A 8051-es mikrokontroller család

Tartalomjegyzék

1 A MIKROKONTROLLEREK ÁLTALÁNOS ÁTTEKINTÉSE 5

1.1 A SZÁMÍTÓGÉP FELÉPÍTÉSE. 5
1.2 A MIKROKONTROLLER MEGJELENÉSE 5
1.3 A MIKROKONTROLLER ÉS A MIKROPROCESSZOR KÖZÖTTI KÜLÖNBSÉG. 5
1.4 A MIKROKONTROLLER ALKALMAZÁSI TERÜLETEI. 6

2 A 8051/8031-ES TÍPUSÚ MIKROKONTROLLER HARDVERKIALAKÍTÁSA 7

2.1 ÓRAJELEGYSÉG. 7
2.1.1 A kvarc, ... *7*
2.1.2 Kerámiarezonátorok. *8*
2.1.3 Oszcillátorkapcsolás *8*
2.1.4 Külső szinkronizálás *8*
2.2 BELSŐ ÉS KÜLSŐ TÁROLÓK 8
2.2.1 Belső programtároló *8*
2.2.2 A 8751-8051-es típus EPROM-mal *9*
2.2.3 Belső adattároló *9*
2.3 A FOGYASZTÁS CSÖKKENTÉSE 10
2.3.1 Power-down üzemmód *10*
2.3.2 Idle üzemmód *11*
2.3.3 A belső RAM áramának biztosítása *11*
2.4 A CPU IDŐZÍTÉSEI 11
2.4.1 Belső időviszonyok *11*
2.4.2 A külső tárolók vezérlőjelei *11*
2.5 IDŐZÍTŐ/SZÁMLÁLÓ 12
2.5.1 A T0 és T1 időzítő/számlálók és üzemmódjaik (0-3) *12*
2.5.2 A T2-es időzítő/számláló. Capture, Auto-Reload és Baudrate üzemmódok *13*
2.6 A SPECIÁLIS FUNKCIÓREGISZTEREK (SFR-EK) 14
2.6.1 Bitcímek *15*
2.6.2 Alapértelmezés szerinti értékek *15*
2.6.3 Veremmutató (Stack-Pointer) *16*
2.6.4 Adatmutató (Data-Pointer) *16*
2.6.5 Fogyasztásvezérlő (Power-Control) regiszter (PCON) *16*
2.6.6 Timer-Control regiszter *16*
2.6.7 Timer-Modus regiszter *16*
2.6.8 Serial-Port-Control regiszter *17*
2.6.9 A soros port adatpuffere *17*
2.6.10 A megszakítás-engedélyező (Interrupt-Enable) regiszter (IE) *18*
2.6.11 Megszakításprioritás- (IP) regiszter *18*
2.6.12 A T2 időzítő vezérlőregisztere (T2CON) *18*
2.6.13 Program-Status-Word (PSW) *19*
2.6.14 Akkumulátor *19*
2.6.15 B regiszter *19*
2.7 A MEGSZAKÍTÁSOK ÉS HASZNÁLATUK 19
2.7.1 Megszakításforrások *20*
2.7.2 A megszakítások műveletei *20*
2.7.3 A megszakítások válaszideje *21*
2.8 SOROSVONAL-ILLESZTŐ 21
2.8.1 Adás, vétel a 0-ás üzemmódban (2.12. ábra) *22*
2.8.2 Adás, vétel az 1-es üzemmódban *23*
2.8.3 A 2-es és a 3-as üzemmódok. Többkontrolleres kommunikáció, adás és vétel *24*
2.8.4 A baudrate előállítása a Tl és a T2 számlálókkal *27*
2.9 A 0-3 PORTOK 28
2.9.1 Egy portra való írás időzítése, jelmeredeksége *30*
2.9.2 A portok terhelhetősége (fan-out) *31*
2.9.3 READ-MODIFY-WRITE utasítások *31*
2.10 KÜLSŐ TÁROLÓK ALKALMAZÁSA 32

2.11 8 VAGY 16 BITES MEMÓRIACÍMZÉS 32
2.11.1 Az ALE jel 32
2.11.2 Az egy-, két- és hárombájtos utasítások végrehajtása 32
2.11.3 A PSEN jel 33
2.11.4 A WR és az RD jelek 34
2.12 A TÁROLÓK HASZNÁLATA 34
2.12.1 Programtároló 34
2.12.2 Adattároló. A belső RAM-terület felépítése 34
2.13 A 8051-ES ÉS A 8052-ES TÍPUSÚ MIKROKONTROLLEREK ÖSSZEHASONLÍTÁSA 36

3 A 8051-ES TÍPUS UTASÍTÁSKÉSZLETE 37

3.1 A BIT- ÉS BÁJTSZERVEZÉSŰ MŰKÖDÉS 37
3.2 AZ UTASÍTÁSOK HOSSZA 37
3.3 FUTÁSI IDŐ 37
3.4 CÍMZÉSI MÓDOK 37
3.4.1 A címzésről általában 37
3.4.2 Regisztercímzés 38
3.4.3 Direkt címzés 38
3.4.4 Indirekt címzés 38
3.4.5 Közvetlen címzés 38
3.4.6 Indirekt regisztercímzés 38
3.5 A KÜLÖNBÖZŐ UTASÍTÁSFAJTÁK 38
3.5.1 Adatátviteli utasítások: általános, akkumulátorral kapcsolatos és a Data-Pointeres adatmozgató utasítások 38
3.5.2 Aritmetikai utasítások: összeadás, kivonás, szorzás és osztás 39
3.5.3 Logikai és Boole-utasítások 39
3.5.4 Vezérlőutasítások: feltétel nélküli CALL és JUMP, feltételes JUMP 40
3.6 ADATÁTVITELI UTASÍTÁSOK 41
3.6.1 MOV 41
3.6.2 MOVC 42
3.6.3 MOVX 42
3.6.4 PUSH 42
3.6.5 POP 42
3.6.6 XCH 42
3.6.7 XCHD 43
3.7 ARITMETIKAI UTASÍTÁSOK 43
3.7.1 ADD 43
3.7.2 ADDC 43
3.7.3 SUBB 43
3.7.4 INC 44
3.7.5 DEC 44
3.7.6 MUL és DIV 44
3.7.7 DA 44
3.8 LOGIKAI ÉS BOOLE-UTASÍTÁSOK 44
3.8.1 Logikai utasítások 44
3.8.2 Boole-utasítások 45
3.9 VEZÉRLŐUTASÍTÁSOK 46
3.9.1 CALL 46
3.9.2 JUMP 46
3.10 AZ UTASÍTÁSOK HOSSZA ÉS VÉGREHAJTÁSI IDEJÜK 48
3.11 A JELZŐBITEKET BEFOLYÁSOLÓ UTASÍTÁSOK 50

Előszó a német kiadáshoz

A számítógépek legfontosabb elektronikus építőelemei, a mikroprocesszorok, változó kivitelűek és teljesítőképességűek lehetnek.
Az áramköröket integrálási technológiával készítik, és egyre több tranzisztorfunkciót tudnak egy chipre (lapkára) felvinni. Ennek eredményeként nagyobb teljesítőképességű (nagyobb "intelligenciájú") mikroprocesszorokat gyártanak. A másik fejlesztési irányzat az, hogy változatlan számítási funkciók mellett, az eddig különálló áramkörökkel megvalósított perifériafunkciókat integrálnak a processzorchipbe. Ilyen áramkör a mikrokontroller. Ezek az áramkörök vezérlik a felvonókat és a televíziókat, "okos" játékokat építenek velük, és megjelennek a járművekben is.
Összefoglalva: A különálló áramkörök funkcióit egy chipbe integrált mikrokontroller a teljes vezérlést elláthatja. Előnyei ellenére kevésbé ismerik, mint a mikroprocesszorokat.
Ebben a könyvben igyekszünk érzékeltetni a mikrokontrollerek és a klasszikus mikroprocesszorok felépítése és funkciói közötti különbségeket. Megismertetjük az olvasókat a nagyon elterjedt 8051-es mikrokontroller-családdal. Leírjuk az egyes típusok tulajdonságait, hardverfunkcióit és utasításkészletét. Megemlítjük előnyeiket és hátrányaikat is. Az olvasó segítséget kap ahhoz, hogy eldönthesse, hol előnyösebb mikrokontrollert alkalmazni a mikroprocesszorral szemben.
A könyv azok számára készült, akik ismerik a mikroelektronika alapfogalmait. Ezért a gyártók katalógusaiban szereplő angol elnevezéseket használjuk.

Thierstein *Roland Dilsch*

RÖVIDÍTETT VÁLTOZAT!

1 A mikrokontrollerek általános áttekintése

1.1 A számítógép felépítése.

A mikrokontroller megértéséhez foglalkozzunk először a mikroszámítógép történetével! Ehhez szorosan kapcsolódik a mikrokontroller története is.
Bár programozható számítógép már korábban is volt mikroszámítógépről valójában csak kb. 1975 óta beszélhetünk. Abban az időben állították elő az első olyan integrált áramköröket, amelyek egy chipben egy számítóegységet és annak vezérlését tartalmazták. Ez volt a mikroprocesszor.
Ahhoz, hogy a mikroprocesszorból működőképes számítógép legyen, számos kiegészítő áramkörre van szükség: pl. órajel-generátorra - rendszerint kvarccal megvalósítva -, amely a mikroprocesszor időalapját adja. Továbbá tárolók (RAM-ok) is szükségesek. Számukat az egyes IC-k kapacitása és a szükséges memória nagysága szabja meg. Ezeken kívül feltétlenül kell még fix tároló (ROM, PROM vagy EPROM), amely az ún. beolvasó- (angolul a Boot) programot tartalmazza. Ha egy számítógépet vezérlési feladatra használunk, akkor majdnem az egész programnak EPROM-ban kell lennie. A már említett memóriák mellett olyan diszkrét logika is szükséges, amely azt biztosítja, hogy a mikroprocesszor mindig a megfelelő memóriába írjon, vagy onnan olvasson.
Az adatokat a külvilágba is ki kell juttatni. Ezen kívül különböző perifériavezérlők is szükségesek (pl. egy terminállal való kommunikáció vezérléséhez). A felhasználásokhoz egyre több áramköri elemet kell alkalmazni, így az analógjelek feldolgozásához A/D átalakítókat.
A diszkrét építőelemekből való rendszerépítésnek sok előnye van. Pl. mindig a feladathoz szükséges optimális méretű memória vagy I/O illesztőegység építhető össze. Ily módon viszont nagyon összetett és drága berendezést kapunk.

1.2 A mikrokontroller megjelenése

A nagyon gyorsan fejlődő integrálási technológia révén egyre több tranzisztorfunkció került be egy chipbe. Eleinte csak a mikroprocesszorok teljesítőképességét növelték. Ezzel párhuzamosan keresték annak a lehetőségét is, hogy az egy chipbe integrált funkciók számát is növeljék. A cél az volt, hogy az egyedi periféria-áramköröket is a processzorchipbe vigyék be, és ezzel egyszerűbbé váljon az áramkörépítés.
Az Intel cég 1975-ben állította elő az első mikrokontrollert. a 8048-ast. Erre a kontrollerre alapozva 1980-ban jelentette meg a piacon az Intel a 8051-es típust. Ezt a mikrokontrollert világszerte felhasználták. Napjainkban számos gyártó - részegységeit továbbfejlesztve - kínálja különböző változatait. Bár azóta már tíz év telt el, napjainkban is sok felhasználó alkalmazza. Ezért vált ez a típus alapkontrollerré. Természetesen más félvezetőgyártók is kínálnak mikrokontroller-családokat (pl. Motorola), ennek ellenére a 8051-es család folyamatosan bővül. Ebben a könyvben csak a 8051-es típusjelű mikrokontrollerrel foglalkozunk részletesen.

1.3 A mikrokontroller és a mikroprocesszor közötti különbség.

A mikrokontroller nem más, mint egy mikroprocesszor és további periféria-áramkörök egyetlen közös egységbe integrálva.
Ma már az összes, ún. perifériafunkciót beintegrálják egy mikrokontroller-IC-be. A legismertebb mikrokontrollerbe - a 8051-esbe - a következő funkciók vannak beépítve:
- 4 Kbájt ROM;
- két 16 bites időzítő/számláló;
- négy 8 bites párhuzamos port, 32 I/O vonalként; - 128 bájt RAM;
- 4 Kbájt programtároló (ROM); - sorosvonal-illesztő.

A nagyobbik testvérnek, a 80515-ös típusjelű mikrokontrollernek ezeken kívül más funkciói is vannak:
- 8 Kbájt ROM, - 256 bájt RAM,
- hat 8 bites port, 48 I/O vonal,
- három 16 bites időzítő/számláló (egy ezek közül különleges tulajdonságokkal),
- négyszintű megszakítás-vezérlés,
- 8 bites A/D átalakító nyolc multiplexelt bemenettel.

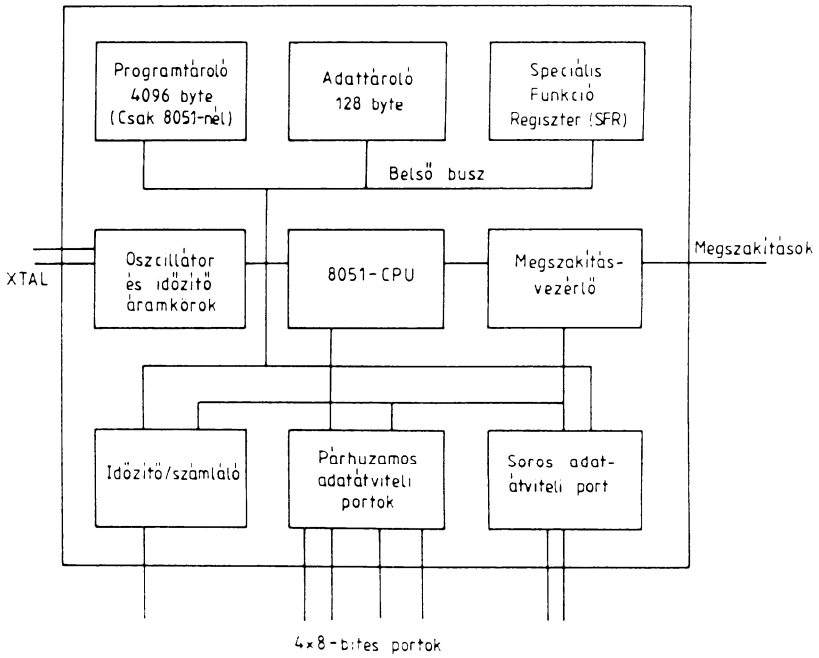
Leírtuk tehát azokat a szakkifejezéseket, amelyeket a gyártók is használnak a katalógusaikban. Az egyes fogalmak jelentését a későbbiekben adjuk meg. Itt csak a 8 bites kontrollerekkel foglalkozunk. A különböző alkalmazásokban a nagyobb számítási teljesítményű 16 bites mikrokontrollereket is használják. Példaként most az Intel cég 8096-os családját említjük meg. A 16 bites kontrollerekkel a kapcsolás és a nyomtatott áramkör összetettebb, mint a 8 bitesekkel.

1.4 A mikrokontroller alkalmazási területei.

A mikrokontrollerek ára a nagyon sok funkció ellenére sem magas. Azzal mindvégig tisztában kell lennünk, hogy a mikrokontrollerek számítási teljesítménye elmarad a mikroprocesszorokétól. A hardverkialakítás sem olyan rugalmas. mint a mikroprocesszoros rendszereknél. Nincs is erre szükség. A mikrokontrollerek klasszikus alkalmazási területe a vezérlés, amely kisméretű, a szükséges számításokra alkalmas készülékeket igényel. Erre viszont igen alkalmas a mikrokontroller. Nagyon kevés külső áramkörrel - pl. programtárolóként használt EPROM vagy néhány meghajtótranzisztorral - egyetlen Európa-kártyán vagy más kivitelben alakítható ki a teljes áramkör. További felhasználási területük a szórakoztatóelektronika (képmagnók, televíziók). Ezekben, pl. a különböző adókra való rugalmas átprogramozást biztosítja, ilyen módon nagyon sok áramköri elemet vált ki. A gyártónak csupán egyszer kell a programot kifejleszteni a vezérelt készülékhez.

2 A 8051/8031-es típusú mikrokontroller hardverkialakítása

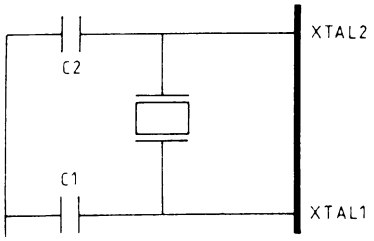
A mikrokontroller egy mikroprocesszor műveletvégző egységéből és kibővített hardverrészekből áll. A bővített funkciókkal a továbbiakban még foglalkozunk. A 8051-es család egyes tagjainak tulajdonságai között nincs lényeges eltérés. Az egyes gyártók által kínált CMOS-változatok alapfunkciói és lábkiosztásuk megegyezik az NMOS-változatokéval. Ha szükséges, ismertetjük a különbségeket.
A 2.1. ábrán a 8051-es típusú mikrokontroller működési blokkvázlata látható.



2.1. ábra. A 8051-es típus blokkvázlata

2.1 Órajelegység.

A 8051-es mikrokontrollernek - a legtöbb mikroprocesszorhoz hasonlóan belső órajelegysége van. Az áramkör egy invertálóerősítő, amelynek a be- és a kimenete az XTAL 1 és az XTAL2 lábakhoz csatlakozik (19 és 18-as lábak). A felhasználáshoz szükséges órajelfrekvenciát a csatlakoztatott külső áramköri elemek határozzák meg. Az invertálóerősítő 180°-os fázistolást okoz. Az erősítő be- és kimenete közé olyan pozitív reaktanciájú elemet kell kapcsolni, amely az önrezgéshez szükséges - további 180°-os fázistolást okoz. Erre a célra többnyire kvarcot vagy kerámiarezonátort használhatunk (a 2.2. ábra szerinti kapcsolásban). Elméletileg induktivitás is használható, de a pontos frekvencia-beállítás így nehéz. A 8051-es külső órajel-generátorral is szinkronizálható!



