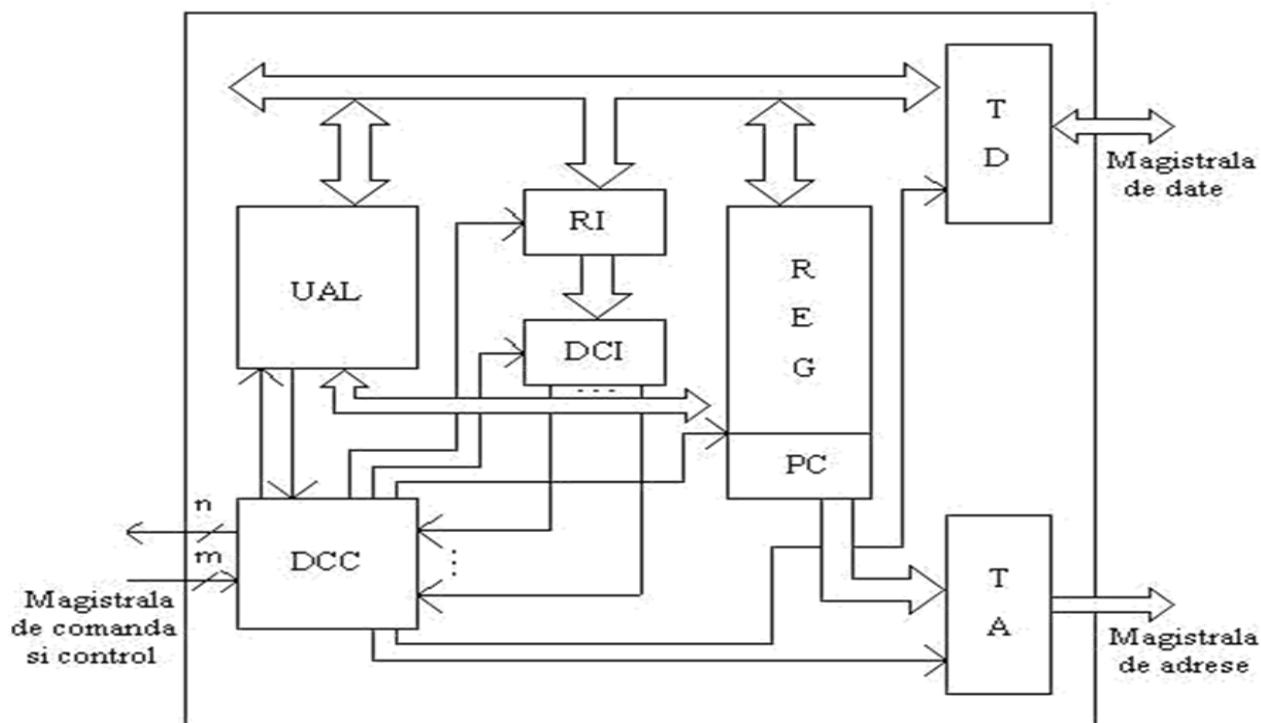


## Teorie PMD.

### 1. Definiti microsistemul digital.

Sistemele de calcul bazate pe microprocesoare sau microcontrolere se numesc micro sisteme digitale; există 2 tipuri de echipamente care sunt micro sisteme digitale: microcalculatoarele și sistemele de calcul orientate pe aplicații.

### 2. Prezentați și comentați structura clasică a unui microprocesor.



UAL- unitatea aritmetică și logică. Este un circuit logic combinațional capabil să realizeze operații aritmetice și logice (+, -, \*, /, și, sau, etc.) asupra operanzilor aplicați la intrări sub formă binară.

DCC- disp de comanda și control.

DCI- decodificator de instrucțiuni (comanda DCC)

RI- Registru de instrucțiuni (le memorează temporar)

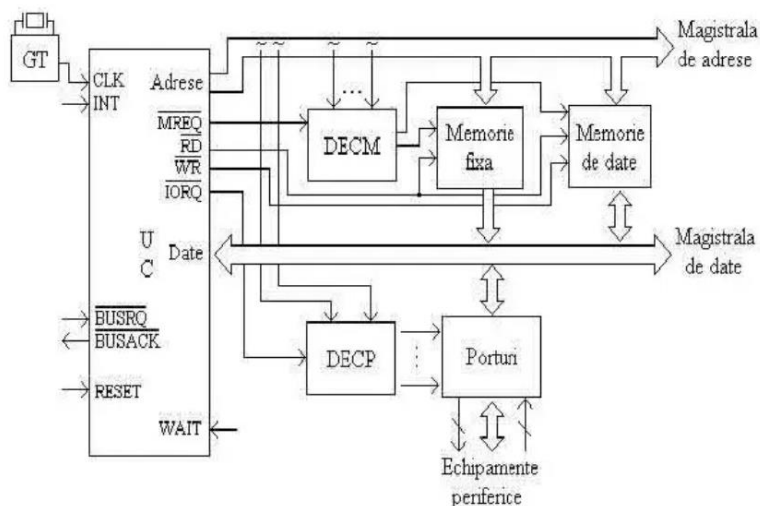
PC- numără adresele.

TD- tampon de date.

TA- tampon de adrese

REG- registre (capacitate mică) este utilizat pentru depozitarea temporară a datelor în CPU, unele registre fiind destinate doar pentru un anumit tip de date.

### 3. Prezentați și comentați schema bloc a unui microsistem digital



UC- up/uc+generator de tact, amplif/demux mag. unitatea de comanda care interpreteaza instructiunile din program, genereaza semnale de comandane necesare pentru executia acestora si supravegheaza executia.

Mem.Fixă-ROM, OTP, EPROM, EEPROM, Flash

Mem.Scr/cit-(S/D)RAM-memorie temporara.

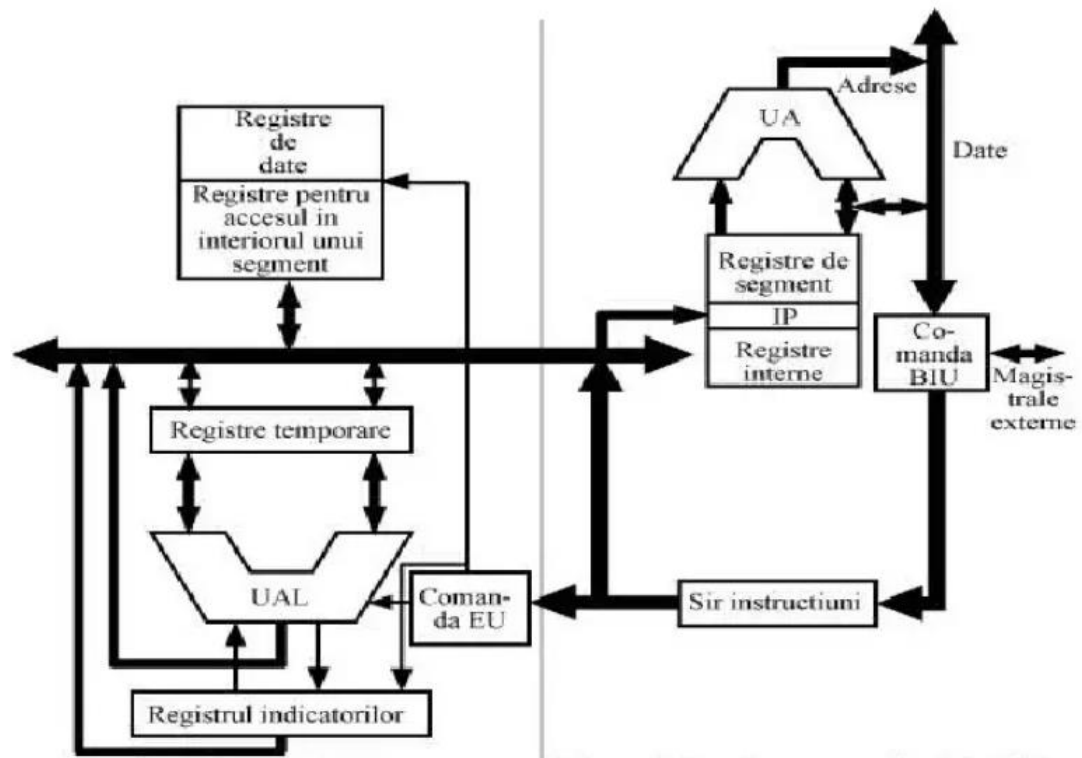
Porturi- Interfata dintre UC si I/O(input/output)

DECM- genereaza semnalele de selectie pentru memorie

DECP-decodifica porturile

4.

5. Prezentați structura internă a microprocesorului 8086. Care este rolul unitatilor interne.



EU(execution unit) executa toate instructiunile ,furnizeaza date si adrese pentru UIU.In timp ce EU isi indeplineste rolul ,UIU extrage noi instructiuni pe care le pune intr-o coada de asteptare (queue).

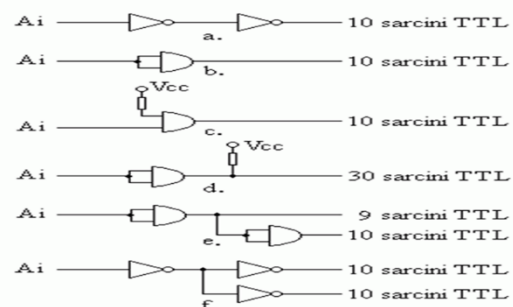
UIU(bus interface user) pregateste executia fiecarei instructiuni astfel : extrage o instructiune din memorie ,o depune in coada de instructiuni si caculeaza adresa din memorie a unui eventual operand.La terminarea executiei unei instructiuni EU are deja o noua instructiune in coada de asteptare construita de UIU.

EU continue ALU si unitate de comanda si control pe 16 biti,registrul indicatorilor de stare (flags),registrul operatorilor (temporari) si registri generali (de date si adrese).

