

MICROCONTROLERUL 80C32

INTRODUCERE

Microcontrolerul 80C32 face parte din familia de microcontrolere MCS-51 a firmei INTEL (S.U.A). Este un microcontroler de uz general pe 8 biți. Frecvența de ceas admisibilă este cuprinsă între 1,2 și 12 MHz.

Folosit cu profesionalism, acest microcontroler oferă multe satisfacții în utilizare, constituind „inima” multor echipamente din domeniul bunurilor de larg consum sau al aparatelor industriale.

Sunt prezentate în lucrare atât particularitățile hardware ale microprocesorului 80C32, cât și instrucțiunile software ale sale.

Circuitele integrate care fac parte din familia de microcontrolere MCS-51 sunt prezentate în tabelul 1.1.

Circuitul integrat	Tehnologia de realizare	Memoria Program (PROM) intern	Memoria de date (RAM) intern(octeți)	<i>Timers/ Counters</i>	Înteruperi
8051	HMOS	4K-ROM	128	2x16 bit	5
8031	HMOS	NU ARE	128	2x16 bit	5
8051H	HMOS	4K-EPROM	128	2x16 bit	5
80C51	CHMOS	4K-ROM	128	2x16 bit	5
80C31	CHMOS	NU ARE	128	2x16 bit	5
80C52	HMOS	8K-ROM	256	3x16 bit	6
80C32	HMOS	NU ARE	256	3x16 bit	6

Microcontrolerul 80C32 se distinge de restul membrilor familiei MCS-51 prin faptul că nu are memorie internă (PROM), dar ca și 80C32 are 256 octeți RAM intern.

Sunt dotate în plus cu un *timer/counter* de 16 biți. Noul timer poate fi folosit ca *timer/counter* sau ca generator de rată a transferului de date (*band rate*) pentru portul serial.

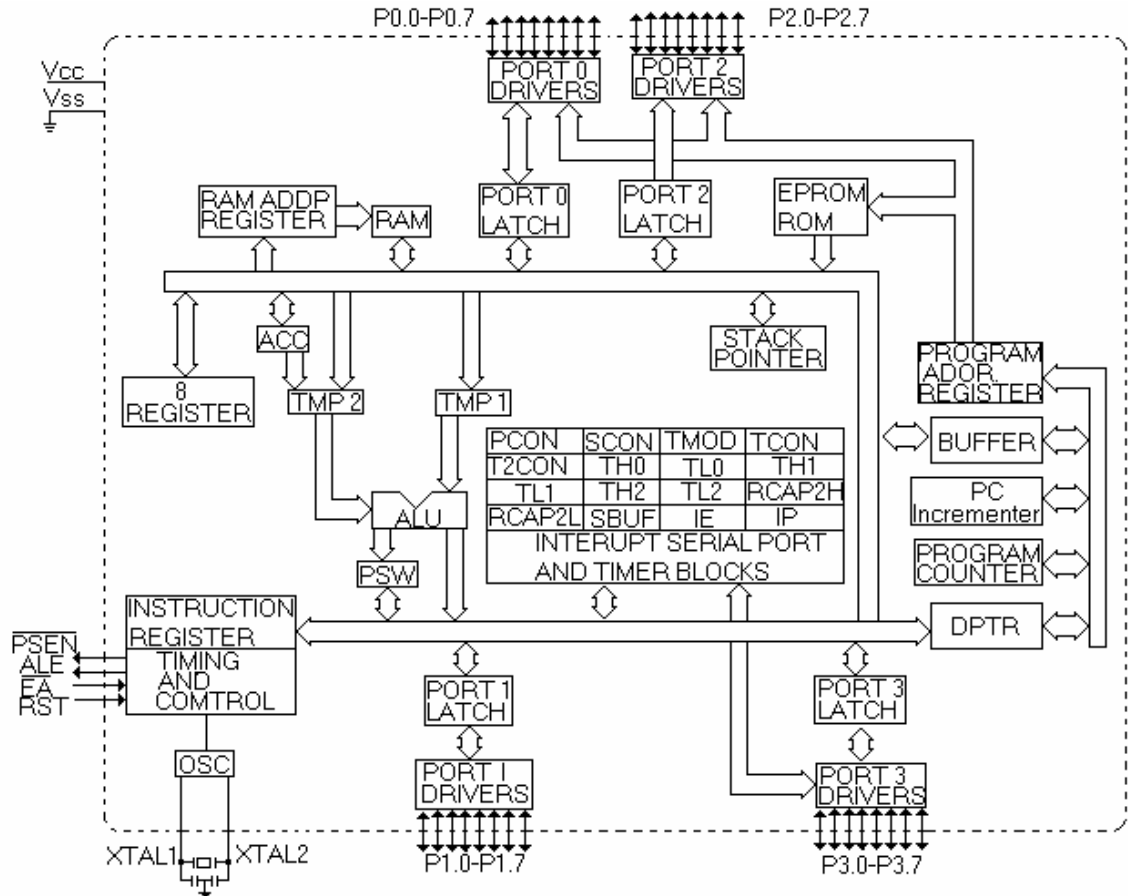
Caracteristicile principale ale familiei de microcontrolere MCS-51 sunt următoarele:

