

Sisteme cu MicroProcesoare

Cursul 2

Arhitectura unui microcontroler

Conf. Gigel Măceșanu

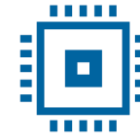


Universitatea
Transilvania
din Brașov

FACULTATEA DE INGINERIE ELECTRICĂ
ȘI ȘTIINȚA CALCULATOARELOR

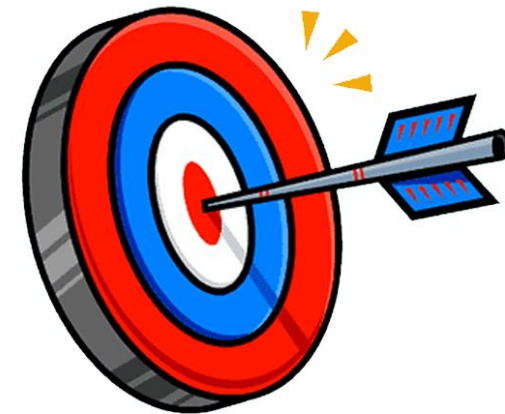


Cuprins

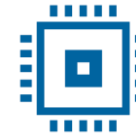


- Reprezentarea datelor
- Memoria unui microcontroler
- Unitatea Centrală de Procesare
- Tipuri de arhitecturi de MC
- Instrucțiuni de asamblare
- Circuite de sincronizare

- Înțelegere modalităților de reprezentare și memorare a datelor, a structurii de bază a microprocesorului, a setului de instrucțiuni specifice unui sistem de calcul cu microcontroler și a modulelor generatoare de semnale de sincronizare:
 - Modul în care datele sunt reprezentate afectează direct eficiența procesării, acuratețea rezultatelor și compatibilitatea între diferite sisteme și componente
 - Memoria este vitală pentru stocarea programelor și datelor necesare pentru execuție. Tipul și structura memoriei influențează performanța și capacitatea sistemului
 - CPU este inima oricărui sistem de calcul. Înțelegerea funcționării sale permite optimizarea performanței și utilizarea eficientă a resurselor de calcul
 - Diferite arhitecturi oferă avantaje și dezavantaje în funcție de aplicație
 - Programarea la nivel de asamblare oferă un control detaliat asupra hardware-ului, permițând optimizarea performanței și utilizarea eficientă a resurselor
 - Circuitele de sincronizare asigură coordonarea și temporizarea precisă a operațiunilor.



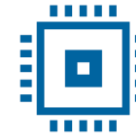
Bit vs Byte



- Un bit este unitatea de bază a informației în sistemul binar (0 sau 1)
- Un byte (octet) reprezintă 8 biți grupați
 - Baza pentru reprezentarea textului (ex. ASCII, UTF-8).
 - Folosit pentru a măsura dimensiunea fișierelor și viteza de transfer.
- Comparatie între dimensiuni:
 - Kilobyte (KB) – 1024 bytes
 - Megabyte (MB) – 1024 KB
 - Gigabyte (GB) – 1024 MB
 - Terabyte (TB) – 1024 GB



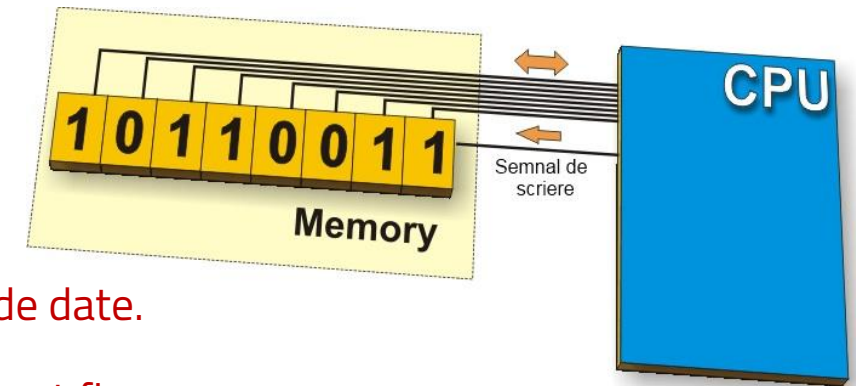
Regiștrii



- Registru ↔ celulă de memorie de viteză mare, integrată în CPU-ul microcontrolerului
- Stochează temporar date și instrucțiuni, permițând accesul rapid al procesorului la informațiile necesare pentru executarea programelor

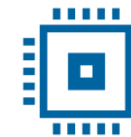
- Tipuri de registre:

- De date: Stochează date temporare necesare în calcule și manipulări de date.
- De adresă: Conțin adresele de memorie de unde pot fi citite sau unde pot fi scrise date.
- De control/configurare: stochează setări pentru diverse componente ale microcontrolerului.
- De stare: Indică starea curentă a procesorului și rezultatul operațiunilor

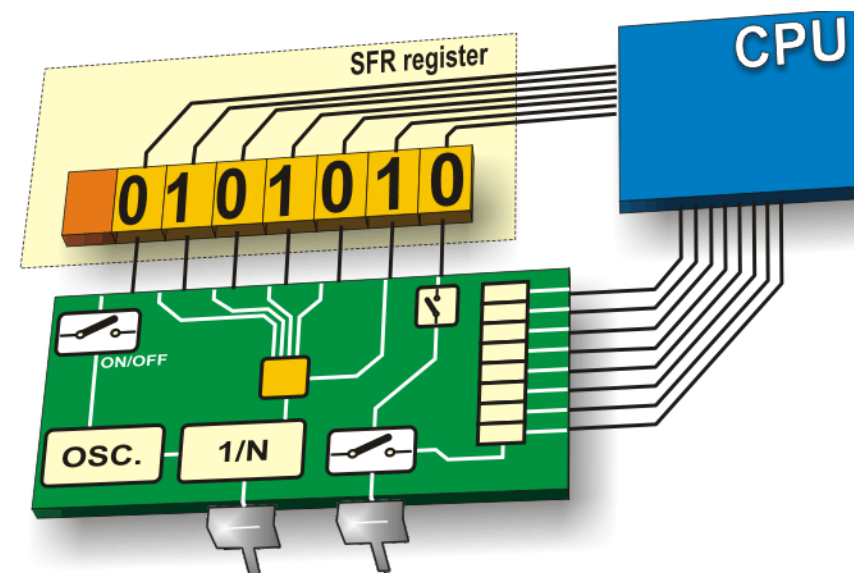


```
if (theory)
    byte
else if (practice)
    registru
```