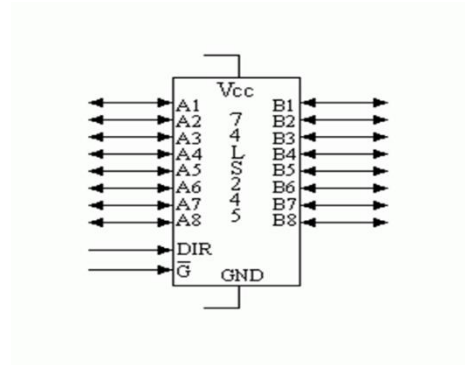
UIU continue IP (instruction pointer),registri segment,un bloc de control si interfatare al magistralei,un sir(coada) de intructiuni extrase de UIU si o unitate aritmetica proprie.

6. Prezentați solutii pentru amplificarea liniilor uni si bidirectionale ale unui microprocesor.

Amplificarea liniilor unidirectionale:

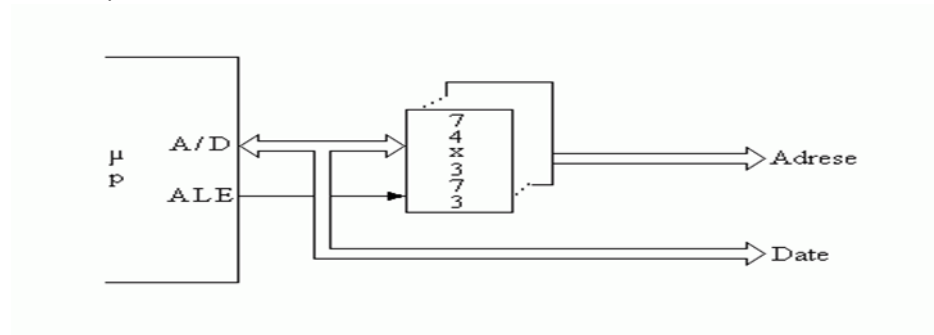


Amplificarea liniilor bidirecționale:

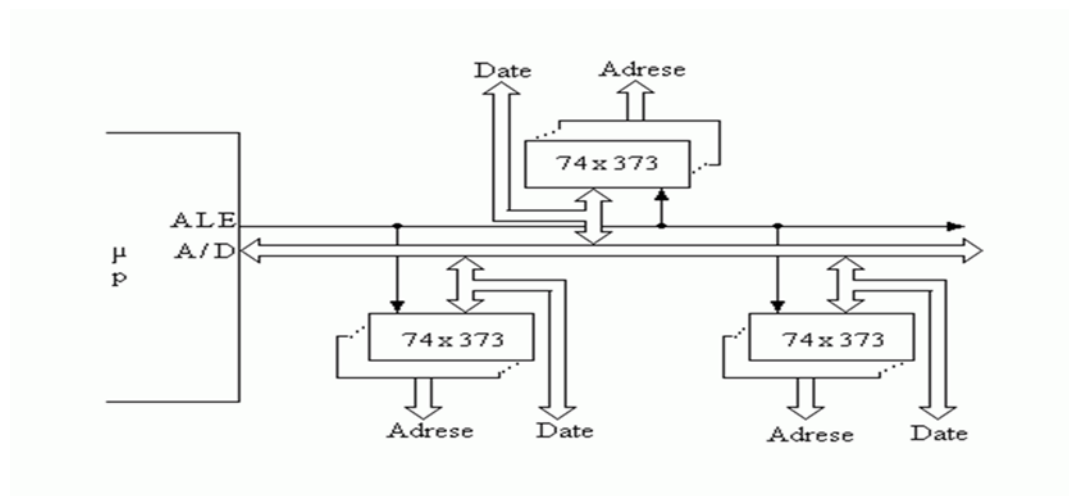


**7. Prezentați soluții principale pentru demultiplexarea liniilor multiplexate ale unui microprocesor. Avantaje și dezavantaje.**

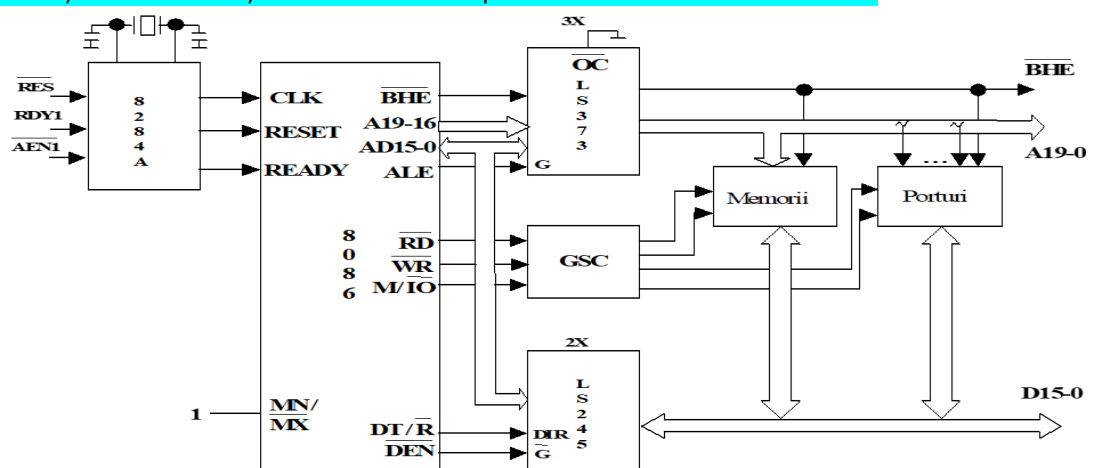
Demultiplexare centrala



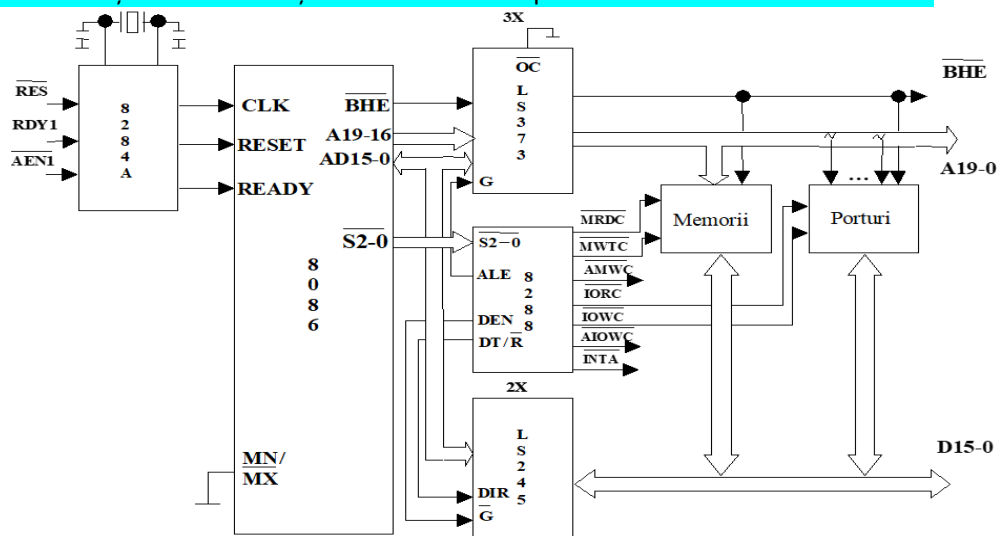
Demultiplexare locala



**8. Prezentați schema unității centrale cu microprocesorul 8086 în modul minim.**

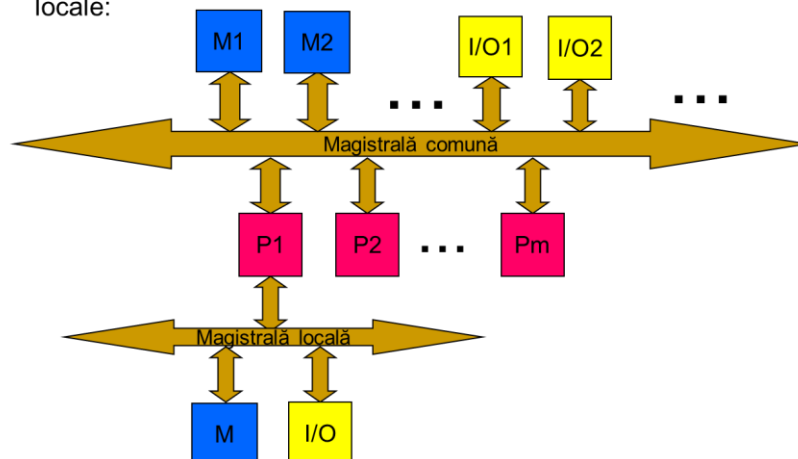


## 9. Prezentați schema unității centrale cu microprocesorul 8086 în modul maxim.



## 10. Prezentați schema clasică a unui sistem multiprocesor cu resurse partajate.

- Sistem multiprocesor cu resurse comune (partajate) și resurse locale:



## 11. Care este rolul unui circuit arbitru într-un sistem multiprocesor?

Acționează

- când arbitrul permite accesul procesorului la magistrala comună, el va activa linia /AEN; aceasta va permite accesul separatoarelor de pe liniile de adrese și date pe magistrala comună; semnalele de comandă vor fi generate, de către circuitul, în funcție de tipul de ciclu executat de procesor;
- când arbitrul nu permite accesul procesorului la magistrala comună, el va dezactiva semnalul /AEN; circuitul va trece toate semnalele de comandă a transferurilor cu resursele de pe magistrala comună în starea lor inactivă, separatoarele de pe liniile de date și adrese între magistralele procesorului și magistrala comună vor fi trecute în a 3-a stare permițând astfel accesul altor procesoare la magistrala comună.