2.2. ábra. Kvarccal stabilizált órajel-generátor

2.1.1 A kvarc,

A kvarc rezgése a piezoelektromos hatáson alapul. A kristályra jutó feszültség mechanikai alakváltozást okoz. Ennek hatására a kristályban elektromos ellenfeszültség keletkezik. A kvarc rezonanciafrekvenciáját alapvetően a mechanikai méretek határozzák meg. A rezgést csak nagyon pontos technológiával lehet jól beállítani. Önmagában a kvarc nem rezeg, ezért további áramköri elemekkel (kondenzátorok vagy ferritgyöngy) kell kiegészíteni. A jó minőségű kvarc frekvenciastabilitása nagy.

2.1.2 Kerámiarezonátorok.

A szórakoztatóelektronikai készülékekben lényeges szempont az alacsonyabb ár, viszont nem alapvető a nagy frekvenciastabilitás. A kerámiarezonátorok kielégítik ezeket az igényeket. Működésük a kvarcokéhoz hasonló, de nehezebb a frekvenciaátfogást külső elemekkel beállítani.

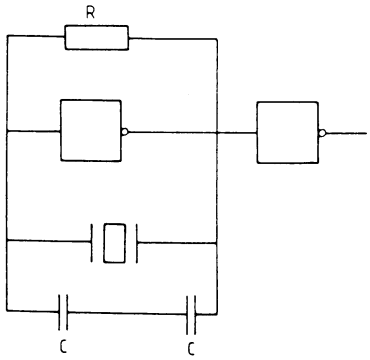
A kerámiarezonátorok olcsóbbak a kvarcoknál, frekvenciastabilitásuk viszont kisebb.

2.1.3 Oszcillátorkapcsolás

A kvarcot vagy a kerámiarezonátort az XTAL1 és az XTAL2 lábak közé kell bekötni. A lábak és a test közé egy-egy kerámiakondenzátort kell csatlakoztatni (CX1, CX2). A kondenzátorok értékeinek változtatásával mintegy 5 %-os frekvenciaváltoztatást lehet megvalósítani. Mind a frekvenciaátfogás, mind pedig a stabilitás függ a kondenzátorok értékétől. A rezgési frekvencia és a frekvenciastabilitás optimális beállítása együttesen nem valósítható meg. A CX1 és CX2 kapacitások értékének növelésekor javul a frekvenciastabilitás, a rezgés periódusideje viszont csökken. Minden alkalmazás esetében meg kell találni a kompromisszumot. Általában a 20 pF-os kapacitások megfelelőek. Legfeljebb 100 pF-ig növelhető ez az érték.

2.1.4 Külső szinkronizálás

A nagyobb frekvenciastabilitáshoz vagy nagyobb rendszerek szinkronozásakor külső - TTL szintű - órajellel is vezérelhető a 8051-es típusú mikrokontroller. Ilyenkor a jelet az XTAL2 bemenetre kell vezetni. Az XTAL1 lábat pedig az U_{ss}-re (föld) kell kötni. Mivel a fázisfordító műveleti erősítő kimeneti ellenállása nagy, ezért nem terheli a csatlakozó órajelet.
A 80C51-es típusjelű, CMOS-változatú mikrokontroller esetén - az előbbiektől eltérően - az órajelet az XTAL 1 lábra kell vezetni, az XTAL2 láb pedig szabadon hagyható. A külső órajel alkalmazásakor figyelembe kell venni, hogy



2.3 ábra Külsõ órajelet elõállító oszcillátor

a 8051-es bemenetei nem TTL-kompatíbilisek. Ezért a TTL jel csatlakoztatásakor még egy - 4,7 kΩ körüli - felhúzó ellenállást is alkalmazni kell.

A 8051-es típusú mikrokontrollerekhez külső órajel is használható.

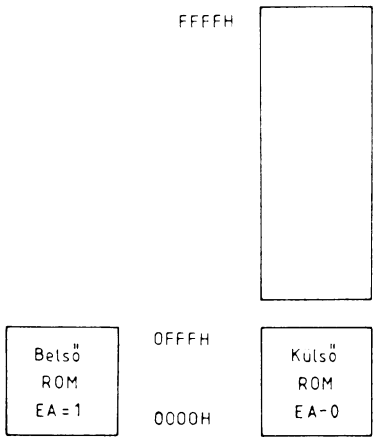
A 2.3. ábrán látható egy példa a külső órajel-generátor felépítésére.

2.2 Belső és külső tárolók

A 8051-es típus 8 bites mikrokontroller. 64 Kbájt külső adatmemóriát és 64 Kbájt programtárolót tud kezelni. Az utóbbi egy része a chipen belül is elhelyezkedhet. Lényeges külön-külön használni a program- és adatmemóriát.

2.2.1 Belső programtároló

Ha van belső programtároló (ROM), akkor 4 Kbájt kapacitású. Két különböző változata: a 8051 típus, amelyben van ROM, ill. a 8031-es típusjelű, amely ROM nélküli (2.4. ábra). Mint ahogy már említettük, a belső programmemória ROM típusú. A ROM-ba a gyártás során maszkolják a programot, és azt a későbbiekben már nem lehet megváltoztatni. A maszkprogramozás előnye, hogy részben, esetleg teljesen elhagyható a külső programmemória. A feladathoz illesztett maszkot és programját azonban drága fejleszteni, ezért ezt a változatot csak nagy darabszám esetén kifizetődő alkalmazni (pl. videomagnókban). A 8051 a 8031-nek megfelelően is használható. A belső programtároló az EA jelű bemenetre (31 láb) adott vezérléssel be- vagy kiiktatható. A lábra adott magas (HIGH) szintű jel esetében a mikrokontroller a belső programtárból hívja az utasításokat, ameddig a programkódok száma kevesebb, mint 4096 (4 Kbájt).



2.4 ábra A belsõ és a külsõ ROM

Az e feletti kódokat pedig már a külső memóriából veszi. Alacsony (LOW) szintű vezérlés esetén mindig a külső programmemóriát használja. A 8031-es típusnál ezt a lábat mindig alacsony szintre kell kötni, mivel nincs belső programtárja. Az adatlap mindenkor ezt a kötést írja elő. A valóság viszont sokszor ettől eltérő. A kereskedelemben kapható 8031-esjelű mikrokontrollerek esetenként olyan 8051-es változatok, amelyekben a belső ROM üres vagy egy tetszőleges programot tartalmaz. Alapvetően ennek az az oka, hogy a gyártásban olcsóbb egy típust - a 8051-est - gyártani. Ha szükséges, akkor programozzák a ROM-ot, egyébként üresen hozzák forgalomba. Sokszor kaphatók felhasználói programmal is 8051-es típusok. Ez semmiképpen sem negatívum, sőt emiatt esetleg olcsóbbak. A mikrokontrollerek specifikált jellemzői minden változatban azonosak.

2.2.2 A 8751-8051-es típus EPROM-mal

A 8051-es áramkör egy másik változata a 8751-es típusjelű áramkör. Ehhez hasonló a 8752-es típus is. Ezekbe a típusokba nem ROM-ot, hanem EPROM-ot integráltak. A programfejlesztésnél a tesztelés és az esetleges módosítás lehetőségét biztosítja az EPROM. A végleges maszk elkészítése előtt, ha programhiba fordul elő, az EPROM törölhető és újraírható. Így csak a valóban helyes program kerül maszkolásra a gyártásban.
A 8751/8752-es típusjelű mikrokontrollerek meglehetősen drágák. Programfejlesztésre is használhatók.

A programfejlesztéshez használt változatokban a ROM helyett EPROM van.

A felsorolt változatok mind a nagy integráltságú áramkörök családjába tartoznak.

2.2.3 Belső adattároló

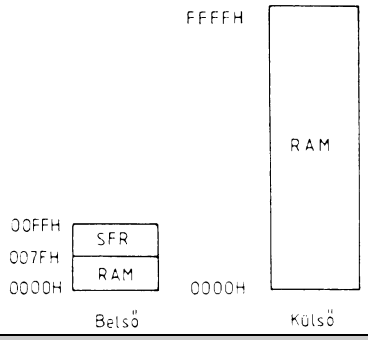
A 8051-es mikrokontroller-család mindegyik tagjában - a belső ROM (EPROM) mellett - RAM is van. A belső adat- és programtároló nagysága a család egyes tagjainál különböző. A fontosabb típusok tárméreteit foglalja össze a 2.1. táblázat.

2.1. táblázat. A 8051-es típusjelű mikrokontroller-család elemeinek adat- és programtároló mérete

Típus	Adattár, bájt	Programtár
8051	128	4 Kbájt ROM
8052	256	8 Kbájt ROM
80C51	128	4 Kbájt ROM
8031	128	Nincs
8032	256	Nincs
80515	256	8 Kbájt ROM
80535	256	Nincs
8751	128	4 Kbájt EPROM
8752	256	8 Kbájt EPROM
80517	256	8 Kbájt
80537	256	Nincs

Már itt felhívjuk a figyelmet, hogy a külső RAM csak közvetetten érhető el (1. a 2.5.2. pontot). A belső RAM-ban számos, ún. speciális funkcióregiszter (SFR) van. Ezek feladatait a 2.6. szakaszban tárgyaljuk részletesen. A belső RAM a 00H és a 7FH címtartományban helyezkedik el. A 8051-es típusú mikrokontrollerben az SFR-ek a 80H és az FFH címek között vannak. A 8052-es típusú mikrokontrollerben az SFR-ekkel párhuzamosan egy második - 128 bájt

kapacitású - RAM is található. Az SFR regiszterek és az ezekkel azonos című memóriacellák más címzési móddal érhetők el. A belső RAM mellett 64 Kbájt kapacitású külső RAM is alkalmazható (a 0H és az FFFFH közti címtartományban). A 2.5. ábra a 8051-es felsorolt RAM-területeit szemlélteti. A címátfedés nem okoz bajt. A belső és a külső RAM más-más címzési móddal érhető el.



2.5. ábra. A RAM részei

A külső adatmemória használatakor figyelembe kell venni, hogy a 8051esnél nincs várakozó állapot (Wait-state).

A rövid hozzáférési idő miatt gyors tárolókat kell alkalmazni.

2.3 A fogyasztás csökkentése

A mikrokontroller nagyszámú elemből összeintegrált, bonyolult áramkör, amellyel nagyon gyakran építenek hordozható készülékeket. Az ilyen készülékekben a tápellátást elem biztosítja, ezért a mikrokontroller áramfelvétele lényeges tényező. Ugyanezen okból készülnek az NMOS-változat mellett CMOS-kialakítású 8051-es változatok is. E technológia lényeges hátránya, hogy nagyobb a chip felülete. A CMOS-változatok magasabb árát viszont ellensúlyozza a kisebb fogyasztás.

A CMOS-változatok áramfelvétele, ezzel együtt a fogyasztás is erősen függ az órajel frekvenciájától.

A következő összehasonlítási értékek az (U_{cc} = 5 V-os tápfeszültséghez tartoznak és csak közelítőek. Ugyanakkor gyártónként is szórhatnak. Az egyes gyártmányok jellemző értékeit az adatlapok tartalmazzák. A CMOS-változatok maximális áramfelvétele 20...30 mA, az NMOS-változatoké pedig 120...160 mA.

Frekvencia, MHz	I_{cc} max ,mA
0,5	2,2
3,5	5,7
8,0	11,0
12,0	16,0
16,0	20,5

Az előzőekben leírtak miatt lényeges a feladathoz feltétlenül szükséges órajelfrekvencia megválasztása. Különösen a hordozható készülékeknél döntő ez a szempont.

A CMOS-változatok alkalmazásakor kis áramfelvétel kis órajelfrekvencia esetén érhető el.

A 8051-es CMOS-változata további két lehetőséget biztosít az átlagos áramfelvétel csökkentésére. Ezek csak akkor alkalmazhatók, ha a kontrollerrel kialakított áramkörnek nem kell folytonosan üzemelnie. Ilyenkor a kontroller Power-down vagy Idle üzemmódban működtethető.

2.3.1 Power-down üzemmód

A teljesítményveszteség csökkentésének egyik lehetősége az ún. Power-down (csökkentett teljesítményű) üzemmód alkalmazása.

A Power-down üzemmódban a belső oszcillátor leáll, ezáltal az összes belső funkció is megszűnik.

A belső RAM tartalma azonban nem változik, vagyis az éppen aktuális értékek megmaradnak. A portok lábain azok az értékek maradnak, amelyek a megfelelő SFR-ben voltak az átkapcsoláskor. Az ALE (Address-Latch-Enable) és a PSEN (Peripherial-Store-Enable) jelek alacsony (LOW) szintűek lesznek. A Power-down üzemmódot csak a hardver RESET jel szünteti meg. Vigyázni kell arra, hogy mivel a RESET az SFR regisztereket is inicializálja, ezért eredeti tartalmuk elvész, ugyanakkor a belső RAM tartalma nem változik meg. Emiatt a Power-down üzemmódba lépés előtt feltétlenül biztosítani kell az SFR regiszterek tartalmának mentését a belső RAM-ba. A Power-down üzemmód beállítása a PD bit (PCON.1) 1-be írásával történik. Ez a PCON (Power Control Register) nevű, speciális funkcióregiszterben van. Ezt az üzemmódot rendszerint a 8051-es bázisú vezérlés feszültségkimaradáskori védelme során használják a következők szerint. Meghatározott feszültségesés egy megszakítást kér. A megszakításrutin a szükséges adatokat a belső RAM-ba menti és beállítja a PD bitet. A tápfeszültség visszatérése RESET jelet ad a kontrollernek, ezzel befejeződik a Power-down üzemmód. A Power-down üzemmódban - kapcsolástechnikai

megoldásokkal - csökkenthető az U_{cc}, ezáltal minimalizálható a fogyasztás. Arról viszont gondoskodni kell, hogy normál üzemben az U_{cc} a specifikáció szerinti értékű legyen.

2.3.2 Idle üzemmód

Az áramfelvétel az Idle üzemmódban is - a Power-down módhoz hasonlóan jelentősen csökkenthető.

Ekkor - a Power-down móddal ellentétben - az órajel tovább működik, a CPU viszont nem működik.

Az időzítő, a megszakítás és a soros adatátviteli port továbbra is működik, ehhez szükséges az órajel. Az Idle üzemmódba kapcsolás a PCON.0 bit 1-be állításával végezhető. Az üzemmód előnye, hogy a 8051-es nemcsak a RESET-tel, hanem megszakítással is visszaállítható a normál üzemmódba. A visszaállító megszakításrutinnak csupán a PCON.0 bitet kell törölnie a mikrokontroller újbóli üzemeltetéséhez.

A 80C51-es típus szoftver úton kapcsolható a Power-down vagy az Idle üzemmódba. Ekkor a kontroller csökkentett áramfelvételű, inaktív állapotba kerül.

2.3.3 A belső RAM áramának biztosítása

A Power-down vagy az Idle üzemmódban a belső RAM tárolja a szükséges adatokat. A tároláshoz - az üzemszünet alatt is - meghatározott nagyságú I_{cc}-t, ún. nyugalmi áramot kell biztosítani. A 8051/52-es típusoknál a nyugalmi tápellátást az RST/ VPD lábra kell kötni. A nyugalmi tápellátás csak a RESET jel megérkezése után kapcsolható le. A 80515-ös típusnak külön VPD (4.láb) bemenete van. A VPD lábra a nyugalmi tápfeszültséget csak akkor kell rákapcsolni, ha a csökkenő U_{cc} a VPD alá kerül. Az egyes mikrokontrollerek nyugalmiáram-szükséglete eltérő. A legrosszabb esetben szükséges értékek (Worst-Case) a következők:

Típus	Áramfelvétel
8051/ 8031	10 mA
8052/8032	15 mA
80515/80535	3 mA
80C51/80C31	50 μA

2.4 A CPU időzítései

2.4.1 Belső időviszonyok

A belső oszcillátor kimenete, ill. a bevezetett órajel egy belső ütemgenerátort hajt meg. Ez állítja elő a szükséges belső ütemeket.

A belső ütemezőjel frekvenciája a mindenkori oszcillátorfrekvencia fele.

A belső jel ütemezi a műveleteket.
A 8051/8031-es típusoknál minden gépi ciklus hatütemű, amelyeket S1-től S6-ig jelölünk. Mindegyik ütem két oszcillátorperiódusból - P1 és P2 - áll. Minden gépi ciklus P, tehát 12 órajel-periódusból áll, vagyis az S1P1-gyel kezdődik és az S6P2-vel fejeződik be. A gépi ciklus periódusideje (P) a következő összefüggés alapján számolható ki: P=12/f.
Az f-et Hz-ben, ill. MHz-ben behelyettesítve, a P periódusidőt s-ban, ill. μs-ban kapjuk. Pl. ha egy kontrollert 8 MHz-es órajellel működtetünk, akkor

$$P = \frac{12}{8 * 10^6} s = 1.5 \mu s$$

hosszú lesz egy gépi ciklus. A ciklusidő ismeretében egy adott program futási ideje is kiszámolható. A 8051/8031-es kontrollerek utasításai 1-3 bájt hosszúak, és végrehajtásuk egy vagy két ciklus alatt történik. Kivétel a szorzás (MUL), ill. az osztás (DIV), amelyeket a mikrokontroller 4 ciklus alatt hajt végre. Az utasítások ciklusszámát a 3. fejezetben adjuk meg egy listában.

2.4.2 A külső tárolók vezérlőjelei

Az Sl-S6 ütemekben a P1, P2 órajelciklusok szerint változik minden külső tárolóáramkörök működtetéséhez szükséges - vezérlőjel. Ezek a következők:

ALE	(Address Latch Enable)
PSEN	(Program Store Enable)
RD	(Read)
WR	(Write)

A jelek ciklusonként egyszer vagy többször jelennek meg, az S1P1-S6P2 ütemek valamelyikében. Az egyes jelek tényleges változásával a 2.10. szakaszban foglalkozunk.

2.5 Időzítő/számláló

A 8051-es család tagjaiban különböző számú időzítő/számláló áramkör van. Ezekhez még további funkciók is kapcsolódhatnak.