- unitatea centrală de prelucrare CPU pe 8 biți;
- oscilator și ceas intern (*on-chip*);
- 32 de linii de intrare/ieșire;
- 64 KO spațiu de adresare pentru memoria program externă;
- 64 KO spațiu de adresare pentru memoria de date externă;
- 2 temporizator/numărătoare (*Timer/Counters*) pe 16 biți;
- structură cu 5 surse de întrerupere cu două nivele de prioritate;
- porturi seriale bidireționale,
- procesor boolean.

1.DESCRIEREA HARDWARE A MICROCONTROLLERULUI

80C32

Schema bloc a microcontrolerului 80C32 este prezentată în figura 1.1,a.



1.1. CONFIGURAȚIA PINILOR

În figura 1.1,b este prezentată configurația pinilor pentru microcontrolerul 80C32.

31	EA/VP	P0.0	39
19	X1	P0.1	38
18	X2	P0.2	37
		P0.3	36
9	RESET	P0.4	35
		P0.5	34
		P0.6	33
		P0.7	32
12	INT0	P2.0	21
13	INT1	P2.1	22
14	T0	P2.2	23
15	T1	P2.3	24
		P2.4	25
1	P1.0/T2	P2.5	26
2	P1.1/T2X	P2.6	27
3	P1.2	P2.7	28
4	P1.3	RD	17
5	P1.4	WR	16
6	P1.5	PSEN	15
7	P1.6	ALE/P	29
8	P1.7	TXD	11
	80C32	RXD	10

pin 20-Vss

pin 40-Vcc

Figura 1.1,b Configurația pinilor pentru microcontrolerul 80C32

1.2. SEMNIFICAȚIA PINILOR

Vcc-Tensiunea de alimentare al circuitului
Vss-Pinul de masă al circuitului

PORT 0 este port pe 8 biți direcțional intrare/ieșire, drenă în gol (*open drain*). Este port de date și adrese (scoate adresa și apoi încarcă datele de pe magistrală).

Dacă pinii portului 0 sunt în „1” logic, ei pot funcționa ca ieșire cu valoarea 1 sau ca intrare cu nivel controlat de un circuit extern și în această stare funcționează ca intrare de înaltă impedanță. Ca port de ieșire fiecare pin al portului 0 poate suporta până la 8 intrări LS TTL.

De asemenea, portul 0 primește octeții de cod în timpul programării EPROM-ului și generează octeții de cod în timpul verificării programului din EPROM.

PORT 1

Portul 1 este port pe 8 biți bidirecțional intrare/ieșire, drenă în gol (*open drain*). Registrele tampon (*buffer*) de ieșire ale portului 1 suportă până la 4 intrări LS TTL.

Când pinii portului 1 sunt în starea „1” logic (*high*) pot fi folosiți ca intrări. Ca intrări, pinii portului 1 care au fost puși pe *low* din exterior vor furniza curent datorită *pull-up*-ului intern.

De asemenea, portul 1 va primi în timpul programării EPROM-ului și în timpul verificării programului octeții cei mai puțin semnificativi din adresă.

Pinii P1.0 și P1.1 ai portului 1 au funcții alternative, ei putând fi folosiți și ca pini ai portului și ca pini cu funcții speciale. Astfel T2 (pinul P1.0) este intrare externă în *Timer*-ul 2, iar T2X (pinul P1.1) este intrarea prin care o „captură” a *Timer*-ului 2 este declanșată.

PORT 2

Portul 2 este port pe 8 biți bidirecțional intrare/ieșire. *Buffer*-ele de ieșire ale portului 2 pot suporta până la 4 intrări LS TTL.

În timpul accesărilor memoriei de date externă care folosesc adrese pe 16 biți, sau al extragerilor din memoria program externă, portul 2 generează octetul celor mai semnificativi biți de adresă. În timpul accesărilor memoriei de date externă, care folosesc adrese pe 8 biți, portul 2 emite conținutul registrului cu funcții speciale P2.