## 12. Care este rolul și poziția unui decodificator de memorii într-un microsistem digital?

❑ rolul unui decodificator de memorii este să genereze semnale de selecție pentru circuitele de memorie din microsistem;

❑ poziția pe care o ocupă, în cadrul microsistemului, este între unitatea centrală și circuitele de memorie;

### 13 Descrieți decodificarea completă. Avantaje și dezavantaje.

- ❑ se construiește un tabel având ca și coloane rangurile de adrese și ca linii configurații care arată adresa de început și de sfârșit a fiecărei zone ocupate de câte un circuit;
- ❑ pentru fiecare circuit, se inspectează coloanele, pornind de la rangurile mai semnificative spre cele mai puțin semnificative și se stabilesc acele ranguri ce rămân nemodificate oricare ar fi locația adresată în respectiva zonă;
- ❑ o funcție combinațională a acestor intrări va individualiza zona respectivă și va constitui ecuația pentru semnalul de selecție a circuitului care o acoperă;
- ❑ exemplu: să se proiecteze decodicatorul pentru următoarea hartă a memoriei:
- ❑ 00000H – 1FFFFH – circuitul de memorie C1, având capacitatea de 64K x 16 biți;
- ❑ 40000H – 4FFFFH – circuitul de memorie C2, având capacitatea de 32K x 16 biți;
- ❑ 70000H – 7FFFFH – circuitul de memorie C3, având capacitatea de 32K x 16 biți.

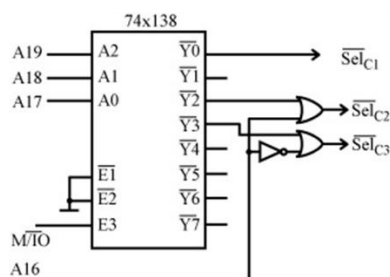
#### ■ Tabelul:

A 19	A 18	A 17	A 16	A 15	A 14	A 13	A 12	A 11	A 10	A9	A8	A7	A6	A5	A4	A3	A2	A1	C
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C 1
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C 2
0	1	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	
0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	C 3
0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	

#### ■ Ecuațiile semnalelor de selecție:

- ❑  $/SEL_{C1} = A19 + A18 + A17$
- ❑  $/SEL_{C2} = A19 + /A18 + A17 + A16$
- ❑  $/SEL_{C3} = A19 + /A18 + /A17 + /A16$

#### ■ Schema decodicatorului:



- ❑  $/SEL_{C1} = A19 + A18 + A17 + A16$
- ❑  $/SEL_{C2} = A19 + /A18 + A17 + A16$
- ❑  $/SEL_{C3} = /A19 + A18 + A17 + A16$
- ❑  $SEL_{C4} = /A19 + /A18 + A17 + A16$

- Ecuațiile semnalelor de selecție în cazul decodificării incomplete:

- ❑  $/SEL_{C1} = A19 + A18$
- ❑  $/SEL_{C2} = A19 + /A18$
- ❑  $/SEL_{C3} = /A19 + A18$
- ❑  $SEL_{C4} = /A19 + /A18$

- Fiecare circuit ocupă 4 zone de memorie, având capacități egale cu cele ale circuitului; de exemplu circuitul C1 ocupă zonele:

- ❑ 00000H – 0FFFFH,
- ❑ 10000H – 1FFFFH,
- ❑ 20000H – 2FFFFH și
- ❑ 30000H – 3FFFFH.

14. Descrieți decodificarea incompletă. Avantaje și dezavantaje.

- Decodificarea incompletă:

- ☐ se construiește tabelul ca în cazul anterior dar funcția sau funcțiile care vor genera semnalele de selecție nu vor utiliza toate liniile de adresă ce rămân nemodificate oricare ar fi locația adresată din zona respectivă;
- ☐ aceasta va duce la o simplificare a decodicatorului dar un același circuit, sau diviziune, va ocupa mai multe zone de memorie de aceeași capacitate ca cea a circuitului;
- ☐ dacă nu se consideră o linie de adresă un același circuit va ocupa 2 zone de memorie, dacă nu se consideră 2 linii de adresă un același circuit va ocupa 4 zone de memorie ș. a. m. d.
- ☐ exemplu: să se proiecteze decodicatorul pentru următoarea hartă a memoriei:
  - 00000H – 0FFFFH – circuitul de memorie C1, având capacitatea de 32K x 16 biți;
  - 40000H – 4FFFFH – circuitul de memorie C2, având capacitatea de 32K x 16 biți;
  - 80000H – 8FFFFH – circuitul de memorie C3, având capacitatea de 32K x 16 biți;
  - C0000H – CFFFFH – circuitul de memorie C4, având capacitatea de 32K x 16 biți.

 Tabelul:

[illegible]

- Ecuațiile semnalelor de selecție în cazul decodificării complete:
  - $/SEL_{C1} = A19 + A18 + A17 + A16$
  - $/SEL_{C2} = A19 + /A18 + A17 + A16$
  - $/SEL_{C3} = /A19 + A18 + A17 + A16$
  - $SEL_{C4} = /A19 + /A18 + A17 + A16$
- Ecuațiile semnalelor de selecție în cazul decodificării incomplete:
  - $/SEL_{C1} = A19 + A18$
  - $/SEL_{C2} = A19 + /A18$
  - $/SEL_{C3} = /A19 + A18$
  - $SEL_{C4} = /A19 + /A18$
- Fiecare circuit ocupă 4 zone de memorie, având capacități egale cu cele ale circuitului; de exemplu circuitul C1 ocupă zonele:
  - 00000H – 0FFFFH,
  - 10000H – 1FFFFH,
  - 20000H – 2FFFFH și
  - 30000H – 3FFFFH.

15. Care tip de memorie se conectează mai ușor la o unitate centrală cu microprocesor: fixă, SRAM sau DRAM? Dar mai greu? De ce?

#### 16. Ce se înțelege prin noțiunea de port și care este diferența față de noțiunea de interfață?

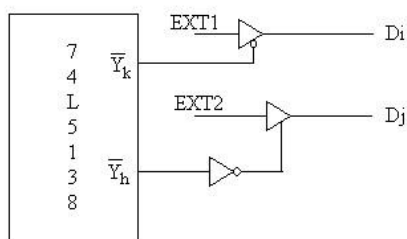
- Prin port de intrare/ ieșire sau, mai scurt, port, se înțelege un circuit, sau grup de circuite, care fac legătura între unitatea centrală a unui microsistem și dispozitivele și echipamentele periferice.
- Porturi + circuite de legătură între ele = interfață

\*\*

17. Se pot conecta porturi la un procesor care nu are sistem de intrare/ieșire? Dacă nu, de ce? Dacă da, cum?

18. Prezentați și comentați soluția pentru portul realizat cu porți. Pentru ce tip de port se recomandă? De ce?

- ☐ Este folosită pentru porturile de intrare;
- ☐ Ieșirile porților se conectează la linii de date ale microprocesorului, ca urmare este necesar să aibă facilitatea de a 3 – a stare;
- ☐ Semnalele care se citesc se conectează la intrările de date ale porților;
- ☐ Intrările de validare ale porților sunt comandate de una sau mai multe ieșiri dintr - un decodificator de porturi;
- ☐ Semnalele care se citesc sunt generate din exterior și se presupune că circuitul care le generează le și memorează până când este necesar; în caz contrar sunt prevăzute bistabile;
- ☐ Citirea se face în timpul ciclului de intrare.

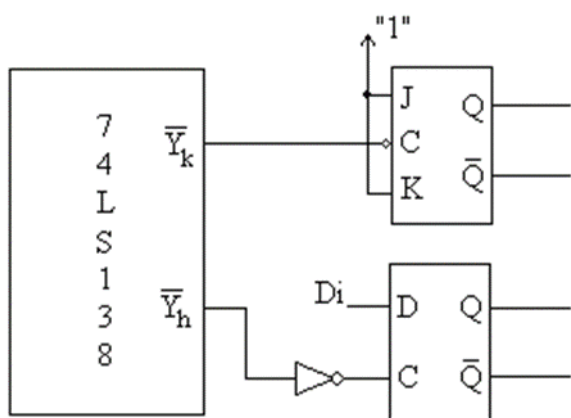


Schema trebuie să fie însoțită de o secvență de instrucțiuni; fie  $i = k = 0$  și  $j = h = 1$ , atunci secvența pentru citirea semnalului EXT1 este:

```
IN    AL,00H
RCR   AL,1
JNC   ZERO; s-a citit 0 logic
      ; s-a citit 1 logic
```

**19. Prezentați și comentați soluția pentru portul realizat cu bistabile. Pentru ce tip de port se recomandă? De ce?**

- ☐ Este folosită pentru porturile de ieșire;
- ☐ Există mai multe variante:
  - cu bistabile:
    - ☐ de tip JK;
    - ☐ de tip D;
  - cu bistabile care comută:
    - ☐ pe nivelul impulsului de tact;
    - ☐ pe frontul impulsului de tact:
      - anterior;
      - posterior.
- ☐ Schema de principiu:



■ Fie  $k = 0$ ,  $h = 1$  și  $i = 7$ :

- ☐ pentru modificarea ieșirii bistabilului de tip JK este suficientă execuția instrucțiunii OUT 00H,AL și bistabilul va trece în starea complementară celeia în care a fost;
- ☐ utilizarea acestei soluții are dezavantajul că trebuie cunoscută starea prezentă a bistabilului înainte de a lansa în execuție instrucțiunea care îi modifică starea;
- ☐ Ieșirea bistabilului de tip D poate fi comandată fără a fi necesară cunoașterea stării prezente întrucât are intrare de date;
- ☐ Următoarea secvență provoacă trecerea bistabilului în starea 1:



MOV AL,80H

OUT 20H,AL

- ☐ Următoarea secvență provoacă trecerea bistabilului în starea 0:

MOV AL,00H

OUT 20H,AL

- ☐ Modificarea stării bistabilelor se face în timpul ciclului de ieșire.

## 20. Ce înseamnă comanda unui semnal prin program?

### ■ Semnifică:

- ☐ Citirea unui semnal din exteriorul microsistemului: soluția este cea prezentată la citirea unui semnal prin intermediul unui port implementat cu o poartă:
  - Semnalul este legat la intrarea de date a unei porți cu 3 stări;
  - Intrarea de validare a porții este comandată de o ieșire a unui decodificator de porturi;
  - Ieșirea porții este legată la o linie a magistralei de date a unității centrale;
  - Citirea se face cu o instrucțiune IN și apoi se face ramificarea.
- ☐ Poziționarea unui semnal la 1 sau 0 logic: soluția este cea prezentată la implementarea unui port cu un bistabil:
  - Valoarea semnalului este transmisă la intrarea de date a unui bistabil prin intermediul unei linii a magistralei de date;
  - Tactul bistabilului este comandat de o ieșire a unui decodificator de porturi;
  - Ieșirea bistabilului este semnalul care trebuie comandat;
  - Poziționarea semnalului se face cu o instrucțiune OUT.