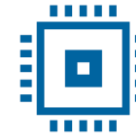
Regiștrii cu funcții speciale



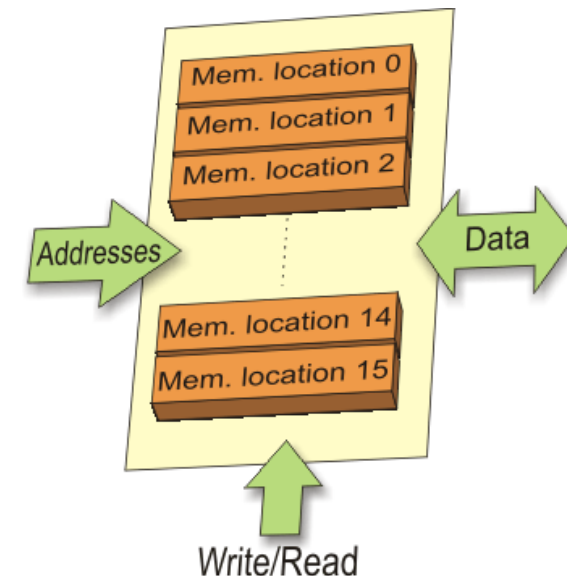
- Sunt conectați la circuite interne specifice: timer, ADC, oscilator
- Fiecare bit al registrului este asignat unei funcții:
 - de control: facilitează activarea sau dezactivarea funcției avute
 - de stare: marchează prin scrierea unei stări logice o anumită stare sau condiție ce a fost îndeplinită (zero, carry, overflow)
- Biții de control operează direct pe hardware
- Biții de stare sunt operați direct de hardware
- În general scriem biții de control și citim biții de stare



