oscilator (Pierce) se obține prin conectarea între pinii XTAL1 și XTAL2 a unui cristal de cuarț sau a unui rezonator ceramic (plus două condensatoare conectate de la cei doi pini la masă). O altă variantă constă în utilizarea unui oscilator extern conectat la pinul XTAL2 pentru varianta de microcontroler NMOS, respectiv la pinul XTAL1 pentru varianta CMOS.

Funcționarea sincronizată a microcontrolerului pentru execuția instrucțiunilor se desfășoară în faze, stări și cicluri mașină, figura 8.9. O fază a microcontrolerului corespunde duratei $T_o = 1/f_o$ a perioadei oscilatorului de tact. Două faze succesive, notate cu P1, P2, definesc o stare a microcontrolerului. În general, operațiile aritmetice și logice se efectuează în fazele P1 ale stărilor, iar transferurile între registrele interne se efectuează în fazele P2 ale stărilor. Un ciclu mașină se desfășoară pe durata a 12 perioade T_o (6 stări) notate S1P1, S1P2, S2P1, ... , S6P2, figura 8.9. Microcontrolerul 8051 execută o instrucțiune în 1 sau 2 cicluri mașină, funcție de tipul instrucțiunii, cu excepția instrucțiunilor de multiplicare *MUL* și divizare *DIV* care se execută în 4 cicluri mașină.

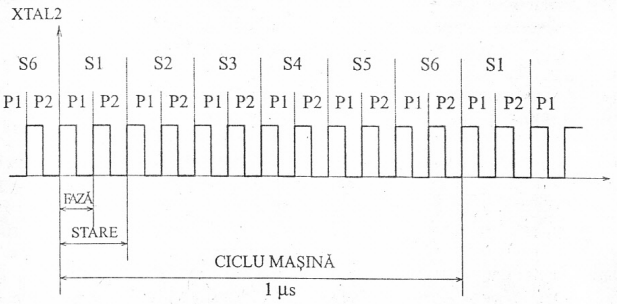


Fig.8.9. Diagrama de timp a semnalului de tact XTAL2.

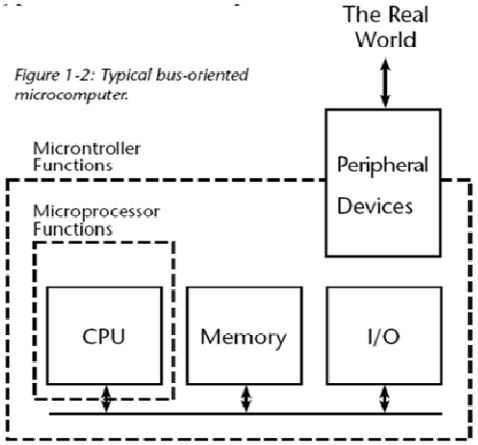
Execuția unei instrucțiuni începe în faza S1P1 a unui ciclu mașină, prin extragerea din memoria de program a primului octet al instrucțiunii și transferul acestuia în registrul de instrucțiuni al microcontrolerului, în faza S1P2. În cazul unei instrucțiuni cu codul mașină de 2 sau 3 octeți, al doilea octet se extrage din memoria de program în starea S4 a aceluiași ciclu mașină. Rezultă că, în funcție de numărul de cuvinte din codul mașină al unei instrucțiuni și în funcție de numărul de cicluri mașină în care se execută, într-un ciclu mașină se pot efectua două extrageri coduri instrucțiuni, o extragere sau nici una, conform exemplelor de mai jos:

INC A	cod mașină	: 1 octet
	execuție	: 1 ciclu mașină
	S1 : extragere cod operație	(PC) + 1 → (PC)
	S4 : - - -	(PC) → (PC)

2. Structura tipică a unui microcontroler. Definiți arhitectura Harward şi von Neumann. Scheme, avantaje şi dezavantaje.

[1], slide nr. 30, 31; [2], pag. 11.

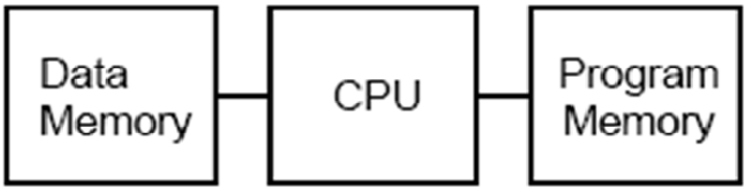
Arhitectura tipica de microcontroler



Microcontrolerul (uC) este un microcalculator pe un singur chip utilizat pentru comanda altor dispozitive si circuite electronice.

