

# **CALCULATOARE NUMERICE**

**Calculatoare 2 2024-2025 Sem. 2**

**CURS 2 Sapt 2 6 martie 2025 12:00-14:00**

**[serban@upit.ro](mailto:serban@upit.ro)**

## **Regulament disciplina**

Nota finala este formata din activitatile:

- Laborator 30%
- Lucrare control (midterm) 20%
- Examen 50%
- Bonus – prezenta activa curs 10%

Conditii pentru promovare:

- Nota de la laborator trebuie sa fie minim 5 (prezenta obligatorie la toate sedintele de laborator);
- Nota la lucrarea de control trebuie sa fie minim 5;
- Nota de la examen trebuie sa fie minim 5.

*In cazul reluarii disciplinei intr-un alt an universitar, activitatile nepromovate trebuie parcurse din nou.*

# **Continut disciplina**

**Structuri digitale folosite în calculatoare numerice**

**Memorii semiconductoare in calculatoare numerice**

**Memoria cache**

**Memoria virtuală**

**Adresarea memoriei in sistemele de calcul**

## Bibliografie

1. **David PATTERSON, John HENNESSY** *Computer Organization and Design The Hardware/Software Interface*, 5th ed., Morgan Kaufmann Elsevier 2012 (a se vedea și traducerea în lb. română David A. Patterson, John L. Hennessy *Organizarea și proiectarea calculatoarelor, Interfața hardware/software*; Editura All, București, 2002);
2. John L. HENNESSY, David A. PATTERSON *Computer Architecture, A Quantitative Approach*, 5th ed., Morgan Kaufmann Publishers, Inc, San Francisco, 2012;
3. Bruce JACOB, Spencer NG, David WANG *Memory Systems Cache, DRAM, Disk* Morgan Kaufmann Elsevier 2008;
4. Miles MURDOCCA, Vincent HEURING *Principles of Computer Architecture*, Prentice Hill, 1999;
5. Jim HANDY *The Cache Memory*, 2nd ed., Academic Press Elsevier, 1998;
6. **Andrew S. TANENBAUM, Todd AUSTIN** *Structured Computer Organization*, 6th ed., Prentice-Hall, Inc., 2013 (a se vedea și traducerea în lb. română Andrew S. Tanenbaum *Organizarea Structurată a Calculatoarelor*, Agora, Tg. Mureș, 2004);
7. **William STALLINGS** *Computer Organization and Architecture: Designing for Performance*, 9th edition, Prentice-Hall Inc., 2013;
8. Vincent HEURING, Harry JORDAN *Computer Systems Design and Architecture*, 2<sup>nd</sup> ed., Person Prentice Hill, 2007;
9. Sajjan G. SHIVA *Computer Organization, Design, and Architecture*, 4<sup>th</sup> ed., CRC Press, Taylor & Francis Group, Boca Raton, USA, 2008;
10. Carl HAMACHER, Zvonko VRANESIC, Safwat ZAKY, Naraig MANJIKIAN *Computer organization and embedded systems*, 6<sup>th</sup> edition McGraw Hill, 2012

# CLASIFICAREA PROCESOARELOR

**1. După numărul de biți** – din care este formată magistrala de date și totodată reprezintă mărimea în biți operanzilor care pot fi procesați printr-o singură instrucțiune de către procesor.

Tipuri de procesoare 4, 8, 12, 16, 32, 64 biți; conceptul BIT SLICE – AMD 2900 și 29000.

**2. După arhitectura internă** – Von Neumann (Princeton) și Harvard.

Memoria unui sistem de calcul este alcătuită din două zone:

- ✓ Memoria de programe în care sunt stocate/memorate instrucțiunile care se execută în cadrul programului. Zona se mai numește și code memory; De obicei, zona memoriei de cod înseamnă o memorie de tip ROM.
- ✓ Memoria de date în care sunt stocați operanzi pe care procesorul îi utilizează în realizarea instrucțiunilor și respectiv, sunt depuse rezultate ca urmare a efectuării instrucțiunilor. Poate fi implementată doar în memorie de tip RAM.

Într-o arhitectură de tip Princeton cele două tipuri de memorii coexistă într-o memorie unitară accesată de către procesor printr-o magistrală unică de date. Acest lucru presupune execuția secvențială a instrucțiunilor:

- ✓ Se preia din memorie codul instrucțiunii de executat;
- ✓ Ulterior din aceeași memorie se preiau și operanzii supuși procesării;
- ✓ Depunerea rezultatului obținut ca urmare a procesării în aceeași memorie (RAM);

Într-o arhitectură Harvard cele două zone de memorii sunt distincte și conectate prin magistrale de date separate la procesor. La execuția unei instrucțiuni se preiau simultan din memoria de programe codul instrucțiunii de executat respectiv din memoria de date operanzii care vor fi procesați. La final rezultatul obținut este depus în memoria de date.

### **3. După numărul instrucțiunilor pe care le pot executa**

Microprocesoare CISC (Complex Instruction Set Computer) ex. x86 INTEL

Microprocesoare RISC (Reduced Instruction Set Computer) ex. ARM

#### **CARACTERISTICI ALE PROCESOARELOR CISC**

Setul de instrucțiuni mare;

Format variabil și durată de execuție variabilă pentru instrucțiuni (cele simple durată mică, cele complexe durată mai mare);

Arhitectură hardware complexă;

Există registre privilegiați în setul de registre.

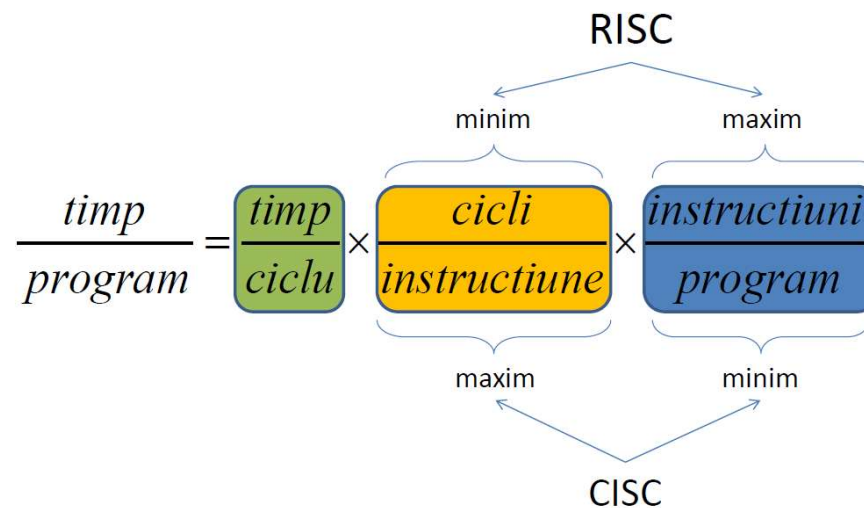
#### **CARACTERISTICI ALE PROCESOARELOR RISC**

Set de instrucțiuni redus;

