

ARHITECTURA UNOR MICROCONTROLERE DIN CATEGORIA PIC. TEHNICI DE ADRESARE A MEMORIEI. STRATEGII DE INTRARE-IEȘIRE

1. Microcontroler contra Microprocesor

Microprocesorul, este unitatea centrală de prelucrare a informației (CPU) a unui calculator sau sistem structurat funcțional, care coordonează sistemul și care, fizic, se prezintă sub forma unui cip electronic.

Microcontrolerul este o structură electronică destinată controlului unui proces sau, mai general, este un microcircuit care încorporează o unitate centrală (CPU), memorii și resurse care-i permit interacțiunea cu mediul exterior.

Pentru a fi folosit, unui microprocesor trebuie să i se atașeze alte componente ca memorie și componente pentru primirea și trimiterea de date. Microcontrolerul este proiectat să cu toate componentele necesare pentru funcționarea și interacțiunea cu mediul exterior.

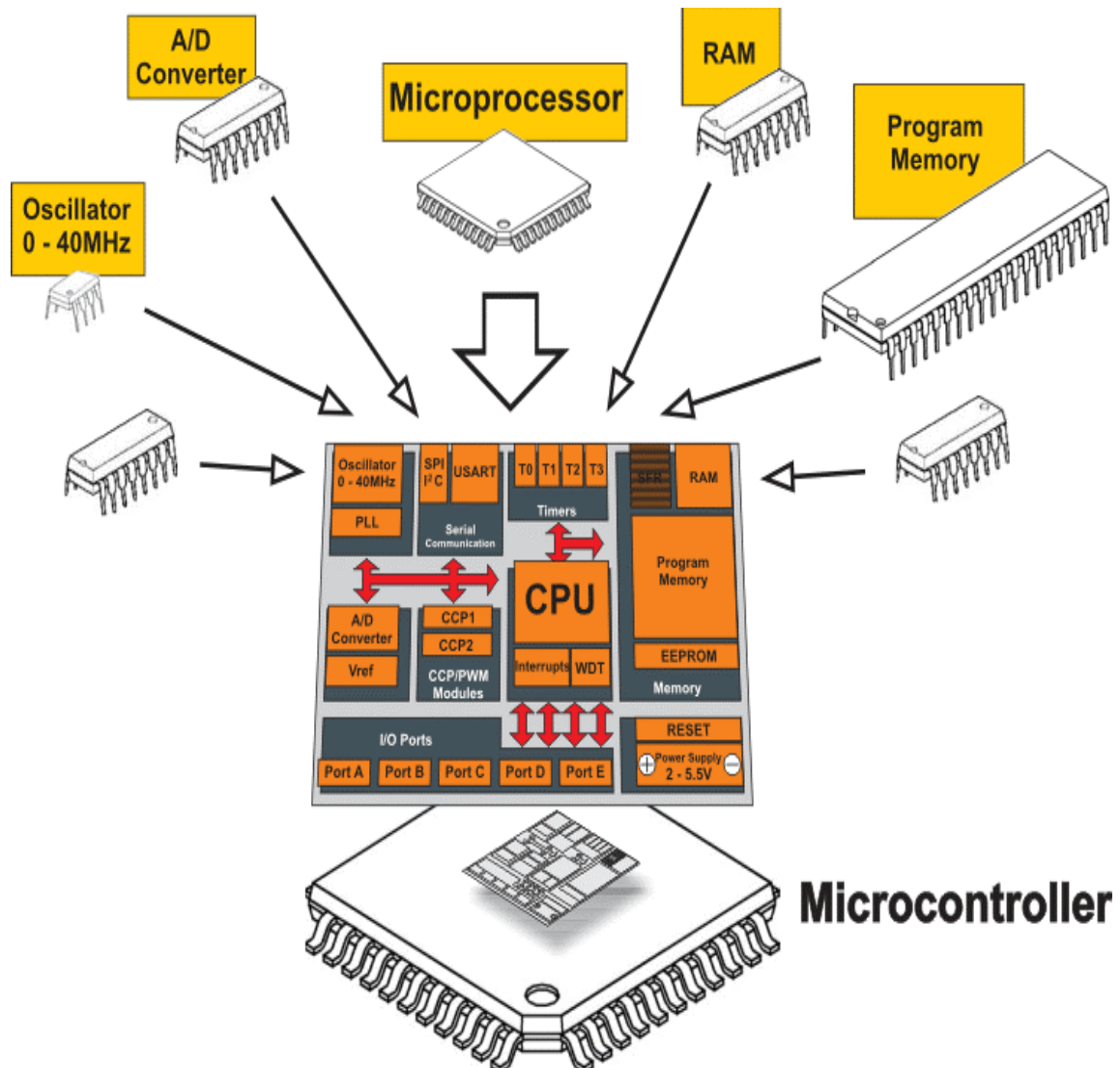


Fig. 1. Microcontroler contra Microprocesor

Printre multe domenii unde utilizarea lor este practic un standard industrial se pot menționa:

