

REGISTRII CU FUNCȚII SPECIALE (Special Function Register -- SFR)

Registrii cu Funcții Speciale se află în spațiul memoriei de date, de la adresa 80H la adresa 0FFH și pot fi accesați **doar prin adresare directă**. Trebuie menționat că tot între aceste adrese (80H și 0FFH), la unele microcontrolere de tipul 80C52, se află fizic o altă memorie de date de uz general, care poate fi accesată **doar prin adresare indirectă**.

Tabelul 1 conține o listă cu toate SFR-urile și adresele lor.

Tabelul 1

Simbol	Nume	Adresa
*ACC	Acumulator	0E0H
*B	Registrul B	0F0H
*PSW	Cuvântul de stare al programului	0D0H
SP	Stack Pointer	81H
DPTR	Data Pointer – este pe 2 bytes:	
DPL	Low Byte Data Pointer	82H
DPH	High Byte Data Pointer	83H
*P0	Port 0	80H
*P1	Port 1	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
TH0	Timer/Counter 0 HighByte	8CH
TLO	Timer/Counter 0 LowByte	8AH
TH1	Timer/Counter 1 HighByte	8DH
TL1	Timer/Counter 1 LowByte	8BH
*SCON	Serial Control	98H
SBUF	Serial Data Buffer	99H
PCON	PCON Power Control	87H

* = Adresabil la nivel de bit

Aceste registre (octeți = bytes) au o anumită valoare la Resetarea microcontrolerului, valoare ce nu întotdeauna este 00H. (vezi foaia nr. 3 aferentă microcontrolerului AduC812 – “SFR MAP AND RESET VALUE”). Cu ajutorul acestor Registre controlăm microcontrolerul. În Tabelul. 1 este prezentată ocuparea “Regiștrilor cu Funcții Speciale” în spațiul de memorie de date cuprins între adresele 80H și 0FFH.

Harta Memoriei SFR 8 Bytes

F8									FF
F0	B								F7
E8									EF
E0	ACC								E7
D8									DF
D0	PSW								D7
C8									CF
C0									C7
B8	IP								BF
B0	P3								B7
A8	IE								AF
A0	P2								A7
98	SCON	SBUF							9F
90	P1								97
88	TCON	TMOD	TL0	TL1	TH0	TH1			8F
80	P0	SP	DPL	DPH				PCON	87

Figura 1 ↑

Regiștrii din prima coloana pot fi adresați la nivel de bit.

Acumulatorul. ACC este registrul acumulator. Mnemonicele pentru instrucțiuni specifice cu acumulatorul se referă la acumulator în mod simplificat ca fiind A.

Registrul B. Registrul B este utilizat pe durata operațiilor de înmulțire și de împărțire. Pentru alte operații poate fi tratat ca alt registru de memorie.

Cuvântul de stare a programului PSW. Registrul PSW conține informații asupra stării programului, așa cum se detaliază în cele ce urmează.

Indicatorul de stivă SP (Stack Pointer). Indicatorul de stivă este un registru de 8 biți. El este incrementat înainte ca data să fie stocată în timpul executării instrucțiunilor PUSH sau CALL. Cum stiva poate fi plasată oriunde în interiorul memorie RAM on-chip, indicatorul de stivă este inițializat la valoarea 07H după reset. Aceasta determină ca stiva să înceapă de la locația 08H.

Indicatorul de date DPTR (Data Pointer). Indicatorul de date (DPTR) este format dintr-un octet superior (DPH) și un octet inferior (DPL). Funcția lor a fost în intenție de a păstra adrese de 16 biți. El poate fi manipulat ca un registru de 16 biți sau ca două registre de 8 biți.

Porturile 0 la 3. P0, P1, P2 și P3 sunt latchurile porturilor 0, 1, 2 și 3. Scrierea unui "1" într-un bit al unui port din SFR (P0, P1, P2 sau P3) determină ca pinul de ieșire corespunzător din port să comute la nivel HIGH. Scrierea unui zero va determina trecerea la nivel LOW a pinului corespunzător. Atunci când este utilizat ca intrare, starea externă a unui pin dintr-un port va fi înscrisă în portul SFR (de exemplu, dacă starea externă a pinului este LOW, bitul corespunzător din portul SFR va conține un "0"; dacă nivelul este HIGH, bitul va conține un "1").