De asemenea, portul 2 primește octetul celor mai semnificativi biți de adresă în timpul programării EPROM-ului și al verificării programului din EPROM.

PORT 3

Portul 3 este port pe 8 biți bidirecțional intrare/ieșire.

Buffer-ele de ieșire ale portului 3 pot suporta până la 4 intrări LS TTL.

Pinii portului 3 aflați în starea „1” logic (*high*) pot fi folosiți ca intrări.

Ca pini de intrare, dacă au fost puși pe *low* din exterior, pinii portului 3 furnizează curent.

Pinii portului 3 pot fi folosiți pentru funcțiile speciale ale microcontrolerului 80C32 astfel:

Tabelul 1.1

Pinul portului	Funcția alternativă
P3.0	RXD-intrare serială a portului pentru recepție
P3.1	TXD-ieșire serială a portului pentru transmisie
P3.2	$\overline{\text{INT0}}$ -întrerupere externă pentru Timer 0
P3.3	INT1- întrerupere externă pentru Timer 0
P3.4	T0-intrare externă în Timer 0
P3.5	$\overline{\text{T1}}$ - intrare externă în Timer 1
P3.6	$\overline{\text{WR}}$ -strob-ul pentru scrierea memoriei de date externă
P3.7	RD- strob-ul pentru citirea memoriei de date externă

Notă: Din punctul de vedere al funcționării electrice funcția negată este activă când se află la potențialul zero volți (*low*).

RST

Intrarea de reset. Dacă pinul rămâne în *high* pe perioada a 2 cicli mașină în timp ce oscilatorul funcționează, microcontrolerul este resetat.

ALE/PROG

ALE (*Address Latch Enable*) furnizează impulsul de ieșire pentru memorarea octetului celor mai puțin semnificativi biți de adresă în timpul accesului la memoria externă.

ALE poate conduce până la 8 intrări LS TTL. În operații normale, ALE este generat de o rată constantă egală cu 1/6 din frecvența oscilatorului și poate fi folosit pentru temporizare sau ceas extern.

De reținut însă că un impuls ALE este omis în timpul fiecărui acces la memoria de date externă. Acest pin este de asemenea pin de intrare pentru impulsul de programare în timpul programării EPROM-ului.

$\overline{\text{PSEN}}$

Program Store Enable este semnalul folosit pentru a valida citirea datelor din memoria program externă. Dacă microcontrolerul accesează memoria program

externă, PSEN este activat de două ori în timpul unui ciclu mașină, cu o excepție: în timpul activării memoriei de date externe.

EA/VP

External Access enable ($\overline{\text{EA}}$). Dacă EA este în starea *high*, CPU ar trebui să folosească memoria program internă, dar având în vedere faptul că microcontrolerul 80C32 nu are memorie program internă, pinul acesta se leagă la masă.

Astfel, EA ținut pe *low* forțează unitatea centrală de prelucrare (CPU) să folosească memoria program externă.

În timpul programării EPROM-ului pinul acesta va primi tensiunea de alimentare de 21V (-VP).

XTAL(X1)

Intrare la inversorul oscilatorului.

XTAL(X2)

Ieșire de la inversorul oscilatorului.

1.3 ORGANIZAREA MEMORIEI

Microcontrolerul 80C32 are spațiile de adresare separate pentru memoria program și pentru memoria de date.

Memoria program are până la 64 KO EPROM de la adresa 0000H la adresa FFFFH.

Microcontrolerul nu dispune de memorie program internă.

Memoria de date are până la 64 KO RAM extern la care se mai adaugă 256 de octeți de RAM intern și un număr de registre cu funcții speciale SFR (*Special Function Registers*) tot interne.

Organizarea memoriei de date este ilustrată în **figura 1.2**.

Spațiul de RAM adresabil doar indirect se află între adresele 80H și FFH și are 128 de octeți.

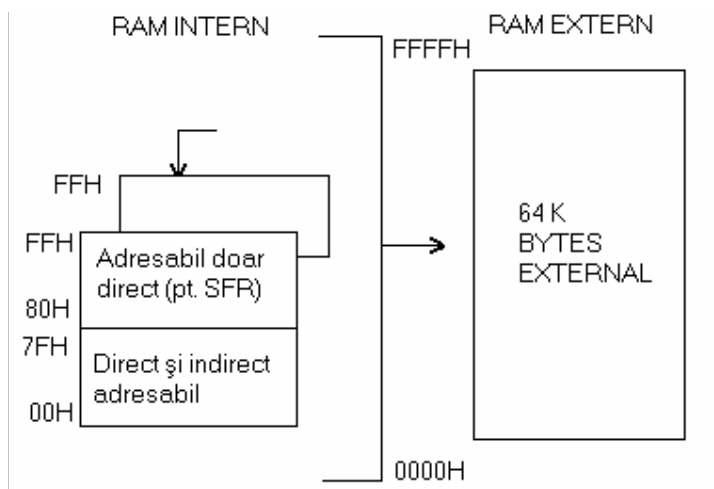


Figura 1.2 Organizarea memoriei de date internă și externă

Se observă că în **figura 1.2** că SFR și segmentul de RAM intern adresabil indirect, au aceleași adrese de la 80H la FFH, dar se impune precizarea că sunt totuși două spații de adresare diferite și sunt accesate în mod diferit.