Format fix și durată fixă de execuție a instrucțiunilor, indiferent de natura acestora;

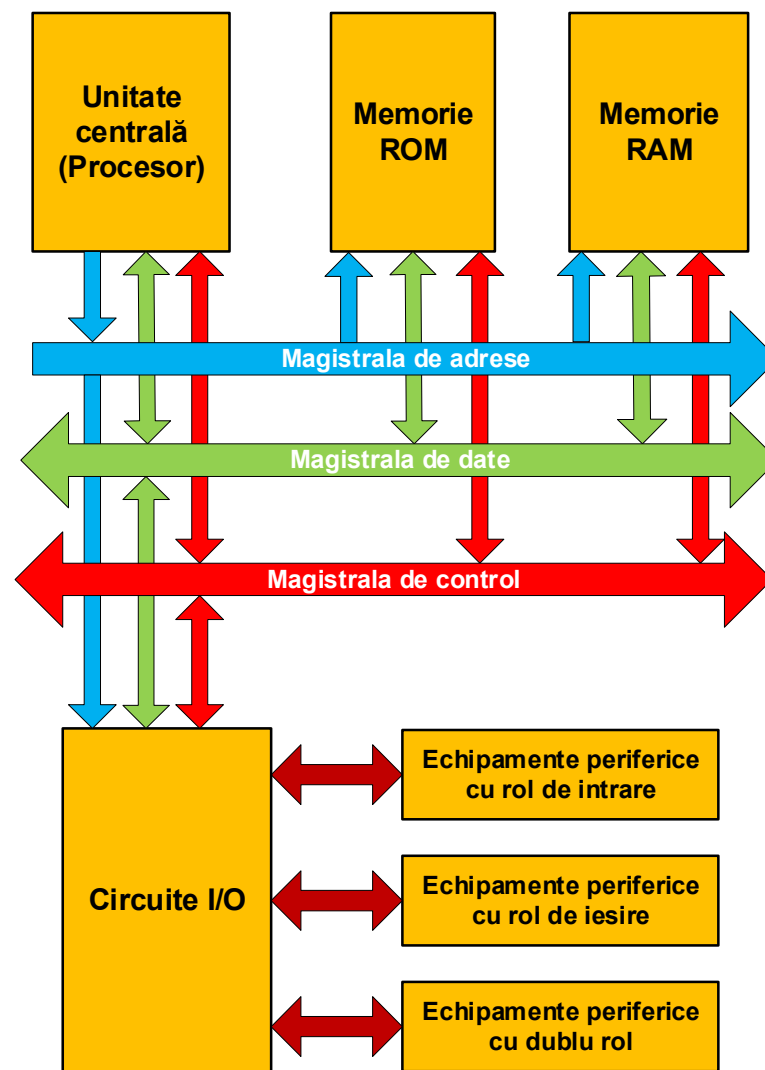
Arhitectură simplificată și viteză de operare ridicată;

Instrucțiunile se execută în mod similar indiferent de regiștrii utilizați din cadrul procesorului (setul de instrucțiuni este ortogonal în raport cu setul de regiștri ai procesorului). Nu există regiștri privilegiați.



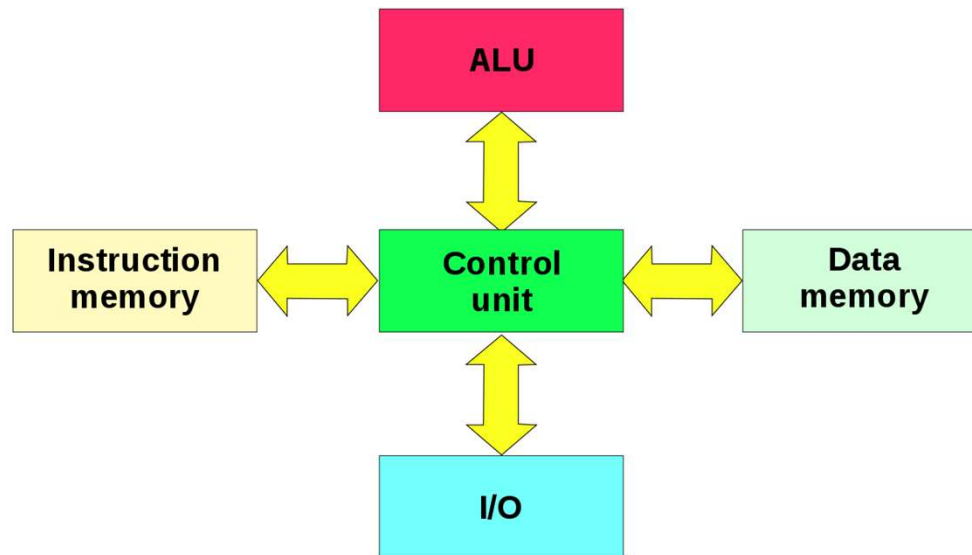
RISC	CISC
<ol style="list-style-type: none"> <li>1. Pune accent pe software</li> <li>2. Instrucțiuni reduse într-un singur ciclu de ceas</li> <li>3. Load &amp; Store ca instrucțiuni separate</li> <li>4. Cod cu multe instrucțiuni</li> <li>5. Complexitate redusă - număr mic de tranzistoare pe chip (lasă loc de periferice)</li> <li>6. Necesită un spațiu mărit de memorie pentru program și date</li> </ol>	<ol style="list-style-type: none"> <li>1. Accent pe hardware</li> <li>2. Instrucțiuni complexe în unul sau mai mulți cicli</li> <li>3. Load/Store încorporate în instrucțiunea complexă</li> <li>4. Cod de lungime redusă</li> <li>5. Complexitate mărită – număr mare de tranzistoare alocate executării instrucțiunilor complexe</li> <li>6. Nu are nevoie de foarte multă memorie</li> </ol>

## Structura unui sistem de calcul - arhitectura Von Neumann (Princeton)





## Structura unui sistem de calcul - arhitectura Harvard



#### **4. Dupa numarul de instructiuni executate in paralel**

Microprocesoare scalare

Microprocesoare superscalare; Tehnica Pipe-line; VLIW – Very Long Instruction Word Processor