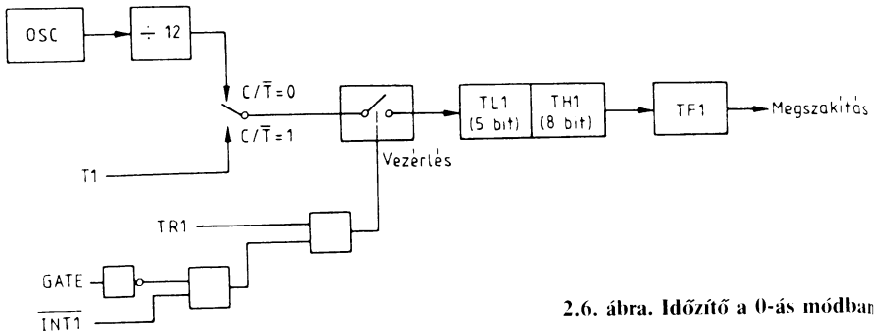
Típus	Időzítő/számlálók száma
8051	2
8052	3
80513	3
80515	3

2.5.1 A T0 és T1 időzítő/számlálók és üzemmódjaik (0-3)

A T0 és a Tl áramkörök időzítőként vagy eseményszámlálóként használhatók. A működési módot a TMOD regiszter 2. és 6. C/T jelű bitjeivel (TMOD.2, ill. TMOD.6) kell beállítani. A logikai 1-hez a számláló üzemmód tartozik.
Az időzítő üzemmódban a számtartalom gépi ciklusonként eggyel nő. Ebből következik, hogy az időzítés alapegysége az oszcillátor periódusidejének a 12-ed része.
A számláló üzemmódban mindegyik számláló a hozzá tartozó bemenetre - T0, Tl, (T2) - kapcsolt jel lefutó élénél inkrementálódik. A 8051-es ezeket a bemeneteket mindig csak az S5P2 ütemben mintavételezi (1. a 2.9. szakaszt). A működésből következik, hogy egy lefutó él (HIGH-LOW szintváltás) érzékeléséhez legalább két gépi ciklus kell. A maximális számlálási frekvencia tehát az órajel-frekvencia 1/24-e. A 12 MHz-es oszcillátor használatakor a számlálási frekvencia legfeljebb 500 kHz. Természetesen ez csak 50 %-os kitöltési tényező esetén érvényes. Eltérő kitöltésnél a rövidebb ideig tartó szint határozza meg a frekvenciahatárt.
Az áramkört kiegészítő, szoftverből vezérelhető kapu előnyösen használható a pontos eseményszámláláshoz. A kapuzás a TR0 és a TR1 bitekkel (TMOD.7 és a TMOD.3) történik. A 0-ás, ill. 1-es jelű időzítő/számláló áramkörök négy különböző üzemmódban működtethetők. Az üzemmódokat az 1-esnél a TMOD.4 és a TMOD.5, ill. a 0-ásnál a TMOD.0 és TMOD.1 bitekkel kell kiválasztani (1. a 2.6.7. pontot). Az egyes üzemmódok a következők:

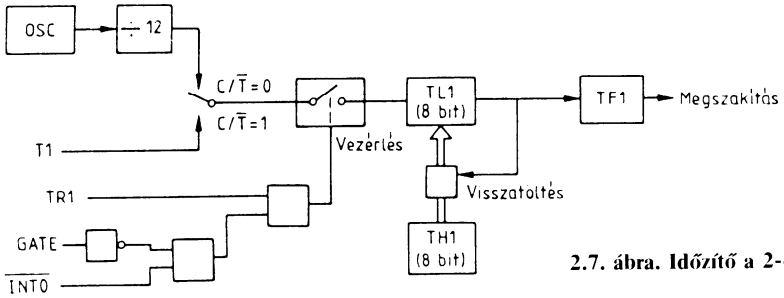
A maximális frekvencia „időzítés” üzemmódban $f_{osc}/12$, számlálós üzemmódban pedig $f_{osc}/24$.

0-ás üzemmód
A 0-ás üzemmód a 8048-as mikrokontroller működésével egyezik meg. A TH0 (TH1) regiszter 8 bites, míg a TL0 (TL1) csak 5 bites. Az áramkörök tehát egy 32-es elosztóval rendelkező. 8 bites számlálóként használhatók (2.6. ábra).



2.6. ábra. Időzítő a 0-ás módban

A számláló túlcsordulása 1-be írja a megszakításjelzőt (Timer-Interrupt-Flag TF0/TF1). A számlálandó impulzusokat a külső és a belső vezérlés együttesen kapcsolja a számlálóra. A 2.6. ábrának megfelelő feltétel, hogy a TR0/TR1 bit 1 szintű, ugyanakkor az üzemmódot vezérlő TMOD regiszter TMOD.3/TMOD.7 bitje alacsony értékű, vagy az INT0/INT1 csatlakozási ponton magas szintű legyen. A számlálók külső (hardver) és belső (szoftver) vezérelhetősége széles körű, rugalmas felhasználást biztosít.

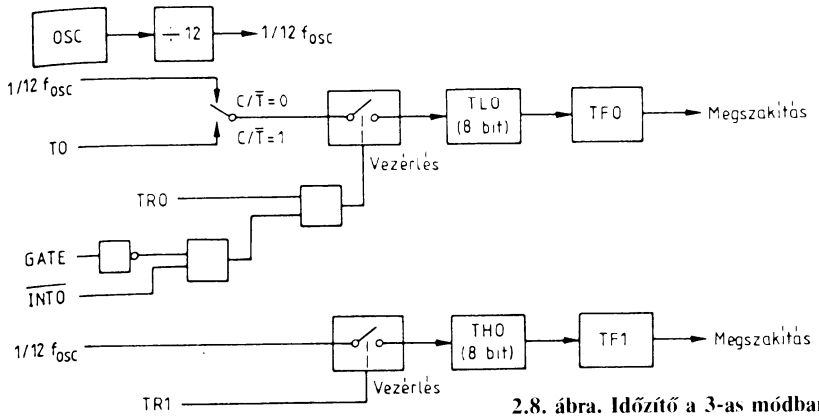


2.7. ábra. Időzítő a 2-es módban

1-es üzemmód
Az 1-es üzemmód hasonló a 0-áshoz, azzal az eltéréssel, hogy a számláló 16 bites. TL és TM sorba (kaszkádba) van kapcsolva.

2-es üzemmód
A számláló a 2-es üzemmódban 8 bites, de automatikus újratöltés is történik (2.7. ábra). A TL0/TL1 regiszter túlcsordulása beállítja a megfelelő megszakításjelzőt (TF0/TF1), ugyanakkor beírja a TH0/TH1 regiszter tartalmát a TL0/TL1 regiszterbe. A TH0/TH1 regiszter tartalma nem változik. Ezek felülírása szoftver úton végezhető. A megoldással programból változtatható késleltetés valósítható meg.

3-as üzemmód
A 3-as üzemmódban a TL0 és a TH0 regiszterek önálló 8 bites számlálóként használhatók és egymástól teljesen függetlenül dolgoznak (2.8. ábra).

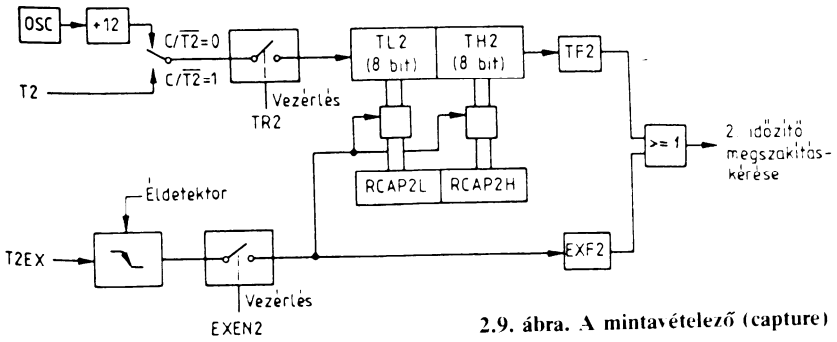


2.8. ábra. Időzítő a 3-as módban

Ebben az üzemmódban az 1-es időzítő/számláló áramkör tartalma változatlan marad. Az üzemmódot akkor használják, ha több funkció is szükséges. (Megjegyzés: A TH0 csak időzítőként működtethető!)

2.5.2 A T2-es időzítő/számláló. Capture, Auto-Reload és Baudrate üzemmódok

T2 16 bites időzítő/számláló áramkör - a 8051-es típus kivételével - a család minden tagjában van. A T2 sokkal összetettebb felépítésű áramkör, mint a T0 és a T1. Alapvetően a T2 is időzítésre vagy eseményszámlálásra használható. Az üzemmódot a T2CON.1 bittel lehet kiválasztani. Ugyanakkor az áramkör további funkciókkal van bővítve. A 80515-ös típus T2 áramkörének - az alaptípusétól eltérő - funkcióbővítéseit a 4. fejezetben tárgyaljuk. A T2 három különböző - Capture, Auto-Reload és Baudrate-Generátor - üzemmódban használható. Az üzemmód kiválasztását a T2CON.0, a T2CON.2, valamint a T2CON.4 és T2CON.5 bitek végzik (részletesen a 2.10. táblázatban). A TR2 bit (T2CON.2) ki- vagy bekapcsolja a T2-t. Amikor az RCLK (T2CON.5) és a TCLK (T2CON.4) jelzőbitek értéke 1, a T2 - a CP/RL2 bit értékétől függetlenül – Baudrate-Generátor üzemmódban

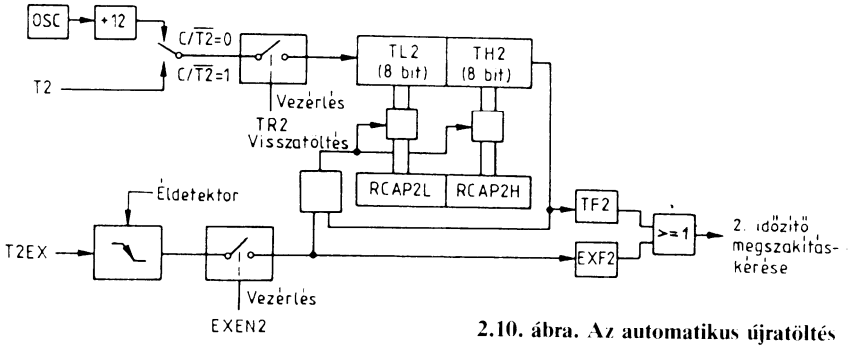


2.9. ábra. A mintavételező (capture) mód

dolgozik. Ha a TCLK és az RCLK bitek értéke 0, akkor a CP/RL2 bit értéke határozza meg a további két üzemmód valamelyikét. A bit 1 értékével a Capture, míg a 0-val az Auto-Reload üzemmódot választjuk ki.

Capture üzemmód
A Capture üzemmódban a TH2 és a TL2 regiszterek tartalma az RCAP2H, ill. az RCAP2L regiszterekbe átírható. Az átírást a T2EX bemenetre (2 láb) adott lefutó él vezérli. A vezérlés az EXEN2 bittel engedélyezhető vagy tiltható (a logikai 1 engedélyez). Az adatátvitellel egyidejűleg a T2 megszakítása is beíródik. Az RCAP2H és a RCAP2L regiszterek tartalma szoftverből állítható. Mivel a megszakítási folyamat hardvervezérelt, ezért ebben az üzemmódban a külső esemény idejét vagy frekvenciáját mérhetjük.

Auto-Reload üzemmód
Az Auto-Reload üzemmódot a 2.10. ábra szemlélteti. Mint látható, további választási lehetőség van



2.10. ábra. Az automatikus újratöltés

Amikor az EXEN2 bit értéke 0, akkor a T2 minden túlcsordulásakor az RCAPx regiszterek tartalma átíródik. A programozott számtartalom 0-tól eltérő, tetszőleges 16 bites érték lehet. Ebben az üzemmódban - az RCAPx regiszterekbe írt értéktől függő - változtatható frekvenciájú jel állítható elő. Az EXEN2 bit logikai 1 értékénél az átírást a T2EX jelű bemenetre érkező negatív él is vezérli.

Baudrate-Generátor üzemmód
Ennek az üzemmódnak a soros adatátvitel ütemezésében van szerepe. Ezért ott tárgyaljuk részletesen.

2.6 A speciális funkcióregiszterek (SFR-ek)

Ezen belső regiszterek egy részének funkciója (mint pl. az Akkumulátor vagy a Program-Status-Word) azonos a mikroprocesszorokban használt regiszterekével. A mikrokontrollerek számos kiegészítő funkcióval bővültek.

2.2 táblázat. A speciális funkcióregiszterek (SFR-ek)

Szimbólum	Jelentés	Cím, hexadecimálisan
*p0	Port 0	80
SP	Stack-Pointer	81
DPL	Data-Pointer; LOW bájt	82
DPH	Data-Pointer; HIGH bájt	83
TCON	Timer/Counter Control	88
TMOD	Timer/Counter-Mode Control	89
TL0	Timer/Counter 0; LOW bájt	8A
TL1	Timer/Counter 1; LOW bájt	8B
TH0	Timer/Counter 0; HIGH bájt	8C
TH1	Timer/Counter 1; HIGH bájt	8D
*P1	Port 1	90
PCON	Power Control	97
*SCON	Serial Interface Control	98
SBUF	Serial Data-Buffer	99
*P2	Port 2	0A0
IE	Interrupt Enable Control	0A8
*P3	Port 3	0B0
*IP	Interrupt Priority Control	0B8
+*T2CON	Timer/Counter 2 Control	0C8
+RCAP2L	Timer/Counter 2 Capture-Register; LOW bájt	0CA
+RCAP2H	Timer/Counter 2 Capture-Register; HIGH bájt	0CB
+TL2	Timer/Counter 2; LOW bájt	0CC
+TH2	Timer/Counter 2; HIGH bájt	0CD
*PSW	Program-Status-Word	0D0
*ACC	Akkumulátor	0E0
*B	B-regiszter	0F0

Ezek vezérléséhez számottevően megnövelték az SFR regiszterek számát is. E regiszterek közül soknak az egyes bitjei is elérhetők egyedileg (bitcímzéssel). A 2.2. táblázatban *-gal jelöltük a bitenként is címezhető regisztereket. A táblázatban + jellel ellátott regiszterek csak a 8052/8032-es típusokban, ill. ezek továbbfejlesztett változataiban vannak. A 8051/8031-es típus ezeket az SFR-eket nem tartalmazza.

A 8051-es család minden belső funkciója az SFR-ek segítségével vezérelhető.

A címterületen szándékosan hagytak lyukakat a későbbiekben fejlesztett változatok - mint pl. 80515-ös vagy a 80517-es - további regisztereinek az illesztése céljából.

2.6.1 Bitcímek

A speciális funkcióregiszterek (SFR-ek) legtöbbjének bitjei külön-külön is címezhetőek. Az egyes bitek jelentését a 2.6.3. ponttól kezdve a 2.3.15. pontig részletesen tárgyaljuk. A bitek megnevezését és címeiket a következő táblázatban foglaltuk össze.

Port 0	P0.7	P0.6	P0.5	P0.4	P0.3	P0.2	P0.1	P0.0
	87H	86H	85H	84H	83H	82H	81H	80H
TCON	TF1	TR1	TF0	TR0	lEl	IT1	IE0	IT0
	8FH	8EH	8DH	8CH	8BH	8AH	89H	88H
Port 1	P1.7	P1.6	P1.5	P1.4	P1.3	P1.2	P1.1	P1.0
	97H	96H	95H	94H	93H	92H	91H	90H
SCON	SM0	SM 1	SM2	REN	TB8	RB8	TI	RI
	9FH	9EH	9DH	9CH	9BH	9AH	99H	98H
Port 2	P2.7	P2.6	P2.5	P2.4	P2.3	P2.2	P2.1	P2.0
	A7H	A6H	A5H	A4H	A3H	A2H	A1H	A0H
IE	AE	-	-	ES	ET1	EX1	ET0	EX0
	AFH			ACH	ABH	AAH	A9H	A8H
Port 3	P3.7	P3.6	P3.5	P3.4	P3.3	P3.2	P3.1	P3.0
	B7H	B6H	B5H	B4H	B3H	B2H	B1H	B0H
IP				PS	PT1	PX1	PT0	PX0
				BCH	BBH	BAH	B9H	B8H
T2CON	TF2	EXF2	RCLK	TCLK	EXEN2 -	TR2C	CP	
	CFH	CEH	CDH	CCH	CBH	CAH	C9H	C8H
PSW	CY	AC	FO	RS1	RS0	OV	-	P
	D7H	D6H	D5H	D4H	D3H	D2H		D0H
ACC	E7H	E6H	E5H	E4H	E3H	E2H	E1H	E0H
B	F7H	F6H	F5H	F4H	F3H	F2H	F1H	F0H

2.6.2 Alapértelmezés szerinti értékek

2.3 táblázat. A mikrokontroller regisztereinek RESET utáni értéke

SFR	Kezdőérték	Regiszterhossz bájtban
PC	0000H	2
ACC	00H	1
B	00H	1
PSW	00H	1
SP	07H	1
DPTR	0000H	2
P0-P3	FFH	1
IP	XXX00000B	1
IE	0XX00000B	1
TMOD	00H	1
TCON	00H	1
TH0	00H	1
TL0	00H	1
TH1	00H	1
TL0	00H	1
SCON	00H	1
PCON (CMOS)	0XXX0000B	1
PCON (NMOS	0XXXXXXXB	1

A mikrokontroller SFR regisztereibe a RESET jel hatására meghatározott alapérték íródik (2.3. táblázat).
A 2.3. táblázatban szereplő X határozatlan (előzetesen nem definiált) értéket jelent. A PCON regiszterbe íródó alapérték a CMOS-, ill. a NMOS változatokban eltérő. Ez abból adódik, hogy Power-down mód csak a CMOS-változatban van. (A 2.3. szakaszban erről már volt szó.) Az SBUF regiszter alapértéke határozatlan.

2.6.3 Veremmutató (Stack-Pointer)

A veremmutató (SP) funkciója a mikrokontrollerekben is hasonló, mint a mikroprocesszorokban. A RESET jel után az SP-be 07H érték íródik. Tehát a veremterület a 08H-nál kezdődik és felfelé növekszik. Ugyanakkor adott a lehetőség arra, hogy az SP regisztert tetszőleges értékkel töltsük fel. A verem a 256 bájtos RAM-terület tetszőleges helyén kijelölhető.

2.6.4 Adatmutató (Data-Pointer)

Az adatmutató (DP) 16 bites regiszter, amely az alacsony (LOW) bájtból (DPL) és a magas (HIGH) bájtból (DPH) áll. A két regiszter tartalma a megfelelő utasításokkal külön-külön bájtként, vagy pedig együtt 16 bites szóként is használható. Rendszerint az adatmemóriába való írásnál vagy olvasásnál közvetett címzéshez használjuk.