EXEMPLE: 1) instrucțiunea: MOV 80H,#0AAH

Scrie valoarea 0AAH în portul 0 care este un SFR

2) secvența de instrucțiuni: MOV R0,#80H

MOV @R0,#0BBH

Scrie valoarea 0BBH în locația 80H din memoria de date RAM

Astfel, după executarea ambelor instrucțiuni, portul 0 va conține 0AAH, iar locația 80H din RAM va conține 0BBH.

Spațiul de RAM adresabil direct se află între adresele 00H și 7FH și cei 128 octeți ai săi sunt împărțiți în trei fragmente astfel:

1. Bancurile de registre 0-3 (*Register Banks* 0-3): adresabile de la adresa 00H la adresa 1FH (32 octeți), fiecare dintre aceste bancuri are câte 8 registre (de opt biți fiecare). La reset, se inițializează indicatorul de stivă SP (*Stack Pointer*) la locația 07H și este incrementat o dată pentru a începe de la locația 08H, care este adresa primului registru din al doilea banc de registre. Această operație este necesară deoarece bancul de registre 1 este folosit de stivă.
2. Spațiul adresabil pe bit (*Bit addressable area*): adresabil de la 20H la 2FH (16 octeți), fiecare din cei 128 de biți ai acestui segment putând fi adresat direct. Pentru acest segment biții pot fi adresați în două moduri diferite:
 - Un mod de adresare este prin referire la adresele lor, de la 0H la 7FH;
 - Al doilea mod de adresare este cu referire la octeții între 20H și 2FH. Astfel biții 0..7 pot fi referiți ca biții 20.0-20.7, biții 8..16 pot fi referiți ca biții 21.0-21.7 și așa mai departe. Oricare dintre cei 16 octeți din acest segment poate fi adresat ca octet.
3. Spațiul de adresare pe octet (*Scratch Pad Area*): adresabil de la 30H la 7FH și disponibil pentru utilizator ca memorie de date RAM. Oricum, dacă indicatorul de stivă (*stack pointer*) este inițializat în această zonă, este necesar să se lase un număr suficient de biți de o parte pentru a preveni distrugerea indicatorului de stivă.

Tabelul 1.2

Registrele cu funcții speciale

Simbol	Denumire	Adrese
*ACC	Accumulator	0E0H
*B	B Register	0F0H
*PSW	Program Status Word	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer 2 bytes	
DPL	Low Byte	82H
DPH	High Byte	83H
*P0	Port 0	80H
*P1	Port1	90H
*P2	Port 2	0A0H
*P3	Port 3	0B0H
*IP	Interrupt Priority Control	0B8H
*IE	Interrupt Enable Control	0A8H
TMOD	Timer/Counter Mode Control	89H
*TCON	Timer/Counter Control	88H
*T2CON	Timer/Counter 2 Control	0C8H
TH0	Timer/Counter 0 High Byte	8CH
TL0	Timer/Counter 0 Low Byte	8AH
TH1	Timer/Counter 1 High Byte	8DH
TL1	Timer/Counter 1 Low Byte	8BH
TH2	Timer/Counter 2 High Byte	0CDH
TL2	Timer/Counter 2 Low Byte	0CCH
RCAP2H	T/C 2 Capture Reg. High Byte	0CBH
RCAP2L	T/C 2 Capture Reg. Low Byte	0CAH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	Power Control	87H

În RAM-ul intern există și zona SFR-urilor, adică a registrelor cu funcții speciale, care are 128 de octeți adresabili după cum am arătat mai sus, între 80H și FFH. În această zonă sunt registrele cu funcții speciale. În **tabelul 1.2** sunt enumerate aceste registre cu adresele corespunzătoare, cu precizarea că registrele marcate cu „*” sunt registre adresabile și pe bit și pe octet.