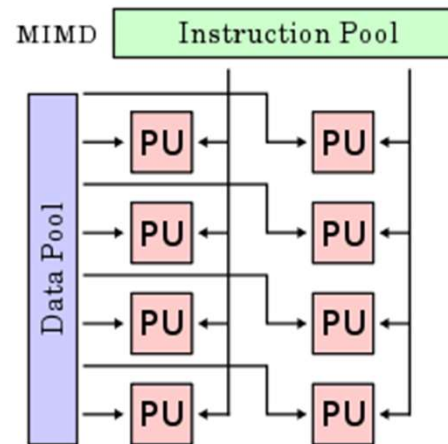
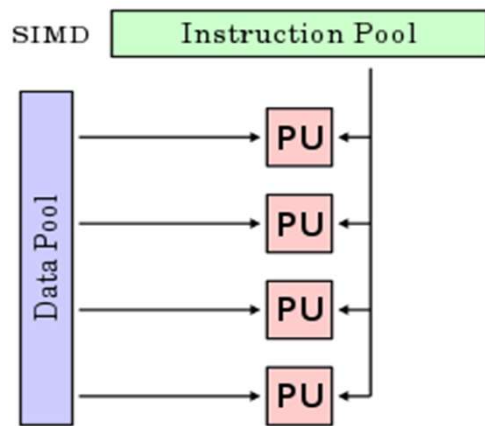
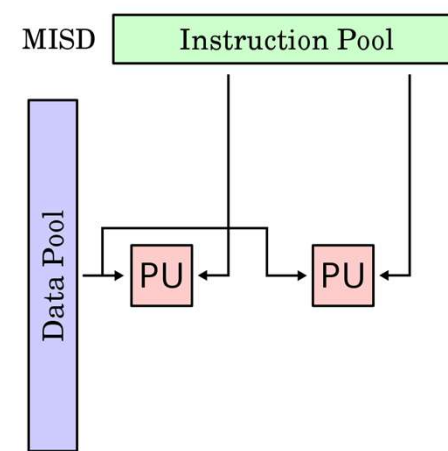
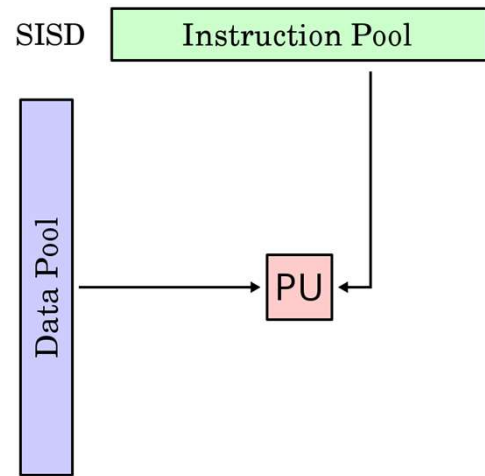
#### **5. Taxonomia (clasificarea) Flynn**

Single instruction stream, single data stream (SISD)

Single instruction stream, multiple data streams (SIMD) – procesoare vectoriale

Multiple instruction streams, single data stream (MISD) – procesoare cu scopuri speciale (flight control)

Multiple instruction streams, multiple data streams (MIMD) – sisteme distribuite, multicore superscalare



## **6. Dupa numarul de core-uri / procesoare separate continute pe chip**

Microprocesoare unice

Microprocesoare multicore – mai multe unitati de procesare separate

## **7. Dupa numarul de thread-uri / fire de executie**

Microprocesoare unithread

Microprocesoare multithread

## **8. Dupa destinatia de utilizare**

Uz general (General Purpose)

DSP – Digital Processing Unit

GPU – Graphic Processing Unit

Network Processor

Embedded System Processor (MCU – MicroController Unit); Sisteme încorporate

# Conventii folosite la notatia circuitelor din structurile digitale

1) Un grup de semnale cu aceiasi semnificatie, notate: A, B, C, ... are urmatoarea interpretare:

- Semnalul notat A reprezinta bcmps (bitul cel mai putin semnificativ – least significant bit - lsb)
- Ultimul semnal din notatia mentionata reprezinta bcms (bitul cel mai semnificativ – most significant bit - msb)
- Citirea numarului se face de la bcms catre bcmps!

Ex. A=1, B=0, C=0 – 100 – se citeste corect 001 adica 1!

2) Similar pentru grupuri de semnale notate  $A_0, A_1, A_2, \dots$  respectiv  $D_0, D_1, D_2, \dots$

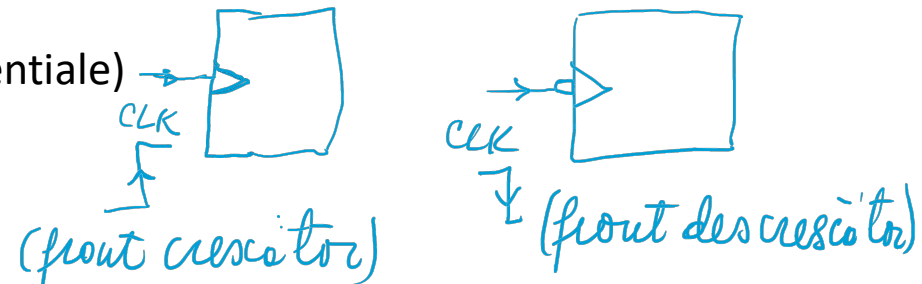
- Semnalul notat cu indexul 0 reprezinta bcmps, iar semnalul notat cu indexul cel mai mare indica bcms. Citirea are loc de la bcms catre bcmps.