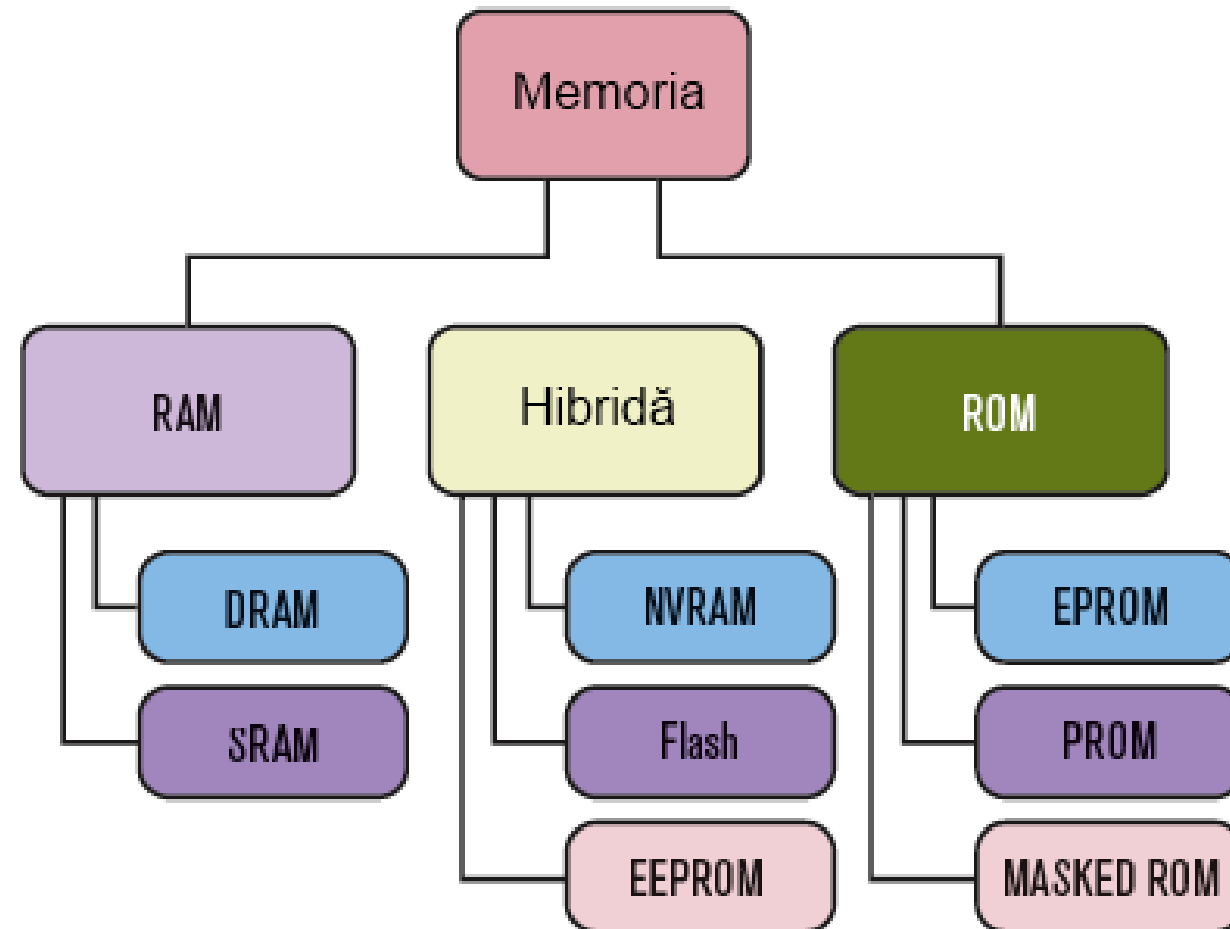
Memoria



- Stochează informațiile necesare pentru funcționarea corectă a microcontrolerului
- În principal sunt 2 tipuri de memorie:
 - Memoria program – conține instrucțiunile sau codul de program necesar pentru a executa sarcinile și funcționalitățile dorite
 - Memoria de date - utilizată pentru a stoca date temporare, variabile și alte informații necesare pentru execuția programului
- Fiecare locație de memorie are o adresă unică
- Pot avea funcții de: **scriere (W)**, **citire (R)**, **ambele**
- Într-un MC pot exista multiple tipuri de memorie



- Principalele tipuri de memorie, disponibile pe dispozitivele integrate sunt următoarele:



■ Memoria RAM (Random Access Memory)

- Utilizată pentru stocarea temporară a datelor
- Locațiile de memorie accesibile în orice ordine, cu aceeași viteză de acces
- Poate fi citită și scrisă de unitatea centrală a microcontrolerului

■ Memoria DRAM (Dynamic Random Access Memory):

- Volatilă, reprogramabilă, necesită controller DRAM,
- Formată dintr-un condensator și un tranzistor per celulă, ieftină, cel mai comun tip de RAM

■ Memoria SRAM (Static Random Access Memory):

- Volatilă, reprogramabilă, operare la viteze mari, scumpă
- 4-6 tranzistoare per celulă, folosită la formarea memoriei cache a procesorului

■ Memoria ROM (Read Only Memory)

- Utilizată pentru a stoca date permanente care nu se pot modifica de către utilizator (sau pot fi dar cu mare dificultate): programe de pornire, firmware și alte date care trebuie să fie disponibile la pornire
- Poate fi doar citită

■ Memoria PROM (Programmable Read Only Memory)

- Nevolatilă, reprogramabilă o singură dată de utilizator, cost redus, robustețe

■ Memoria EPROM (Erasable PROM)

- Nevolatilă, reprogramabilă (cu ultraviolete), robustă

■ Memoria MaskedROM

- Circuitele conțin o mască (software) care este adăugată memoriei în procesul de producție

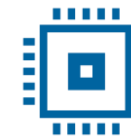
■ Memoria Flash

- Nevolatilă și reprogramabilă
- Nu pot fi șterse zone individuale (ștergere neselectivă)
- Poate fi ștearsă/scrisă de un număr foarte mare de ori
- Majoritatea MC actuale utilizează această memorie pentru stocarea programului

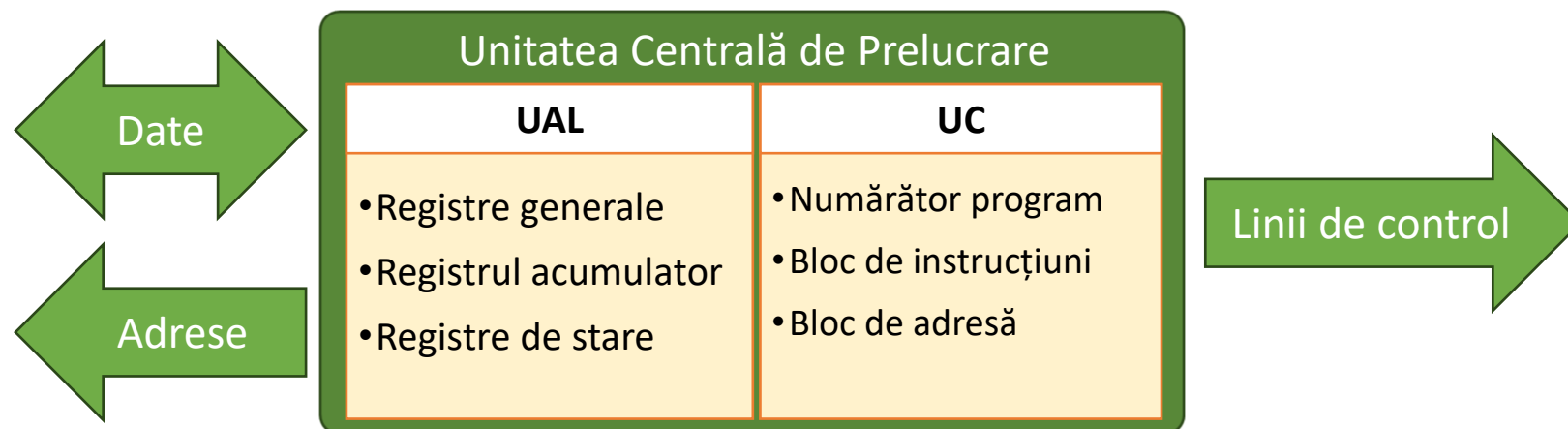
■ Memoria EEPROM

- Nevolatilă, reprogramabilă, folosită pentru stocarea datelor de configurație
- Pot fi șterse zone individuale (ștergere selectivă)
- Număr limitat de scrieri/ștergeri
- Viteză mică de scriere

Unitatea Centrală de Procesare - UCP



- Monitorizează și controlează toate procesele din interiorul unui MC
- Principalele două unități funcționale ale CPU sunt:
 - Unitatea Aritmetică și Logică
 - Unitatea Centrală



- Registre generale <-> memorie locală
- Registrul Acumulator:
 - Utilizat pentru stocarea datelor până când anumite operații vor fi efectuate (șiftare, adunare, etc.)