Registrul acumulator (*Accumulator*) – în program este referit și cu A și cu ACC, în funcție de instrucțiune;

Registrul B (*B register*) – este folosit în timpul executării operațiilor de înmulțire și de împărțire;

PSW- Cuvânt de stare al programului (*Program status word*)- registru ce conține informații despre starea programului. Configurația acestui registru este prezentată în **figura 1.3**;

Cel mai semnificativ bit
MSB (*Most significant bit*)

Cel mai puțin semnificativ bit
LSB (*Less significant bit*)

CY	AC	F0	RS1	RS0	OV	-	P
----	----	----	-----	-----	----	---	---

Figura 1.3 Registrul cuvântului de stare a programului PSW (*Program Status Word Register*)

Registrul indicator de stivă SP (*Stack Pointer*)- este un registru pe 8 biți și este incrementat înaintea stocării datelor, în timpul instrucțiunilor PUSH și CALL;

Registrele P0, P1, P2, P3 (*Ports 0..3*)- sunt registrele cu funcții speciale (*SFR*) ale porturilor 0, 1, 2 și 3;

Registrul tampon (*buffer*) serial de date SBUF (*Serial data buffer*) – ce reprezintă de fapt două registre: un registru tampon (*buffer*) de transmisie și un registru tampon (*buffer*) de recepție. Când se pun date în SBUF ele merg în registrul tampon (*buffer*) de transmisie unde se păstrează pentru transmisia serială. Când se preiau date din SBUF ele provin din registrul tampon (*buffer*) de recepție.

Registre temporizator/numărătoare (*Timer/Counter registers*) – pereche (TH0, TL0), (TH1, TL1) și (TH2, TL2) sunt numărătoare pe 16 biți pentru temporizator/numărător (*Timer/Counters*) 0, 1 și 2;

Registrele pereche de captură (*Capture register*) – (RCAP2H, RCAP2L), pentru modul *captură* al *Timer*-ului 2. În acest mod de lucru, ca răspuns la o tranziție pe pinul T2X, conținutul registrelor TH2 și TL2 este copiat în RCAP2H, respectiv RCAP2L. Dacă se folosește *Timer 2* ca registru pe 16 biți în modul auto-reîncărcare (*auto-reload*), în RCAP2H și RCAP2L se pune valoarea de reîncărcare pentru acest mod de lucru.

Registrele de control (*Control registers*) – IP, IE, TMOD, TCON, T2CON, SCON și PCON conțin biții de control și stare pentru sistemul de întreruperi, pentru temporizator/numărătoare (*Timer/Counters*) și pentru portul serial.

Simbol	Poziție	Funcție
CY (Carry flag)	PSW.7	Indicator de transport
AC (Carry flag auxiliar)	PSW.6	Indicator de transport auxiliar. Este folosit pentru operații BCD (binar codat zecimal).
F0 (Flag 0)	PSW.5	Este disponibil utilizatorului pentru scopuri generale.
RS1	PSW.4	Biți de control al selecției bancurilor de registre (<i>Register bank</i>).Setați/ resetați prin software pentru a determina bancul de registre selectat, conform cu NOTA de mai jos.
RS0	PSW.3	
OV (Overflow flag)	PSW.2	Indicator de depășire.Bit indicator de depășire.
-	PSW.1	Bit rezervat.
P	PSW.0	Bit de paritate. Este setat/resetat prin hardware la fiecare ciclu instrucțiune pentru a indica un număr impar/par de biți cu valoare logică „1” în acumulator. Se folosește pentru verificarea corectitudinii informațiilor.

Notă: Valorile prezentate de RS0 și RS1 pentru selectarea bancului de registre corespunzător sunt următoarele:

RS1	RS0	Bancul de registre	Adresa
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

1.4 CIRCUITUL OSCILATOR ȘI DE CEAS

Pinii 19 – XTAL1 (X1) și 18 – XTAL2 (X2) ai microcontrolerului 80c32 sunt pinii de intrare, respectiv de ieșire, pentru ca un inversor care poate fi configurat pentru a fi folosit ca oscilator intern legând între cei doi pini două condensatoare și un cristal de cuarț.

Structura pentru un asemenea oscilator este prezentată în **figura 1.4**.

De asemenea, există posibilitatea folosirii unui circuit oscilator extern. În acest caz pinul XTAL2 se lasă neconectat, iar pinul XTAL1 se conectează la oscilatorul extern ca în **figura 1.5**.

În oricare dintre aceste cazuri (folosirea oscilatorului intern sau utilizarea unui oscilator extern) oscilatorul comandă generatorul de ceas intern. Generatorul de ceas intern furnizează *chip*-ului semnalele de ceas interne. Semnalele de ceas sunt la jumătate din frecvența oscilatorului și definesc fazele, stările și ciclul-mașină interni.