## 21. Definiți și comentați noțiunea de circuit specializat programabil.

### 5.1 Definiție și caracteristici

- Standardizarea soluțiilor pentru diversele tipuri de aplicații rezolvabile cu sisteme cu microprocesoare → circuite specializate pe o problemă.
- Cerințe:
  - ☐ Flexibilitate: pentru a acoperi toate modurile de lucru;
  - ☐ Programabilitate: pentru comunicare cu programatorul.
- Un circuit specializat este programabil în sensul că i se poate cere, prin intermediul unuia sau mai multor cuvinte de comandă, alegerea unor particularități ale tipului de aplicație pentru care circuitul a fost conceput.
- Cuvintele de comandă sunt transmise circuitului de unitatea centrală.

- Oferă informație de stare legată de ultimul transfer.

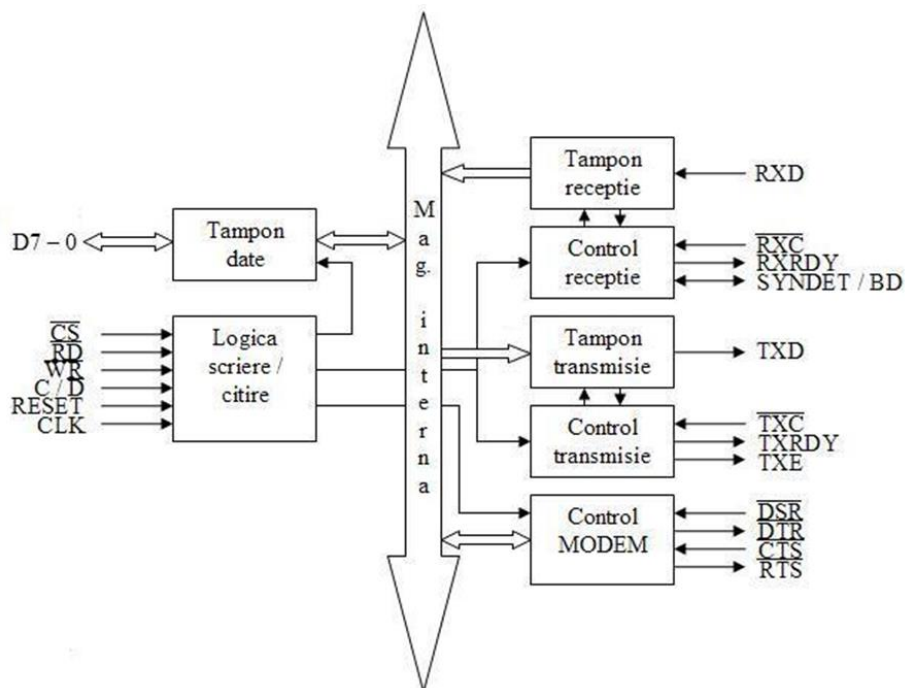
## 22. Prezentați și comentați caracteristicile de bază ale interfeței RS232 (fără controlul fluxului).

Nivelele de tensiune sunt EIA și au următoarele valori:  $-25V \div -3V$  pentru "1" logic și  $+3V \div +25V$  pentru « 0 » logic. Este necesar folosirea de circuite de conversie din TTL în EIA și EIA în TTL : 232. Direcția de deplasare a datelor se face simplex, adică unidirecțional, semi-duplex sau half duplex (bidirecțional dar la momente diferite de timp) și duplex sau full duplex (bidirecțional ce pot avea loc simultan). Viteza este mică aceasta fiind măsurată în bit pe secundă - bps și baud dacă transferul se face digital. Transferul serial de tip RS232 este util în momentul în care acesta se face pe distanțe mai mari, peste 3 m, între componentele care comunică. Motivele pentru care se recomandă este costul și rezistența la perturbatii.

## 23. Ce este circuitul 8251? Prezentați caracteristicile sale.

Circuitul specializat programabil 8251 este specializat în transferurile seriale, face parte din categoria de tipul USART, iar modul de lucru este sincron și asincron. Acesta poate primi un octet în paralel de la o unitate centrală, să-l serializeze și apoi să îl trimită la un echipament serial. Poate să preia de pe linie, de la un echipament periferic serial un octet să îl assembleze și să îl predea în paralel unității centrale. Are transmisie și recepție cu dublu tampon.

## 24. Prezentați și comentați structura internă a circuitului 8251.



□ C//D este intrarea prin care se alege unul din cele 2 porturi interne ale circuitului: portul de comandă/stări, ptr. C = 1 și portul de date ptr. C = 0; aici se leagă o linie din magistrala de adrese (de obicei A1 sau A2);

Pentru creșterea vitezei, a fost implementat conceptul de suprapunere.

- Tamponul recepție este alcătuit din 2 registre, RA și RB;

- Datele sunt preluate de pe linia serială în registrul RB, se elimină informația de cadrare, și, după asamblarea unui caracter, acesta este transferat în registrul RA și, simultan, este activat terminalul RXRDY și rangul cu același nume din registrul de stare.
- Prin acest semnal, circuitul anunță microprocesorul că are un caracter asamblat pe care poate să i – l ofere.
- Microprocesorul va trebui să preia conținutul registrului RA în timpul cât se assemblează un nou caracter în registrul RB.
- Dacă nu este îndeplinită această cerință, vechiul caracter se pierde, întrucât în RB se va transfera noul caracter și circuitul va detecta eroare de ritm și va poziționa un rang din registrul de stare, anunțând astfel exteriorul că a apărut această eroare.
- În acest fel, s – a realizat suprapunerea între timpul necesar preluării și asamblării unui caracter de pe linia serială cu timpul necesar citirii caracterului asamblat de către microprocesor.
- Semnalul RXRDY este dezactivat doar de un semnal activ pe linia /RD, ceea ce corespunde unei citiri din partea microprocesorului.

#### Transmisia:

- Suprapunerea;
- Tamponul transmisie este alcătuit din 2 registre, TA și TB.
- Unitatea centrală încarcă un caracter în registrul TA, circuitul îl transferă în registrul TB, de unde caracterul este serializat și transmis pe linie.
- După transferul conținutului registrului TA în TB, circuitul activează semnalul TXRDY comunicând microprocesorului că poate prelua un nou caracter.
- Simultan este activat și rangul cu același nume din registrul de stare.
- Dezactivarea terminalului TXRDY se face de primul front al semnalului /WR care corespunde operației de scriere a unui nou caracter de către microprocesor.

- Terminalul TXRDY este activat la îndeplinirea condițiilor descrise mai sus doar dacă transmisia poate avea loc, adică rangul TXE din cuvântul de comandă este 1 și terminalul /CTS = 0;
- De remarcat că rangul TXRDY din registrul de stare este activat indiferent de starea rangului TXE și a terminalului /CTS.

25. Cum se programează circuitul 8251 (nu este necesară structura cuvintelor ci doar ce anume i se comunică circuitului)?

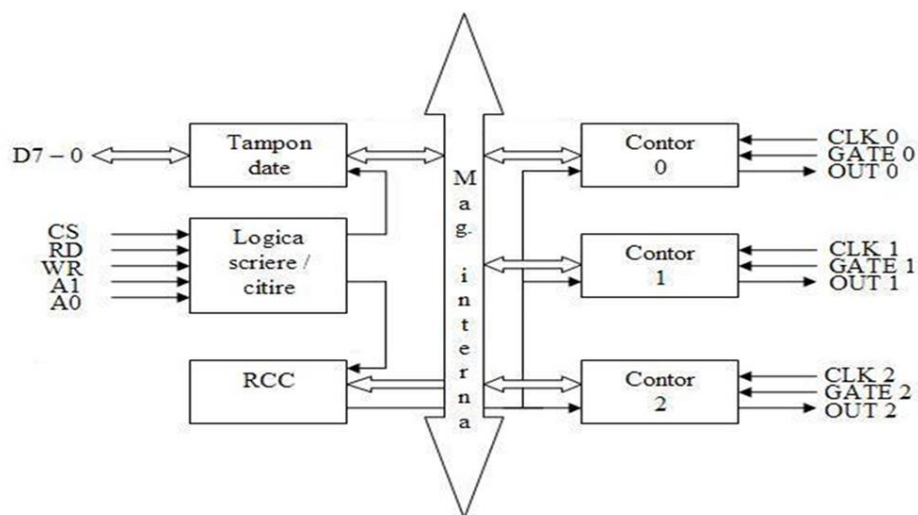
☐ Circuitul comunică unității centrale când are un caracter gata pentru ea sau când a terminat de transmis un octet și poate prelua altul; poate comunica:

- ☐ Prin program;
- ☐ Prin întreruperi;

26. Pentru ce clasă de aplicații se folosește circuitul 8253?

- ☐ generarea de întârzieri;
- ☐ generarea de impulsuri cu frecvență programabilă;
- ☐ generarea de impulsuri singulare;
- ☐ ceas de timp real;
- ☐ monostabil numeric;
- ☐ numărarea de evenimente externe.

27. Prezentați și comentați structura internă a circuitului 8253.



- ☐ Tamponul de date asigură legătura între circuit și magistrala de date a sistemului; transferă cuvinte de comandă, constante de divizare precum și conținutul contoarelor, în cazul unei operații de citire a lor.

☐ Logica scriere/ citire conduce întregul transfer de date din interiorul circuitului; are ca intrări semnale de comandă din partea microprocesorului care îi comunică ce fel de operații se desfășoară.