Un microcontroler include pe chip:
- **Unitatea centrala de procesare (CPU),**
- **Memorii,**
- **Interfete de intrare/iesire,**
- **Dispozitive periferice integrate (interne),**
- **Convertoare A/D si D/A.**

Microcontroler cu arhitectura HARWARD



Avantaje arhitecturii HARWARD:
- **Viteza de executie mai ridicata,**
- **Siguranta sporita in functionare**

Observatie: de regula, cele doua memorii sunt conectate la CPU printr-o singura magistrala.

Viteza de execuţie a instrucţiunilor este funcţie de frecvenţa semnalului de la generatorul de tact al SPN. În general, o operaţie de bază se efectuează pe durata a unei perioade sau a mai multor perioade ale semnalului de tact. Intervalul corespunzător efectuării unei operaţii de bază se numeşte un ciclu maşină al SPN. Ciclurile maşină corespunzătoare unei instrucţiuni definesc un ciclu instrucţiune. Execuţia unei instrucţiuni durează câteva cicluri maşină incluzând: ciclu de extragere cod operaţie, ciclu de decodificare şi, apoi, funcţie de tipul instrucţiunii, cicluri de citire/scriere, cicluri de intrare/ieşire şi cicluri de execuţie. Efectuarea acestor cicluri poate implica componente diferite din structura SPN. Rezultă posibilitatea ca SPN să efectueze simultan cicluri diferite din execuţia unei instrucţiuni sau din execuţia unor instrucţiuni succesive. Această tehnică de suprapunere în timp a execuţiei ciclurilor maşină (operare paralelă) se numeşte tehnică pipeline şi utilizarea ei în funcţionarea unui SPN conduce la mărirea vitezei de lucru a acestuia.

În general, realizarea unui SPN se bazează pe utilizarea unui circuit integrat pe scară largă de tip microprocesor, microcontroler sau procesor numeric de semnal, care conţine, pentru orice tip, o unitate centrală de prelucrare UCP cu următoarele componente din structura generală a unui SPN: unitate aritmetică şi logică, registru cu indicatorii de condiţii, registru numărător de adrese, registru de instrucţiuni, circuite de decodificare a instrucţiunilor, unitate de control şi sincronizare şi registre cu funcţii dedicate. Toate aceste componente se numesc interne, relativ la circuitul integrat utilizat ca bază pentru realizarea SPN. În acest sens, registrele din structura SPN se numesc registre interne. Un circuit de tip microcontroler conţine toate componentele din structura generală a unui SPN, figura 1.1, incluzând memorie internă, porturi de I/E, precum şi alte periferice.

1.3. FUNCŢIONAREA UNUI SPN PENTRU EXECUŢIA INSTRUCŢIUNILOR DE TRANSFER AL CONTROLULUI

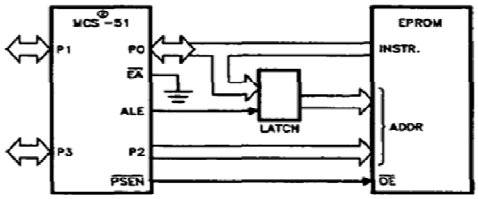
Transferul controlului constă în efectuarea unui salt în procesul de extragere a codurilor instrucţiunilor dintr-un spaţiu de adresare al memoriei de program în altul. Rezultă că transferul controlului se realizează prin modificarea conţinutului registrului numărător de adrese PC. Instrucţiunile de transfer al controlului sunt instrucţiunile de salt, apel de subrutine şi revenire din subrutine, prin execuţia cărora se realizează, în principal, încărcarea registrului PC cu adresa de salt, de început a subrutinei şi, respectiv, de revenire din subrutină. În cazul instrucţiunilor de salt şi de apel de subrutine, aceste adrese sunt indicate prin codurile maşină ale instrucţiunilor.

Subrutina este definită de programator ca un grup de instrucţiuni care realizează o funcţie dată, funcţie necesară a fi executată de mai multe ori în cadrul unui program. Execuţia grupului de instrucţiuni ca subrutină permite scrierea codurilor maşină corespunzătoare o singură dată în memoria de program. Astfel, programul

3. Conectarea unei memorii program externe. Schema de conectare, semnale utilizate.

[1], slide nr. 46; [4], pag. 135.