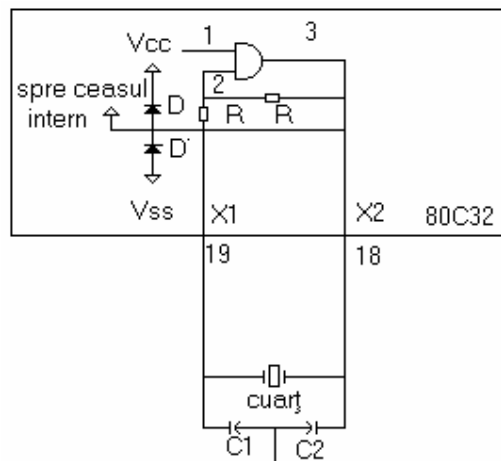


Figura 1.4 Modul de utilizare a oscilatorului intern

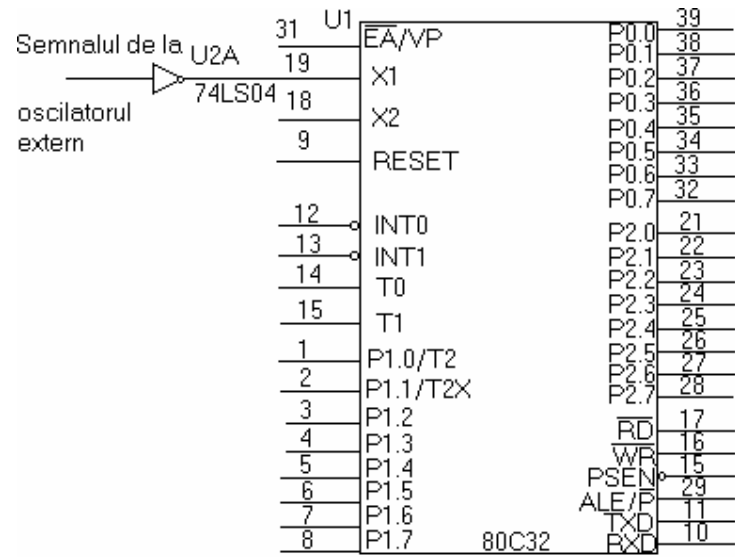


Figura 1.5 Conectarea microcontrolerului 80C32 la un oscilator extern

1.5 DIAGrameLE DE TIMP ALE FUNCȚIONĂRII CPU (CPU TIMING)

Un ciclu mașină constă din 6 stări (12 perioade de oscilator), fiecare stare fiind împărțită în două: prima jumătate (Phase 1) în care primul impuls de ceas este activ și a doua jumătate (Phase 2) în care al doilea impuls de ceas este activ.