■ Registre de stare:

- Utilizate pentru a stoca informații despre starea curentă a procesorului, rezultatul ultimei operațiuni aritmetice sau logice și pentru a controla anumite aspecte ale execuției programului
- Aceste registre includ și diverse flag-uri: Zero, Carry, Sign, Overflow, Parity

■ Numărătorul de program (PC):

- Conține întotdeauna adresa următoarei instrucțiuni de executat

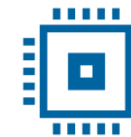
■ Bloc de instrucțiuni

- Preluarea (fetch) instrucțiunilor din memoria principală
- Decodificarea acestora pentru a determina ce operație trebuie să fie efectuată
- Blocul de instrucțiuni generează semnalele de control necesare pentru a coordona activitățile celorlalte componente ale CPU-ului, cum ar fi ALU, registrele și unitatea de memorie

■ Bloc de adresă:

- Responsabil pentru calcularea adreselor necesare pentru accesul la memorie
- Gestionează accesul la memorie
- Interpretează modurile de adresare specificate de instrucțiuni, cum ar fi adresarea directă, indirectă, relativă, indexată etc

Semnale la interfața UCP cu exteriorul



■ Pot fi grupate funcțional în trei categorii:

■ Magistrala de adrese:

- Liniile de pe această magistrală se folosesc pentru adresarea locațiilor de memorie și a porturilor de intrare - ieșire
- Transmite doar semnale de ieșire din microprocesor, fiind deci o magistrală unidirecțională
- Lățimea magistralei de adrese determină capacitatea maximă de memorie care poate fi adresată de CPU. De exemplu, o magistrală de 32 de biți poate adresa până la 2^{32} locații de memorie (4 GB)

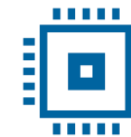
■ Magistrala de date:

- Este utilizată pentru transferul efectiv al datelor între CPU și memoria principală sau alte periferice
- Liniile magistralei de date pot transmite bidirecțional informațiile (intrare sau ieșire din UCP)
- Lățimea magistralei de date indică cantitatea de date care poate fi transferată simultan. De exemplu, o magistrală de 64 de biți poate transfera 64 de biți de date într-un singur ciclu.

■ Magistrala de control:

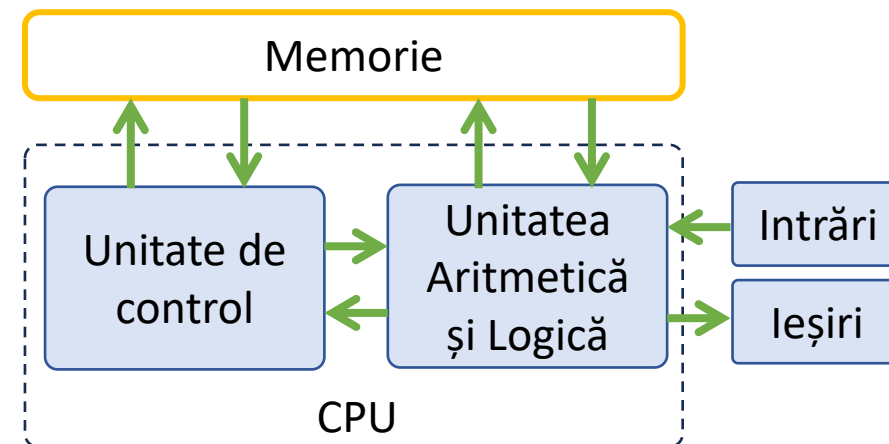
- Transportă semnalele de control care coordonează și gestionează operațiunile de citire și scriere, sincronizarea și alte funcții necesare pentru coordonarea activităților între CPU și restul sistemului
- De obicei, magistrala de control este bidirecțională, permițând atât CPU-ului cât și altor dispozitive să trimită semnale de control.

Arhitectura generală a unui MC

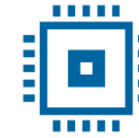


■ Arhitectura VON-NEUMANN

- O memorie comună pentru date și instrucțiuni
- Conține trei unități de bază:
 - CPU: responsabil pentru preluarea instrucțiunilor din memorie, decodificarea și executarea acestora
 - Memorie: utilizată pentru a stoca temporar datele și instrucțiunile pe care CPU-ul le folosește în timpul execuției programelor.
 - I/O: unități de intrare și ieșire care permit interacțiunea cu dispozitive externe, cum ar fi tastaturi, monitoare și imprimante
- Sunt necesare două etape pentru execuția unei instrucțiuni:
 - Extragere instrucțiune (fetch) din memorie, pe baza adresei din PC și decodificarea acesteia
 - Prelucrare dată din memorie și execuție instrucțiune decodificată. Rezultatul este salvat în registre speciale CPU sau în memorie