☐ RCC, Registrul de comandă și control, va prelua cuvintele de comandă transmise de către microprocesor care arată contoarele implicate în operațiile curente și modurile de lucru ale acestora; nu poate fi citit.

☐ Contoarele 0, 1 și 2 sunt identice și independente în funcționare; fiecare este un numărător pe 16 ranguri, cu preîncărcare, care numără în jos, având ca intrare de tact linia CLK<sub>i</sub>; în faza de programare, contorul este preîncărcat cu o constantă iar cuvântul de comandă care îi corespunde îi va fixa modul de lucru; fiecare contor poate lucra în unul din 6 moduri; dacă se numără evenimente, impulsurile care le descriu vor fi preluate de contor la intrarea CLK proprie iar dacă se dorește generarea de întârzieri atunci la intrarea CLK a unui contor se va lega tactul sistemului; conținutul contoarelor poate fi citit de microprocesor; conținutul contoarelor poate fi citit.

## 28. Cum se programează circuitul 8253 (nu este necesară structura cuvintelor ci doar ce anume i se comunică circuitului)?

Fiecare contor al circuitului va fi programat individual, fiind privit, de către microprocesor, ca un port de intrare/ ieșire; ordinea poate fi oricare;

Fiecare contor va fi programat cu un cuvânt de comandă și cu o constantă de preîncărcare, pe 1 sau 2 octeți.

## 29. Descrieți un mod de lucru al unui contor al circuitului 8253.

☐ Modul 0 sau generarea unei cereri de întrerupere la sfârșitul numărării:

☐ După programarea contorului, ieșirea acestuia trece la nivel 0 logic;

☐ Se menține la această valoare până când conținutul contorului va ajunge la valoarea 0, în urma numărării impulsurilor sosite pe linia CLK, când va trece la 1 logic;

☐ Rămâne în această stare până la o nouă încărcare;

☐ Numărarea poate fi inhibată prin activarea intrării GATE;

☐ Fig. prezintă modificarea ieșirii unui contor programat să lucreze în modul 0:

☐ Cifra din paranteză indică valoarea constantei;

☐ Decrementarea începe după încărcarea constantei și după trecerea unui impuls la intrarea de tact.

## 30. Câte linii sunt necesare pentru un transfer paralel? Justificați.

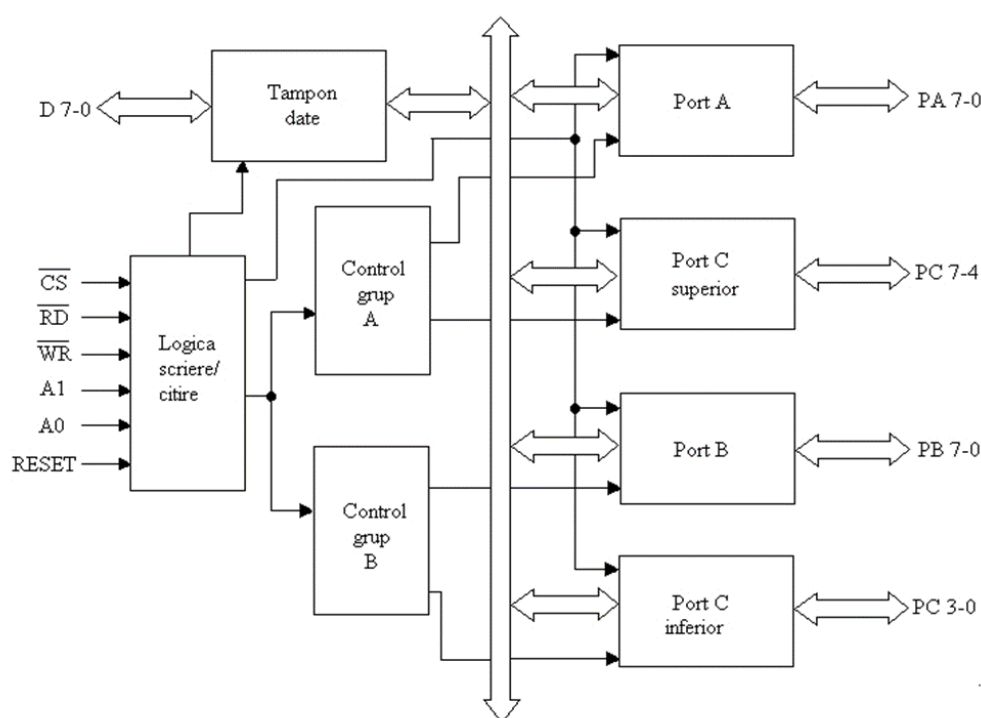
Dacă ne referim la 2 sisteme atunci e nevoie de n linii de date (și anume câți biți transmitem) și o linie de control pentru validare dacă s-au transmis respective recepționat datele.

31. Ce este circuitul 8255? Prezentați caracteristicile sale.

Circuit integrat realizat în tehnologie NMOS, 40 de terminale, cu o singură tensiune de alimentare, +5 V. Toate intrările și ieșirile sunt compatibile TTL.

- ☐ Dispune de 24 linii de intrare/ ieșire care pot fi configurate în mai multe feluri în funcție de modul de lucru ales:
- ☐ 2 grupe de câte 12 linii de intrare sau ieșire, fără semnale de dialog;
- ☐ 2 grupe de câte 8 linii de intrare sau ieșire, cu semnale de dialog sau
- ☐ o grupă de 8 linii bidirecționale, cu semnale de dialog.
- ☐ Din cele 24 linii de intrare/ ieșire 16 au posibilități de memorare iar 8 nu au această facilitate.
- ☐ Comunicarea cu circuitul 8255 se face prin intermediul a 4 adrese de port, corespunzătoare porturilor A, B, C și portului pentru cuvântul de comandă.

### 32. Prezentări și comentarii structura internă a circuitului 8255.



- ☐ Tamponul date asigură legătura între circuit și magistrala de date a sistemului: transferă cuvinte de comandă, de stare și date.
- ☐ Logica scriere/ citire are rolul să conducă întregul transfer de date din interiorul circuitului:
- ☐ Are ca intrări semnale de comandă din partea microprocesorului care îi comunică ce fel de operații se desfășoară;
- ☐ Acest bloc comandă cele 2 jumătăți ale circuitului: grupul A și grupul B;
- ☐ Blocurile de control pentru grupele A și B primesc comenzi de la blocul Logică scriere/ citire, primesc cuvintele de comandă și generează comenzile pentru porturile corespunzătoare;
- ☐ Porturile A, B și C au câte 8 biți:
- ☐ Portul A poate fi ieșire, în modurile 0, 1 sau 2, cu posibilități de memorare sau intrare, în modurile 0, 1 sau 2, cu posibilități de memorare doar în modul 1 sau 2;

☐ Portul B poate fi ieșire, în modurile 0 sau 1, cu posibilități de memorare sau intrare, în modurile 0 sau 1, cu posibilități de memorare doar în modul 1;

☐ Portul C are funcțiuni multiple: poate fi ieșire, doar în modul 0, cu posibilități de memorare sau intrare, doar în modul 0, fără posibilități de memorare; dacă porturile A și B lucrează în modurile 1 sau 2, atunci ranguri din portul C vor fi folosite pentru semnale de dialog și întregul port C poate fi folosit pentru a obține informație de stare de la circuit;

### 33. Cum se programează circuitul 8255 (nu este necesară structura cuvintelor ci doar ce anume i se comunică circuitului)?

☐ Se realizează prin transmiterea unui cuvânt de comandă la adresa portului cuvântului de comandă; se poate cere porturilor să lucreze în unul din următoarele moduri:

☐ modul 0, numit și mod intrare/ ieșire, pentru porturile A, B și C,

☐ modul 1, numit și mod intrare/ ieșire cu dialog, pentru porturile A și B și

☐ modul 2, numit și mod bidirecțional cu dialog, doar pentru portul A.

☐ Există și posibilitatea ca, printr – o singură instrucțiune de ieșire, microprocesorul să comande ieșiri individuale ale portului C; se face printr – un cuvânt de comandă în care rangul 7 este la valoarea 0 logic; particularitatea este cunoscută sub numele mod “bit set/ reset” și este utilă în aplicații de control.

### 34. Descrieți unul din modurile de lucru cu dialog ale circuitului 8255.

☐ Modul 1:

☐ Se mai numește și mod de intrare/ ieșire cu dialog și se folosește pentru operații de intrare/ ieșire cu dialog. Caracteristicile modului sunt următoarele:

☐ se pot comanda independent grupul A și grupul B;

☐ grupul A conține portul A și 3 ranguri ale portului C iar grupul B conține portul B și alte 3 ranguri ale portului C;

☐ rangurile nefolosite ale portului C pot fi comandate prin modul bit set/ reset;

☐ porturile A și B pot fi declarate ca intrări sau ieșiri având și posibilități de memorare;

☐ portul C are și rol de registru de stare.

### 35. Care sunt posibilitățile de conectare a porturilor la o UC cu microprocesor?

Posibilități de conectare a porturilor la UC:

☐ Intrare/ ieșire programabilă (PIO): transferul se desfășoară prin intermediul UC; folosește intensiv timpul UC;

☐ Întreruperi: procesorul este întrerupt la transferul fiecărui cuvânt (octet); multe întreruperi care nu întotdeauna pot fi acceptate; rată de transfer limitată

☐ Accesul direct la memorie (DMA): transfer direct memorie – periferic sau periferic – periferic sau memorie – memorie (neutilizat pentru că este mai rapid prin UC); nu implică UC dar cere hardware suplimentar conectat pe magistrale;

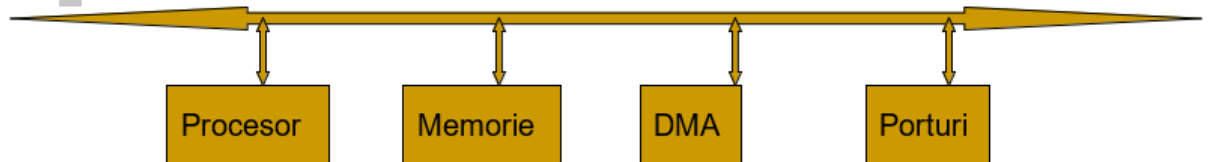
☐☐Controler DMA: 8237A

### 36. Prezențați 3 configurații DMA. Caracteristici

Configurații DMA:

- Magistrală comună, controler DMA separat,
- Magistrală comună, controler DMA integrat,
- Magistrală de intrare/ ieșire distinctă.