Bufferul de date seriale. Bufferul de date seriale este format în realitate din două registre separate, un buffer transmițător și un buffer receptor. Când data este transferată în

SBUF, ea merge în bufferul transmițător și este păstrată pentru transmisia serială. Când data este transferată din SBUF, ea provine de la bufferul receptor.

Registrele timerelor de bază ale 80C51. Registrele pereche (TH0, TL0) și (TH1, TL1) sunt registre numărătoare de 16 biți pentru circuitele timer / numărător 0, respectiv 1.

Registrele de control pentru 80C51. Registrele de funcții speciale IP, IE, TMOD, TCON, SCON și PCON conțin biții de control și de stare pentru sistemul de întreruperi, timere / numărătoare, pentru portul serial și pentru controlul alimentării.

Registrul cuvânt de stare program (PSW)

MSB						LSB	
CY	AC	F0	RS1	RS0	OV	–	P

BIT	SIMBOL	FUNCȚIE
PSW.7	CY	Flag transport
PSW.6	AC	Flag transport auxiliar
PSW.5	F0	Flag 0 (disponibil utilizatorului pentru aplicații generale)
PSW.4	RS1	Bit 1 de control pentru selecția bancului (vezi nota)
PSW.3	RS0	Bit 0 de control pentru selecția bancului
		Setat/șters software pentru a preciza bancul de registre (vezi nota)
PSW.2	OV	Flag depășire
PSW.1	–	Flag ce poate fi definit de utilizator
PSW.0	P	Flag de paritate
		Setat/șters prin hardware la fiecare ciclu de instrucțiune pentru a indica un număr par / impar de biți "1" din acumulator, și anume par

Valorile din RS0 și RS1 selectează bancul de regiștri corespunzător tabelului de mai jos:

RS1	RS0	Register Bank	Adresa
0	0	0	00H-07H
0	1	1	08H-0FH
1	0	2	10H-17H
1	1	3	18H-1FH

Registrul de control al alimentării (PCON)

MSB						LSB	
SMOD	–	–	–	GF1	GF0	PD	IDL

BIT	SIMBOL	FUNCȚIE
PCON.7	SMOD	Dublează rata de bit. Când este setat la valoarea 1 și timerul 1 este folosit pentru a genera rata Baud și portul serial este folosit în modurile 1, 2 sau 3
PCON.6	–	Rezervat
PCON.5	–	Rezervat
PCON.4	–	Rezervat
PCON.3	GF1	Flag de uz general
PCON.2	GF0	Flag de uz general
PCON.1	PD	Bit alimentare redusă. Prin setarea acestui bit se activează funcționarea cu alimentare redusă
PCON.0	IDL	Bit de mod mers în gol. Prin setarea acestui bit se activează funcționarea în mod mers în gol

Moduri de adresare

Modurile de adresare din setul de instrucțiuni 80C51 sunt următoarele:

Adresarea directă. În adresarea directă, operandul este specificat de un câmp de adresă de 8 biți în instrucțiune. Adresarea directă se face numai pentru memoria internă RAM și pentru registrele de funcții speciale SFR.

Exp. MOV 32H, 31H

Adresare imediată. Valoarea unei constante poate urma unui cod de operare în memoria program. De exemplu:

Exp. MOV A, #25

încarcă acumulatorul cu valoarea zecimală 25. Același număr poate fi specificat în hex prin 19H.

Adresarea indirectă. În adresarea indirectă, instrucțiunea specifică un registru care conține adresa operandului. Atât memoria internă cât și cea externă RAM pot fi adresate indirect.

Registrul de adresă pentru adresele de 8 biți pot fi R0 sau R1 din bancul selectat sau indicatorul de stivă (*Stack Pointer*). Registrul de adresă pentru adrese de 16 biți poate fi numai registrul de 16 biți DPTR (*Data Pointer*) - indicator de date.