Arhitectura generală a unui MC

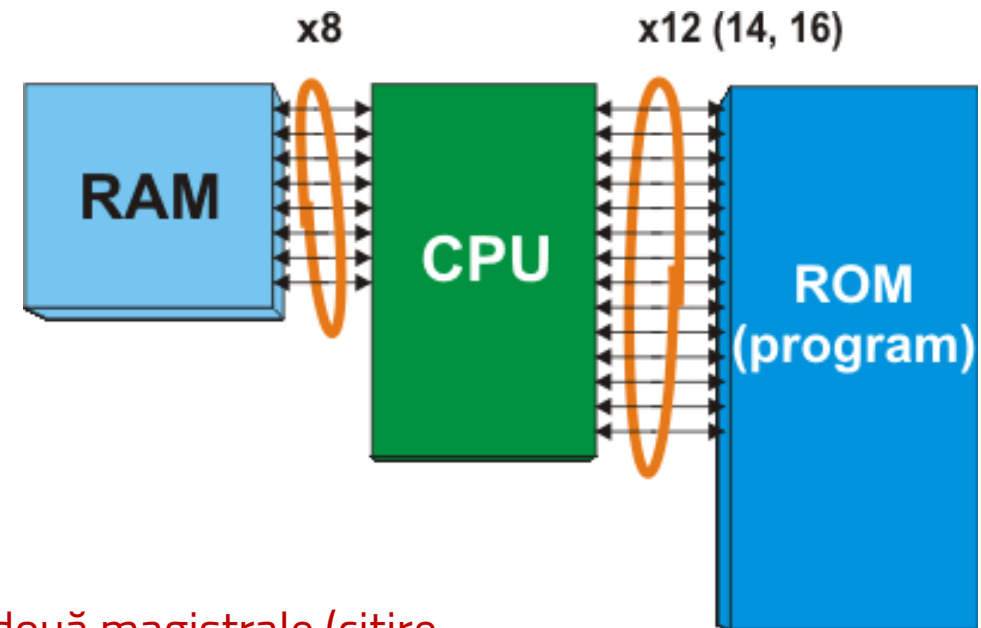


■ Arhitectura Harvard

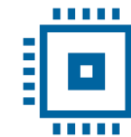
- Spațiu de memorie separat pentru instrucțiuni și date
- Magistrale proprii (utilizate exclusiv pentru instrucțiuni sau pentru date)
- Execuția unei instrucțiuni necesită un ciclu de clock

■ Avantajele arhitecturii Harvard

- Performanță sporită datorită separării magistrelor
- Transfer îmbunătățit de date
- Posibilitatea execuției cvasiparalele a informațiilor pe cele două magistrale (citire instrucțiune și accesare memorie)



Arhitectura generală a unui MC

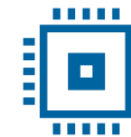


■ Execuție cvasiparalelă – conceptul de „pipeline”

- Mai multe instrucțiuni se execută simultan, dar decalat
- Acest lucru mărește considerabil viteza de procesare a unui procesor
- Implementarea unui pipeline este complexă și necesită o sincronizare precisă între diferitele stadii
- Hazarduri: pot apărea trei tipuri de hazard:
 - De date: o instrucțiune depinde de rezultatul unei instrucțiuni anterioare care nu a fost încă completată
 - De control: apar în cazul instrucțiunilor de ramificare (branch instructions), unde rezultatul unei condiții de ramificare nu este cunoscut imediat, ceea ce poate duce la execuția instrucțiunilor incorecte
 - Structurale: când două sau mai multe instrucțiuni necesită aceeași resursă hardware în același timp

	Ciclul 1	Ciclul 2	Ciclul 3	Ciclul 4	Ciclul 5
Instrucțiunea 1	Extrage	Decodează	Execută		
Instrucțiunea 2		Extrage	Decodează	Execută	
Instrucțiunea 3			Extrage	Decodează	Execută


Setul de instrucțiuni al unui MC



■ Setul de instrucțiuni reprezintă mulțimea de comenzi de bază pe care un microcontroler le înțelege.

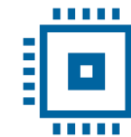
■ Tipuri de instrucțiuni prezente la toate MC:

- Instrucțiuni aritmetice și logice
- Instrucțiuni de decizie care implementează o ramificație a grafului (bifurcație)
- Instrucțiuni pentru transferul datelor
- Instrucțiuni de test și instrucțiuni la nivel de bit



```
...    ...  
...    ...  
movlw  0x3F  
movwf  TEMP1  
btfsc  MAX3,7  
goto   check  
btfsc  MAX3,6  
goto   opening  
btfsc  MAX3,5  
goto   closing  
...    ...  
...    ...
```

Setul de instrucțiuni al unui MC

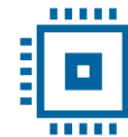


- MC pot avea 2 tipuri de seturi de instrucțiuni: RISC și CISC
- RISC (Reduced Instruction Set Computer):
 - Recunoaște și execută doar operații de bază
 - Operațiile complicate efectuate prin combinarea celor de bază
 - Execuție rapidă și eficientă (instrucțiunea este simplă și executată într-un singur ciclu de ceas)
- CISC (Complex Instruction Set Computer)
 - Peste 80 de instrucțiuni, atât de bază cât și complexe
 - Pot fi specializate pentru funcții specifice, optimizate pentru anumite tipuri de sarcini
 - Multe din aceste instrucțiuni sunt foarte diferite între ele

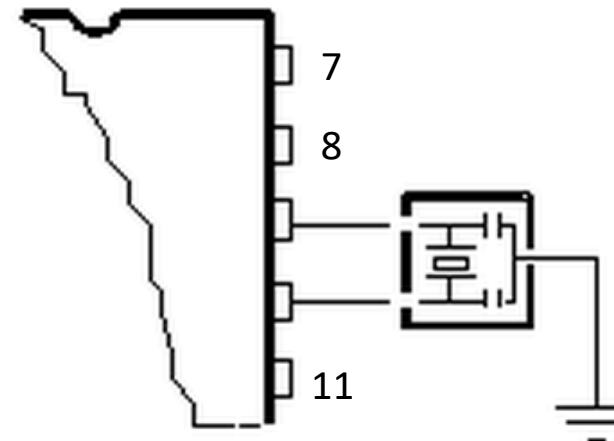
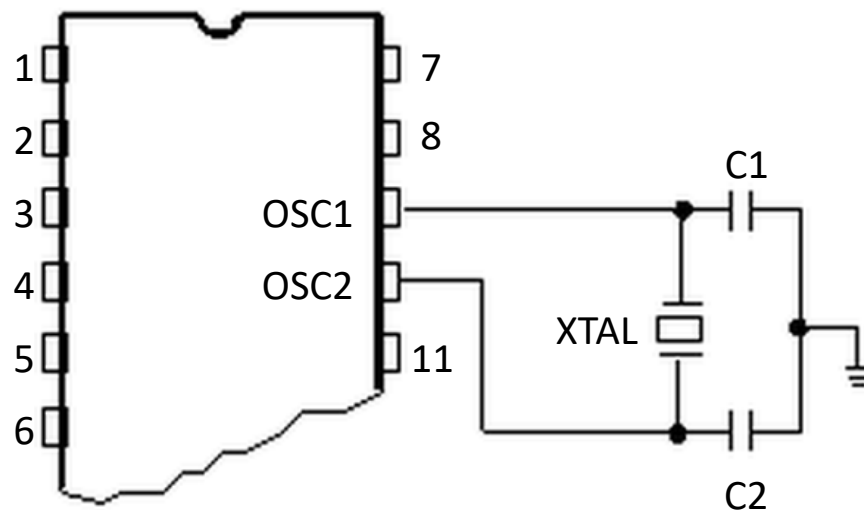
Arhitectura Atmega 328P