- în industria de automobile (controlul aprinderii motorului, climatizare, diagnoză, sisteme de alarmă, etc.);

- în așa zisa electronică de consum (sisteme audio, televizoare, camere video și videocasetofoane, telefonie mobilă, GPS-uri, jocuri electronice, etc.);
- în aparatura electrocasnică (mașini de spălat, frigidere, cuptoare cu microunde, aspiratoare);
- în industria aerospațială;

2. Structura unui microcontroler

2.1. Unitatea centrală de procesare

Unitatea centrală de procesare, (CPU) este structurată pe locații de memorie (registri) având capacitatea de a realiza operații logice și aritmetice (înmulțire, împărțire, scădere și adunare) precum și mutarea conținutului dintr-o locație de memorie în alta.

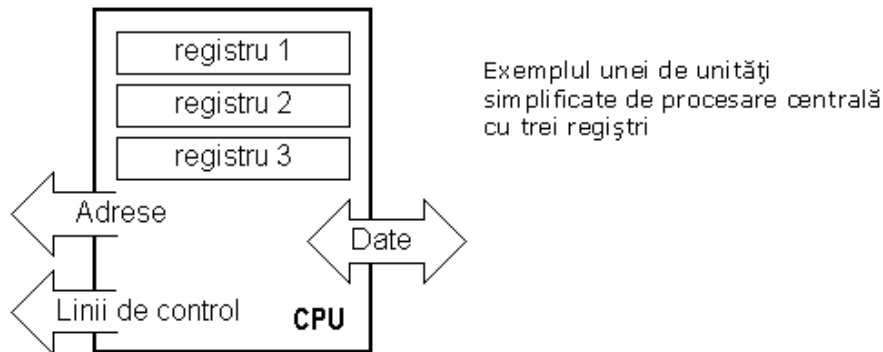


Fig. 2. Unitatea centrală de procesare

Regiștrii sunt deci locații de memorie al căror rol este de a ajuta prin executarea a variate operații matematice sau a altor operații cu date.

2.2. Unitatea de memorie

Memoria este o componentă a microcontrolerului a cărei funcție este de a stoca date.

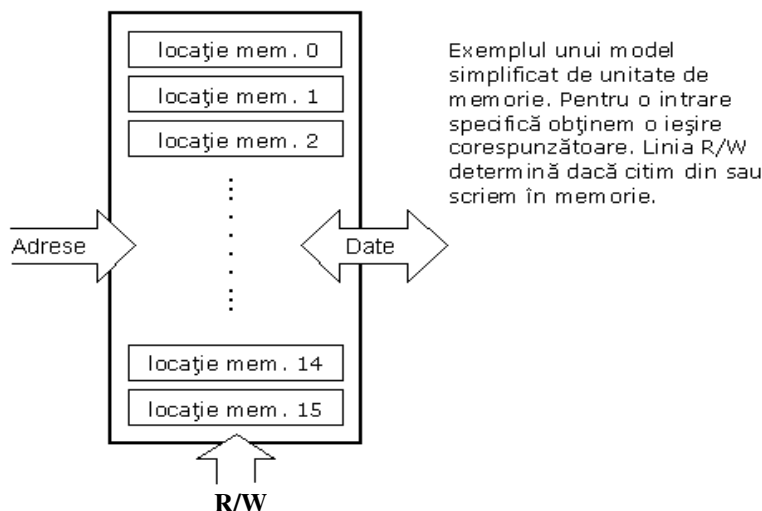


Fig. 3. Unitatea de memorie

Memoria este alcătuită din toate locațiile de memorie, iar adresarea acestora nu este altceva decât selectarea uneia din ele.