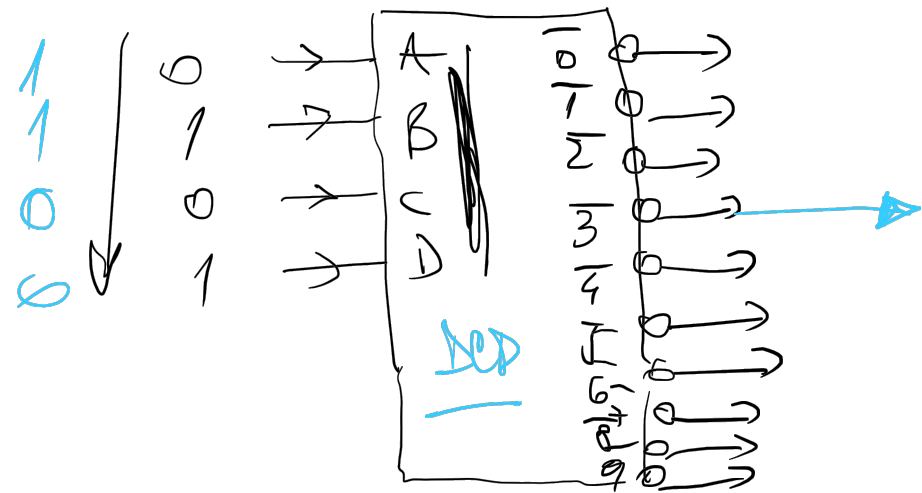
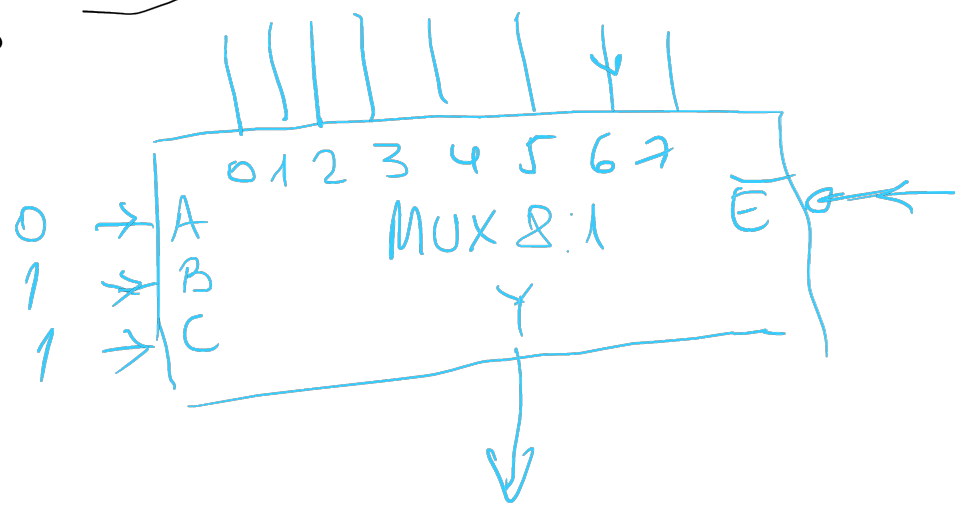
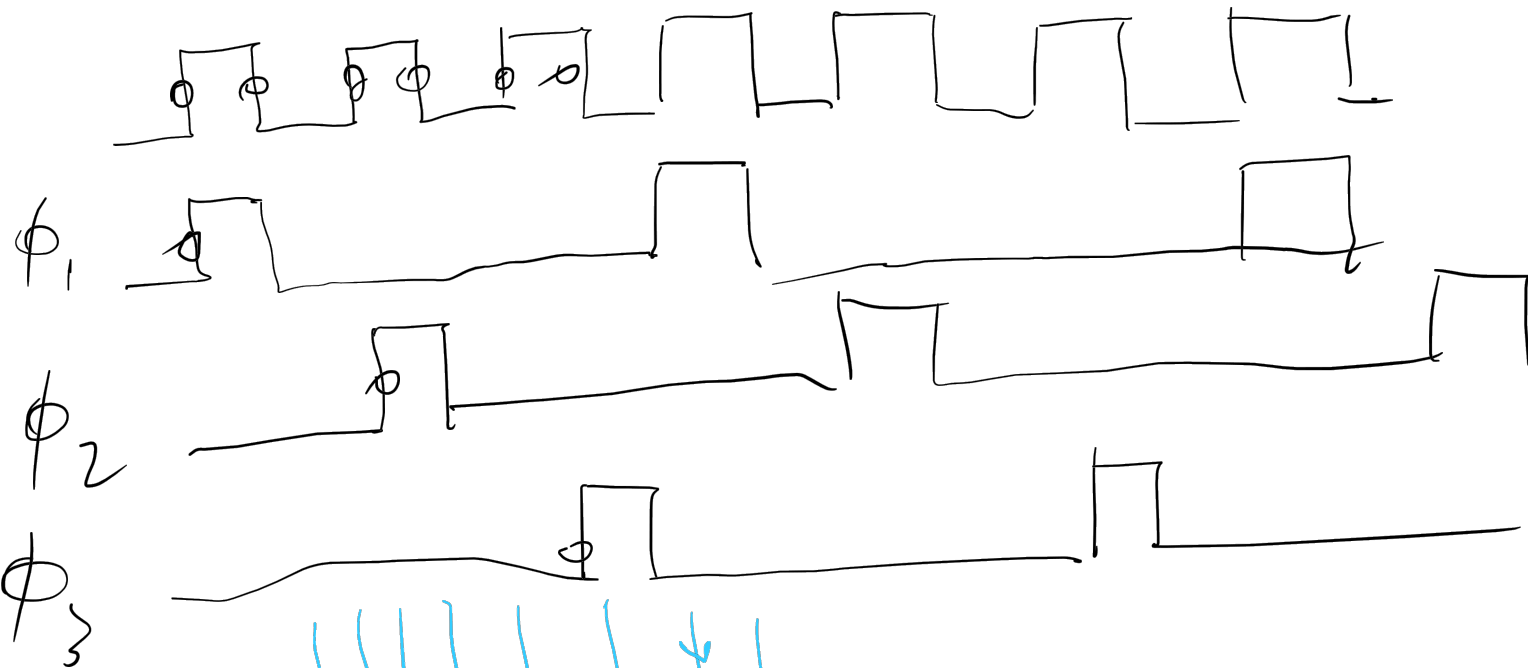
3) Semnale notate prin cifre 0, 1, 2, ... au interpretarea:

- Semnalul notat 0 indica bcmps, iar semnalul notat cu cifra cea mai mare reprezinta bcms.

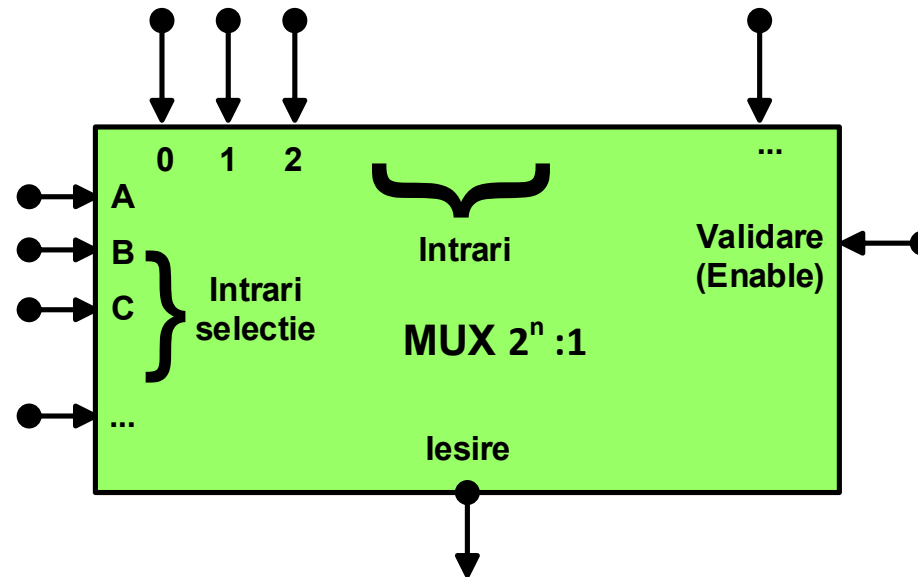
4) Un semnal notat cu bara de supraliniere (ex.  $\overline{EN}$ ) indica faptul ca respectivul semnal este activ 0-logic sau altfel spus, isi indeplineste functia logica in starea 0-logic (functia de validare a circuitului).