- Arhitectura Harvard modificata
- Memorie de date (2KB de memorie SRAM):
 - utilizată pentru stocarea datelor temporare, variabilelor şi altor informaţii necesare pentru execuţia programului
 - 5 tipuri diferite de adresare
 - Accesul intern la memorie se face în 2 clk de CPU
- Memorie de date (1KB de memorie EEPROM):
 - utilizată pentru stocarea datelor care trebuie să rămână salvate chiar şi după oprirea alimentării microcontrolerului
- Memorie de program (32 KB de memorie Flash):
 - utilizată pentru stocarea codului de program. În timpul execuţiei, instrucţiunile din memoria de program sunt citite şi executate de către procesorul AVR
 - 2 zone: Boot Loader şi Aplicaţie
- Program Counter pe 14 biţi

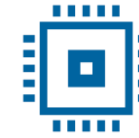
Circuitul oscilator



- Furnizează un semnal ceas (clock) necesar pentru sincronizarea proceselor interne
- Oscilatorul cu cuarț (XT): generează un semnal electric cu o frecvență foarte stabilă.
 - Circuit individual + condensatori (pentru a stabili și a ajusta frecvența de oscilație)
 - Încapsulat (rezonator)
 - Cristalul de cuarț vibrează la o frecvență precisă când este excitat electric datorită efectului piezoelectric

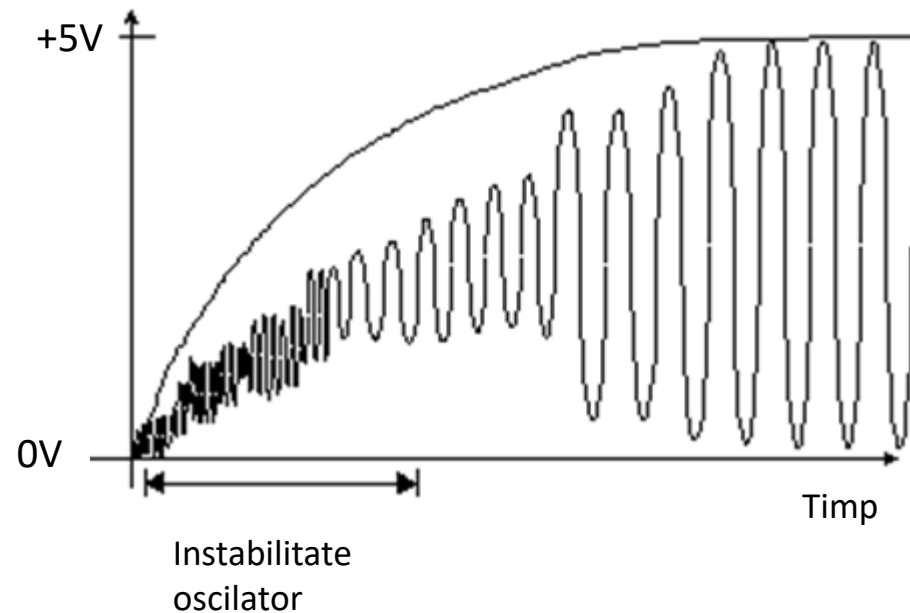
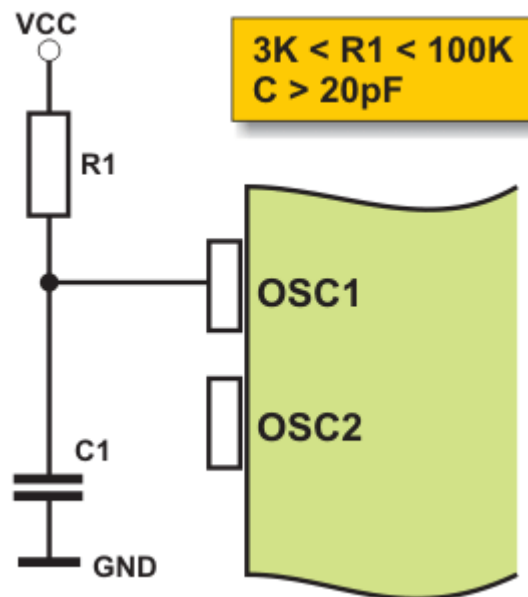


Circuitul oscilator

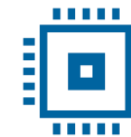


■ Oscilatorul Rezistor – Condensator (RC):