În afară de citirea dintr-o locație de memorie, memoria trebuie de asemenea să permită scrierea în ea. Aceasta se face prin asigurarea unei linii de control R/W (citește /scrie).

Linia de control este folosită în următorul fel: dacă R/W=1, se face citirea, și dacă opusul este adevărat atunci se face scrierea în locația de memorie.

2.3. Bus-ul

Bus-ul sau magistrală reprezintă un grup de 8, 16, sau mai multe fire. Sunt două tipuri de bus-uri: bus de adresă și bus de date.

Magistrala de adrese servește la transmiterea adreselor de la CPU la memorie, iar magistrala de date permite transmiterea datelor la componentele microcontrolerului.

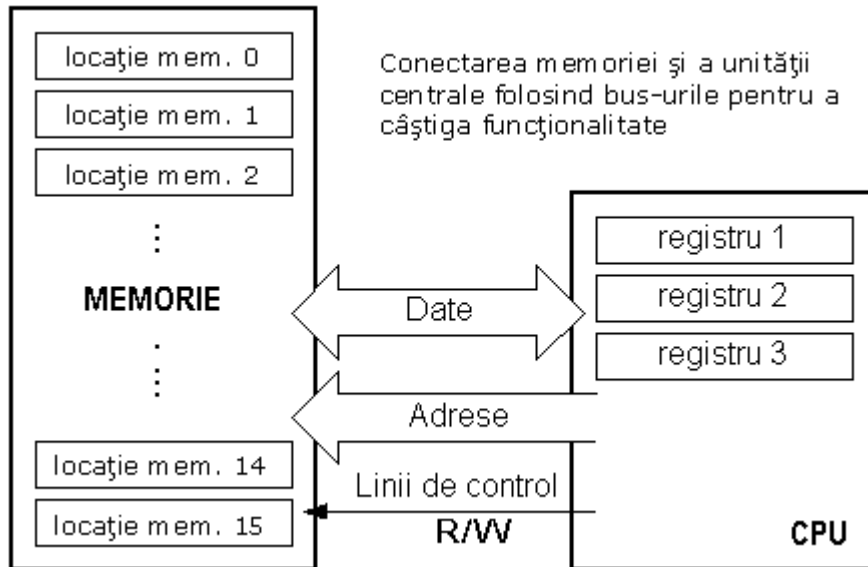


Fig. 4. CPU, memoria și magistralele

Tehnici de adresare a memoriei

Adresarea memoriei se face prin magistrala de adrese. Dacă magistrala de adrese are n linii atunci mulțimea adreselor este 2^n .

Pentru a rezolva o problema, procesorul trebuie să cunoască în fiecare moment atât operația pe care urmează să o execute, cât și datele care participă în operație. Aceste operații sunt comunicate procesorului prin intermediul instrucțiunilor.

Instrucțiunea reprezintă o succesiune de cifre binare prin care se indică procesorului operația de executat și amplasamentul (locul) operanzilor.

OPCODE	ID	ADROP1	...	ADROPn
Codul instrucțiunii	Modalitatea de accesare a memoriei	Adresă operand 1		Adresă operand n

Principalele modalități de adresare a memoriei sunt:

- adresarea în registru* - în acest caz informația asupra căreia acționează instrucțiunea se află localizată într-unul din regiștri microprocesorului.
- adresarea imediată* - informația referință în acest caz se află în memoria program la octetul imediat următor după OPCODE și ID.
- adresarea absolută* - informația este localizată printr-o adresă completă care se găsește în formatul instrucțiunii, imediat după OPCODE și ID.
- adresarea informației în stivă* - stiva constituie un segment de memorie separat. Accesarea informației din stivă se face utilizând registrul care conține adresa bazei segmentului de stivă și regiștrii care conțin baza și respectiv vârful stivei.

În ceea ce privește funcționalitatea microcontrolerului, situația s-a îmbunătățit, dar o nouă problemă a apărut: avem o unitate ce este capabilă să lucreze singură, dar nu poate interacționa cu mediul exterior.

Pentru a înlătura această deficiență, se mai adăugă un bloc ce conține câteva locații de memorie al căror singur capăt este conectat la bus-ul de date, iar celălalt are conexiune cu liniile de ieșire la microcontroler ce pot fi văzute cu ochiul liber ca pini la componenta electronică.

