

3.MÉRÉS

Vezérlési és bit-műveletes feladatok programozása

1. A mérés célja:

A mikrokontroller bit-kezelő utasításainak használata logikai feladatok, vezérlések programjaiban.
A programfejlesztés, programélesztés gyakorlása

2. A szükséges ismeretek:

A 8051/es mikrokontroller család közvetlen bit műveletei lehetővé teszik, hogy a logikai algebrai formában megfogalmazott feladatok - bájtos maszkolások nélkül is - programozhatóak. A bit műveletek operandusai számára a belső memória bit címezhető területein, összesen 256 bit áll rendelkezésre.

2.1. Az általános célú belső memóriában

összesen 128 bit használható bit-operandusként. A memória 20H - 2FH címtartományában elhelyezkedő 16 bájt minden bit-je külön - külön, **abszolút**, vagy **bájt-relatív** címmel érhető el.

Az **abszolút** címezésnél **00H** című a **20H** címen található bájt **0**-ás bitje, a **01H** ugyanezen bájt **1**-es bitje. A számozást folytatva a **08H** című lesz a **21H** bájt **0**. bitje. A **7FH** értékkel (a **2FH** című bájt **7** -es bitje) fejeződik be az itt található bitek címe.

A **bájt-relatív** címezésben megadjuk a **bájt címét**, és **'.'** - al elválasztva a **bit sorszám** - át.

00H	->	20H.0
	.	
07H	->	20H.7
08H	->	21H.0
	.	
7FH	->	2FH.7

2.2. Az SFR regiszterek között

is 128 bit használható bit-operandusként. Az abszolút címsor 80H - FFH között van. A **törvény-szerűség**: a **8** - al osztható című regiszterek bitjei kezelhetők külön - külön, és a bájt címe és a 0 -ás bitjének a címe azonos.

80H	->	80H.0
	.	
87H	->	87H.7
88H	->	88H.0
	.	
FFH	->	FFH.7

2.3. Bitek és szimbólumok összerendelése.

A program olvashatósága érdekében célszerű az operandusokhoz, így a **bitek**-hez is - a funkcióra utaló - **megnevezést**, **szimbólumot** rendelni. Az esetek döntő többségében az önálló bit-oprandus és a bájtot is kell használnunk. Például perifériaillesztők mindig bájtosan kezelhetők, ezek egyes bitjeit már a programban külön - külön is alkalmazhatók. Ilyenkor nem bit szegmenst kell nyitni, hanem adatszegmenst a bit címezhető területre. A deklarációkat az alábbiak mutatják.

;***** Szegmensek deklarálása *****

VALTB SEGMENT DATA BITADDRESSABLE ;bitcímezhető változók

;***** ADAT szegmens bitcímezhető változóknak *****

	RSEG	VALTB	
	DS	2	
BEM:	DS	1	;bemeneti memória
KIM:	DS	1	;kimeneti memória
JELZ:	DS	1	;jelző bájt
NG0BIT	BEM.0		;NG0 nyomógomb
NG6BIT	NG0+6		;NG6 nyomógomb
LD2	BIT	KIM.2	; 2. LED
JLZB1	BIT	JELZ.0	;1.jelzőbit
JLZB2	BIT	JELZ.1	;2.jelzőbit

2.4. Logikai függvény programozása

A mikrokontroller bit műveleteinek használatával egy logikai feladat függvénye közvetlenül programozható. Példának vegyük, hogy az előzőekben deklarált bitek felhasználásával a

$$LD2 = \overline{NG0} * \overline{NG6} + NG0 * NG6$$

kizáró-vagy művelet programját.

MOV	C,NG0	; első ÉS művelet
ANL	C,/NG6	
MOV	JELZB1,C	; részeredmény tárolása
MOV	C,NG6	; második ÉS művelet
ANL	C,/NG0	
ORL	C,JELZB1	; VAGY művelet

3. Házi feladatok

Írja meg a 3.1., 3.2., és 3.3. feladatok programjait ugyanazon forrásnyelvi fájlba! Az egyes feladatokhoz tartozó részprogramokat két NOP utasítással válassza el. A 3.4. feladatot, - önállóan futó programként - külön állományba írja!

- 3.1. A gyakorló LED-során jelezze ki binárisan a P1 porthoz csatlakoztatott peremkerekcs-kódkapcsolón beállított értékben lévő 1-ek számát.
- 3.2. A nyomógomb-sor alsó három gombjaihoz rendeljük a bináris szám helyiértékeit. A gombok lenyomásainak lehetséges kombinációinál a megfelelő helyiértékű LED világítson.
- 3.3. A középső nyomógomb lenyomásakor a P1 porthoz csatlakoztatott peremkerekcs kódkapcsoló beállított értékét írja a V1 változóba, felengedésekor pedig a V2 változóba. A gomb felengedése után a LED-soron binárisan jelenjen meg a V1-V2 különbség abszolút értéke!
- 3.4. Világítson a legnagyobb helyiértékű LED az $F=D(B+/C)+ABC$ logikai függvény IGAZ értékénél. Az A,B,C,D bemeneti változók a legkisebb helyiértékű nyomógombok adják

A programozáshoz szükséges hardver jellemzők:

- a LED-sor címe: **0C001H** és a megfelelő LED **0** vezérlésre világít,
- a nyomógomb-sor címe: **0C000H** és a nyomógomb lenyomásakor ad **0**-t értéket.

4. Mérési feladatok:

- 4.1.** Az első három házi feladatot tartalmazó fájlt másolja be saját alkönyvtárába. Végezze el a fordítást, szerkesztést és a hibátlan program HEX konverzióját.
- 4.2.** A program letöltése után - az unassembler belső parancs segítségével - határozza meg az egyes programrészek kezdő címeit, majd lépésenkénti futtatással ellenőrizze a programrészek helyes működését.
- 4.3.** A negyedik feladatot megvalósító programot élessze fel, és ellenőrizze a működést mind szimulátorral, mind valós idejű működtetéssel. Hibás működés esetén végezze el a szükséges korrekciókat!

5. Kérdések

- 5.1.** Milyen módon lehet a biteket címezni?
- 5.2.** Melyik memória területeken találhatók egyedileg írható-olvasható bitek?
- 5.3.** A bit műveleteknél mi a " bit akkumulátor " ?
- 5.4.** Milyen logikai műveletek programozhatók közvetlen utasításokkal?
- 5.5.** Milyen ugró utasítások használhatók bit operandussal?