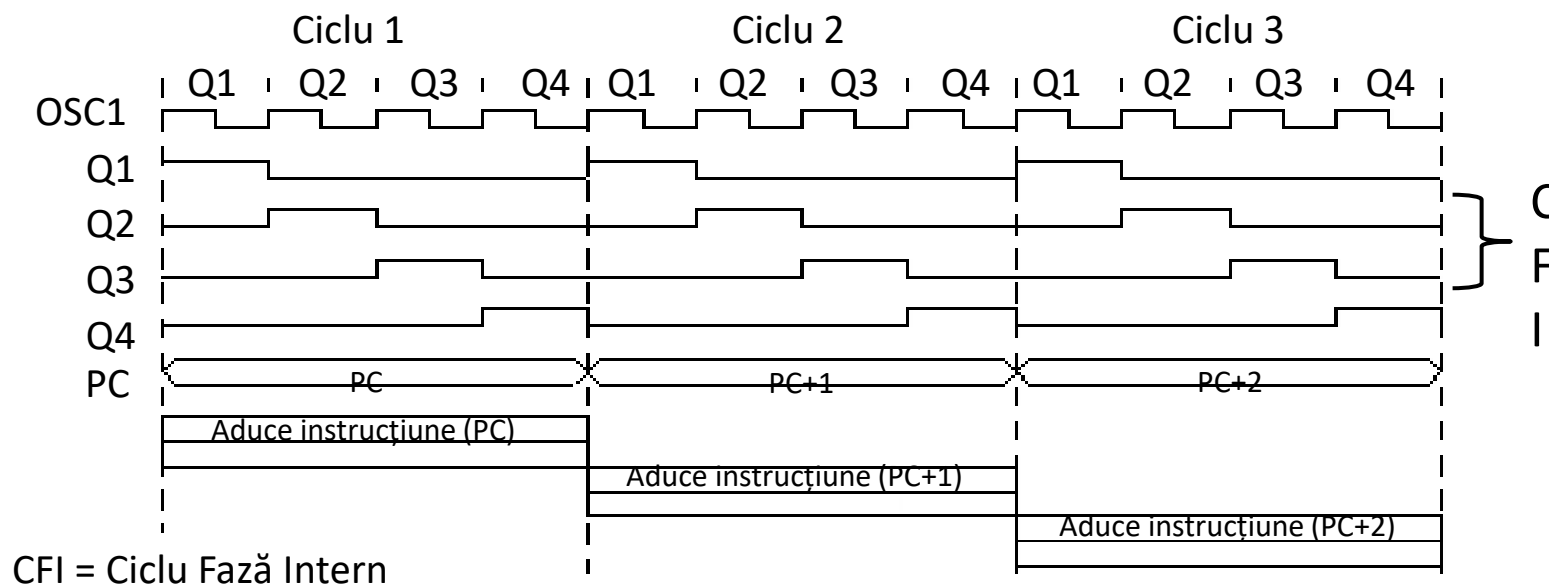
- Nu are o precizie foarte bună
- Frecvența generată de un oscilator RC poate varia odată cu schimbările tensiunii de alimentare



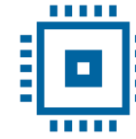
Ciclu instrucțiune



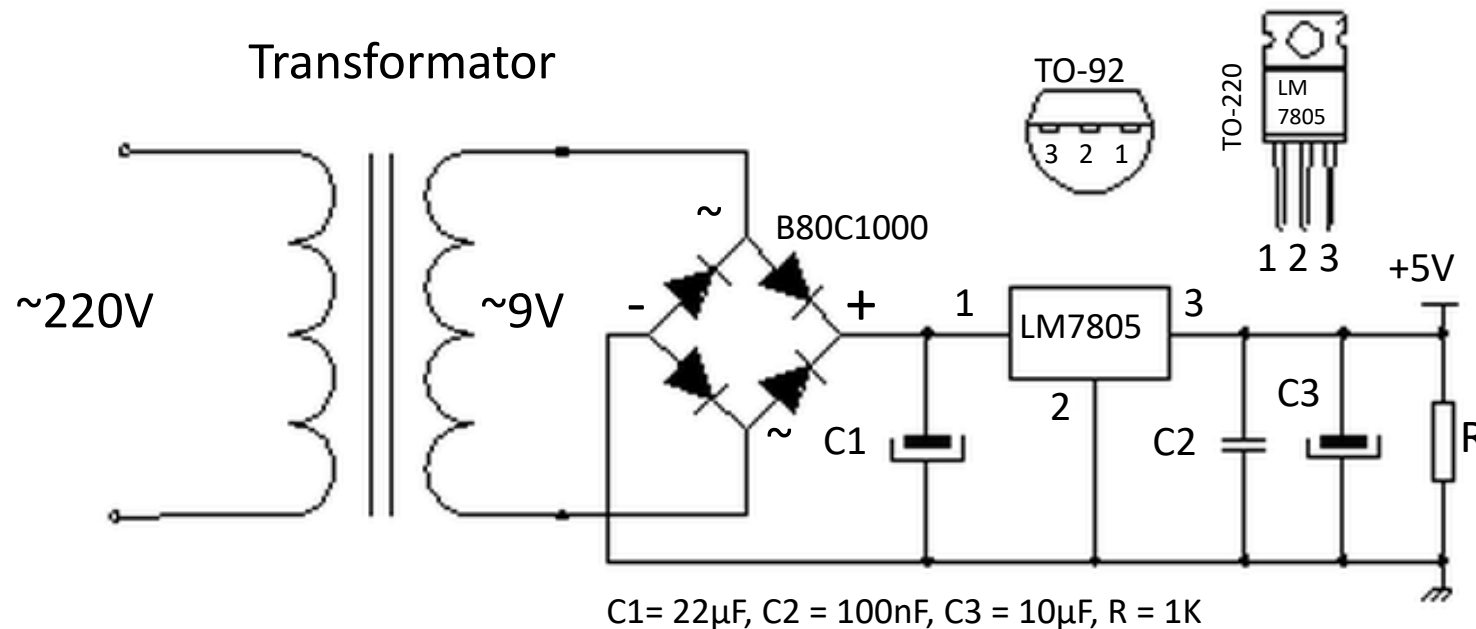
- Este obținut de la circuitul oscilator
- Este obținut din "n" clock-uri egale (de ex pentru $n=4$: Q1, Q2, Q3, Q4)
- Exemplu de etape ale unui ciclu instrucțiune:
 - Q1: Fetch (aducere instrucțiune) instrucțiune, PC++
 - Q2+Q3: Decodare instrucțiune și identificare operanzi (Aducere date din memorie pentru operanzi)
 - Q4: Executare instrucțiune și salvare rezultat