2.6.5 Fogyasztásvezérlő (Power-Control) regiszter (PCON)

A PCON regiszterrel különböző csökkentett fogyasztású üzemmódok állíthatók be. A regiszter egyes bitjeinek funkcióját a 2.4. táblázatban foglaltuk össze.

2.4. táblázat. A Power-Control regiszter (PCON) tartalma

Szimbólum	Bit	Jelentése
IDL	PCON.0	Idle.Mode bit. 1-be írása bekapcsolja az Idle módot.
PD	PCON.1	Power-down bit. Ez a bit kapcsolja a Power-down módot.
GF0	PCON.2	Általános célú jelzőbit (General-Purpose-Flag) 0.bit
GF1	PCON.3	Általános célú jelzőbit (General-Purpose-Flag) 1.bit
-	PCON.4	Foglalt
-	PCON.5	Foglalt
-	PCON.6	Foglalt
SMOD	PCON.7	E bit 1-be írása a soros port átviteli sebességét (Baudrate) az 1, 2 és 3-as üzemmódokban megkettőzi

Az IDL és a PD bitek egyidejű beállításakor a mikrokontroller a Power-down üzemmódba kerül. Mindkét GF jelű bit tetszőleges célra használható. A regiszter PCON.4, 5 és 6 bitjeit nem szabad felülírni, mert ezáltal a kontroller meghatározatlan állapotba kerül. A RESET után a PCON regiszterbe a 0XXX0000 bináris érték íródik.

2.6.6 Timer-Control regiszter

A TCON regiszter feladata a T0 és T1 időzítő/számlálók vezérlése.

2.5. táblázat. A Timer-Control regiszter (TCON) tartalma

Szimbólum	Bit	Jelentése
IT0	TCON.0	Az Interrupt 0 vezérlőbitje. 1-be írt érték esetén a megszakítás a külsőjel negatív élénél megy végbe, egyébként pedig LOW szintnél. Írása, törlése szoftverből történik.
IE0	TCON.1	Az Interrupt 0 élflagje. Az INT0 bemenetre jutó negatív él írja 1-be. Törölni a megszakítást kiszolgáló rutinban kell.
IT1	TCON.2	Az Interrupt 1 vezérlőbitje. Megegyezik az IT0 funkciójával.
IE1	TCON.3	Az Interrupt 1 élflag. Megegyezik az IT0 funkciójával.
TR0	TCON.4	A Timer 0 futását vezérlő bit. Szoftverből írható/törölhető. A bit 1 értéke a Timer/Counter 0-t indítja, a 0 pedig leállítja.
TF0	TCON.5	A Timer 0 Overflow-flagje.
TR1	TCON.6	Funkciója megegyezik a TR0-éval. de a Timer 1-re vonatkozik.
TF1	TCON.7	Funkciója megegyezik a TF0-éval

2.6.7 Timer-Modus regiszter

A T0 és Tl időzítő/számlálók különböző üzemmódjai állíthatók be a TMOD regiszter egyes bitjeivel (2.6. táblázat). Ezekről már írtunk a 2.4.1. pontban.

2.6. táblázat. a Timer-Modus regiszter (TMOD) tartalma

Szimbólum	Bit	Jelentése
M0	TMOD.0	A T0 időzítő/számláló üzemmódját meghatározó bitek.
M1	TMOD.1	
C/T	TMOD.2	A T0-t időzítő vagy számláló funkcióba állítja. Logikai l értéknél a T0 külső jelet számlál.
Gate	TMOD.3	Kapuvezérlő bit. A bitet 1-be írva működik a T0. ha az INT0 és TR0 is 1 szintű (pl. l. a 2.6. ábrát).
M0	TMOD.4	A T1 időzítő/számláló üzemmódját meghatározó bitek.
M 1	TMOD.5	
C/T	TMOD.6	A T1-et időzítő vagy számláló funkcióba állítja. Logikai 1 értéknél a Tl külső jelet számlál.
Gate	TMOD.7	Kapuvezérlő bit. A bitet 1-be írva működik a Tl, ha az INT1 és a TRl is 1 szintű (l. pl. a 2.6. ábrát).

Az M1 és az M0 jelű bitek kombinációja határozza meg a T0 (TMOD.0 és TMOD.1), ill. a T1 (TMOD.4 és TMOD.5) időzítő/számlálók üzemmódját. A 2.5. szakaszban ezt már ismertettük. Itt csupán - magyarázat nélkül - táblázatba foglaljuk a leírtakat.

M1	M0	Üzemmód
0	0	0-ás mód
0	1	1-es mód
1	0	2-es mód
1	1	3-as mód

2.7. táblázat. A Serial-Port-Control-Regiszter (SCON) tartalma

Szimbólum	Bit	Jelentése
RI	SCON.0	A vételi megszakítás jelzője. A hardver állítja 1-be. A megszakítást kiszolgáló rutinban kell törölni![1]
TI	SCON.1	Az adás megszakításjelzője. A stopbit kezdete. ill. az M0 üzemmódban a 8. bit írja 1-be. A megszakítást kiszolgáló rutinban kell törölni.[2]
RB8	SCON.2	A vett adat 9. bitje a 9 bites adatátviteli üzemmódban.
TB8	SCON.3	A küldött adat 9. bitje a 9 bites adatátviteli üzemmódban.
REN	SCON.4	E bit beírásával vagy törlésével lehet szoftverből indítani, ill. leállítani a vételt.
SM2	SCON.5	A beállított üzemmód különböző funkcióinak beállítását végzi.
SM1	SCON.6	Az SM0 és SMl bitek állítják be a soros port üzemmódjait.
SM0	SCON.7	Szoftver kezeli.

2.6.8 Serial-Port-Control regiszter

A 8051-es család minden tagjában van szabványos soros adatátviteli port. Az adatátviteli üzemmódok az SCON regiszter egyes bitjeivel állíthatók be (2.7. táblázat). Az SCON regiszterben találhatók a 9 bites adatátviteli üzemmód 8-as adatbitjei (TB8 az adásnál és RB8 a vétel esetén).

2.6.9 A soros port adatpuffere

Az SBUF regiszter tárolja az adás előtt a kiviendő bájtot, ill. ide íródik be a soros portról a vett bájt. A regiszter tehát mind a vételnél, mind pedig az adásnál csak átmenetileg tárol egy bájtot. Vételkor a felhasználónak kell arról gondoskodni, hogy a vétel után azonnal kiolvassa az SBUF regiszter tartalmát, és más helyre tárolja. Ennek elmulasztása esetén a soros vonalon érkező újabb bájt felülírja az SBUF-t, és így annak korábbi tartalma így elvész. A teljes bájt vétele, ill. az üzemmódtól függő kilencedik bit után a megszakítást kérő flag (Interrupt-Request-Flag) 1-be íródik. A vett bájt átvitelét a megszakítást kiszolgáló rutinnak kell végrehajtania.
A következőkre kell figyelni:

Az SBUF regiszternek csak egy címe van, bár fizikailag két különböző SBUF regiszter szolgál a vétel és az adás adatainak tárolására.

Az SBUF regiszterbe kerülő nyolc adatbit mellett egy kilencedik belső bit a vezérlést szolgálja. Az adóregiszternél ez a SEND, amelynek 1-be írásakor indul a tényleges adás. A vevőregiszternél ez a RECEIVE bit (l. később, a 2.12. ábrán).

[1] RI=1 A vételi SBUF megtelt
[2] TI=1 Az adási SBUF kiürült

2.6.10 A megszakítás-engedélyező (Interrupt-Enable) regiszter (IE)

A 8051-es mikrokontroller megszakításainak engedélyezése az IE regiszter tartalmának megfelelő beírásával végezhető. Az EA (IE.7) bit az általános megszakítás-engedélyező. Amikor EA = 0, akkor mindegyik megszakítás tiltott. Az általános engedélyezés (EA = 1) mellett az IE regiszter megfelelő bitjét (2.8. táblázat) is be kell írni a kívánt megszakítás engedélyezéséhez.

2.6.11 Megszakításprioritás- (IP) regiszter

A 8051-es megszakítás-vezérlő része két prioritásszintet különböztet meg. Az egyes megszakításforrások prioritását az IP regiszter megfelelő beírásával végezhetjük. Az 1 jelenti a magasabb prioritást (2.9. táblázat).

2.8. táblázat. A megszakítás-engedélyező regiszter (IE) tartalma

Szimbólum	Bit	Jelentése
EX0	IE.0	a külső INT0 engedélyezése
ET0	IE.1	a T0 időzítő túlcsordulásmegszakítás engedélyezése
Exl	IE.2	a külső INT1 engedélyezése
ET1	IE.3	a T1 időzítő túlcsordulásmegszakítás engedélyezése
ES	IE.4	a sorosvonal-megszakítás engedélyezése
ET2	IE.5	a T2 időzítő túlcsordulásmegszakítás engedélyezése
-	IE.6	Foglalt
EA	IE.7	általános megszakítás-engedélyezés

2.9. táblázat. A megszakításprioritás-regiszter (IP) tartalma

Szimbólum	Bit	Jelentése
PX0	IP.0	a külső INT0 prioritása
PT0	IP.1	a T0 időzítő prioritása
PXl	IP.2	a külső INTI prioritása
PT1	IP.3	a T1 időzítő prioritása
PS	IP.4	a soros vonal prioritása
PT2	IP.5	a T2 időzítő prioritása (csak a 8052-nél)
-	IP.6	Fenntartva
-	IP.7	Fenntartva

A kevés prioritási szint megnehezíti a bonyolultabb feladatok megoldását. A mikrokontroller-család későbbi tagjainál a megszakításszintek számát növelték.

2.6.12 A T2 időzítő vezérlőregisztere (T2CON)

A T2 időzítő/számláló működését a T2CON regiszter tartalma vezérli. A 8051-es típusnál nincs T2-es időzítő/számláló, és így természetesen T2CON regiszter sincs. A T2 funkcióival a 2.4. szakaszban már foglalkoztunk. A 2.10. táblázatban foglaltuk össze az egyes üzemmódok beállítási lehetőségeit.

2.10. táblázat. .A Timer2-Control-Regiszter (T2CON) tartalma

Szimbólum	Bit	Jelentése
CP/RL2	T2CON.0	Capture/Reload-flag. 1 értéke választja ki a Capture üzemmódot. 0-nál az újratöltés (Reload) érvényes. A bit csak akkor hatásos. ha az EXEN2 értéke 1.
C/T2	T2C0N.1	Üzemmódválasztó bit. 0-nál késleltető (az f_{osc}/12 belső jelet számlálja), míg 1-nél eseményszámláló (a P 1.0 bemenetre érkező 1-0 éleket számlálja).
TR2	T2CON.2	1 indítja. míg a 0 leállítja a T2-t.
EXEN2	T2CON.3	A T2 külső engedélyezőflag, 1-gyel engedélyezi a T2 külső vezérlését. A T2EX bemenetre érkező 1-0 átmenet Capture, Reload funkciót vált ki.
TCLK	T2C0N.4	Adásórajel flag. Ha 1. akkor T2 túlcsordulása az adás ütemezőjele, ellenkező esetben csupán a T1 túlcsordulása. A soros vonal 1. és 3. üzemmódjában van szerepe.
RCLK	T2CON.5	Vételórajel flag. Az adáshoz hasonlóan a bit l értékénél a T2, míg 0-nál T1 túlcsordulása a soros vétel ütemező órajele.
EXF2	T2CON.6	T2 külső flag. Akkor íródik 1-be, amikor a T2EX bemenetre érkező negatív él Capture vagy Reload működést eredményez. Ekkor indulhat a megszakítást kiszolgáló rutin. A bitet szoftverben kell törölni.
TF2	T2CON.7	A T2 túlcsordulásbitje. A számláló túlcsordulása írja 1-be, és szoftverben kell törölni. Ezt a funkciót nem használják, amikor a T2 az RCLK, ill. a TCLK jelet szolgáltatja.

2.11. táblázat. A Program-Status-Word (PSW) tartalma

Szimbólum	Bit	Jelentése
P	PSW.0	Paritásflag; beíródik, ha az akkumulátorban az egyesek száma páros
-	PSW.1	Foglalt
OV	PSW.2	Túlcsordulás- (Overflow) flag
RS0	PSW.3	az RS0 és az RSl bitekkel választható az aktuális regiszter-
RS1	PSW.4	Bank
F0	PSW.5	a felhasználó által szabadon használható jelzőbit (flag)
AC	PSW.6	Közbenső átvitelflag (Auxiliary-Carry); a BCD műveletek használják
CY	PSW.7	Átvitel (Carry) flag

2.6.13 Program-Status-Word (PSW)

A PSW-ben különböző jelzőbitek vannak, amelyek többek között meghatározott programugrások feltételeiben szerepelhetnek (2.11. táblázat).
Az RSl és az RS0 bitek segítségével választható ki a 2.12. táblázatban adottak szerint az aktuális regiszterbank.

2.12. táblázat. A regiszterbankok kiválasztása

RS1	RS0	Regiszterbank
0	0	Bank 0 (0000H-0007H)
0	1	Bank 1 (0008H-000FH)
1	0	Bank 2 (0010H-0017H)
1	1	Bank 3 (0018H-001FH)

A CY, AC, OV és P flageket az aritmetikai utasítások (részletesen a 3. fejezetben) írják 1-be, vagy pedig törlik. Az egyes flagek funkciója a következő:

CY az összeadás átvitelekor, ill. a kivonás áthozatakor íródik 1-be;

AC közbenső átvitel;

OV aritmetikai túlcsordulást jelző bit, amely 1-be íródik, ha az összeadás, ill. a kivonás eredménye a -128 és a +127 tartományon kívül esik;

P a paritást jelző bit. 1-be íródik, ha az akkumulátorban páros számú 1-es van, és törlődik páratlan számú 1-es esetében; P értéke minden gépi ciklusban újraképződik.

2.6.14 Akkumulátor

A mikrokontroller és a mikroprocesszor akkumulátorainak funkciói azonosak. Külön magyarázatra ezért nincs szükség.

2.6.15 B regiszter

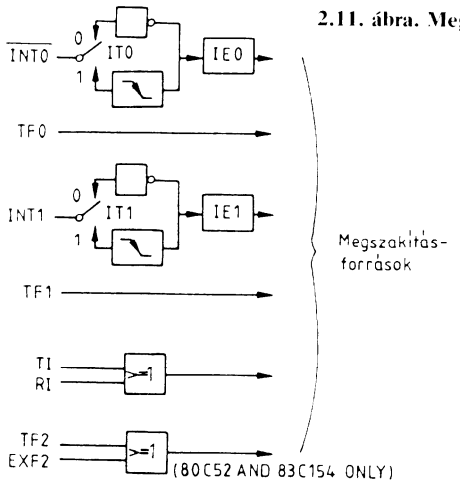
A B regiszternek csak a szorzáskor és az osztáskor van meghatározott szerepe. Egyébként adattárolóként használható.
A program során a B-be írt adatot csak a már említett szorzó- vagy az osztóutasítások írják felül.

2.7 A megszakítások és használatuk

A mikrokontroller megszakításaival oldható meg az események valós idejű kezelése.

A megszakítás (Interrupt) egy bekövetkező esemény kiszolgálására szolgál. A kiszolgálást kérő jel a hozzá tartozó megszakításkérő (Interrupt-Request) flaget 1-be állítja.

Megszakítást külső (pl. a megfelelő bemenetre érkező feszültségszint vagy szintváltás), ill. belső esemény (pl. az egyik számláló túlcsordulása) kérhet.



2.11. ábra. Megszakításforrások (F — élérzékelő)

Mindegyik megszakításhoz meghatározott prioritás rendelhető. Amikor egy megszakítást kérőjel érkezik, akkor az éppen futó program végrehajtása felfüggesztődik és a megszakítást kiszolgáló (Interrupt-Service) rutin fog futni. Ha a mikrokontroller éppen egy azonos vagy magasabb prioritású megszakítást szolgál ki, akkor csak a futó rutin befejezése után hívja az újabb megszakításrutint.

A 8051-es típusú mikrokontroller megszakításrutinjainak kezdőcíme - a legtöbb mikroprocesszorral ellentétben - állandó.

Erre a címre egy ugróutasítást kell beírni, amely a megszakításrutinra adja át a vezérlést. A rutinból a főprogramba való visszatérést a RETI utasítás (1. a 3. fejezetet) hajtatja végre.
A 8051-es mikrokontroller megszakításkezelése - a modern 16 bites mikroprocesszorokéhoz képest - meglehetősen korlátozott. Ez egyrészt a fix kezdőcímből, ill. abból adódik, hogy csak két prioritási szintet lehet beállítani. A 8051-es család tagjainál egyre több hardverfunkciót integráltak a chipbe, és így a megszakítás-vezérlő is belül van. A későbbi változatokban a megszakítás-kiszolgálást jelentősen továbbfejlesztették (részletesebben erről a 4. és az 5. fejezetben lesz szó).

2.7.1 Megszakításforrások

A 8051, ill. a 80C51-es típusoknál öt megszakításforrás és két prioritási szint van. A 8052-es típusú mikrokontrollerbe még egy további megszakításforrást (2.11. ábra) integráltak.
Megkülönböztetünk külső és belső megszakításforrást. A külső megszakításokat (INT0, INT1) a megfelelő bemenetre (12, ill. 13lábak) adott feszültségszint vagy negatív él indítja. Az egyes bemenetek szint- vagy élérzékenységét a TCON (Timer-Control) regiszter IT0, ill. IT1 bitjeivel lehet megválasztani (részletesen a 2.6.6. pontban tárgyaljuk). Az egyes megszakítások végrehajtását a TCON.1 (IE0), ill. a TCON.3 (IE1) bitek 1-be írásával engedélyezhetjük. A belső megszakításokat - az előzőekkel ellentétben - nem külső, hanem a tokon belüli esemény kezdeményezi. A felhasználó ezért csak közvetetten kezdeményezheti ezeket. A belső megszakításra példa az ES, amelyet a soros vonal indít az RI vagy a TI beírásával. További belső megszakításforrás a TF0 és a TFl. Ezeket a megfelelő időzítő/számláló (T0, ill. T1) túlcsordulása indítja.
A 2.13. táblázatban összefoglaltuk a megszakításforrásokat és kezdőcímeiket.