5) Notatia frontului activ la semnalele de ceas (circuite secventiale)





## Circuitul MUX



Mux-ul transfera o informatie digitala aplicata la una dintre intrarile circuitului catre unica iesire a acestuia, in conditiile activarii circuitului prin aplicarea unui semnal de validare pe intrarea ENABLE, respectiv prin selectia intrarii mentionate folosind o combinatie de semnale digitale aplicate pe intrarile specializate de selectie.

In situatia unui numar de  $n$  intrari de selectie se va putea selecta una dintre cele  $2^n$  intrari. Intrarile de selectie se noteaza cu A, B, C, ... unde A reprezinta bcms, iar litera cea mai departata de inceputul alfabetului reprezinta bcms.

Intrarile normale sunt notate 0, 1, 2, ...

De obicei intrarea ENABLE este activa 0-logic.

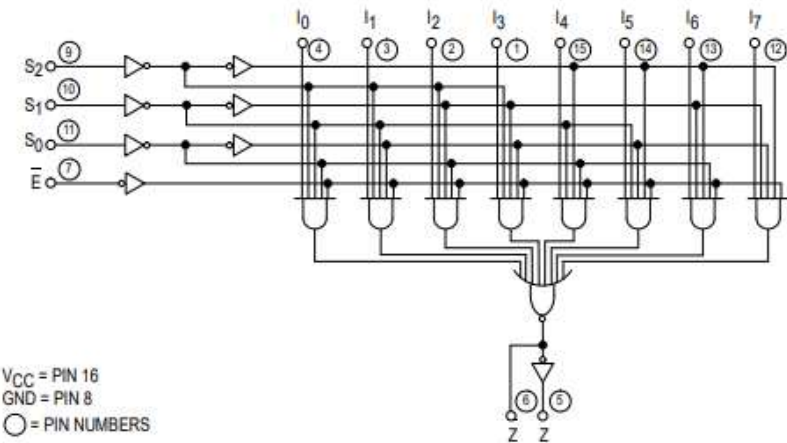
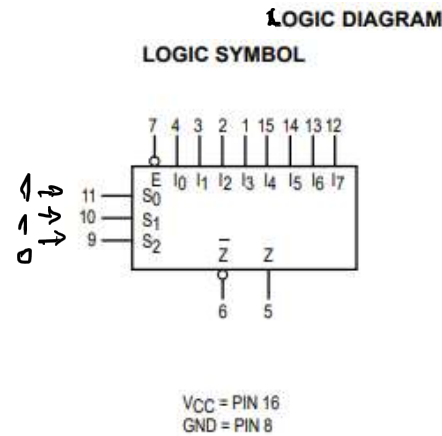
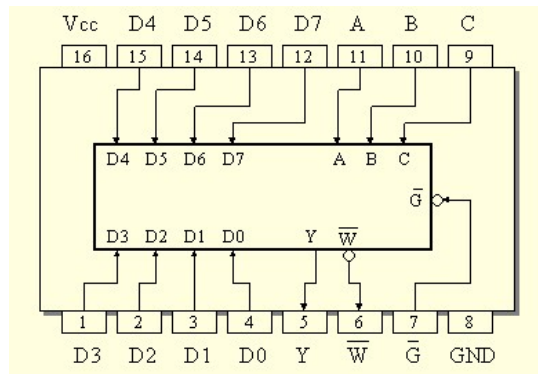
## Circuitul MUX

Pentru functionarea MUX-ului sunt valabile urmatoarele afirmatii:

- MUX-ul este un circuit digital care selecteaza un semnal digital aplicat la una din intrarile acestuia si il transfera catre unica iesire in conditiile in care circuitul este activ (ENABLE validat) si exista o combinatie de semnale digitale aplicate pe intrari specializate de selectie, care permite identificarea intrarii de la care are loc transferul mentionat.
- **MUX-ul este un selector** si notatia  $2^n:1$  reprezinta raportul de selectie.



## Circuitul MUX 74151 (de tip 8:1)



Circuitul 74151 are particularitatea dată de faptul că are 2 ieșiri, una dintre acestea fiind inversa celeilalte.

MUX-ul 8:1 semnifică faptul că se selectează una dintre cele 8 intrări și semnalul respectiv se transferă către unica ieșire a circuitului, iar numărul intrărilor de selecție este de 3.

MUX-ul 16:1 semnifică faptul că se selectează una dintre cele 16 intrări și semnalul respectiv se transferă către unica ieșire a circuitului, iar numărul intrărilor de selecție este de 4.

