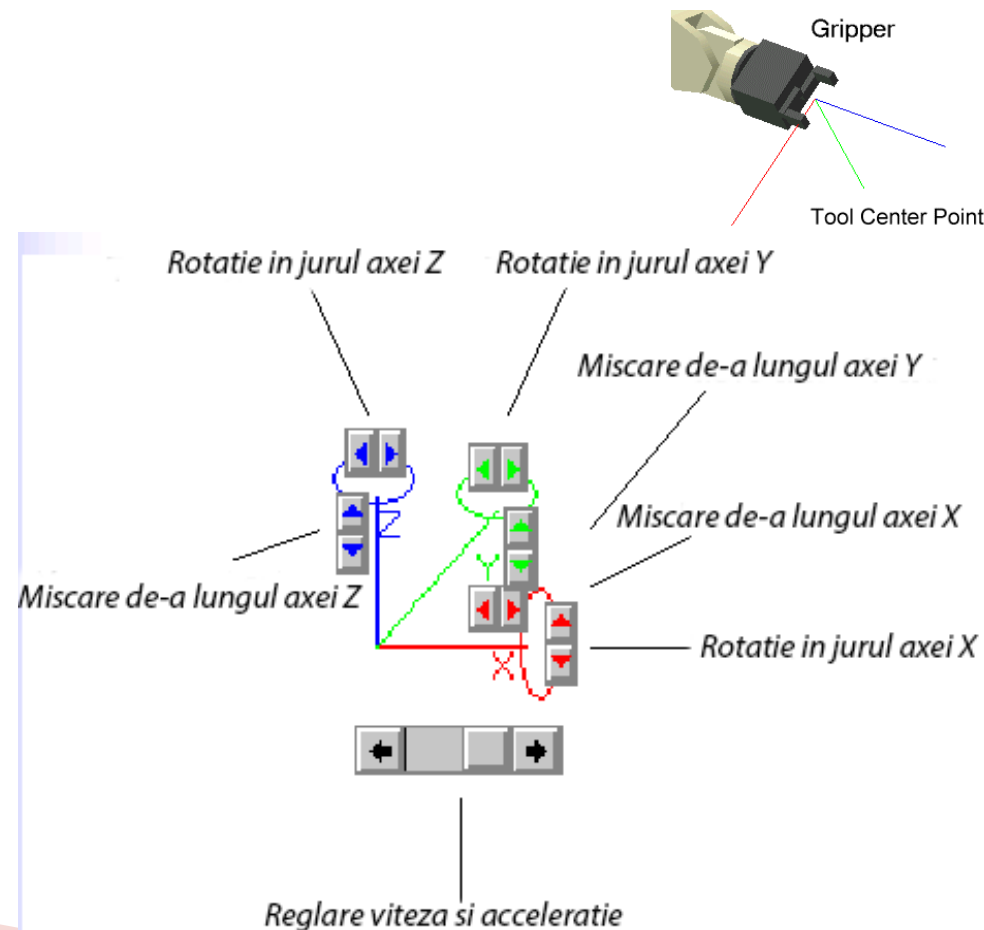
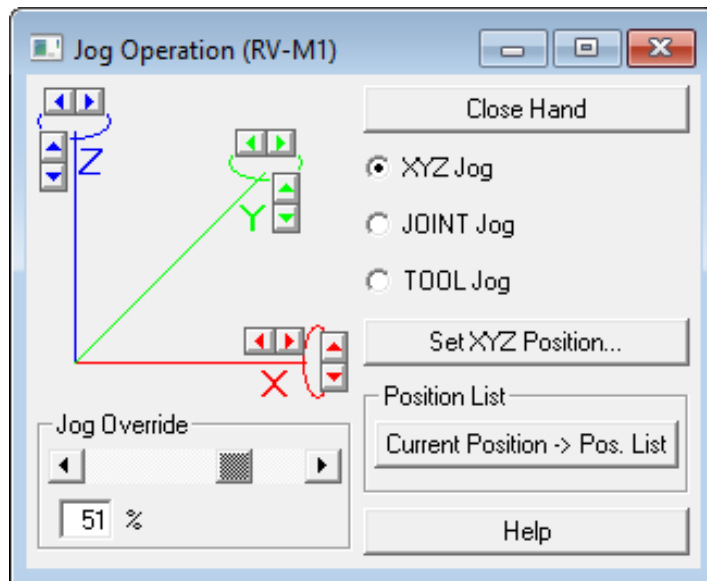
- selectarea unei articulatii;
- selectarea vitezei de miscare;
- inchidere griper;
- alegerea modului de lucru;
- setarea coordonatelor unui punct;
- identificarea coordonatelor unui punct.