Magistrală comună, controler DMA separat:

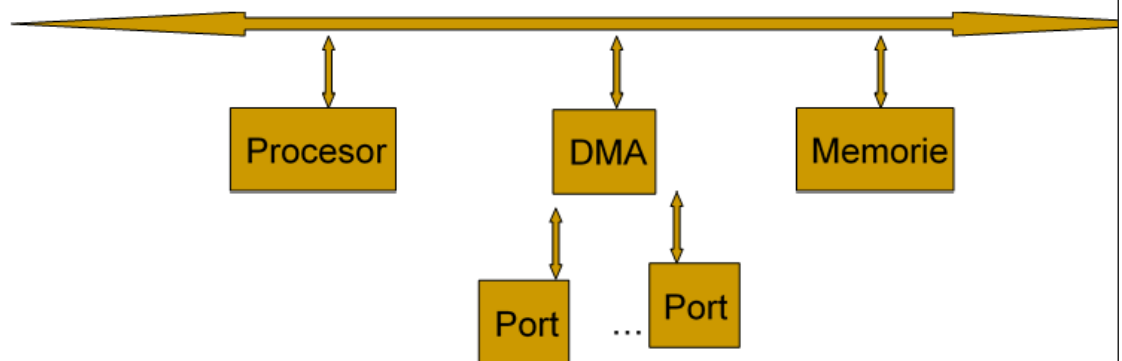


■ Fiecare transfer folosește magistrala de 2 ori:

- ☐ Port cu DMA,
- ☐ DMA cu memoria

■ UC este oprit de 2 ori/ transfer.

■ **Magistrală comună, controler DMA integrat:**



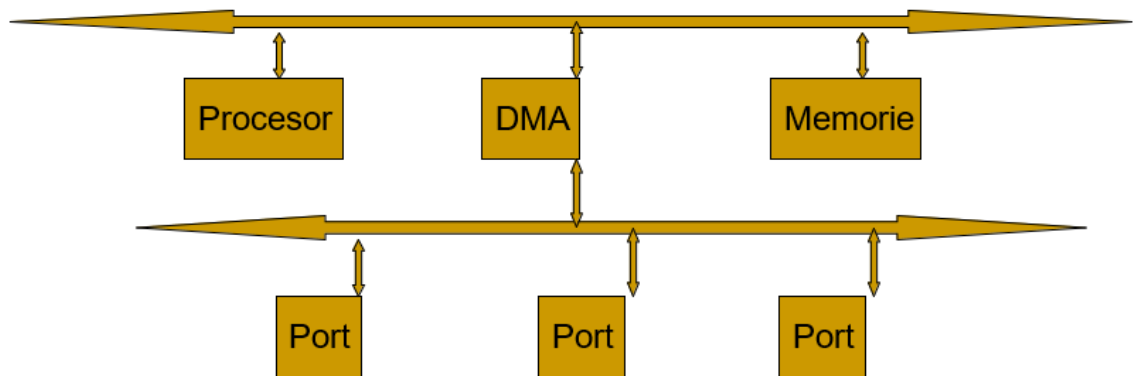
■ Controlerul DMA este conectat la mai multe porturi.

■ Fiecare transfer folosește magistrala o singură dată: DMA cu memoria.

UC este oprit o singură dată/ transfer.



■ **Magistrală de intrare/ ieșire distinctă:**



■ Fiecare transfer folosește magistrala comună o singură dată: DMA cu memoria.

■ UC este oprit o singură dată/ transfer.

**37. Ce este circuitul 82C37A? Prezentați caracteristicile sale.**

. Controlerul DMA 82C37A (o variantă mai modernă a clasicului 8237A)

- ☐ Dispune de 4 canale independente,
- ☐ Poate fi cascadat pentru a obține oricâte canale,
- ☐ Control individual (activare/ dezactivare și polaritate) asupra fiecărei cereri DREQ și fiecărui răspuns DACK,
- ☐ Permite transfer memorie – memorie,
- ☐ Rată de transfer ridicată: până la 4 MO/ sec. sau până la 6.25 MO/ sec. în funcție de frecvența tactului;
- ☐ Asigură incrementarea sau decrementarea adresei,
- ☐ Se folosește cu un registru extern pentru jumătatea mai semnificativă a adresei,
- ☐ Poate lucra în mai multe moduri,
- ☐ Compatibil TTL/ CMOS.

**38. Descrieți întreruperile externe.**

- Sunt provocate de evenimente externe microprocesorului care cer atenție imediată din partea acestuia.
- Cel care cere întreruperea trebuie să:
  - ☐ Țină cererea activată până când microprocesorul răspunde;
  - ☐ Se identifice atunci când microprocesorul o cere.
- Pot exista cereri simultane din partea mai multor surse => prioritizare:
  - ☐ Fie cu o logică de prioritizare:
    - Serială ("daisy – chain") sau

- Paralelă;
- ☐ Fie cu un PIC;
- ☐ Prioritizarea poate fi:
  - Fixă;
  - Programabilă sau
  - Rotativă.

### 39 . Descrieți întreruperile interne

- Se găsesc la microprocesoarele peste 8 biți și la toate microcontrolerele;
- Se numesc și excepții la unele microprocesoare;
- O parte din ele se numesc și “traps” la unele microprocesoare;
- Provocate de cauze interne procesorului;
- La microcontrolere pot fi mascate, la microprocesoare nu;
- 2 tipuri de cauze (la microprocesoare):
  - ☐ Evenimente interne speciale sau
  - ☐ Instrucțiuni dedicate (se mai numesc și întreruperi software).
- Întreruperi interne provocate de evenimente speciale:
  - ☐ Pot fi generate înainte de execuția unei instrucțiuni: de ex.: o violare de privilegiu sau cod ilegal etc.;
  - ☐ Pot fi generate după execuția unei instrucțiuni: de ex.: o divizare la 0, obținerea unui rezultat în afara unui domeniu, pas cu pas etc.

### 40. Ce este circuitul 8259? Prezentați caracteristicile sale.

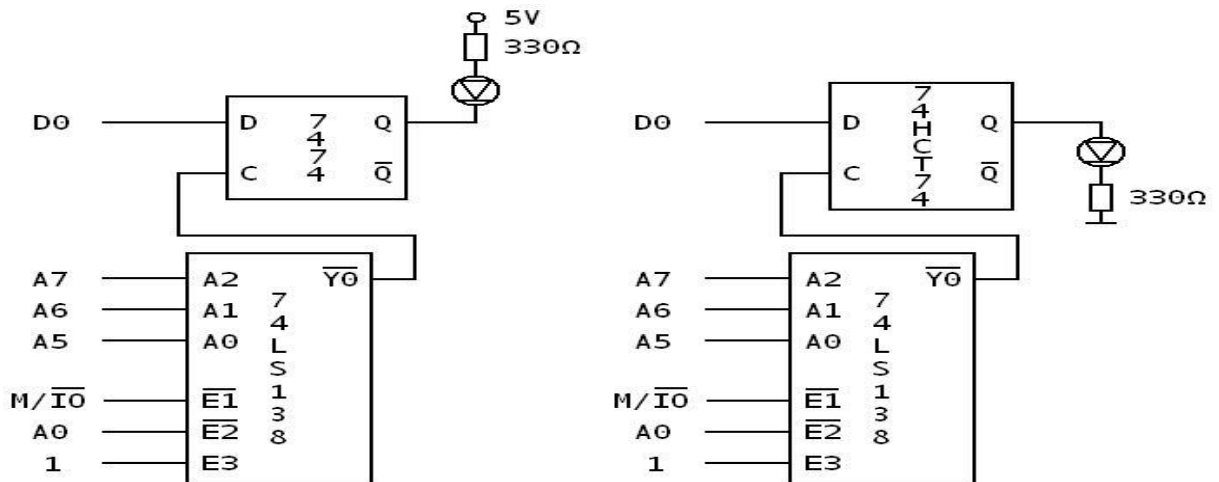
. Controlerul pentru întreruperi (PIC) 8259A

- Poate gestiona și prioritiza 8 cereri de întrerupere;
- Poate fi legat în cascadă cu alte 8 circuite similare asigurând astfel gestionarea și prioritizarea a 64 cereri de întrerupere;
- Posibilitate de mascare individuală a cererilor;
- Mai multe moduri de gestionare a întreruperilor:
  - ☐ Fully nested,
  - ☐ Rotating priority,
  - ☐ Special mask și
  - ☐ Poll.
- Necesită 2 - 4 cuvinte de inițializare (ICW1 – 4) și 4 cuvinte de operare (OCW1 – 4);

- Generează vectorul de întrerupere asociat cererii luată în considerare de procesor.

#### 41. Justificația existența unui circuit de memorare (bistabil, registru) în schema de comandă a unui led prin program.

- Conectarea la o UC se face prin intermediul unui port de ieșire;
- Elementul de afisare poate fi un bistabil sau un registru.