# Circuitul MUX 74153 (de tip 4:1)

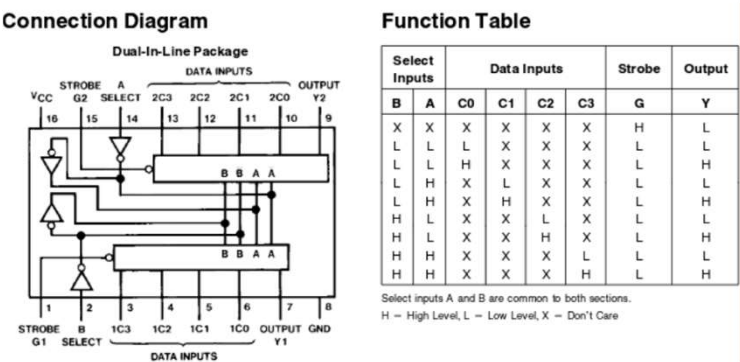


Figure 2. Function Table and Connection Diagram of 74153

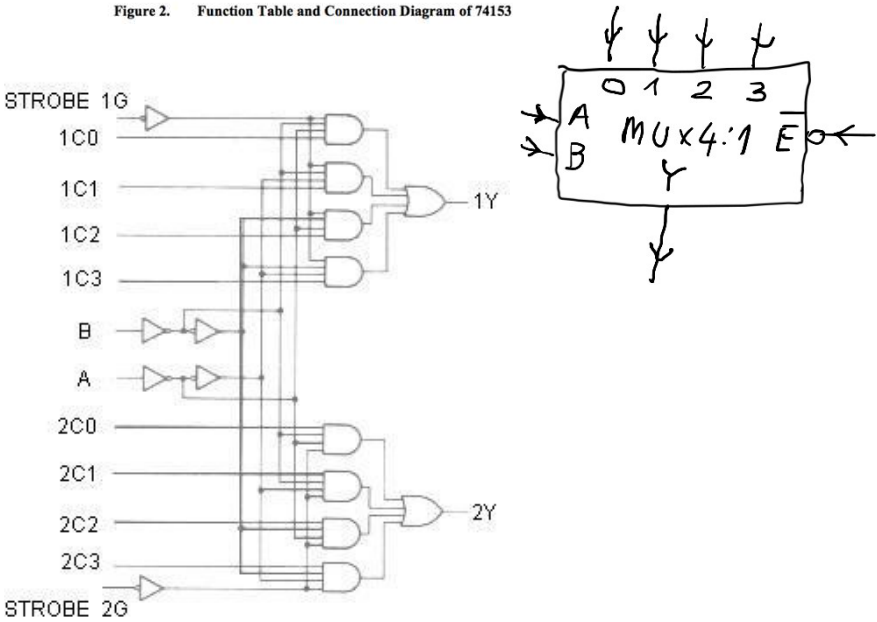
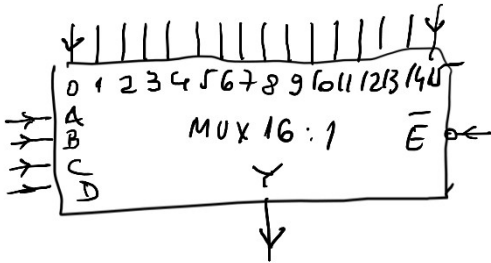
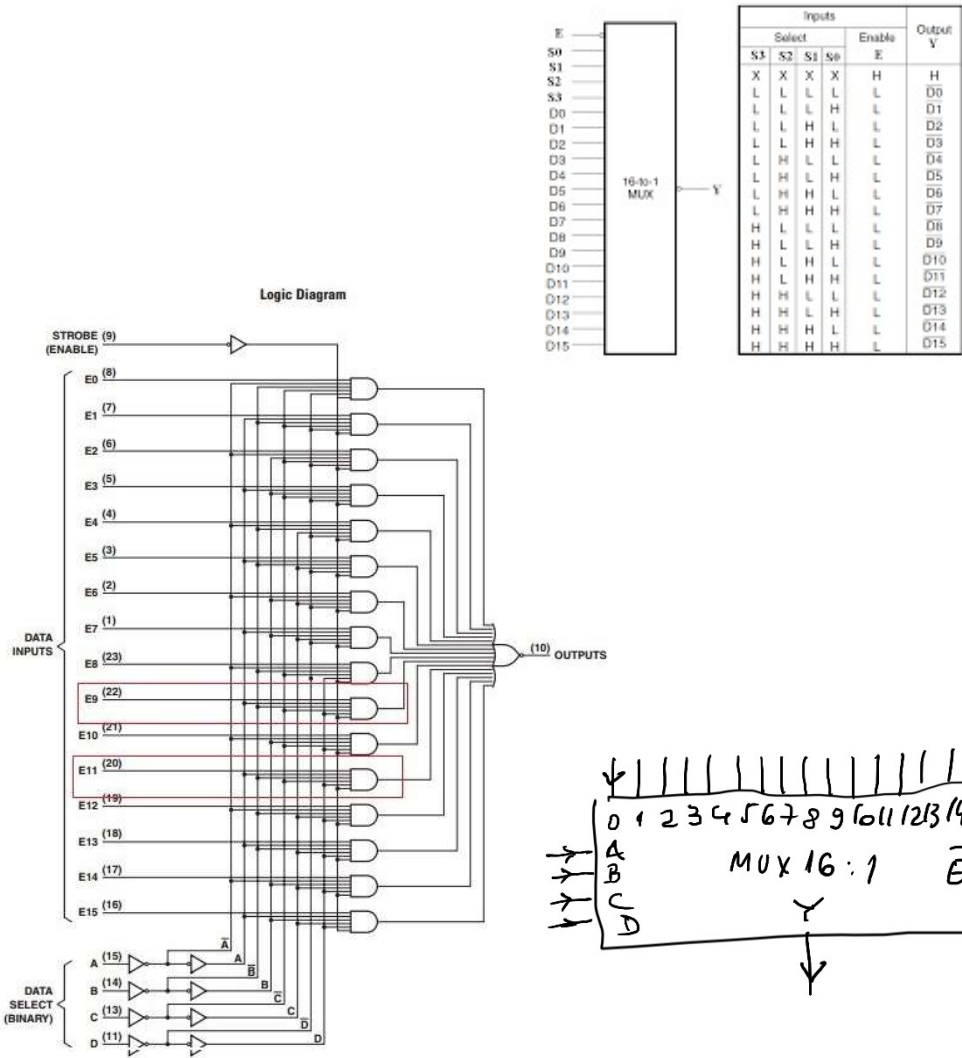
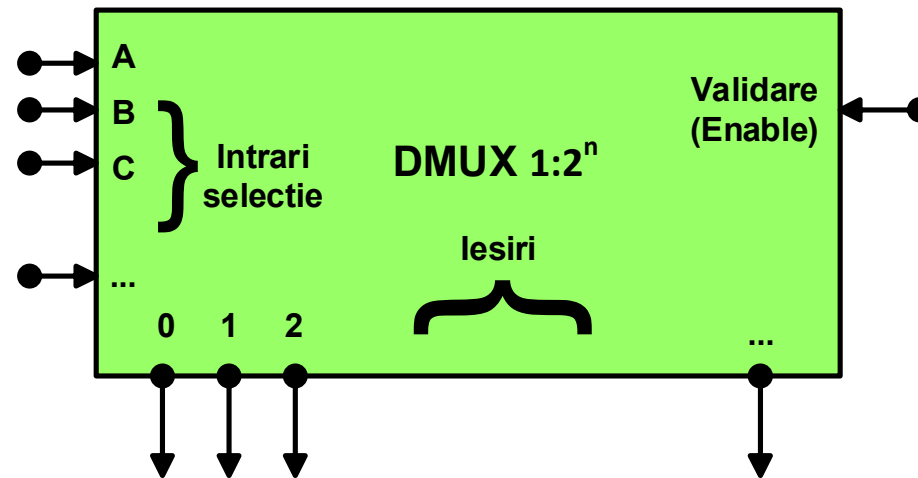