2.4. Unitatea intrare-ieșire

Unitatea intrare-ieșire este structurată pe locații de memorie denumite „porturi”. Porturile fizic sunt pini microcontrolerului. Sunt diferite tipuri de porturi: intrare, ieșire sau porturi pe două-căi.

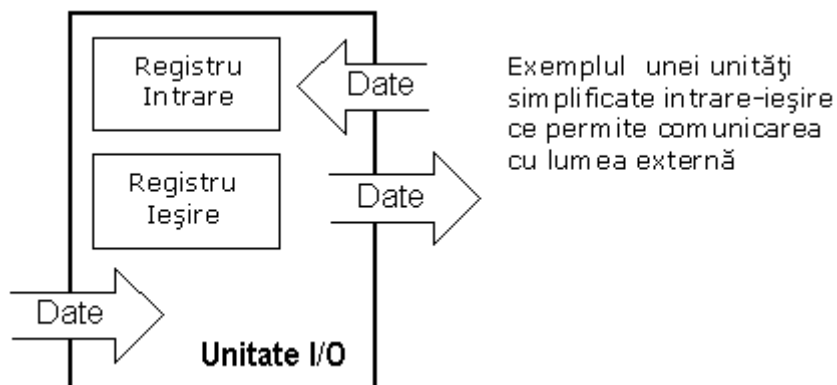


Fig. 5. Unitatea intrare-ieșire

Strategii intrare - ieșire

Din punct de vedere al aplicației software implementată pe microprocesor pot exista trei metode de realizare a operației de intrare-ieșire: metoda de interogare continuă, utilizarea întreruperilor și transferul direct cu memoria.

a. *Tehnica de interogare continuă* - microprocesorul când are nevoie de citirea unei date de la portul de intrare trimite adresa acestuia pe magistrala de adrese și așteaptă activarea semnalului READY de la portul respectiv. În intervalul de timp de la lansarea operației de citire de la porturi și până când este activat semnalul READY microprocesorul așteaptă, neputând să desfășoare nici o altă acțiune.

b. *Tehnica întreruperilor* – atunci când se activează o întrerupere, semnalul READY este activat, procesorul va încheia execuția instrucțiunii în curs și va realiza citirea datei disponibile pe portul de intrare. În acest caz, procesorul nu mai pierde timp în așteptarea semnalului READY, deci în așteptarea datelor de la portul de intrare.

c. *Tehnica de acces direct la memorie* - aceasta tehnică de intrare-ieșire este foarte frecvent utilizată pentru volume mari de date, de exemplu trimiterea datelor către o imprimantă sau citirea datelor de la o unitate optică (în cazul calculatoarelor). Atunci când se dorește transferul unui volum mare de date de la memorie la unul din periferice sau de la periferice la memorie, microprocesorul inițializează controlerul de acces direct la memorie DMA cu adresa de start a domeniului care trebuie să facă transferul, numărul de octeți și sensul în care se face transferul. Un astfel de mecanism de transfer este foarte eficient atunci când microprocesorul este prevăzut cu memorie cache. În intervalul de timp în care se face transferul de date microprocesorul poate continua activitatea utilizând instrucțiunile și datele stocate în memoria cache, deci apare o îmbunătățire substanțială a gradului de utilizare a timpului de calcul al microprocesorului.

2.6. Unitatea timer

Unitatea de bază a timer-ului este un contor liber (free-run) care este de fapt un registru a cărui valoare numerică crește cu o unitate la intervale egale, așa încât luându-i valoarea după intervalele t_1 și t_2 , pe baza diferenței lor să putem determina cât timp a trecut.

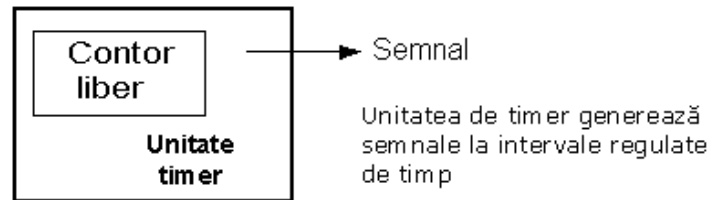


Fig. 6. Unitatea timer

2.7. Watchdog-ul

Watchdog-ul este de fapt un alt contor liber (free-run) unde programul nostru trebuie să scrie un zero ori de câte ori se execută corect. În caz că programul se „blochează”, nu se va mai scrie zero, iar contorul va reseta singur microcontrolerul la atingerea valorii sale maxime.