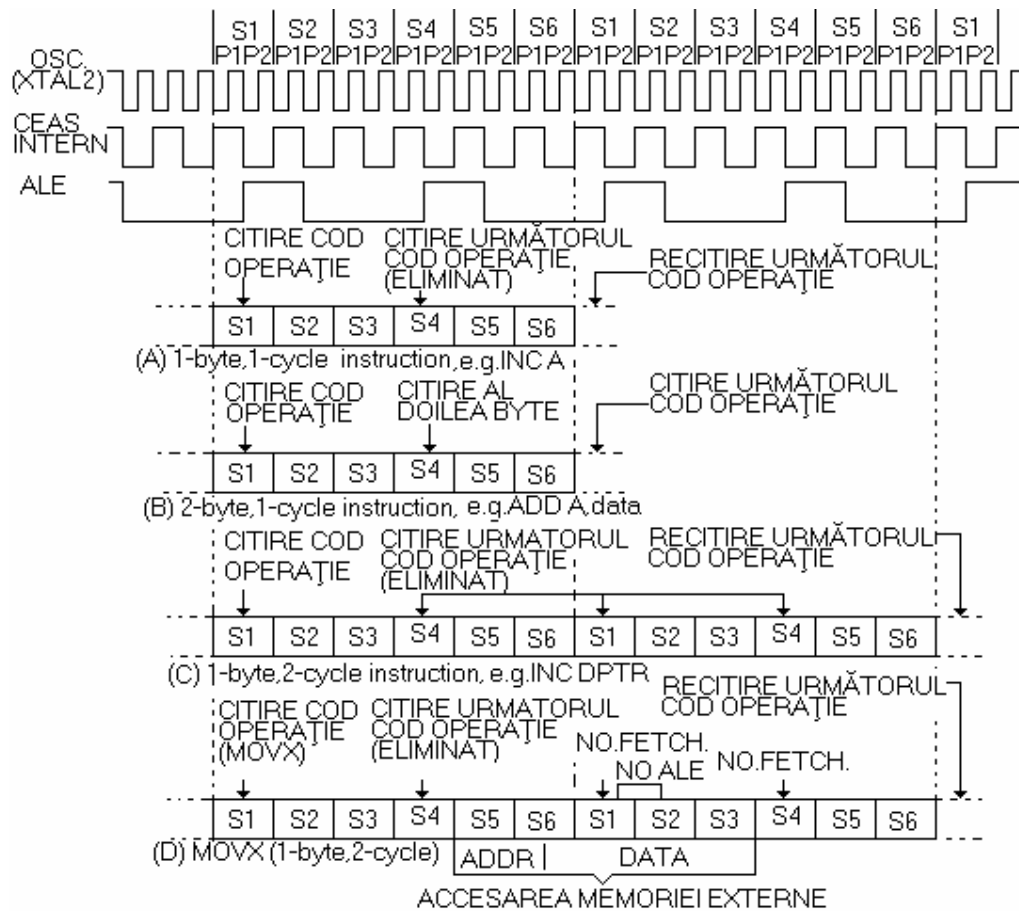


Figura 1.6 Secvențele de extragere/execuție (fetch/execute)

Astfel, un ciclu mașină constă din 12 perioade de oscilator numerotate astfel: S1P1 (State 1, Phase 1), S1P2 (State 1, Phase 2), S2P1 (State 2, Phase 1), S2P2 (State 2, Phase 2) și așa mai departe până la S6P2 (State 6, Phase 2), unde fiecare stare durează două perioade de oscilator. În general operațiile aritmetice și logice au loc pe durata fazei 1, iar operațiile de transfer registru-la-registru au loc pe durata fazei 2.

În **figura 1.6** se poate urmări secvența de extragere/execuție raportată la stări și faze.

Deoarece semnalele de ceas intern nu sunt accesibile utilizatorului, se folosesc ca referință externă semnalele XTAL și ALE. În mod normal ALE este activat de două ori în timpul unui ciclu mașină: prima dată pe perioada S1P2-S2P1 și a doua oară pe perioada S4P2-S5P1.

Execuția unui ciclu instrucțiune începe în S1P2, când codul operației (*opcode*) este deja încărcat în registrul de instrucțiuni. Pentru instrucțiunile pe doi octeți, al doilea octet este citit în S4P2 al aceluiași ciclu mașină. Pentru instrucțiunile pe un octet, a doua citire care se face în cadrul aceluiași ciclu mașină în S4P2, este citirea codului operației următoare, dar octetul citit este ignorat și numărătorul de program PC (*Program Counter*) nu este incrementat.

În oricare din aceste două cazuri, indiferent dacă instrucțiunea este pe un octet sau pe doi octeți, execuția este completă la sfârșitul lui S6P2. În mod normal, pe durata unui ciclu mașină se extrag din memoria program două coduri de octeți. Singura excepție care apare este atunci când se execută instrucțiunea MOVX. Instrucțiunea MOVX, de tipul un octet doi cicli, accesează memoria de date externă.

În timpul unei MOVX sunt omise două extrageri (*fetch*) în timpul adresării și strobării memoriei de date externe. **Figurile 1.6,c** și **1.6,d** prezintă diagramele de timp pentru o instrucțiune normală de tipul un octet doi cicli și respectiv pentru o instrucțiune MOVX.

1.6 PORTURI – STRUCTURĂ ȘI FUNCȚIONARE

Microcontrolerul 80C32 are 4 porturi bidirecționale. Fiecare din aceste porturi este format dintr-un registru cu funcții speciale (respectiv P0, P1, P2, P3), un *driver* de ieșire și un *buffer* de intrare.

Driver-ele de ieșire ale porturilor 0 și 2, și *buffer*-ele de intrare ale portului 0, sunt folosite în accesul la memoria de date externă astfel: Portul 0 scoate octetul celor mai puțin semnificativi biți adresei memoriei externe multiplexat în timp cu octetul scris sau citit, iar portul 2 scoate octetul celor mai semnificativi biți ai adresei memoriei externe dacă această adresă este pe 16 biți; dacă adresa este pe 8 biți, pe pinii portului 2 se transmite în continuare conținutul registrului cu funcție specială P2.