Conectarea unei memorii program externe



PSEN→ Program Store Enable
P0 → Adrese low (A7-A0) multiplexate in timp cu date (D7-D0)
ALE→ Address Latch Enable
P2 → Adrese high (A15-A8)
Memorii program externe sunt rar utilizate

iuni, continutul
termină în faza

ırea memoriilor
i la tehnicile de

ıstrucțiuni, prin
(A) + (DPTR)
tă se realizează
n conectarea la
ogic 1, accesul
h ÷ FFFFh. În
internă.
rin adresare cu
ctiunile MOVX
semnificativ al
i.
ram sau de date
tin semnificativ
semnificativ al
riei externe se
plificatorul de
adresare cât și
are în timp în-
a octetului mai
portului P0 în
ținerea adresei
e memorare în
ALE, generat

rogram externă
16 biti, rezultă

necesitatea selecției (validării) separate a celor două tipuri de memorii. În acest scop, microcontrolerul generează semnalul /PSEN de validare a memoriei de program externă și semnalele /RD și /WR de validare citire/scriere din/în memoria de date externă, semnale generate la pinii corespunzători, figura 8.1.

Conectarea unor memorii externe de date și program cu capacitate maximă, de câte 64 Kocteți, este prezentată în figura 8.10. Registrul pentru memorarea octetului mai putin semnificativ al adresei este comandat la intrarea STB cu semnalul ALE generat de microcontroler. Registrul funcționează ca amplificator de magistrală repetor dacă intrarea de control STB este la nivel logic 1, memorează starea liniilor magistralei de intrare din momentul frontului de coborâre de la intrarea STB și încarcă magistrala de ieșire cu această stare dacă intrarea STB este la nivel logic 0. Funcționarea circuitelor din figura 8.10 în cicluri de extragere coduri instrucțiuni din memoria de program externă, citire din memoria de date externă și scriere în memoria de date externă este prezentată, pe bază de diagrame de timp, în figurile 8.11, 8.12 și respectiv 8.13. Se precizează că diagramele de timp din figura 8.11 sunt caracteristice funcționării microcontrolerului în sensul că semnalele corespunzătoare se generează tot timpul, cu excepția ciclurilor de acces la memoria de date externă și la memoria de program internă. În acest ultim caz, linia /EA a microcontrolerului se comandă cu nivel logic 1. Astfel, memoria de program internă este accesată în spațiul de adresare 0 ÷ 0FFFFh și în intervalele de acces la memoria de program internă linia /PSEN este comandată de microcontroler în stare inactivă. Incrementarea conținutului numărătorului de adrese, citirea sau ignorarea cuvintelor coduri instrucțiuni în diferite faze ale ciclurilor mașină sunt functie de instrucțiunile executate de microcontroler, conform exemplelor de instrucțiuni prezentate în acest paragraf.

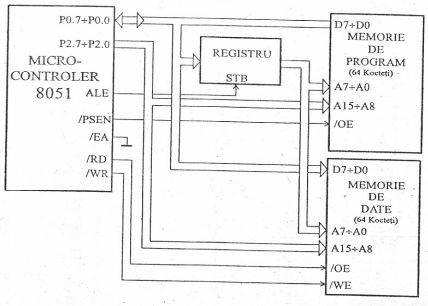


Fig.8.10. Conectarea memoriilor externe de date și program la un microcontroler 8051.

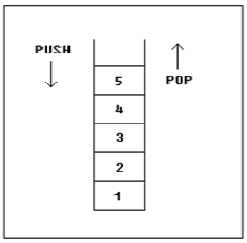
4. Memoria stivă. Definiție, funcționare, tipuri de memorie stivă, inițializare.

[1], slide nr. 59, 60; [2], pag. 13.

Memoria stiva (1)

Este o zona de memorie amplasata in memoria interna RAM si care stocheaza temporar urmatoarele tipuri de date:

- **Automat, adresele de revenire din subrutine (de tratare a intreruperilor sau subrutine apelate prin instructiuni CALL),**
- **Prin program, continutul unor registri utilizati in subrutine si care trebuie recontuit inainte de revenirea in progeamul principal,**
- **Datele sunt manipulate cu instructiunile PUSH (incarca date in stiva) si POP (extrage date din stiva).**



- **Stiva este o memorie LIFO (last in, first out – "ultimul intrat primul iesit",**
- **Stiva poate creste "in sus" ca in figura sau "in jos",**
- **Adresele de scriere/citire sunt date de registrul Stack Pointer (SP); continutul acestuia indica adresa ultimei locatii ocupata din stiva.**

Memoria stiva (2)

- **Stabilirea zonei din RAM intern alocata stivei se face prin initializarea continutului registrului SP.**
- **Exixta posibilitatea prevenirii citirii/scrierii inafara limitelor memoriei stiva, utilizind 2 registri: Stack Overflow respectiv Stack Underflow, care contin adresele limita ale stivei. La atingerea lor sunt generate intreruperi.**

Exemple de utilizare a memoriei stiva

```
CSEG AT 23H              SBRT:  PUSH A
       PUSH A                   PUSH PSW
       PUSH PSW                 ----------
       CALL SBRT                ----------
       POP PSW                  POP PSW
       POP A                    POP A
       RETI                     RET
```