- În soluția a s-a folosit un bistabil în tehnologie TTL, ca urmare comanda va fi în 0 ( $I_{OL} = 16 \text{ mA}$ ,  $I_{OH} = 0,8 \text{ mA}$ );

- Secvența ca LED – ul să lumineze este:

```
MOV     AL,00H
```

```
OUT     00H,AL
```

- Secvența ca LED – ul să nu lumineze este:

```
MOV     AL,01H
```

```
OUT     00H,AL
```

- În varianta b bistabilul este în tehnologie HCT, ca urmare comanda se poate face fie în 0 fie în 1;

- Secvența ca LED – ul să lumineze (comandă în 1):

```
MOV     AL,01H
```

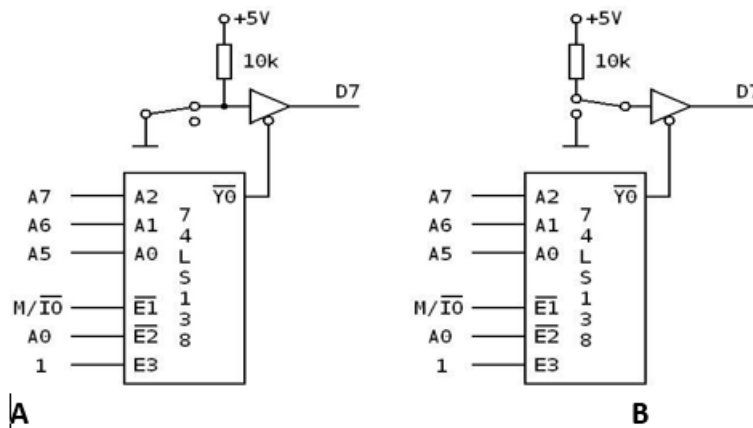
```
OUT     00H,AL
```

- Secvența ca LED – ul să nu lumineze (comandă în 0):

```
MOV     AL,00H
```

```
OUT     00H,AL
```

#### 42. Descrieți o soluție pentru citirea stării unui comutator prin program.



■ Secvența pentru soluția a:

```
IN      AL,00H
AND     AL,80H
JZ      SUS ; ramura corespunzătoare poziției din figură
; ramura corespunzătoare poziției contrare
```

■ Secvența pentru soluția b:

```
IN      AL,00H
AND     AL,80H
JNZ     SUS ; ramura corespunzătoare poziției din figură
; ramura corespunzătoare poziției contrare
```

#### 43. Care este diferența între arhitectura și microarhitectura unui microprocesor?

##### ■ Arhitectură vs. microarhitectură

- ☐ Arhitectură: setul de instrucțiuni, registrele, structura datelor în memorie; sunt informații accesibile programatorului;
- ☐ Microarhitectură: implementarea arhitecturii pe suportul fizic; este îmbunătățită continuu;

44. Dați exemple de arhitecturi ale seturilor de instrucțiuni ale microprocesoarelor. Caracteristici.

##### ☐ Von Neumann

- Codul și datele sunt reprezentate în memorie la fel și sunt accesate la fel;
- Se folosesc aceleași magistrale; rezultă simplitate;

- Întrucât toate accesele se fac la aceeași memorie și memoria este mai lentă ca procesorul (la toate nivelele tehnologice), aceasta va frâna execuția programului;

☐ Harvard

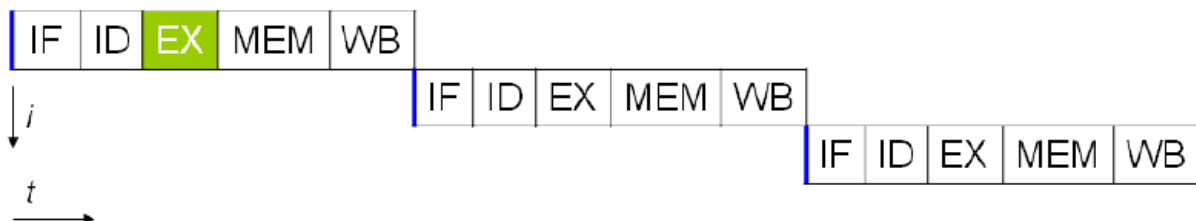
- Codul și datele sunt păstrate în memorii diferite; există 2 spații de adrese;
- Accesul la cod și date se face prin magistrale diferite; rezultă viteză mare dar scheme mai complicate;
- Structura Princeton: un singur spațiu de adrese dar 2 blocuri de memorii;

**45. Descrieți evoluția soluțiilor implementate în microprocesoare pentru execuția instrucțiunilor (fără hyperthreading).**

- 1 ciclu instrucțiune = 5 cicluri mașină:

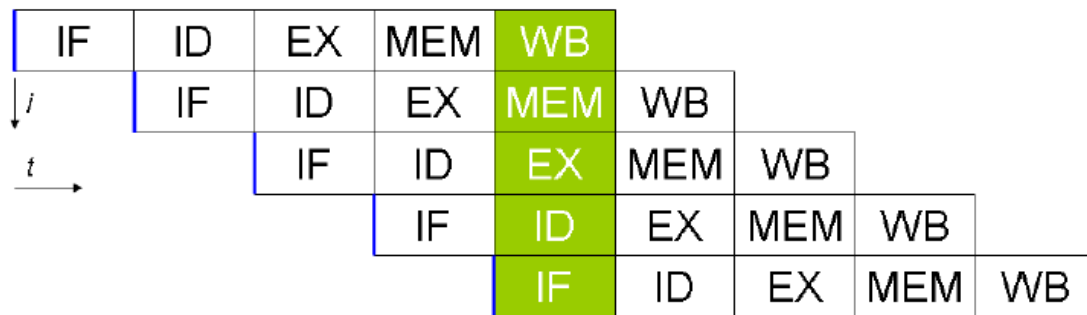
- ☐ IF: aducere cod de instrucțiune (instruction fetch);
- ☐ ID: decodificare cod de instrucțiune (instruction decode);
- ☐ EX: generare semnale de comandă și control (execution);
- ☐ MEM: citire operanzi;
- ☐ WB: scriere rezultat;

- Procesoare cu execuție secvențială:



- ☐ Hardware ineficient folosit, viteză mică;
- ☐ Simplitate, ușor de stabilit durata secvențelor de cod;

Procesoare cu bandă de asamblare (pipeline):



- ☐ Rata de execuție crește de 5 ori (banda de asamblare are 5 nivele);
- ☐ Hardware eficient folosit; hardware complex;
- ☐ Latența este dată de cel mai lent nivel;

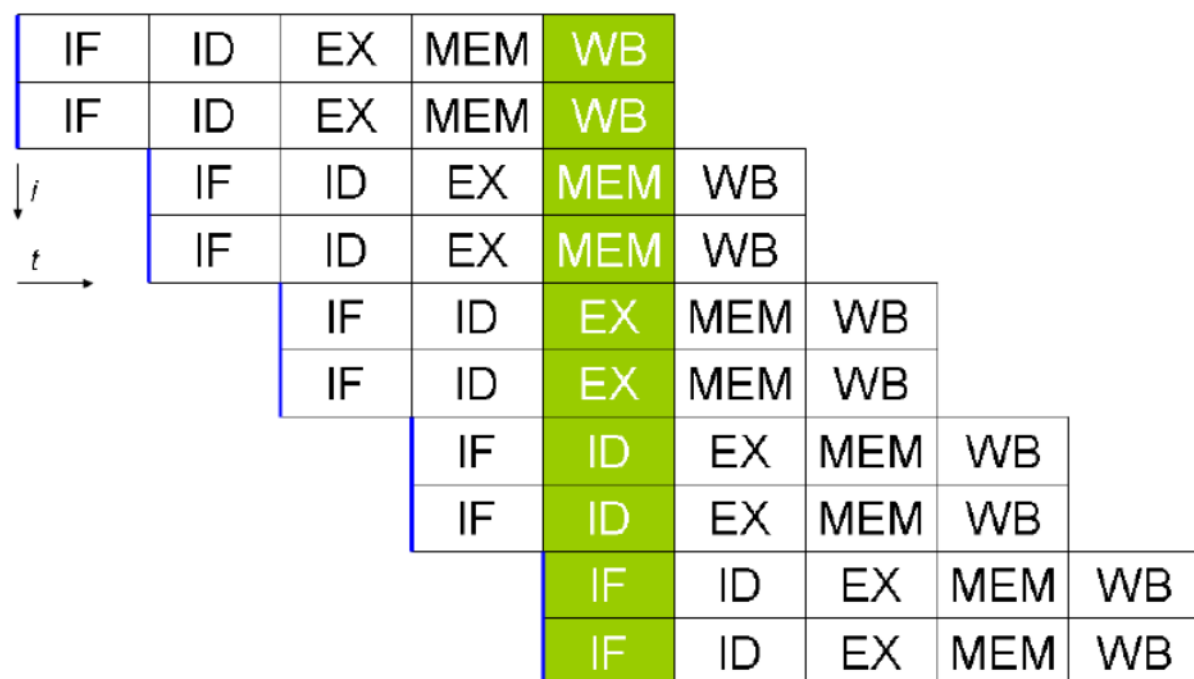
Procesoare cu superpipeline: Pentium 4 cu pipeline cu 20 nivele;



- ☐ Nivelele mai lente sunt divizate în nivele cu latență mai mică;
- ☐ Hardware complex;

#### ■ Procesoare superscalare:

- ☐ Au un singur nucleu dar mai multe resurse de execuție (multiple ALU, multiple FPU etc.);
- ☐ Execută simultan mai multe instrucțiuni;



46. Care este cauza limitării eficienței benzii de asamblare? Prezentați soluții pentru creșterea eficienței benzii de asamblare.

☐ Creșterea eficienței benzii de asamblare

☐ Scade la apariția instrucțiunilor care rup secvențialitatea unui program: salturi, apeluri, întreruperi; pierderea este mare la apariția instrucțiunilor de salt sau apel condiționat;

☐ Soluții:

☐ Execuția speculativă: se ghicește ramura unui salt/apel condiționat care va fi continuată; în cazul ghicirii eronate, banda trebuie golită; în cazul ghicirii corecte, nu se pierde timp;

☐ Întârzierea execuției: după instrucțiunea de salt/apel condiționat se plasează o instrucțiune care va fi executată indiferent de ramura care va continua; dacă nu sunt dependențe de date, această instrucțiune poate fi NOP; se câștigă timp pentru ca instrucțiunea de salt/apel condiționat să stabilească ramura pe care va continua;

☐ Predicția ramurii: se ghicește ramura care se va executa dar nu aleator ci bazat pe anumite considerente, de obicei istoria salturilor;

☐ Ex.: execuția unei bucle; o buclă este executată de mai multe ori înaintea terminării ei, însemnând că au loc mai multe ramificări în o direcție și doar o singură ramificare în cealaltă direcție; fie secvența în pseudocod HLL:

47. Care sunt diferențele principale între microprocesoarele unice, multimicroprocesoare, microprocesoarele multitasking și cele multicore?