Fig. 32. - Schéma logique du circuit intégré 74153

# Circuitul MUX 74150 (de tip 16:1)



## Circuitul DMUX

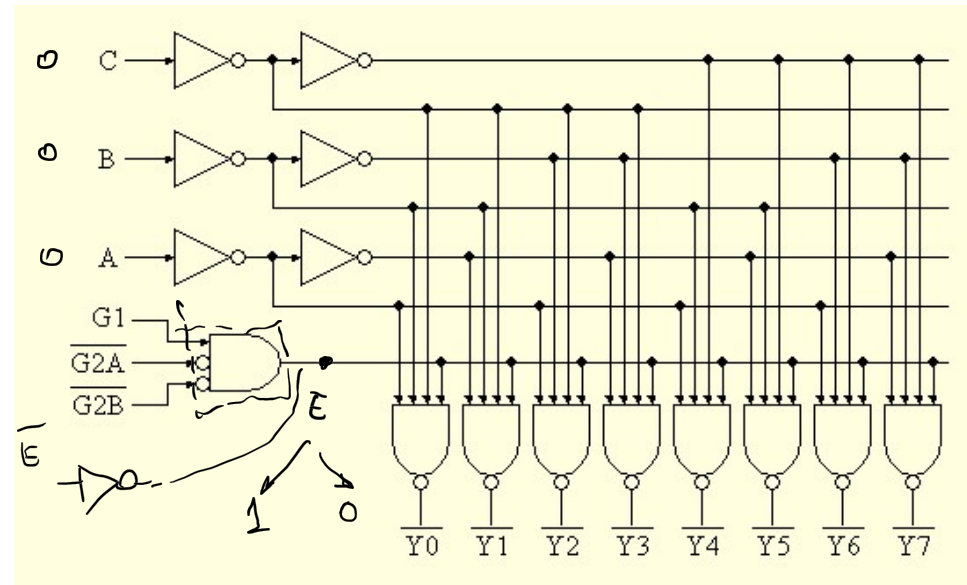
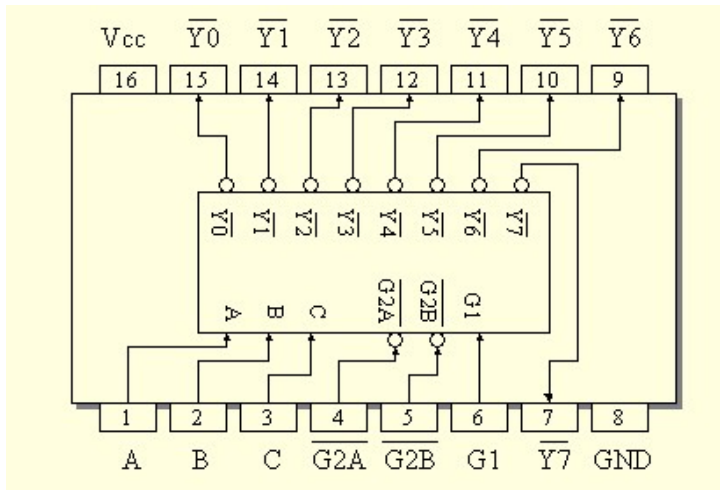


Pentru functionarea DMUX-ului sunt valabile urmatoarele afirmatii:

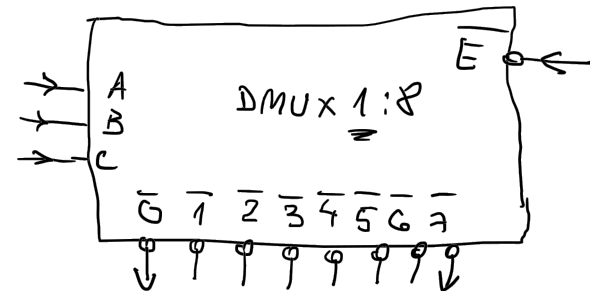
- DMUX-ul este un DCD prevazut cu intrare ENABLE; un DCD este un circuit digital care recunoaste un cod binar aplicat intrarilor prin activarea unei iesiri corespunzatoare;
- DMUX-ul este un circuit digital care repartizeaza un semnal digital aplicat unei intrari specializate, notata ENABLE, catre una dintre iesirile circuitului, selectata printr-o combinatie digitala aplicata unor intrari de selectie.

**DMUX-ul este un repartitor** si notatia  $1:2^n$  reprezinta raportul de repartitie.

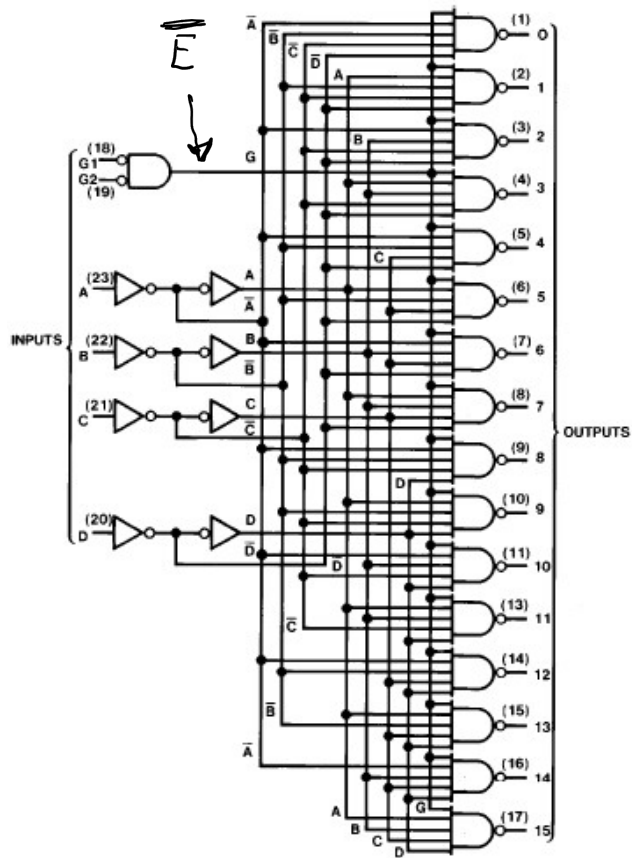
## DMUX 1:8 74138



Circuitul DMUX 74138 de tip 1:8 are particularitatea data de faptul ca are 3 intrari de validare (ENABLE sau notate  $G1$ ,  $\overline{G2A}$ ,  $\overline{G2B}$ ). Toate cele 3 intrari de validare trebuie sa fie active simultan pentru ca circuitul sa fie activ! La un moment dat cel mult o iesire este activa (in starea 0-logic).