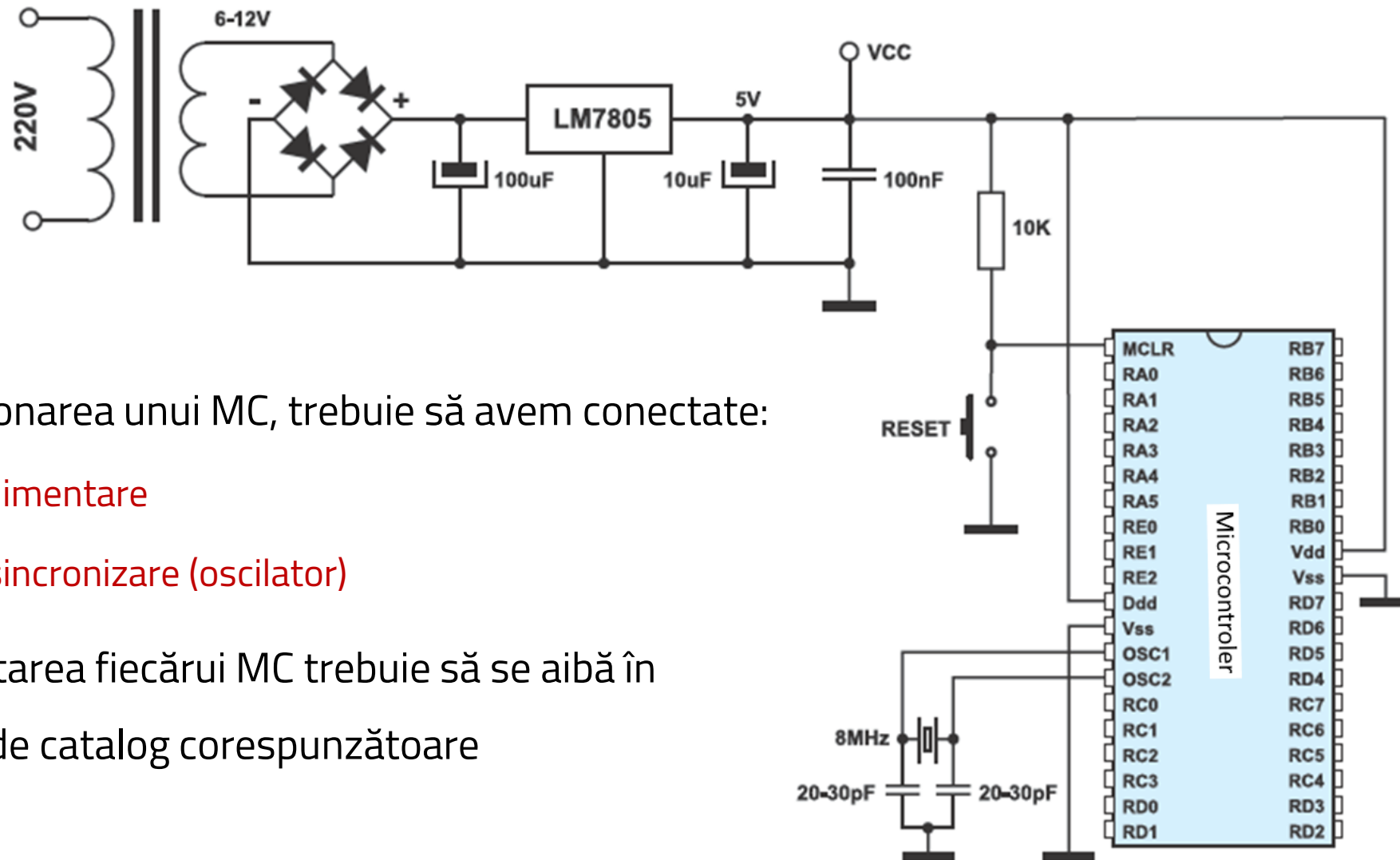
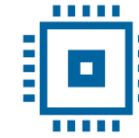
Alimentarea MC



- Tensiunea de alimentare: în general între 2.0÷6.0V
- Cea mai simplă schemă utilizează LM7805
- La pinul 1 (LM7805) tensiune de intrare între 7÷24V cc
- Pentru curent până la 1A TO-220 cu răcire adițională
- Pentru curent până la 100mA TO-92 (capsulă mică)



Exemplu conectare Microcontroler

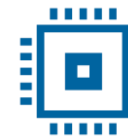


■ Pentru funcţionarea unui MC, trebuie să avem conectate:

- Sursă de alimentare
- Modul de sincronizare (oscilator)

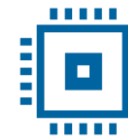
■ Pentru conectarea fiecărui MC trebuie să se aibă în vedere foaia de catalog corespunzătoare

Concluzii



- Modalitățile de reprezentare și memorare a datelor
 - Registre de bază, cu funcții speciale, tipuri de memorie
- Structura de bază a unui microprocesor
 - Unitate Centrală de Procesare și tipuri de arhitecturi
- Instrucțiunile specifice unui sistem de calcul cu MC
 - Clase de instrucțiuni și ciclu instrucțiune
- Module generatoare de semnale de sincronizare
 - Oscilatoare și rezonatoare





Contact:

Email: gigel.macesanu@unitbv.ro

elearning.unitbv.ro - Sisteme cu Microprocesoare