☐ Microprocesoare unice

☐ Execuție secvențială a instrucțiunilor;

☐ Facilități: pipeline, out-of-order execution (instruction level parallelism)

☐ Performanța limitată de interdependențele între instrucțiuni;

☐ Performanța pipeline-ului limitată de instrucțiunile care modifică secvențialitatea;

☐ Multimicroprocesoare: sisteme cu mai multe microprocesoare + circuite, cu acces comun la resurse;

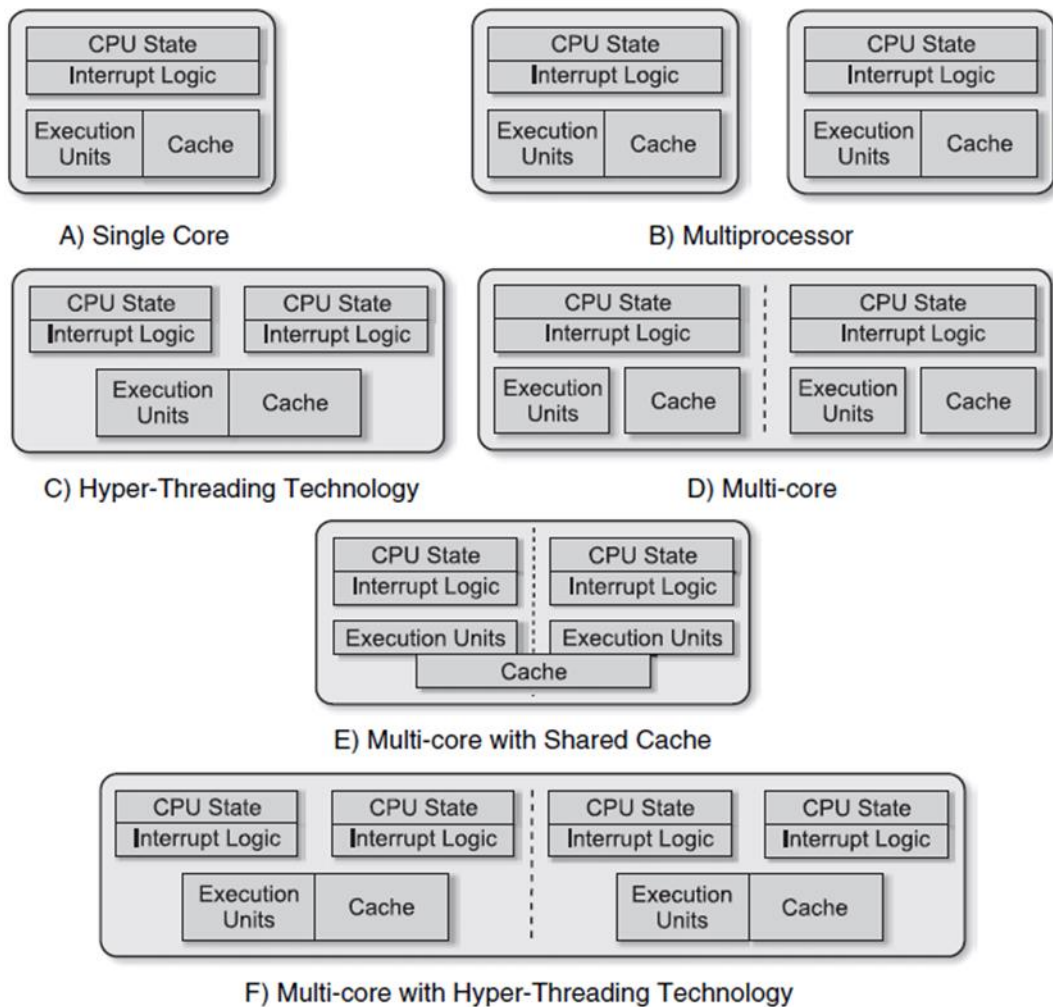
☐ Problemă: accesul la resursele comune;

☐ Microprocesoare multitasking (multithreading)

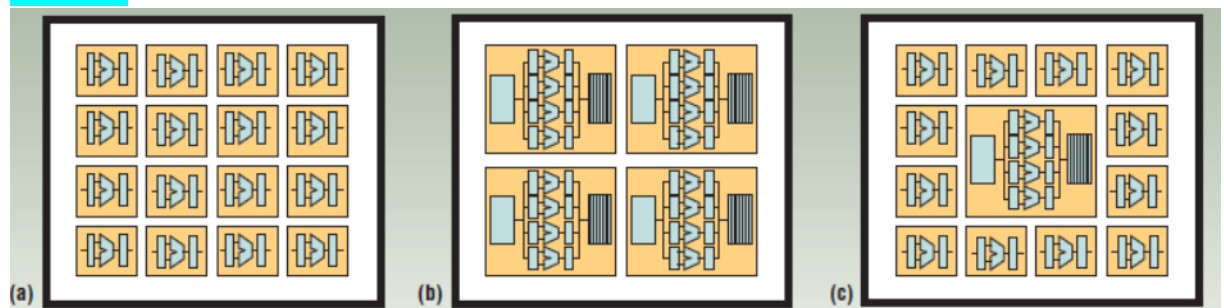
☐ Un unic microprocesor fizic este văzut de software ca mai multe microprocesoare logice;

- ☐ Fiecare microprocesor logic execută un task;
- ☐ Problemă: comutarea taskurilor;
- ☐ Microprocesoare multicore
- ☐ Mai multe microprocesoare fizice (core (nucleu)) execută fiecare câte un task;

48. Prezentați schema bloc a unui microprocesor multicore cu hyperthreading.



49. Prezentați structura unui microprocesor multicore simetric și a unui asimetric.



- ☐ a: multicore simetric cu 16 nuclee cu 1 BCE;
- ☐ b: multicore simetric cu 4 nuclee cu 4 BCEs;
- ☐ c: multicore asimetric cu 12 nuclee cu 1 BCE și 1 nucleu cu 4 BCEs;
- ☐ Procesoarele multicore pot fi:
- ☐ Simetrice: toate nucleele sunt la fel (resurse, performanțe, cost);



☐Asimetrice: nuclee diferite (resurse, performanțe, cost);

☐Dinamice: nucleele își pot modifica caracteristicile în timpul rulării;

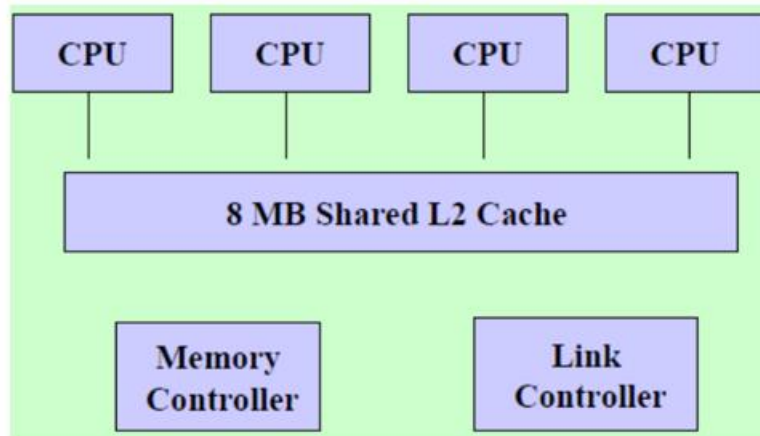
sau:

☐Omogene: toate nucleele au aceeași funcționalitate;

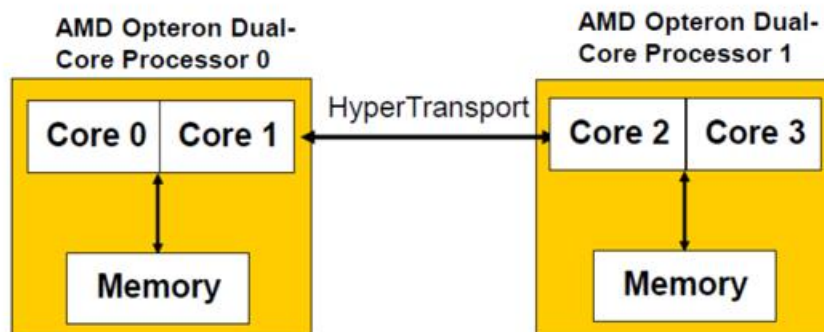
☐Heterogene: o parte din nuclee au funcționalități diferite, sunt neconvenționale (GPU, FPGA etc.);

**50. Prezentați schema bloc a unui microprocesor Intel Quad Core și a unui microprocesor AMD Dual sau Quad Core. Care sunt diferențele principale?**

Quad Core Intel



AMD Duo Core



Diferențe :

Diferențe :

Nr nuclee

Nr cpu

Memorie mai mare la quad core

Performanța (poate rula mai multe taskuri deodată)

**50. Descrieți 3 inovații ale microarhitecturii Intel Core.**

☐ Intel Intelligent Power Capability

■ Gestionează puterea în mod dinamic;

- Se bazează pe Power-Gating; întârzierea introdusă de etajul de comandă a alimentării este minimizată;

☐ Intel Advanced Smart Cache:

- Prin crearea unei memorii cache L2 comune, crește probabilitatea ca fiecare nucleu să acceseze informația dorită din memoria cache;
- Dacă un nucleu accesează rareori memoria cache comună, alt nucleu poate să o utilizeze mult mai intens;

☐ Intel Advanced Digital Media Boost:

- Permite creșterea performanței la execuția operațiilor de tip SIMD, cu instrucțiuni pe 128 biți (extensia SSE – Streaming SIMD Extensions); instrucțiunile pe 128 biți pot fi executate în un singur ciclu față de 2 cicluri în lipsa acestei caracteristici;
- Util la aplicații multimedia, video, sunet, procesare foto, criptare, calcule ingineresti etc.;