Pinii P1.0 și P1.1 ai portului 1 și toți pinii portului 3 sunt multifuncționali. Ei pot fi folosiți ca pini de port sau pini cu funcții alternative. Funcțiile alternative ale acestor pini sunt următoarele:

Pinul de Port	Funcția alternativă
P1.0	T2-intrare externă în <i>Timer/Counter 2</i>
P1.1	T2EX- <i>Trigger</i> -ul pentru capturare/reîncărcare al <i>Timer/Counter 2</i>
P3.0	RXD- intrare serială de port pentru recepție
P3.1	TXD- ieșire serială de port pentru transmisie
P3.2	INT0- întrerupere externă
P3.3	INT1- întrerupere externă
P3.4	T0- intrare externă în <i>Timer/Counter 0</i>
P3.5	T1- intrare externă în <i>Timer/Counter 1</i>
P3.6	WR- <i>Strob</i> -ul pentru scrierea memoriei de date externă
P3.7	RD- <i>Strob</i> -ul pentru citirea memoriei de date externă

Fiecare din aceste funcții alternative este activată doar dacă bitul din *latch*-ul corespunzător din SFR-ul portului este 1; dacă bitul respectiv este 0, pinul portului este fixat la 0.

1.6.1 SCRIEREA ÎNTR-UN PORT

La execuția unei instrucțiuni care schimbă valoarea în *latch*-ul unui port, noua valoare ajunge la *latch* în S6P2 al ciclului final al instrucțiunii. Oricum, conținutul registrelor portului este de fapt eșantionat de *buffer*-ele de ieșire ale portului doar în timpul fazei 1 a oricărei perioade de ceas. (În timpul fazei 2, *buffer*-ele de ieșire păstrează valoarea văzută în faza 1 anterioară). În consecință, noua valoare din *latch*-ul portului nu va apărea la pinul de ieșire până la următoarea fază 1, care va fi în S1P1 al următorului ciclu mașină.

Dacă schimbarea cere o tranziție din 0 în 1 în portul 1, 2 sau 3, este activată o rezistență suplimentară de sarcină conectată la +Vcc, pe durata lui S1P1 și S1P2 a ciclului în care are loc tranziția. Acest tip de conectare la +Vcc (*pull-up*) furnizează un curent de 100 de ori mai mare decât cel pe care-l poate furniza o conectare normală.

1.6.2 ÎNCĂRCAREA ȘI INTERFAȚAREA UNUI PORT

Buffer-ele de ieșire ale porturilor 1, 2 și 3 pot să conducă, fiecare dintre ele, câte 4 intrări LS TTL. În tehnologia HMOS, în care este realizat microcontrolerul, aceste porturi pot fi conduse în mod normal de orice circuit TTL sau NMOS.

Buffer-ele de ieșire ale portului 0 pot, fiecare dintre ele, să conducă 8 intrări LS TTL. Pentru a conduce intrări NMOS, au nevoie de o rezistență de *pull-up* externă, exceptând cazul în care sunt folosite ca magistrală de adrese/date.

1.6.3 FACILITATEA CITIRE-MODIFICARE-SCRIERE (READ-MODIFY-WRITE)

Unele instrucțiuni de citire a portului citesc *latch*-ul, iar altele citesc pinul. Instrucțiunile care citesc *latch*-ul, sunt cele care citesc valoarea, pot să o schimbe și apoi să o rescrie în *latch* și sunt numite instrucțiuni citire-modificare-scriere.

Instrucțiunile de tip citire-modificare-scriere sunt următoarele:

-ANL:	AND logic	exemplu: ANLP1,A
-ORL:	OR logic	exemplu: ORLP2,A
-XRL:	XOR logic	exemplu: XRLP3,A
-JBC:	salt dacă bit=1 și reset	exemplu: JBCP1.1,LABEL
-CPL:	Complement bit	exemplu: CPLP3,0
-INC:	incrementează	exemplu: INCP2
-DEC:	decrementează	20

25

exemplu: DECP2

- MOV PX.Y,C: mută bitul de transport (carrz) în bitul Y al portului X
- CLR PX.Y: resetează bitul Y al portului X
- SET PX.Y: setează bitul Y al portului X

Ultimele trei instrucțiuni sunt și ele instrucțiuni *read-modify-write*, citesc octetul din port, toți cei 8 biți, modifică bitul adresat, apoi scriu noul octet înapoi în *latch*.

Motivul pentru care instrucțiunile *read-modify-write* sunt direcționate mai degrabă spre *latch* decât spre pin este că astfel se evită o posibilă interpretare greșită a nivelului de tensiune la pin. De exemplu, un bit din port poate fi folosit pentru a comanda baza unui tranzistor – când bitul este „1” logic tranzistorul este deschis. Dacă CPU citește același bit din port de la pin, și nu din *latch*, el va citi tensiunea din baza tranzistorului și o va interpreta ca „0” logic. Citind *latch*-ul și nu pinul, va întoarce valoarea corectă „1” logic.