2.13. táblázat. Megszakításforrások és kezdőcímeik

Megszakításforrás	Kérőflag	Kezdőcím
0-s külső megszakítás	IE0	0003H
Timer 0 megszakítás	TF0	000BH
1-es külső megszakítás	IE1	0013H
Timer 1 megszakítás	TF1	001BH
Sorosvonali megszakítás	RI vagy TI	0023H

A 8052-es típusú mikrokontroller kibővített megszakítása:

Timer 2 megszakítás	TF2 vagy	002BH
Vagy külső töltés	EXF2	

2.7.2 A megszakítások műveletei

A 8051-es család minden tagjára igaz, hogy a megszakítást kérő jelzőbiteket minden gépi ciklusban csak egyszer, mégpedig az SSP2 ütemben vizsgálja (2.4. szakasz). A leírt működés miatt

a külső megszakításforrás megszakítást kérő jelének LOW vagy HIGH szintje minden esetben legalább egy teljes gépi ciklus időtartamáig kell, hogy fennálljon. Csak ezzel biztosítható a megszakításkérés biztonságos felismerése.

A 8051-es megszakítás-vezérlő logikája bármelyik megszakítási kérelem felismerése után egy szubrutinhívást generál. Az utasításról bővebben a 3. fejezetben lesz szó. A megszakítás lekezelése tehát a következő lépésekből áll:
- Annak ellenőrzése, hogy kiadható-e a szubrutinhívás.

Az ellenőrzés arra vonatkozik, hogy a kérés pillanatában azonos vagy magasabb prioritású megszakítás kiszolgálása folyik-e. Ha igen, akkor a kontroller nem végez szubrutinhívást. Az alacsonyabb szintű megszakítás-végrehajtást egy magasabb prioritású megszakítja.

Ha a megszakításkérés utasítás végrehajtása közben érkezik, akkor először az utasítás befejeződik, és csak utána megy végbe a megszakításrutin hívása. A megszakításkérés
- Elhelyezi a programszámláló pillanatnyi értékét a verembe.
- Feltölti a programszámlálót az aktuális megszakítás-kiszolgáló rutin kezdőcímével (2.13. táblázat).
- A megszakításrutint egy RETI utasításnak kell lezárnia. Ennek hatására folytatódik a megszakított program végrehajtása.

Az itt röviden leírtakat a 3. fejezetben kibővítjük. A RET utasítás ugyancsak biztosít visszaugrást egy szubrutinból, de megszakításrutinnál nem szabad használni!

A megszakítás-kiszolgáló rutinokat minden esetben a RETI (Return From Interrupt) utasítással kell befejezni. Ez jelzi a megszakítás-vezérlő rendszernek, hogy egy kiszolgálás befejeződött.

Ha RET utasítást használunk, akkor a megszakítási rendszer lefagyhat. A megszakításokat vezérlő egység nem kap jelzést a befejezésről, emiatt a továbbiakban az azonos és az alacsonyabb szintű kérések nem kezdeményezhetnek kiszolgálást.

2.7.3 A megszakítások válaszideje

A valós idejű események vagy kritikus időpillanatok lekezelésekor fontos ismernünk, hogy a megszakítás kezdeményezése után mikor következik be a kiszolgálás. Ezt az időtartamot nevezzük a megszakítás válaszidejének (Interrupt-Response-Time).
A következőkben vizsgáljuk meg, hogy az éppen aktív megszakítás-kiszolgáláson kívül még mi akadályozhatja a megszakítást kiszolgáló rutinra való azonnali ugrást.
Amikor egy esemény megszakítást kér, akkor a megfelelő flag az éppen futó gépi ciklus S5P2 ütemében íródik be. Ezt a bitet a mikrokontroller majd csak a következő ciklusban vizsgálja meg. A legjobb esetben tehát a kezdeményezés után 2 ciklusidő múlva következik az LCALL utasítás, vagyis a tényleges megszakításrutin végrehajtása leghamarabb három ciklus után indul. A leírtaknál még azt is feltételeztük, hogy a megszakításkérés időpontjában az aktuális utasítás-végrehajtás utolsó fázisában van. Az esetek többségében úgy kell vennünk, hogy a válaszidő az éppen futó utasítás végrehajtási ideje plusz három ciklusidő. A leghosszabb idő a MUL és a DIV utasítások végrehajtásához kell. Mindkettő négy gépi ciklus időtartamú.

Egy megszakításhoz tartozó rutin legalább három, de legfeljebb hét ciklusidő múlva indul.

Mind ahogyan azt már említettük, a leírtak csak akkor igazak amikor más megszakítás-kiszolgálás nem akadályozza a legújabb hívást. Különösen a valós idejű eseménykezelésnél kell figyelni arra, hogy csak akkor valósak az imént definiált válaszidők, ha az adott megszakítás a legmagasabb prioritású.

2.8 Sorosvonal-illesztő

A 8051-es család tagjainak egyetlen sorosvonal-illesztő egysége van. Ezen keresztül külső készülékkel vagy rendszeren belüli processzorral adatátvitel valósítható meg.

A sorosvonal-kezelő teljes duplex, vagyis egyidejűleg folyhat az adás és a vétel is.

Az egyidejűség annak ellenére lehetséges, hogy mind az olvasásnál, mind pedig az írásnál az SBUF speciális funkcióregisztert címezzük. Fizikailag az SBUF két különálló regiszter, és így elérésük egymástól független.

A soros vonal végfokozata kb. 50 cm hosszú összeköttetést tud meghajtani.

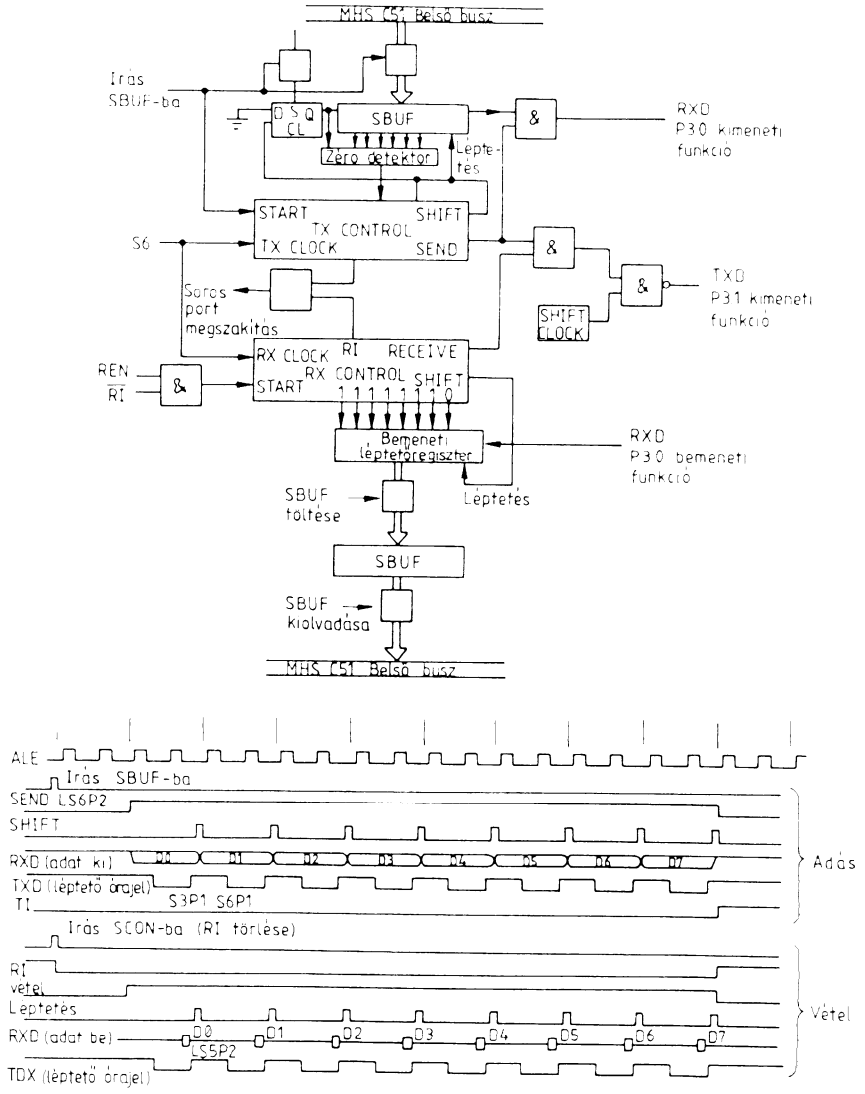
Egy rendszeren belül levő áramkörök közötti adatátvitelhez általában elégséges ez a távolság. Ha nagyobb távolságú adatátvitelt kívánunk kiépíteni, akkor illesztőáramköröket (TTL-kapukat, szintáttevőket, meghajtótranzisztorokat) kell beiktatni. A MAX232 vagy az ICL232-es típusú sorosvonal illesztők alkalmazásával építhető ki az RS232 (V.24., V.28) szabvány szerinti illesztés. E megoldás alkalmazásával nagyobb távolságra elhelyezett perifériákkal is létrehozható a soros vonali adatátvitel.
A 8051-es soros kommunikációja négy különböző üzemmód szerint állítható be. Az üzemmód kiválasztása az SM1 (SCON.6) és SM0 (SCON.7) bitekkel történik (2.6.8. pont).

Üzemmód	SM0	SM1	A bitek száma	Baudrate
0	0	0	8	$F_{osc}/12$
1	0	1	10	Változtatható
2	1	0	11	$F_{osc}/32$ vagy $f_{osc}/64$
3	1	1	11	Változtatható

Az adatátvitel - mindegyik üzemmódban - a legkisebb helyi értékű (LSB) bittel kezdődik.

2.8.1 Adás, vétel a 0-ás üzemmódban (2.12. ábra)

Ebben az üzemmódban az adatátvitel sebessége (baudrate) csak $f_{osc}/12$ lehet. Az adatok adása és vétele is az RXD (P3.0) lábon keresztül folyik. A szinkronozást a TXD (P3.1) lábon kiadott ütemezőjel végzi. Az adatok 8 bitesek.



2.12. ábra. A sorosvonal-illesztő a 0-ás módban

Adás

Az adás folyamata akkor kezdődik, amikor az SBUF adó regiszterébe írjuk az átviendő 8 bitet. A gépi ciklus S6P2 ütemében (l. később) a regiszter kilencedik bitje 1-be íródik, amely aktiválja a belső SEND jelet. A SEND jel hatására a SBUF adóregiszterének kimenete a P3.0 lábra csatlakozik, majd a P3.1 lábra kerül a SHIFT CLOCK jel. Ez a jel szinkronozza a P3.0 lábon megjelenő adatbiteket.

Minden egyes SHIFT CLOCK jel az adó-, ill. a regiszter tartalmát egy hellyel (bittel) lépteti.

A későbbiekben tárgyalt belső ütemezések szerint a SHIFT CLOCK jel egy gépi cikluson belül az S3, S4 és az S5 ütemben LOW, míg az S6, S1 és az S2 ütemek alatt HIGH szintű. Ebből adódik az $f_{osc}/12$ értékű adatátviteli sebesség (baudrate). Az egyes bitek az S6P2 időpillanatban jutnak a kimenetre. Az SBUF adóregisztere mindegyik gépi ciklusban egy bittel jobbra (a legkisebb helyi érték irányába) lép. Balról a regiszterbe mindig 0 lép be. Az adás folyamata a

beírástól számított tizedik gépi ciklus S1P1 ütemében fejeződik be, ekkor a SEND jel inaktív lesz. A művelet kilencedik ciklusában az SBUF 9. bitje 1, a többi pedig 0 értékű. Az átvitel befejezésének pillanatában a TI flag (a SEND jel visszaállításával egyidejűleg) 1 értékű lesz. Az üzemmódot léptetőregiszteres üzemmódnak is nevezik, mivel az SBUF léptetőregiszterként dolgozik.

Az adat kivitele a soros vonalra, az SBUF-ba irányuló írásutasítással kezdődik, és a tizedik gépi ciklus S1P1 ütemében fejeződik be. Ekkor lesz 1 a TI flag.

Vétel

A vételt a REN (SCON.4) bit 1 értéke és az RI (SCON.O) bit 0 értéke engedélyezi. Az engedélyezést követő gépi ciklus S6P2 ütemében a bemeneti léptetőregiszterbe 1111 1110B = FEH érték íródik. A vétel lefolyásának időzítése a következő:

Amikor a RECEIVE belső jel aktív, akkor - a SEND jel hatásához hasonlóan - a SHIFT CLOCK ütemezőjel a P3.1 lábra jut. A SHIFT CLOCK jel váltásai az adásnál már leírtak szerint mennek végbe. A negatív él az S3P 1, míg a pozitív él az S6P 1 ütemben lesz. Az S6P2 ütemben történik a vevőregiszter tartalmának balra léptetése egy helyi értékkel. Ugyanekkor a P3.0 lábon érvényes jel a regiszterbe kerül. A léptetési-beolvasási folyamat ismétlődik. A vétel akkor fejeződik be, amikor a kezdetkor beírt 0 a bemeneti regiszter "bal" végére kerül. Ez jelzi a vezérlőnek, hogy már csak egy léptetés és beolvasás szükséges. Befejezésül a bemeneti regiszter tartalma átíródik az SBUF regiszterbe. A vétel az SBUF címről való olvasást (ennek hatására törlődik az RI bit) követő tizedik gépi ciklus S1P1 ütemében fejeződik be. Ekkor kapcsolódik ki a belső RECEIVE jel és kerül 1-be az RI flag. A programozónak kell gondoskodnia a beolvasott adat továbbításáról. Ezt az RI által hívott megszakítási rutin végzi. A vétel folytathatóságához be kell állítani a REN bitet és törölni az RI-t.

A 0-ás üzemmód passzív partnerrel létrehozott adás, vétel lebonyolítására alkalmas.

A 0-ás üzemmódban mind az adás, mind a vétel ütemezőjelét a 8051-es szolgáltatja (P3.1 láb).

2.8.2 Adás, vétel az 1-es üzemmódban

Ebben az üzemmódban a TXD jelű láb az adóvonal, míg az RXD jelű a vevő. Az adatátvitel 10 bites (a startbit 0, 8 adatbit, stopbit 1). A stopbit az RB8 (SCON.2) bitbe is átíródik.

Az adatátvitel a Timer 1 segítségével, annak beállítási határán belül (2.5 szakasz) változtatható. Mind az adást, mind pedig a vételt a T1 túlcsordulási jel ütemezi. A 8032/52-es típusú mikrokontrollerekben a T2 is használható az adás vagy a vétel szinkronozásához. Azokban a változatokban, amelyekben T2 van, az adás és a vétel eltérő sebességű ütemezése megoldható, mégpedig úgy, hogy az egyik számláló az adást, míg a másik pedig a vételt szinkronozza.

Adás

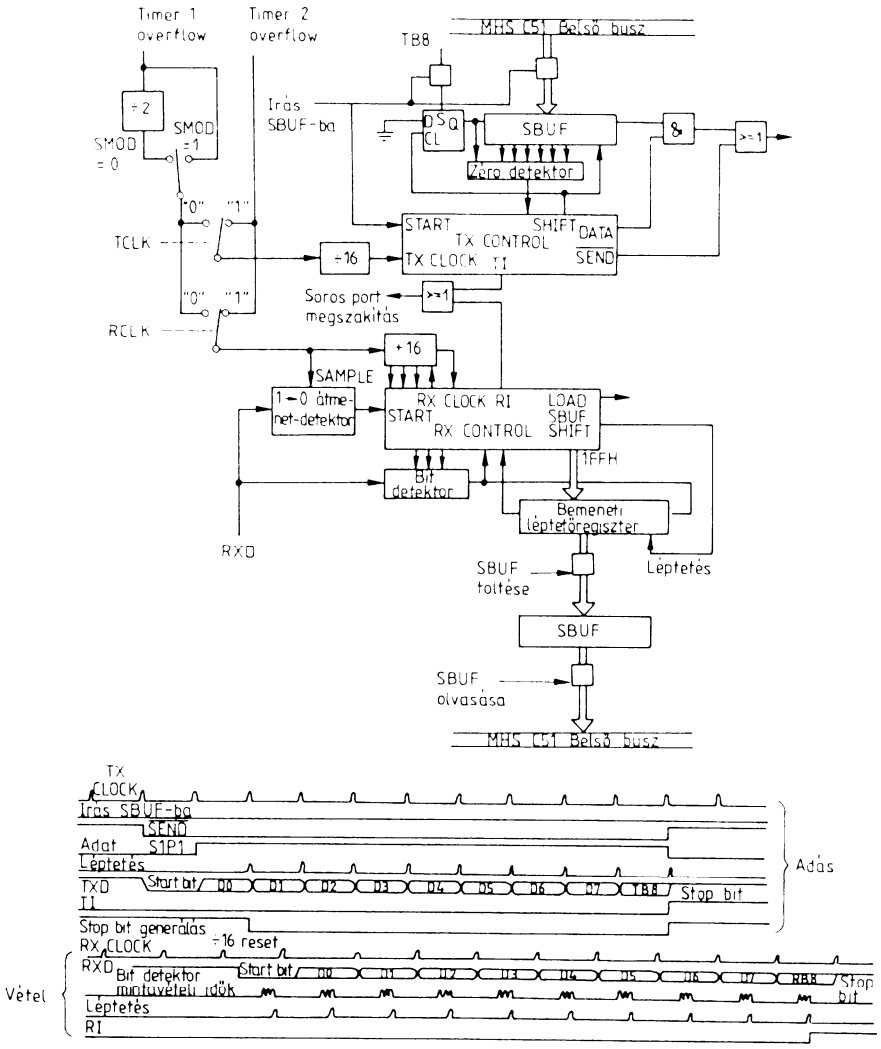
Az adási folyamat indítása azonos a 0-ás üzemmódnál megismerttel. Az adás tehát az SBUF címre való írással kezdődik. A 0-ás módban az adóregiszter kilencedik bitjének beírása állítja be a SEND jelet. Itt viszont az adatátvitel csak a számláló első túlcsordulását követő gépi ciklus S1P1 ütemében indul. Minden további bit kivitelét a soron következő számlálótúlcsordulás hajtja végre. A léptetés után a regiszterbe - itt is - 0 lép be.

Vétel

Az adat vételét az RXD (P3.0) lábra érkező startbit indítja.