Această componentă este importantă pentru că asigură fiabilitate mare a programului fără supravegherea omului.

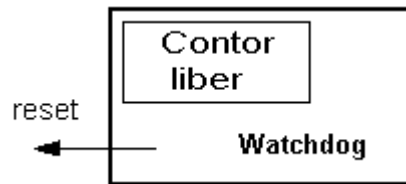


Fig. 7. Watchdog-ul

2.8. Convertorul Analog-Digital

Pentru că semnalele de la periferice (senzori, traductoare) sunt în general semnale analogice, substanțial diferite de cele pe care le poate înțelege microcontrolerul semnale numerice (zero și unu), ele trebuie convertite.

Această conversie este asigurată de convertor analog-numeric.

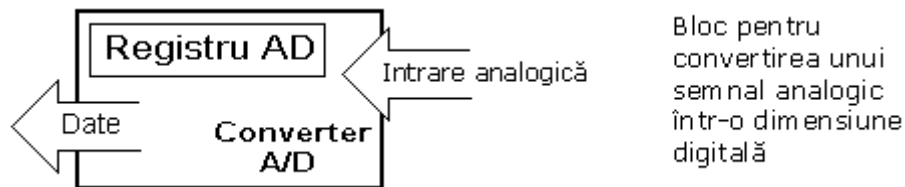


Fig. 8. Convertorul Analog-Digital

2.9. Comunicația serială

Interfața de comunicație serială este un periferic deosebit de util pentru realizarea conectării microcontrolerului cu alte echipamente (de exemplu un alt microcontroler sau un calculator).

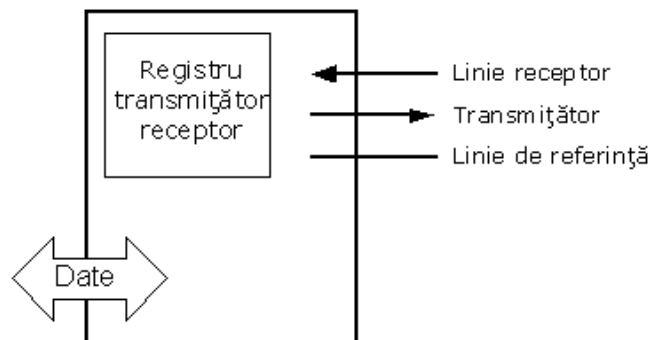


Fig. 9. Unitatea serială folosită pentru a trimite date, dar numai prin trei linii

Schema următoare reprezintă secțiunea centrală a microcontrolerului.