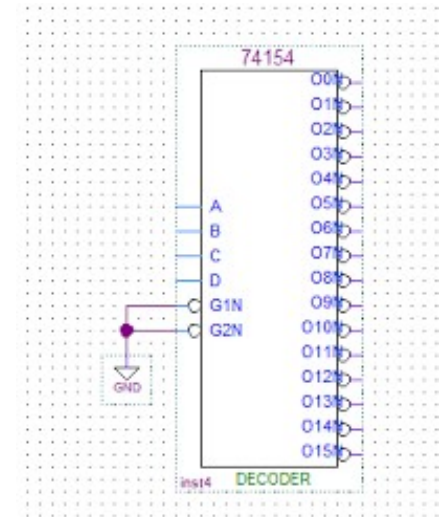


## DMUX 1:16 74154



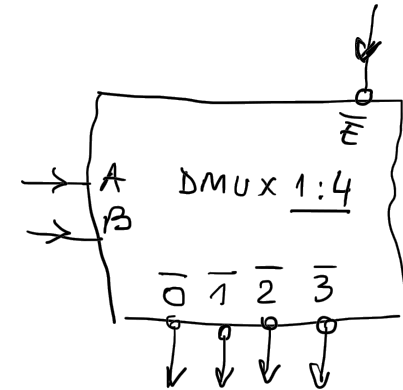
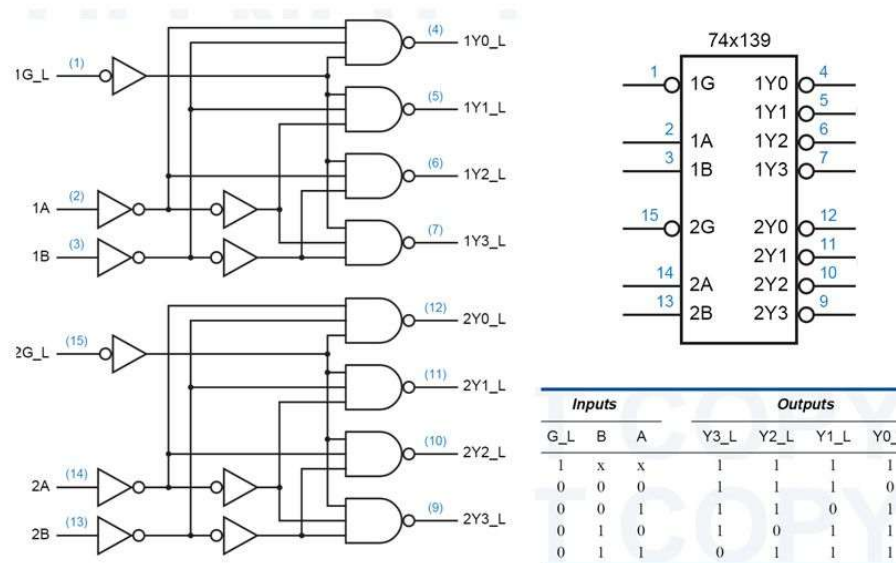
INPUTS					OUTPUTS																
G1	G2	D	C	B	A	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
L	L	L	L	L	L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	L	H	H	H	H	L	H	H	H	H	H	H	X	H	H	H	H	H	H
L	L	L	H	L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	L	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	L	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H
L	L	H	L	L	L	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H	H
L	L	H	L	L	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H	H	H
L	L	H	L	H	L	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H	H
L	L	H	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H	H	H
L	L	H	H	L	L	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H	H
L	L	H	H	H	L	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H	H
L	L	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	L	H
L	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	L	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H
H	H	X	X	X	X	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H	H

H = high level, L = low level, X = irrelevant



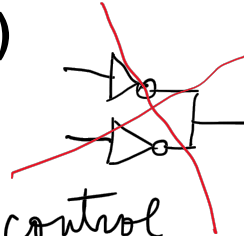
## DMUX 1:4 74139

### 74x139 dual 2-to-4 decoder

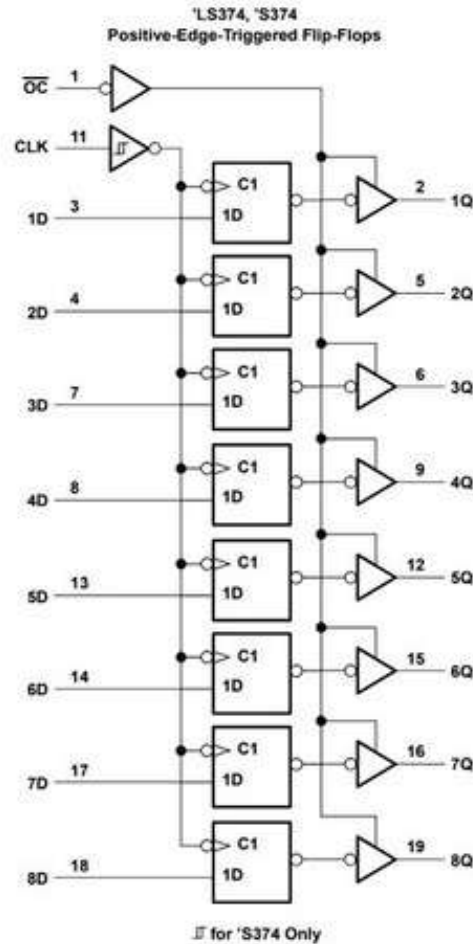
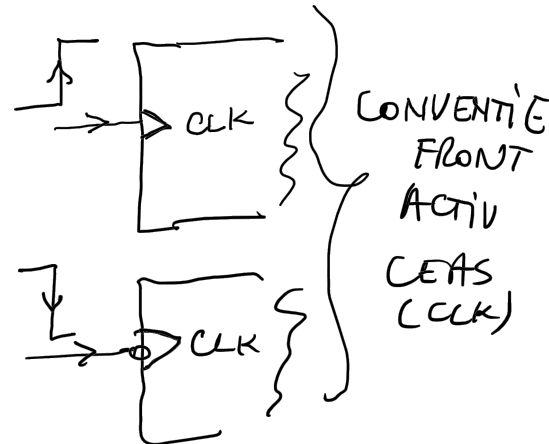
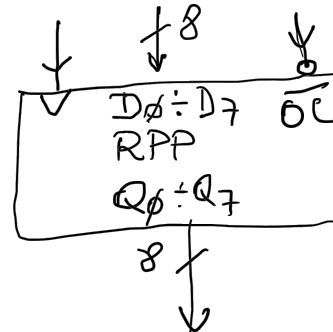


## Registru Paralel-Paralel octal (RPP)

74374  
74574



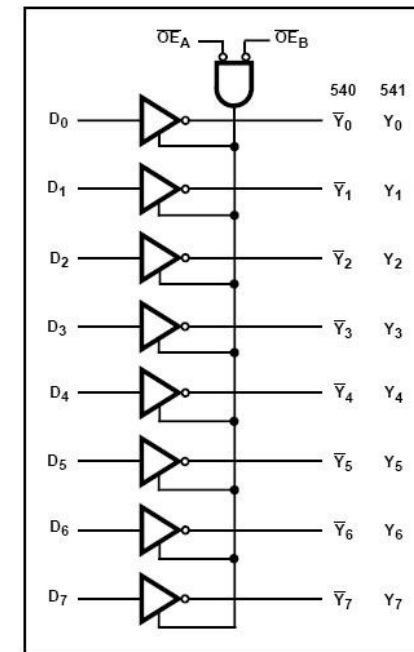
$\overline{OC}$  - output control



## Buffer octal

$\overline{OE}$  - output enable

74HCT541 Octal Buffer  
Logic Diagram



END 6 03 2025