A kontrollernek negatív (lefutó) élt kell érzékelnie. Ehhez a vezérlő a P3.0 bemenetet a baudrate frekvencia 16-ával mintavételezi. Amikor a negatív szintváltást érzékeli, akkor a 16-os mintavételezés újból indul, és a bemeneti regiszterbe 01 1111 1111 (lFFH) értéket ír. A bitidő (két egymást követő bit közötti idő) 16 részre oszlik. A 7., 8. és a 9. ütemben, egymástól függetlenül vizsgálja a szintet. Az eljárás előnye, hogy az esetleges elektromos zavart ki lehet szűrni azáltal, hogy

azt a szintet érzékeli valósnak. amelyik a három vizsgálat közül legalább kétszer érvényes.



2.13. ábra. A sorosvonal-illesztő az 1-es módban

A megoldás a startbit érzékelésének egyszerű módja. A vizsgálat mindaddig ismétlődik. amíg negatív élt nem érzékel. Megérkezésekor a startbit beíródik a bemeneti léptetőregiszter - legnagyobb helyi értéktől (MSB) számított - kilencedik bitjébe. Az ettől jobbra levő 1-ek továbblépnek balra. A 8051-es soros adatátvitele mindig az LSB bittel kezdődik, és sorba lépnek be a további helyi értékű bitek. Amikor a startbit a vevőregiszter 7. bitjébe (RB7) kerül, ezután még egy léptetés következik. Ekkor lép a 0 (startbit) az RB8 helyre, és az utolsó beolvasott bit a vevőregiszter RB0 helyére. Befejezésül a beolvasott tartalmat (RB7-RB0) egy belső (LOAD SBUF)jel átírja az SBUF-be. és egyúttal az RI flag is 1-re vált. Az átíráskor a legelőször beolvasott bit kerül az SBUF LSB helyére. és így tovább a belső adatbusz bitkiosztásának megfelelően. Az SBUF-bó1 - a megszakítást kiszolgáló rutinban kiadott - utasítással kell az adatot kiolvasni. A rutinban kell az RI flag törlését is végrehajtani.

A LOAD SBUF jel csak az RI = 0 utolsó ütemét jelzi. Kiegészítésül szükséges, hogy az SM2 = 1, vagy pedig a vétel stopbitje 1 legyen.

Az előzőekben leírtak nem biztosítanak védelmet az utolsó vett adat elvesztése ellen. Minden esetben - a vett adatbájt értékétől függetlenül - a vételi vezérlőlogika újból kezdi figyelni a következő negatív élt.

A 0-ás és az 1-es üzemmódokban nincs Handshake lehetőség és paritásbit-figyelés. A felhasználónak kell a szoftverben gondoskodnia arról, hogy a vett adatokat időben kiolvassa.

2.8.3 A 2-es és a 3-as üzemmódok. Többkontrolleres kommunikáció, adás és vétel

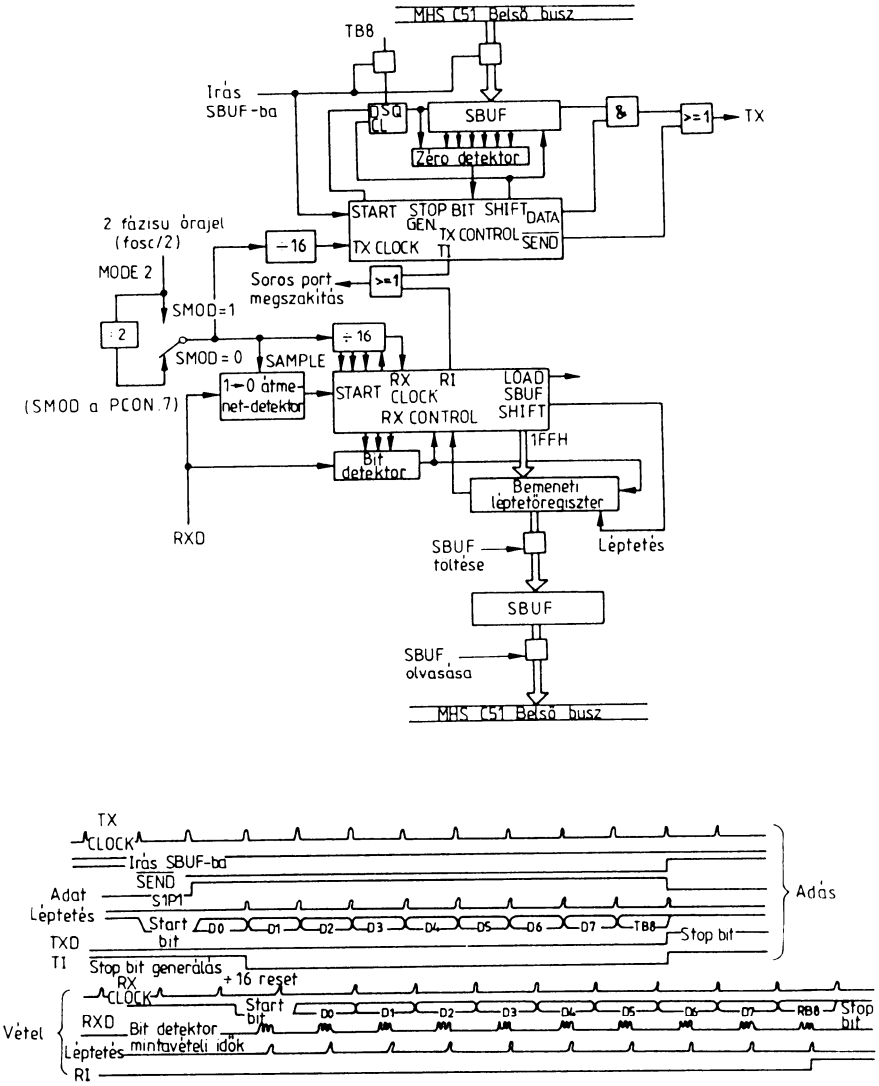
A 2-es és a 3-as üzemmódok (2.14. és 2.15. ábra) csak a baudrate választásában térnek el egymástól. A 3-as módban változtatható az ütemezés frekvenciája. Mindkét átviteli módozat 11 bites; 0 szintű startbit, 9 adatbit (az LSB-vel kezdve) és 1 szintű stopbit. A kilencedik adatbit paritásjelző bitnek is használható. Ilyenkor az adás előtt a programstátusz szó P bitjét (PSW.0) - amely a paritás értékét jelzi - át kell írni a TB8 bitbe. Az adás a TXD, ill. a vétel az RXD lábon keresztül történik.

A 2-es üzemmódban csak meghatározott baudrate - $f_{osc}/32$ vagy $f_{osc}/64$ - választható. A 3-as üzemmódban a T1 számláló szabadon programozható.

A 8052-es vagy a 80515-ös típusok esetében a T2 is felhasználható az ütemezőjel előállítására. Ezáltal az adás és a vétel sebessége különbözhet is. A 2.14. és a 2.15. ábrán mutatjuk be a soros vonali egység felépítését a jelzett módokban. A vétel mindkettőnél megegyezik az 1-es üzemmódnál ismertetett működéssel.

Többkontrolleres kommunikáció

A működéskiesés elleni védelmet alapvetően a több önálló egységből álló rendszer biztosíthatja. Az egyik egység leállásakor, annak funkcióit részben vagy egészében átveheti egy másik egység. Az ilyen felépítés esetében feltétlenül biztosítani kell az egységek mikrokontrollerjei közötti kommunikációt. A soros vonali egység 2-es és 3-as üzemmódjaiban ez megvalósítható. A többkontrolleres rendszerek általában ún. master-slave felépítésűek. Ebben egy master és több slave kontroller dolgozik együtt. Mindegyik slave-nek önálló címe van. A master és a slave-ek közötti adatátvitel két lépésben zajlik. Először a címet hordozó bájt kerül a soros vonalra. A csatlakozó kontrollerek közül csak a címzett fogja a következő bájtot is venni. A helyes működés előfeltétele, hogy a cím- és az adatbájtot egyértelműen meg lehessen különböztetni. A 8051-es rendszerekben erre használható a kilencedik adatbit. Amikor ez a bit 1. akkor cím, és amikor 0,
akkor adat a további 8 bit. Az SM2 bit 1-be írásával a slave kontrollerek címként kezelik a vett adatot. Mindegyik kontroller egyidejűleg vizsgálja meg a vonalra küldött címbájtot. Az a slave kontroller törli az SM2 bitjét, amelyik a saját címét azonosította. Ezután csak ez veszi az adatbájtot.



2.14. ábra. A sorosvonal-illesztő a 2-es módban

Adás

Az adási folyamatot ezekben az üzemmódokban is az SBL; F regiszterbe író átviteli utasítás indítja. A „WRITE TO SBUF” jel egyrészt az SBUF-be írt 8 bites adatot, másrészt a TB8 bitet is az adóregiszterbe írja. Ha paritásjelzőnek

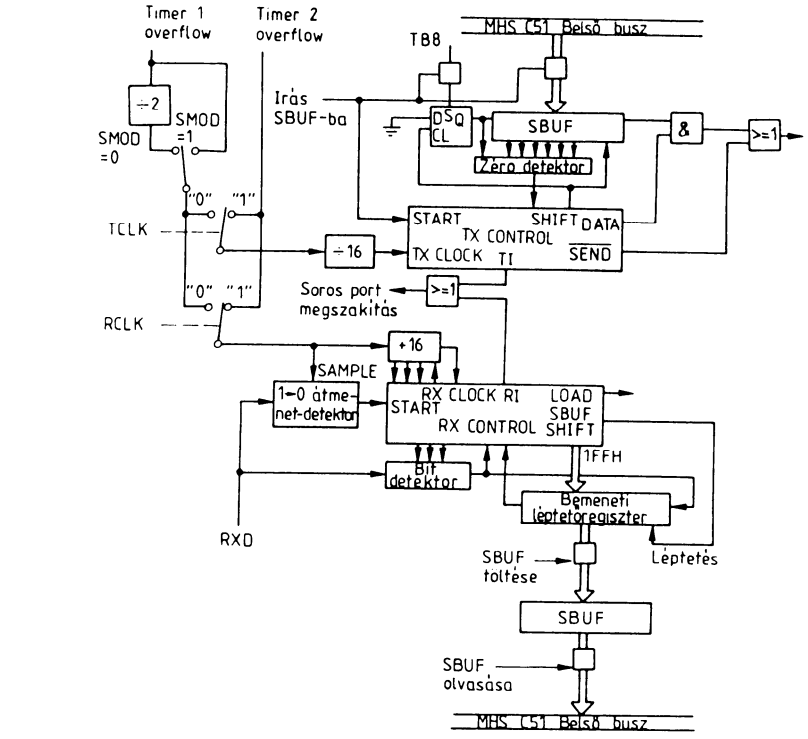
használjuk a TB8 bitet, akkor a felhasználói programban kell gondoskodni a helyes érték beállításáról. Az adatátvitel az átírást követő első TX CLOCK jelhez tartozó gépi ciklus S1P1 ütemében kezdődik. A 2-es módban a processzor órajele, míg a 3-as módban a kiválasztott számláló (T1 vagy T2) túlcsordulása határozza meg a TX CLOCK jelet. A tényleges ütemezőjel egy számlálón (16-os osztó) keresztül jut a vezérlőegységbe. A TX CLOCK bekapcsolja a SEND jelet. E jel hatására a startbit a TXD kimenetre kerül. A következő TX CLOCK az adatkimenetet kapcsolja egy ÉS kapun keresztül a TXD kimenetre. A belső léptetés a TX CLOCK jellel szinkronban zajlik. A léptetőjelek sorrendben 8 adatbitet juttatnak a kimenetre, majd kilencedikként a TB8 tartalmát. Ezt követően egy 1-et, a stopbitet adja ki. A regiszterbe 0-k lépnek be. Amikor az összes bit 0, akkor az adást vezérlő egység
- még egy SHIFT CLOCK jelet ad;
- azután kikapcsolja a SEND jelet és egyidejűleg
- a TI-flaget beírja a programmegszakításhoz.

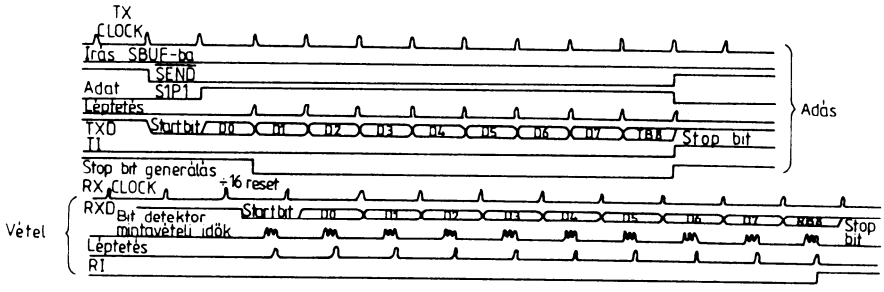
Vétel

A vétel a 2-es és 3-as üzemmódokban közel azonos az 1-es üzemmódéval, ezért csak az eltéréseket írjuk le.

Az alapvető eltérés az 1-es üzemmódhoz képest az, hogy az átvitel 11 bites (startbit 0, 8 adatbit, egy kiegészítő bit, stopbit 1)

A 8 bites adatot az SBUF regiszterbe kell beírni. A kiegészítő kilencedik bitnek (amely rendszerint a paritást jelző bit) az RB8-ba kell kerülnie. A véget az RI flag jelzi. Az egyetlen stopbitről nem kell külön gondoskodni. A 2-es üzemmód - az 1-es és 3-as módokkal ellentétben - sebességét a belső órajel határozza meg, nem programozható szabadon. Ekkor van egy szabad időzítő/számláló az egyéb feladatok számára.
A leírtakból láthattuk, hogy ugyanaz a hardverkonfiguráció különböző funkciókra programozható. Ily módon valósítható meg az éppen szükséges működésmód, és ugyanakkor nem kell kiegészítő áramköröket felhasználni.





2.15. ábra. A sorosvonal-illesztő a 3-as módban

2.8.4 A baudrate előállítása a Tl és a T2 számlálókkal

A T1 használata baudrate generátorként
A soros átviteli csatorna adás-vételi sebessége (baudrate) az alkalmazott oszcillátorfrekvenciától és a programozási módtól függ.

A 0-ás és a 2-es üzemmódokban a baudrate állandó érték. Az 1-es és a 3-as üzemmódokban a T1 (ahol van, a T2) használható az ütemezőjel előállítására. Ennek értékét programozni lehet.

Ha egy számlálót használunk baudrate generátorként, akkor azt rendszerint a 2-es üzemmódban érdemes működtetni. Ez természetesen nem kötelező előírás. Más üzemmód is nyújthat előnyös ütemezőjelet. Az átvitel szinkronozásához a bemenőjel a T1 számláló túlcsordulása (Timer-Overflow TF1) lesz. Az átvitel baudrate értékét a következő összefüggés segítségével számíthatjuk ki:
Baudrate = 2^{SMOD}/ 32 * (T1 túlcsordulása)
Amikor a T1-et az Auto-Reload módban időzítőként alkalmazzuk, akkor az összefüggés a következő lesz:
Baudrate = $f_{osc}/12 * 2^{SMOD}/[32 * [256-(TH1)]]$
Az egyes értékek kiszámolása után megállapíthatjuk, hogy az 1,2 kBaud és 19,2 kBaud közé eső szabványos értékek nem állíthatók be a 12 MHz frekvenciájú oszcillátorral. Ilyen esetekben 11,059 MHz-es kvarcot célszerű alkalmazni. A 2.14. táblázatban foglaltuk össze az összetartozó adatokat, amelyek az 1-es, ill. a 3-as módokban használhatók. A 2.14. táblázatban feltételeztük, hogy a T1 számláló üzemmódban (C/T = 0) működik. A 0-ás és a 2-es üzemmódokban a 12 MHz-es órajel esetében nagyobb átviteli sebességet lehet elérni. A 0-ás módban ez közel 1. MBaud (f_{osc}/ 12), és a 2-es módban 375 kBaud vagy 187,5 kBaud.

2.14. táblázat. Baudrate az 1-es és 3-as módokban

Baudrate. kHz	f_{osc} MHz	SMOD	T1 mód	Beírt érték hexadecimálisan
62.5	12,0	1	2	00FF
19,2	11,059	1	2	00FD
9.6	11,059	0	2	00FD
4,8	11,059	0	2	00FA
2,4	11,059	0	2	00F4
1,2	11,059	0	2	00F8
0,1375	11,986	0	2	001D
0,110	6,0	0	2	0072
0,110	12,0	0	1	FEFB

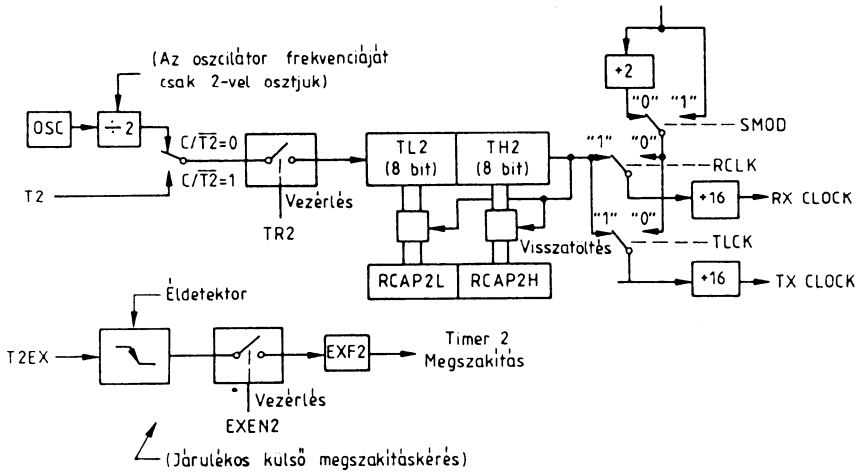
A T2 használata baudrate generátorként
A T2 időzítő ugyanúgy használható ütemezőjel-adónak, mint a T1. Ekkor az adáshoz a TCLK, míg a vételhez az RCLK-bitet kell a T2CON regiszterben 1-be írni (2.6.12. pont). T2 használatakor a következő választás lehetséges:

A Tl vagy a T2 túlcsordulását választva a soros átvitel vezérlésére, megoldható az adás és a vétel független ütemezése.