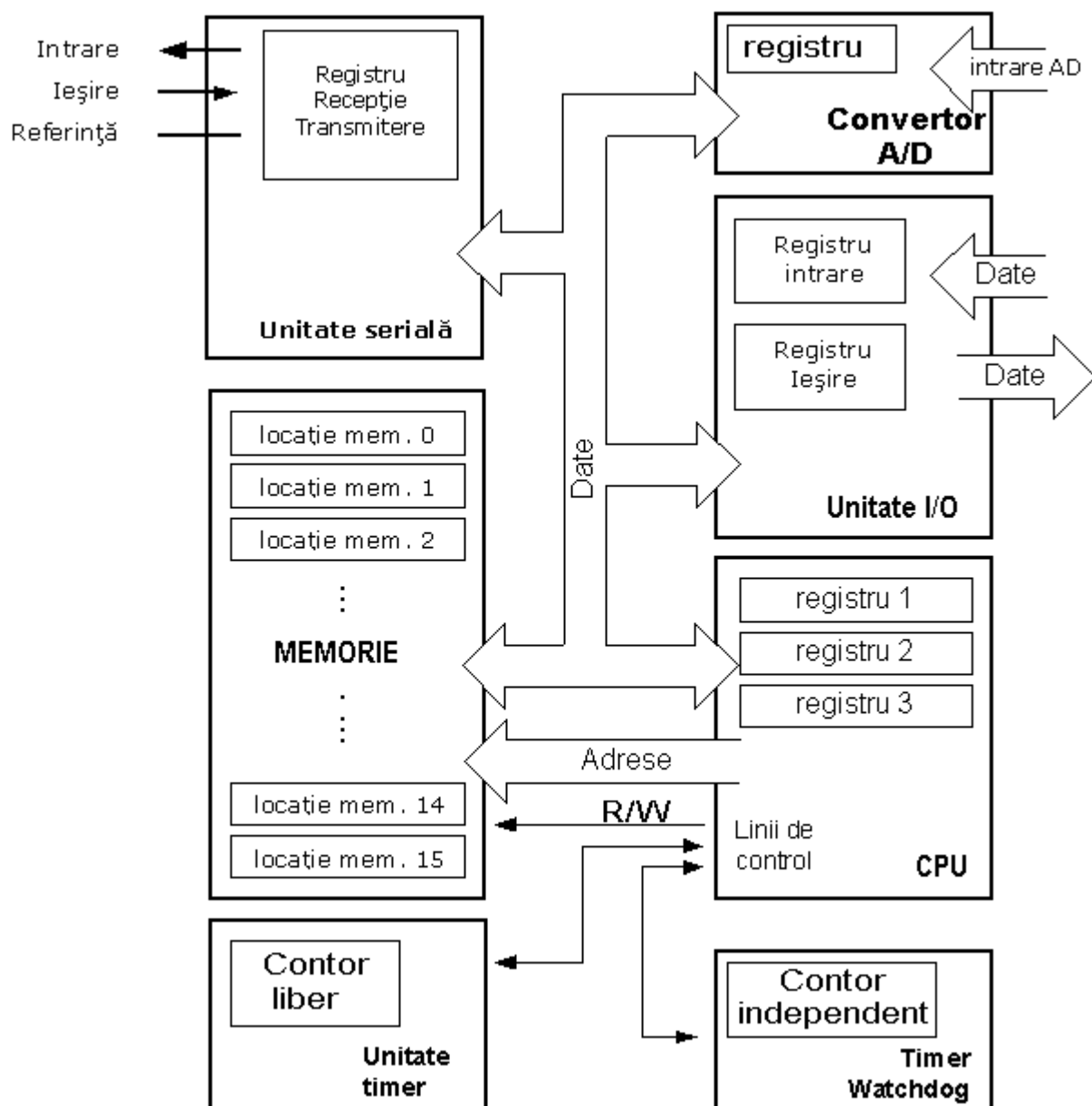


Fig. 10. Elementele de bază și conexiunile interne ale microcontrolerului

Ceea ce deosebește fundamental un microcontroler de un circuit integrat analogic sau digital, este faptul că el nu poate face nimic dacă nu este programat. Programul software conferă microcontrolerului, abilitatea de a realiza funcții diferite cu aceeași configurație hardware.

Scrierea programului se realizează de obicei într-un editor ce permite salvarea liniilor de comandă introduse.

Există mai multe opțiuni pentru scrierea programului de control al aplicației și anume:

- cod mașină (cod hexadecimale);
- limbaj de asamblare;
- limbaj de nivel înalt (C, Pascal, Basic etc).

Comenzile recunoscute de microcontroler sunt cele scrise în cod mașină. Limbajul de asamblare și limbajele de nivel înalt sunt mai evoluat, conțin instrucțiuni ce sunt ușor de reținut, dar pentru transformarea acestora în cod mașină avem nevoie de un compilator.

Compilatorul este program software, de obicei oferit gratuit de producătorii microcontrolerelor. Pentru a transfera codul hexadecimale rezultat în urma compilării, în memoria program a microcontrolerului este nevoie de un programator. Programatorul este compus dintr-un modul electronic care asigură interfațarea între aplicația ce conține microcontrolerul și calculator (PC), și un program software ce rulează pe PC.

3. Microcontrolerul PIC 16F690

Microcontrolere PIC proiectate de Microchip Tehnology sunt o alegere bună pentru începători. **PIC16F690** are arhitectură Harvard și aparține unei clase de microcontrolere de 8 biți tip RISC.

PIC16F690 este un microcontroler RISC (Reduced Instruction Set Computer), aceasta înseamnă că are un set redus de instrucțiuni, mai precis 35 de instrucțiuni (de ex. microcontrolerele INTEL și Motorola au peste 100 de instrucțiuni).