## Mediul virtual COSIMIR

### Deplasarea in sistem cartezian

Miscările sunt paralele cu axele sistemului de coordonate cartezian, schimbarea poziției și orientării TCP (Tool Center Point) se face cu perechile de butoane de la sfârșitul fiecărei axe

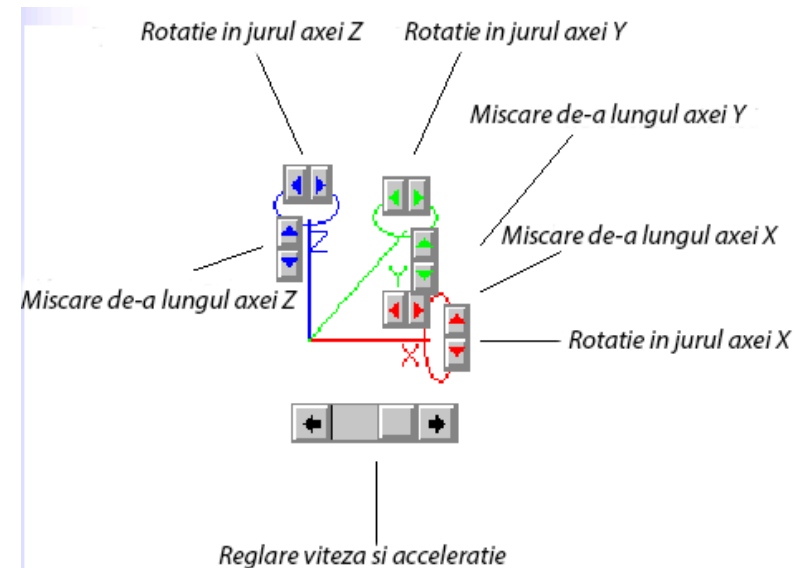
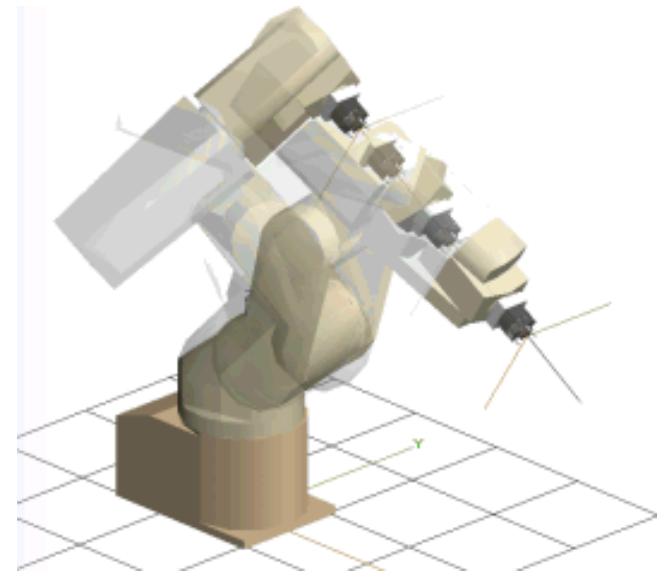
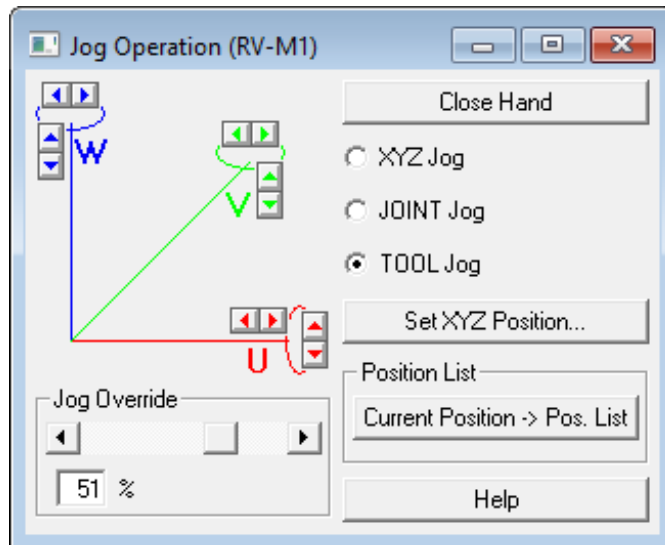




# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Mediul virtual COSIMIR

### Tool Coordinates



## Mediul virtual COSIMIR

### Microinstrucțiuni

SP1 - setare viteza;

MO1 - deplasare în poziția 1

TI 20 - temporizare (2 sec.)

GC - închidere gripper

ED - sfârșit program

### Program aplicație

SP1

MO1

MO2

MO4

TI20

SP4

MO1

MO2

MO5

MO3

GC

SP2

MO1

MO5

ED

# Mașini, Roboți și Echipamente pentru Sisteme Flexibile de Fabricație 2

## Mediul virtual COSIMIR

### Film simulare



## Bibliografie

1. D. Catrina, G. Carutasu, N. Carutasu, A. Totu, C. Cotet, A. Dorin, S. Velicu, Sisteme flexibile de productie, Universitatea Politehnica Bucuresti, 2008
2. O. Spatari, Manualul absolventului de profil electric – Electronica digitala, ULBS 2003
3. T. Muresan, Circuite integrate numerice, Editura: Editura de Vest, 2005
4. R. Morris, Proiectarea cu circuite integrate TTL, Editura: Editura Tehnica, 1974
5. S. Viorel, Roboti industriali. Aplicatii. Volumul I, Editura: Editura Universitaria, 2003
6. J. Horvath, Mastering 3D Printing, Editura: APress, 2014
7. M. Rabaey, A. Chandrakasan, B. Nikolic, Digital Integrated Circuits (2nd Edition), ISBN-13: 978-0130909961, 2003
8. D. Nicula, Electronica Digitala, Editura Universitatii Transilvania din Brasov, 2015
9. N. Jha, Handbook of Flexible Manufacturing Systems, Editor: Academic Press, 2012
10. T. Sobh, Prototyping of Robotic Systems: Applications of Design and Implementation, Editor: IGI Global, 2012