A 2.16. ábrán látható az elvi felépítés.
Amikor a T2 baudrate generátor, akkor Auto-Reload üzemmódba kell beállítani. A számláló túlcsordulása vezérli az RCAP2H és RCAP2L regiszterek tartalmának ismételt betöltését. Mivel a T2-nél 16 bites a beállítható modulus, ezért sokkal szélesebb tartományban változtatható a baudrate, mint a T1-gyel.
A T2 mind késleltetőként, mind pedig számlálóként működtethető. A megfelelő működés kiválasztása a C/ T2 flaggel történik. Az esetek többségében időzítőként használják a T2-t (C/T2=0). Lényegében a baudrate generátor üzemmód is késleltetés, azzal a különbséggel, hogy

a T2 normál késleltetőkénti alkalmazásakor csak minden gépi ciklusban (f_{osc}/12), míg baudrate generátorként minden státusváltásnál (f_{osc}/2) megy végbe az inkrementálás.

Az ütemezés frekvenciáját a következő egyenlőség alapján számíthatjuk:
Baudrate = az időzítő túlcsordulása /16
A T2-re alkalmazva:
Baudrate = $(f_{osc}/32) * 1/[65536-(RCAP2H, RCAP2L)]$
összefüggést kapjuk. Az RCAP2H és RCAP2L 8 bites regiszterek tartalma a programozható 16 bites szó. Ebben az üzemmódban a számláló túlcsordulása nem írja be a TF2 bitet, ezért nem indít megszakítást sem. Ezért a T2 megszakítást tiltani sem szükséges. Ha az EXEN2 bitet 1-be írjuk, akkor a T2EX bemenetre érkező negatív él megszakítást fog kezdeményezni, de nem tölti újból a T2 regisztereit.



2.16. ábra. A T2-es számláló a baudrate üzemmódban

A baudrate generátoros üzemmódban a T2EX bemenet különálló megszakításként alkalmazható.

A leírt alkalmazási módban tehát a TH2 és TL2 regiszterek tartalma minden gépi ciklusban hatszor inkrementálódik. A regisztertartalmakat időnként célszerű ellenőrizni.

A soros vonali kommunikáció ütemezését a regiszterekbe írt adat határozza meg.

Az RCAP2H és RCAP2L regiszterek elvileg írhatók és olvashatók. A felhasználónak viszont tisztában kell lennie azzal, hogy egy felülírás megváltoztatja a soros vonal működési sebességét. Ez a tulajdonság viszont felhasználható változtatható frekvenciájú négyszögjel előállítására is.

2.9 A 0-3 portok

A 8051-es mikrokontroller-család egyes tagjai esetében különböző a 8 bites portok száma. A 80517-es típusnak a 8 bites kialakítás mellett még egy 4 bites portja is van. Az egyes típusok portszámait foglalja össze a 2.15. táblázat.

2.15. táblázat. A portok száma

Változat	A 8 bites portok száma
8051/8031	4
8052/8032	4
80215/80315	4
80513/80533	4
80515/80535	6
80517/80537	9 (+ egy 4 bites)

A portokon keresztül megy végbe a mikrokontroller és a külvilág közötti adatcsere. A 8051-es család tagjainál egyre több perifériafunkciót integráltak egy tokba. A 8051-es portjai eltérnek a hagyományos számítástechnikai I/O áramköröktő1. A mikrokontroller esetében egy 8 bites port több funkciót is elláthat. Így pl. buszmeghajtást vagy soros adatátviteli interfész feladatát. Ebbó1 adódóan a portok áramköri felépítése összetettebb, mint a klasszikus illesztőké. A továbbiakban a 8051-es négy portjának felépítésével és használatával foglalkozunk. A 80515-ös változatét a 4. fejezetben, míg a 80517-ét az 5. fejezetben ismertetjük részletesen.

A négy port tulajdonképpen 32 önálló, egybites port, amelyek bitenként címezhetők is. Ez azt jelenti, hogy bármelyik portbit egyedileg írható, olvasható. A portok mindegyik bitjéhez egy-egy tároló (Latch) tartozik, amely a megfelelő SFR (speciális funkcióregiszter) bitje. Ezek mindegyikéhez kimeneti meghajtó- és bemeneti illesztőáramkör kapcsolódik. A legtöbb porthoz - az alapfunkció mellett - egy-egy alternatív funkció is tartozik. Így pl. a P0-ás és a P2-es portokon keresztül érhetők el a külső tárolók, ezért a P0 multiplexelten üzemel. E porton keresztül megy végbe az adatírás, ill. - olvasás, a hagyományos adatbuszfeladat (2.2., ill. a 2.9. szakasz). A P3 port különböző be-, ill. kimenőjeleket illeszt (2.16. táblázat).

A 8051 típus P1-es portja csak I/O funkciót lát el. A 8052-es típusnál a P1.0, ill. a P1.1 jelű lábak a T2 időzítő/számlálóhoz tartozó jelek (a T2 számlálóbemenet. ill. a T2 capture/reload vezérlőbemenet) csatlakoznak.

2.16. táblázat. A P3 port egyes lábainak funkciója

Láb	Szimbólum	Funkcióleírás
P3.0	RXD	a soros vonal bemenete
P3.1	TXD	a soros vonal kimenete
P3.2	INT0	0-ás külső megszakítás
P3.3	INT1	1-es külső megszakítás
P3.4	T0	0-ás időzítő/számláló bemenete
P3.5	T1	1-es időzítő/számláló bemenete
P3.6	WR	Írójel
P3.7	RD	Olvasójel

Ha az egyes portokat alternatív funkciójukban kívánjuk használni, akkor a következőkre kell figyelni:

A portoknak van belső felhúzó (Pull-up) ellenállásuk. Az alternatív funkció csak akkor működik, ha a megfelelő porttárolóban HIGH szint van.

Ha a port 1-be írása elmarad, akkor a megfelelő lábat egy belső tranzisztor 0 szintre húzza le. Ilyen esetben egy külső U_{cc}(HIGH) szintű jel tönkreteheti az áramkört.
Az egyes portokhoz kapcsolt alternatív funkciók miatt ezek nem használhatók a klasszikus értelemben vett I/O portként. A többfunkciós kialakítás, alapvető oka, hogy a mikrokontroller lábainak száma korlátozott (a 8051/8052-esnél tokformától függően 40 vagy 44, a 80515-ösnél 68).
Az egyes portok bitjeinek áramköri felépítését szemléltetik a 2.17.-2.20. ábrák. A különböző felépítésű portok között az az azonosság, hogy mindegyik bithez egy-egy tároló tartozik. Ezek D flip-flopok, amelyeknek mindegyike valamelyik SFR egyik bitje. A belső buszról az adatot a WRITE LATCH (a D flip-flop clockja) jel írja be a flip-flopba. A tárolóba írt tartalom - a P1 portot kivéve - a belső vezérlőjeltől függően jelenik meg a hozzá tartozó lábon, vagyis csak az alapfunkcióban. Az alternatív funkció esetében a mikrokontroller a másik alternatív SFR tartalmát kapcsolja a lábakra.
Fordított irányban, vagyis amikor egy jelet olvasunk be, két lehetőség van.

Vagy a kiválasztott lábon lévő jelet (READ PIN), vagy a tároló (READ LATCH) jelét olvassuk be.

Egyes utasítások a tároló olvasására, mások pedig a lábon levő jel beolvasására alkalmasak. (Részletesen a 2.9.3. pontban lesz szó erről.)
Ügyelni kell továbbá a következőkre is:

A tárolók csak a kimenőjeleket tárolják. A bemenőjeleket a beolvasás után külön tárolni kell, egyébként elvesznek.

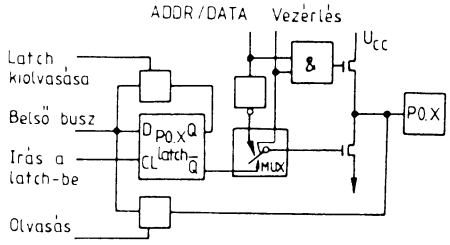
Felhasználáskor a programban kell gondoskodni a jelek változásának figyeléséről. Erre az ismételt lekérdezés (polling) megfelelő. Ha ez nem alkalmazható, akkor külső hardverrel kell megszakítást kérni a figyelt szintváltásnál.
A portokon keresztül nem csak külső tároló jelét lehet beolvasni. Természetesen más perifériaillesztő áramkör (pl. kiegészítő megszakítás-vezérlő) is csatlakoztatható a mikrokontrollerhez.

A buszillesztőként használható portok a 8085-ös típusú mikroprocesszor buszkezelésével kompatibilisek.

A 8085-ös mikroprocesszorhoz kifejlesztett interfészek közvetlenül csatlakoztathatók a 8051-es család kontrollerjeihez.

A 0-ás (P0) port

A P0 alternatív funkciója a külső tárolók és a kontroller közötti adatforgalomhoz kapcsolódik (2.17. ábra). Ez a port adja egyrészt a címbusz alsó 8 bitjét, másrészt pedig a klasszikus értelemben vett adatbuszt. A belső CONTROL jel kapcsolja a portfunkciót (CONTROL=0), ill. a külső memóriakezelői funkciót. Mielőtt a porton keresztül a külső memóriából olvasás megy végbe, a tárolóba 1 íródik. Ezzel a port eredeti tartalma elvész.

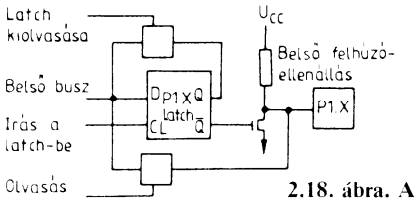


2.17. ábra. A P0 port egy lábának belső kapcsolása

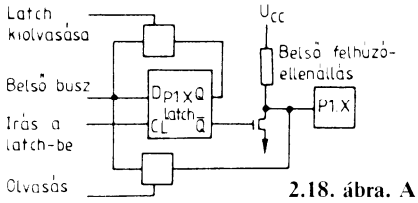
A P0 port az egyetlen, amelynek nincs belső felhúzó ellenállása, ill. a feladatot egy felhúzótranzisztor látja el. Ez a tranzisztor csak akkor vezet, ha a CONTROL jel és az adott címbit is HIGH szintű, vagyis amikor az alternatív funkcióban 1 értékű címbitet ad ki. Egyéb felhasználáskor nyitott kollektoros a kimenet. Ebből következik, hogy a P0 kimeneti portként való használata esetén külső felhúzó ellenállásra van szükség. A tároló 1-be írásakor csak így kapunk meghatározott feszültségszintet a kimeneten. Ha nincs külső ellenállás, és a lábat bemenetként használjuk (a tároló tartalma 1), akkor a bemenőjelet nem terheli belső ellenállás, és ez téves beolvasáshoz vezethet.

A P0 az egyetlen valódi kétirányú (bidirectional) port. Amikor a kimenet feszültségszintjét rögzítjük, akkor csak "kvázi" kétirányú.

Az 1-es (P1) port
A 8051-es típusjelű mikrokontroller egyetlen portja, amelyikhez nincs alternatív funkció is rendelve. Felépítése is ezért egyszerű. Egy D tárolóból, két ÉS kapuból, valamint egy végtranzisztorból és felhúzó ellenállásból áll. Az ÉS kapuk a lábról (READ PIN), ill. a tárolóból (READ LATCH) való olvasást vezérlő belső jeleket kapuzzák (2.18. ábra).



2.18. ábra. A P1 port egy lábának belső kapcsolása



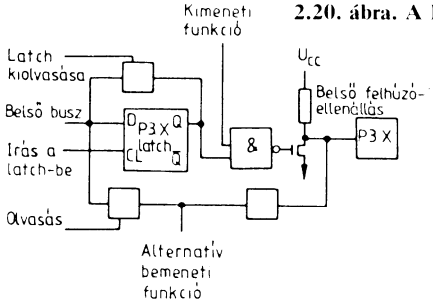
2.18. ábra. A P1 port egy lábának belső kapcsolása

A 2-es (P2) port
A P2 port alternatív funkciója - a P0-hoz hasonlóan - a külső memória és a mikrokontroller közötti adatíráshoz, - olvasáshoz kapcsolódik.
Ilyenkor ez a port adja a címbuszra a cím felső 8 bitjét. Áramköri kialakítása egyszerűbb, mint a P0 felépítése. A címszolgáltatási, ill. a hagyományos portfunkciót - a P0-hoz hasonlóan - a belső CONTROL jel határozza meg. Ha nem használjuk ezt a portot bemenetként, akkor nem is kell a tárolóba 1-et írni (2.19. ábra).

A port tárolóiba írt információ a címzési funkció végrehajtása alatt változatlan. Ezért a belső CONTROL jel inaktiválódásakor ismét a lábakra kerül az eredetileg beírt információ.

A 3-as (P3) port
A P3 port alternatív funkciója bitenként más és más. Egyesek be-, mások pedig kimenőjeleket szolgáltatnak (2.20. ábra). A P0 és a P2-es portokkal ellentétben, itt bitenként külön használhatók az alternatív vagy az alap portfunkciók. Egy lábról való olvasáskor nincs szétválasztva áramkörileg az alternatív és a portfunkció. Ha egy bemenőjelet (pl. az INT0-t) kívánunk használni, akkor azt a programban inicializálni kell, vagyis a lábhoz tartozó D tárolóba 1-et kell írni. A tároló kimenete és az alternatív jel között ÉS kapcsolat van. Ezért alapfunkció - normál port - használatakor előbb az alternatív jelet kell 1-be írni. Az alternatív funkció érvényesülése viszont csak a tároló 1-be írásával biztosítható.

2.20. ábra. A P3 port egy lábának belső kapcsolása



Az alternatív funkció csak akkor kapcsolódik az adott lábhoz, ha előbb a hozzá tartozó tárolóba 1-et írtunk.

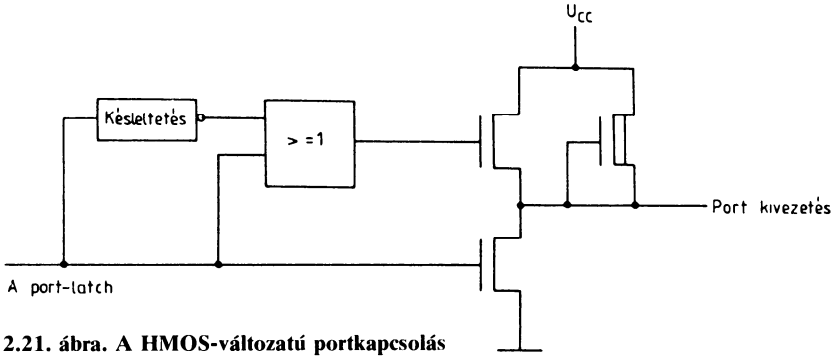
2.9.1 Egy portra való írás időzítése, jelmeredeksége

Időzítés
Mindenekelőtt rögzíteni kell, hogy egy port lábára nem írhatunk közvetlenül, hanem csak a hozzá tartozó tárolóba. Nagyon sok felhasználás esetén egy kimenőjel beírása vagy törlése a vezérlési feladat által megszabott pontos időzítést igényel. A belső időzítések ismerete segít a feladat megoldásában.

A belső buszról az adat az íróutasítás utolsó gépi ciklusának S6P5 ütemébe íródik be a port regiszterébe. Különösen fontos ez a többciklusú utasítások esetén.

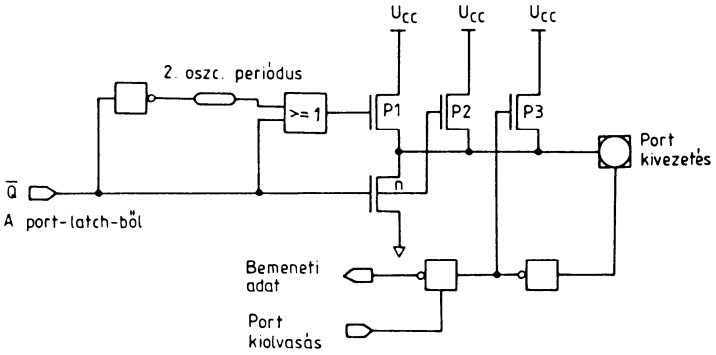
A tároló tartalmát a kontroller minden fázis P1 ütemében olvassa ki. A leírtak alapján egyértelmű, hogy a kürt információ mindig az utasítást követő gépi ciklus SlP1 ütemében jelenik meg a megfelelő kimeneti lábon.

Jelmeredekség
A Pl, P2 és a P3 portok "kvázi" kétirányúak. Minden kimenetet egy felhúzó ellenállás emel magas szintre. Egy 0-1 szintváltást ez az ellenállás "húz" fel. Ténylegesen ezek nem ohmos ellenállások, hanem olyan FET tranzisztorok, amelyeknek a gate-je és a source-a össze van kötve (a HMOS-változatokban). Ezek szélesebb területen használhatók, mint az ohmos ellenállások. Használatuknak általában hátránya, hogy az áramot erősen korlátozzák. I_{max} 0,25 mA. Gyors jelváltáskor ez az áram tölti fel a kimenethez kapcsolódó kapacitást. Az átkapcsolás sebessége ezzel lelassul. A váltás idejét egy párhuzamosan kapcsolt második - vezérelt - tranzisztorral lehet megnövelni úgy, hogy ez a tranzisztor rövid ideig 30 mA-t ad a kimenetre. A 0-1 szintváltáskor a belső vezérlés az SlP1 és SlP2 ütemek alatt nyitja a kiegészítő tranzisztort (2.21. ábra).



2.21. ábra. A HMOS-változatú portkapcsolás

A CMOS-változatokban a kimenet egy kicsit eltér a leírttól. Ezekben három párhuzamosan kapcsolt pFET végzi a szintfelhúzást (2.22. ábra). A T3 jelű pFET a tényleges felhúzóellenállás. A T2 pFET - az NMOS-változathoz hasonlóan - csak a 0-1 szintváltásnál, az SlP1 és SlP2 ütemek alatt vezet, és ezzel megnő az átkapcsolás meredeksége. A T3 tranzisztort (csak a CMOS-változatban) nem vezérli közvetlenül belső jel, hanem a kimenetről egy inverteren keresztül kap vezérlést. Akkor kapcsol be, ha a kimenet már 1-be íródott. Tulajdonképpen egy flipflop, amely tárolja a kiírt 1-et. Mivel a T3 viszonylag nagy belső ellenállású, ezért egy negatív tűimpulzus már visszabillenti. Ezt akadályozza meg a T2 tranzisztor. A T2 belső ellenállása tízszerese a T3-énak, és egy belső jel akkor kapcsolja be, amikor a T2 kikapcsolna.



2.22. ábra. A CMOS-változatú portkapcsolás

2.9.2 A portok terhelhetősége (fan-out)

Mivel az egyes portok áramköri kialakítása különböző, ezért az általuk vezérelhető TTL bemenetek száma is eltér.