Tabel 1. Caracteristicile Microcontrolerului PIC16F690

Tipul caracteristicii	Valoare
Arhitectură	Harvard
Mărimea magistralei de date	8 Biți
Memorie EEPROM	256Bytes
Memorie SRAM	256 Bytes
Dimensiune program memorie	7 kB
Tip memorie program	Flash
Frecvența maximă	20 MHz
Număr pini	20
Număr de intrări/ieșiri	18
Număr de timere	3
Tensiune normală de funcționare	5 V
Tensiune minimă de funcționare	4,5 V

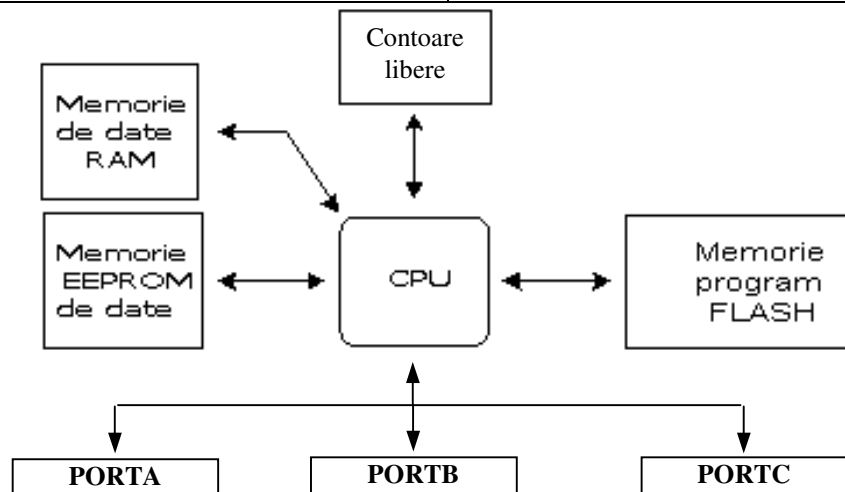


Fig. 11. Schița microcontrolerului PIC16F690

Memoria program (FLASH) – este utilizată pentru memorarea unui program scris. Pentru că memoria este făcută în tehnologia FLASH poate fi programată și ștearsă mai mult decât odată, aceasta face microcontrolerul potrivit pentru dezvoltarea de serie.

EEPROM - memorie de date folosită pentru memorarea de date importante ce nu trebuie pierdute dacă sursa de alimentare se întrerupe dintr-o dată.

RAM - memorie de date folosită de un program în timpul executării sale. În RAM sunt memorate toate rezultatele intermediare sau datele temporare ce nu sunt cruciale și care se pot pierde la întreruperea sursei de alimentare.

PORTUL A, PORTUL B și PORTUL C sunt conexiuni fizice (pini) între microcontroler și exterior. Portul A are 6 pini, portul B are 4 pini, iar portul C are 8 pini.

TIMER-LE LIBERE (FREE-RUN) sunt contoare libere de 8 biți în interiorul microcontrolerului ce lucrează independent de program. La fiecare al patrulea impuls de ceas al oscilatorului își încreează valoarea lui până ce atinge maximum (255), și apoi începe să numere tot din nou de la zero. După cum știm timpul exact dintre fiecare două incrementări ale conținutului timer-ului, poate fi folosit pentru măsurarea timpului ce este foarte util la unele componente.

UNITATEA DE PROCESARE CENTRALĂ (CPU) are rolul unui element de conectivitate între celelalte blocuri ale microcontrolerului. Coordonează activitatea acestor blocuri și execută programul utilizatorului.