Exp. MOV R0, #32H

MOV @R0, 31H

Instrucțiuni registre. Bancurile pentru registre, conținând registrele R0 la R7, pot fi accesate prin anumite instrucțiuni ce conțin 3 biți de precizare a registrului în interiorul codului de operare al instrucțiunii. Instrucțiunile ce accesează registrele în acest mod sunt coduri eficiente, întrucât acest mod elimină un octet de adresă. Când instrucțiunea este executată, este accesat unul din cele opt registre din bancul selectat. La momentul executării instrucțiunii este selectat unul din cele patru bancuri de registre de către cei doi biți de selecție a bancului din PSW.

Exp. ADD A,Rn

Instrucțiuni registre speciale. Câteva instrucțiuni sunt specifice unor anumite registre. De exemplu, câteva instrucțiuni operează întotdeauna cu acumulatorul, cu indicatorul de date (*Data pointer*) etc., și nu mai este necesar un octet de adresă pentru precizare. Codul de operare însuși realizează acest fapt. Instrucțiunile ce se referă la acumulator ca A, asamblează coduri de operare specifice pentru acumulator.

Exp. INC A

DEC A

MUL AB

Adresarea indexată. Cu adresare indexată se poate accesa numai memoria program și numai în modul citire.

Acest mod de adresare este conceput pentru citirea tabelilor de informații din memoria program. Un registru bază de 16 biți (fie DPTR sau Contorul Program) fixează baza tabelului, iar acumulatorul este încărcat cu numărul de intrare în tabel.

Adresa de intrare în tabelul din memoria program se formează prin adunarea datei din acumulator la indicatorul bazei.

Exp. MOV DPTR,#TAB1

MOV A,#01H

MOVC A,@A+DPTR

MOV P0,A

Utilizarea programului uVision2 - Keil Software (Demo Version)

Programul folosit în laborator este o variantă gratis oferită de compania Keil Software, având limitare de asamblare (compilare) a programelor ce depășesc 2K octeți. Acest program este un simulator de microcontrolere, ajutând programatorul să-și testeze programul, în timp real sau pas cu pas, înainte de a-l înscrie în microcontroler.

Utilizarea programului:

Se pornește de la crearea unui „proiect” : „project” → „new project” . Se dă nume proiectului și se salvează. În acest moment apare „Select device for target” , și se alege din librărie microcontrolerul pentru care se editează programul. Apoi se creează o pagină nouă „file” → „new”. Se salvează această pagină cu extensia: „.asm”, apoi din „target 1” → „source group 1” , se dă click dreapta la mouse și se alege „Add files to Group ‘source group 1’ ”, se deschide o nouă fereastră de unde se alege fișierul mai devreme salvat. Atenție la extensia fișierului! Odată ales fișierul, se apasă butonul „close” pentru a închide fereastra de dialog.

Începând din acest moment, se poate trece la editarea programului. Programul oferă o interfață ușoară de lucru cu utilizatorul, de exp. mnemonicele din program se vor colora în albastru.

După ce s-a scris programul, se va trece la asamblarea acestuia, apăsând „F7” (build target). În acest moment se assemblează programul și dacă nu sunt erori se poate trece la simularea programului.

```

name test_2delay          ;nume program
org 0000H                 ;inscrie programul in memoria program incepand de la adresa 0000H
LED EQU P3.4             ;eticheteaza pinul de port P3.4
START:
    CLR LED               ;"stinge LED-ul"
    CALL DELAY1           ;apeleaza o interziere mai mare
    SETB LED              ;"aprinde LED - ul"
    CALL DELAY2           ;apeleaza o interziere mai mica
    JMP START             ;repete

DELAY1:                   ;subrutina de intarziere mare
    MOV R7,#6
    MOV R6,#100
    MOV R5,#100
DELAY10:
    DJNZ R5,DELAY10
    DJNZ R6,DELAY10
    DJNZ R7,DELAY10
    RET                   ;revenire din apelare

DELAY2:                   ;subrutina de intarziere mai mica
    MOV R7,#1
    MOV R6,#100
    MOV R5,#100
DELAY20:
    DJNZ R5,DELAY20
    DJNZ R6,DELAY20
    DJNZ R7,DELAY20
    RET                   ;revenire din apelare
END                       ;sfarsit de program
    
```