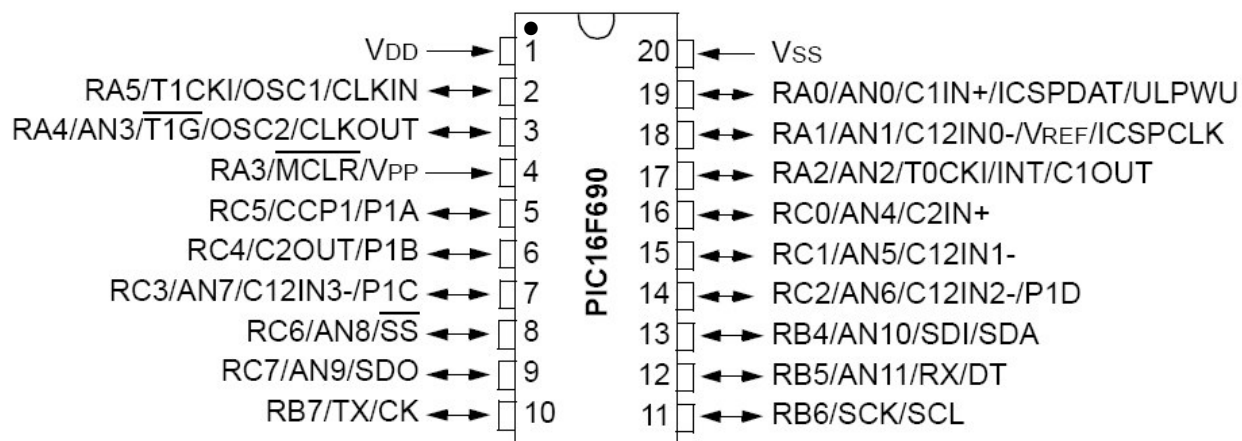


Fig. 12. Pini microcontrolerului PIC16F690

Pin nr.1 VDD - polul pozitiv al sursei.

Pin nr.20 VSS – masa (polul negativ al sursei).

Pin nr. 17 RA2/AN2/T0CKI/INT/C1OUT (RA2 – al doilea pin intrare/ieșire la portul A, AN2 – canal 2 de intrare pentru conversia analog digitală, T0CKI – timer 0, INT – întrerupere din exterior, C1OUT – comparator C1).

RA0÷RA5 – cei 6 pini ai portului A.

RB4÷RB7 – cei 4 pini ai portului B.

RC0÷RC7 – cei 8 pini ai portului C.

AN0÷AN11 – canale de intrare pentru conversia analog digitală.

4. Microcontrolerul PIC 16F877

PIC16F84 aparține unei clase de microcontrolere de 8 biți cu arhitectura RISC (35 de instrucțiuni). Microcontrolerul are 5 porturi: A, B, C, D, E

Tabel 2. Caracteristicile Microcontrolerului PIC16F877

Tipul caracteristicii	Valoare
Arhitectură	Harvard
Mărimea magistralei de date	8 Biti
Memorie EEPROM	256Bytes
Memorie SRAM	368 Bytes
Dimensiune program memorie	8 kB
Tip memorie program	Flash
Frecvența maximă	20 MHz
Număr pini	40
Număr de intrări/ieșiri	33
Număr de timere	3
Tensiune normală de funcționare	5 V

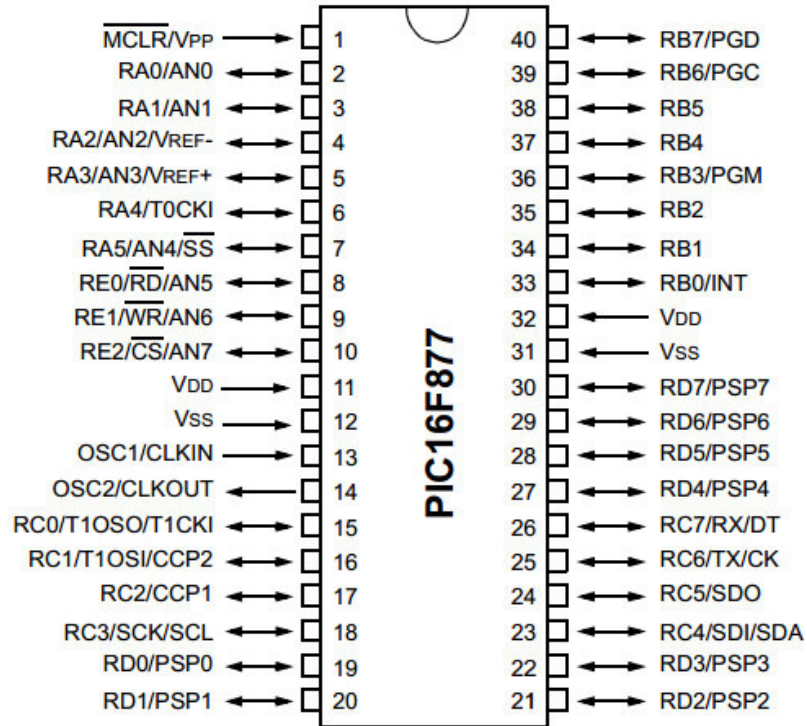


Fig. 13. Pini microcontrolerului PIC16F877

RA0÷RA5 – cei 6 pini ai portului A.

RB0÷RB7 – cei 8 pini ai portului B.

RC0÷RC7 – cei 8 pini ai portului C.

RD0÷RD7 – cei 8 pini ai portului D.

RE0÷RE2 – cei 3 pini ai portului E.