A P0 maximálisan 8 LS-TTL bemenetet tud meghajtani. A P1-P3 portok átlagosan négy egységgel terhelhetők.

Ügyelni kell arra, hogy a P0 port kimenetéhez - kivéve, ha csak külső tárolók eléréséhez használjuk - felhúzó ellenállást kell csatlakoztatni.

2.9.3 READ-MODIFY-WRITE utasítások

Már a 2.9. szakaszban szóltunk arról, hogy a 8051-es családnál mód van arra, hogy csak egy lábon lévő jelet (READ PIN), vagy csak a hozzá tartozó tároló tartalmát olvassuk be (READ LATCH). Az olvasási módok között szoftverben választhatunk. Azokat az utasításokat, amelyek a tárolók tartalmát olvassák ki, READ-MODIFY-WRITE (olvas – változtat - ír) utasításoknak nevezik. Ezekről a 3. fejezetben bővebben lesz szó.

A READ-MODIFY-WRITE típusú utasítások a címzett tároló tartalmát kiolvassák, és az adott értékre változtatva írják vissza azt.

A kimenetekhez kapcsolódó áramkör indokolja a láb közvetlen, ill. a tároló kimenetének külön-külön olvashatóságát. Ha pl. egy porthoz emitterkövető kapcsolódik, akkor a láb közvetlen olvasásakor hibás értéket kapunk. Az 1 szintű

meghajtás a tranzisztort telítésbe viszi, és az emitter-bázis feszültség 0,7 V lesz. A port kivezetésének közvetlen olvasásakor ez a feszültség 0-nak számít.

2.10 Külső tárolók alkalmazása

Az alkalmazások széles körében nem elég a belső tárolók memóriakapacitása. Ilyenkor kell a 8051-et külső adat- és/vagy programtárolóval kiegészíteni. Használatuknál feltétlenül figyelni kell arra, hogy

a 8051 nem használ várakozó állapotot (Wait-State), ezért csak megfelelő sebességű külső tárolók alkalmazhatók.

A kontroller megkülönbözteti a külső adat- és programmemóriát. A kommunikáció is eltérő formájú. Más vezérlőjeleket használ a program-, ill. az adatmemória eléréséhez. Egy adatmemória elérésekor meg kell különböztetni, hogy a kontroller adatot olvas vagy ír a memóriába. Az írást a WR (WRITE), ill. az olvasást az RD (READ) jel vezérli. A külső programmemóriából csak olvasni kell, ezt a PSEN jel hajtja végre.

2.11 8 vagy 16 bites memóriacímzés

A külső adat-, ill. programmemória megkülönböztetése mellett eltérő a 8, ill. a 16 bites címzéses adatátvitel is.

A külső programmemóriát csak 16 bites címzéssel lehet elérni. A külső adatmemóriánál mind a 8, mind pedig a 16 bites címzés használható.

A felhasználás során az utasításokkal különböztethetjük meg a kétféle címzési módot.

Az adatmemória és a mikrokontroller közötti adatátvitelnél a DPTR (Data-Pointer) segítségével lehet 16 bites, míg az R0, ill. R1 révén a 8 bites címzést megvalósítani.

A 8 bites címzés esetén a címet kizárólag a P0 port adja. A P2 portot ilyenkor nem használja címzéshez. A port tárolója változatlanul a kimeneti lábakhoz kapcsolódik. A 16 bites cím használata eltér az előzőektől. A cím magasabb bájtját (A8-A15) minden esetben a P2 adja ki. A cím egy teljes gépi ciklus idejéig változatlan. A 2.19. ábrán látható, hogy egy belső kapcsoló csatlakoztatja a kimenetre a címet vagy a port tárolóját. A tároló tartalmát nem kell megváltoztatni.

A címzési művelet befejezése után ismét megjelenik a lábakon az eredeti szint (a hozzá tartozó tároló tartalma).

A P0 port - a címzési módtól függetlenül - mindig időmultiplex módon adja ki a cím alsó bájtját, ill. betölti az adatbusz szerepét.

Ha a P0 csak külső memóriát vezérel, akkor nem kell külső felhúzó ellenállást alkalmazni.

Amíg a P2 tartalma a címzés befejezte után változatlan, addig a P0 tárolójába - a külső memória elérése után - 0FFH íródik. Emiatt az eredetileg tárolt tartalom elvész.

2.11.1 Az ALE jel

Mint ahogyan azt már említettük, a 8051-es külső memóriát is kezelhet. A lábak számának korlátozása miatt az adatbusz kezelése időmultiplex üzemű. Ilyen buszkezelési megoldással más Intel processzoroknál is találkozhatunk. A megoldás a gyártó specialitása. A 8051-es típus fejlesztése során is ezt alkalmazták. A 16 bites címbusz és a 8 bites adatbusz 24 lábat igényelne. Az adat és a cím alsó bájtjának időmultiplex módon való kezelése viszont 8 csatlakozás használatával is megoldható. A működés során először a cím alsó bájtja kerül ki, amelyet egy külső tárolóba kell beírni. A tárolás vezérlésére használható az ALE jel (Address Latch Enable). Az ALE jel aktív szintje 1.
A 8051-es a cím alsó bájtját a 32-39-es lábakon, míg a felső bájtot a 21-28-as lábakon adja ki. A cím az ALE jellel írható a külső tárolóba. A kontroller a cím alsó bájtját az ALE jel negatív éléig tartja. Ezután a 32-39-es lábakon megy végbe az adatátvitel. Léteznek olyan speciális áramkörök (pl. 80C54), amelyek a multiplexelt buszt közvetlenül használják. Ezekben a címtároló az áramkörbe van integrálva, ezért kiegészítő tárolóra nincs szükség.
Az ALE jel egy gépi ciklusban kétszer jelenik meg. A jel az utasítás lehívásakor is megjelenik. A külső adat olvasásakor viszont elmarad az ALE.

Olyan rendszerben, amelyben nincs külső memória, az ALE órajelként is használható. Frekvenciája: $f_{osc}/6$.

2.11.2 Az egy-, két- és hárombájtos utasítások végrehajtása

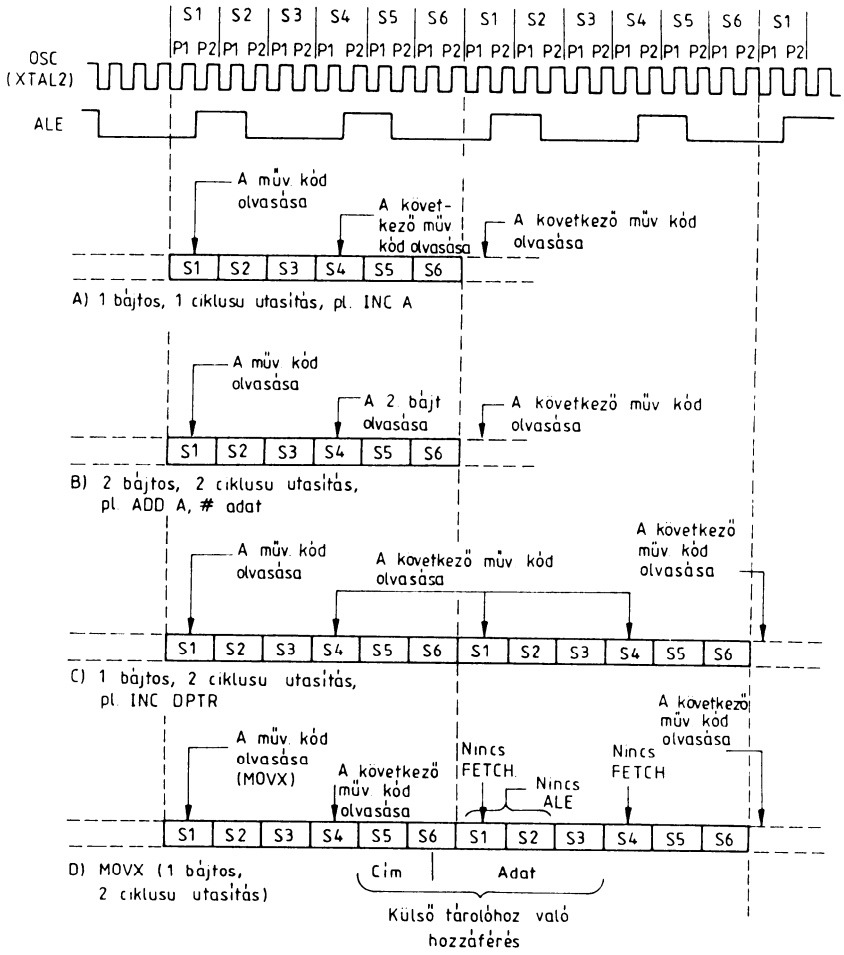
A 2.23. ábra szemlélteti a 8051/8052-es típusjelű mikrokontroller belső időzítését. Az S1-S6, ill. a P1 és P2 csak belső jelek, kívülről nem mérhetők. Az ábra alapján végigkövethetjük a különböző hosszúságú utasítások behívási és feldolgozási folyamatát. Mivel az ALE jel gépi ciklusonként kétszer aktív, ezért a programtárolóból legfeljebb két bájtot lehet ciklusonként lehívni.

Azok az utasítások, amelyek nem tartalmaznak adatot is, legfeljebb kétbájtosak. Ezért végrehajtásuk egy műveleti ciklus alatt befejeződik.

A következő utasítás címe a programszámlálóban tárolódik. Innen közvetlenül, önálló CPU művelet nélkül kapcsolódik a címbuszra. A folyamat időzítése a következő. Az utasítás első bájtját a mikrokontroller az S1P2 ütemben a programtárolóból a saját utasítástárolójába olvassa. Itt dekódolja, és ha egybájtos az utasítás, akkor végre is hajtja. Ezzel egyidejűleg az utasításszámláló tartalmát megnöveli. Az S4P2 ütemben beolvassa a következő bájtot a programtárolóból.

Kétbájtos utasítás esetében ez lesz az utasítás második fele. Egybájtos utasítás esetében - a most beolvasott bájt a következő utasításhoz tartozik. Ekkor ezt az értéket nem dekódolja, és a programszámlálót sem inkrementálja. A következő S1P2 ütemben ugyanaz a bájt kerül beolvasásra. A leírtak adják a feldolgozás időzítését.

Egy gépi ciklus alatt tehát legalább egy utasítás dekódolása és végrehajtása lezajlik.



2.23. ábra. A CPU időzítései

A leírtak alól kivétel a MUL, a DIV és a MOVX utasítás. A MUL és DIV utasítások teljes végrehajtásához négy gépi ciklus szükséges.

A MOVX típusú utasításokat a kontroller két ciklus alatt hajtja végre.

Az első ciklusban a programtárolóból, a második ciklusban pedig az adattároló és kontroller között megy végbe adatátvitel.

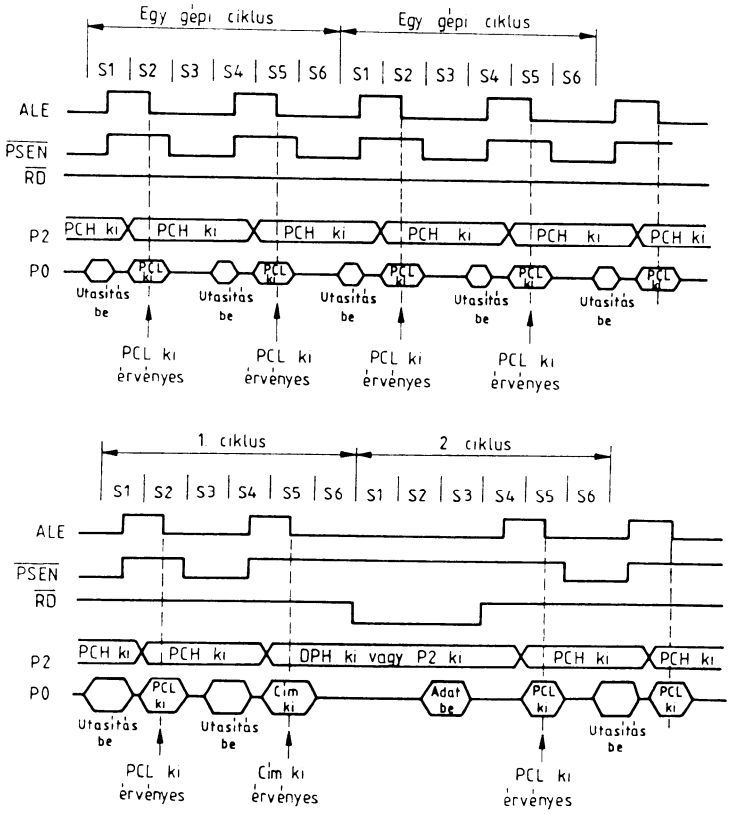
A címnek az ALE jel negatív élekor kell stabilnak lennie. Az utasítás feldolgozása alatt folyik a programszámláló inkrementálása. A következő utasításhoz tartozó cím pedig az S5P2 ütemben kerül a címbuszra.

2.11.3 A PSEN jel

A 8051-es alkalmazásaiban a külső adat- és programmemória együttes használatával is találkozhatunk. E két memória párhuzamosan csatlakozik a cím-, ill. az adatbuszra. A két memória megkülönböztetéséhez egy vezérlőjel szükséges. A 8051-es mikrokontroller-családnál a PSEN (Program Store Enable) jel látja el ezt a feladatot. Ezt a tok 2~es lábán találjuk. Aktív szintje pedig a 0, ami azt jelenti, hogy

a PSEN jel 0 szintje esetén a programtárolóhoz, 1 szintje esetében pedig az adatmemóriához szól a kontroller.

A PSEN - normál esetben - hat oszcillátorperiódus alatt aktív. Egy ciklus alatt tehát kétszer kerül sor a programtároló lekérdezésére. A legtöbb felhasználásban ezt a jelet az EPROM OE (Output Enable) lábára kell csatlakoztatni. Mivel a programtárolóból csak olvasás megy végbe, ezért nem kell megkülönböztetni az adatátvitel irányát. A jelek időzítését a 2.24. ábrán láthatjuk.



2.24. ábra. A külső memóriaelérés időzítései

PSEN jel csak a külső memória kezelésekor létezik. A belső programmemóriából való olvasáskor nem jelenik meg.

2.11.4 A WR és az RD jelek

A külső adattárolónál - a programtárolóval ellentétben - külön kell választani a tárolócellába való írást, ill. az onnan való olvasást. Ezt a feladatot látják el a WR (Write: 16-os láb) és az RD (Read: 17-es láb) jelek. A RD az olvasást, a WR pedig az írást vezérli. Mindkét jel 0-val aktív.

2.12 A tárolók használata

Amint a korábbiakban már említettük, a 8051-es különböző memóriaterületeket tud kezelni. Ezek:
- 64 Kbájt programtároló;
- 64 Kbájt adattároló;
- 256 bájt belső adatmemória.

A belső memória mérete egyes változatokban nagyobb is lehet.

2.12.1 Programtároló

A programtároló logikailag két területre osztható. Az egyik a felhasználói programot tartalmazza, a másikban vannak a megszakítást kiszolgáló rutinok kezdőcímei. Ez utóbbi a programtároló alsó részében helyezkedik el. A 8051-esnél a 0000H-tól a 0023H-ig terjed, míg a 8052-es típusnál ez a 0000H-002BH tartományt foglalja le. Megszakításonként 7 bájt áll a rutin rendelkezésére.

Ez a terület rendszerint nem elegendő a kiszolgáló rutin számára. Ilyenkor ide csak egy ugróutasítást kell beírni, amely a tényleges végrehajtó rutinra adja át a megszakítást.

A programtároló lehet csak külső, vagy részben belső is (2.10. szakasz). A belső programmemória a 8051-esnél a 0000H-OFFFH, ill. a 8052-esnél a 0000H-1FFFH címtartományba esik.

Amikor a programszámlálóban magasabb cím van, mindig a külső memóriából megy végbe az olvasás. Az alacsonyabb címek esetén az Ef1 lábra adott szint határozza meg, hogy belső vagy külső memória-hozzáférés következik-e.

A külső memória-hozzáférést az EA lábra adott 0 szint vezérli.

2.12.2 Adattároló. A belső RAM-terület felépítése

A programmemóriához hasonlóan az adatmemória is egy belső és egy külső részből áll.

A külső adatmemória maximálisan 64 Kbájt lehet. A belső RAM-terület mindig használható.

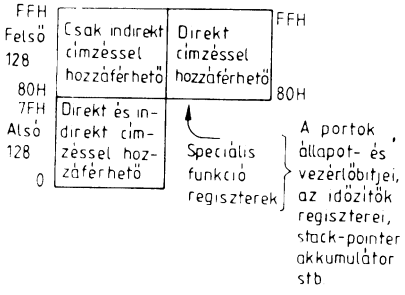
A két terület elérése között az utasításfajtákkal lehet különbséget tenni.

A MOVX utasítások szolgálnak a külső RAM eléréséhez.

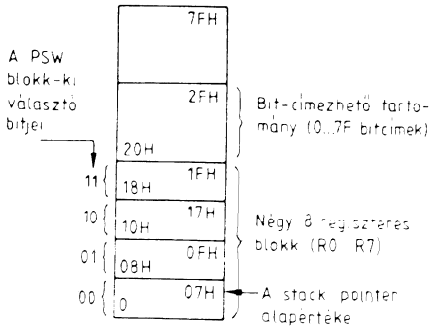
A belső RAM felépítése

A belső adatmemória három különböző területre oszlik. A 2.25. és a 2.26. ábra szemlélteti a megoszlást. Ezek a következők:

- 00H-7FH címtartomány:
  Ez a 128 bájt RAM-terület a 8051-es család mindegyik tagjában megtalálható.
- 80H-0FFH címtartomány:
  A második 128 bájtnyi RAM-terület csak néhány típusban van (2.1. táblázat).
- Speciális funkcióregiszterek (SFR):
  Ezek ugyancsak a 80H-OFFH címterületen érhetők el.



2.25. ábra. A belső RAM területei



2.26. ábra. A belső regiszterbankok

A két azonos című RAM-ot a címzési mód különbözteti meg. A következő módszer mind a 8051-es, mind pedig továbbfejlesztett változataira érvényes.

Az SFR-eket közvetlen címzéssel lehet elérni. A párhuzamos RAM-területhez regiszter-indirekt címzést kell használni.

Részletesebben a 3. fejezetben foglalkozunk a címzési módozatokkal.

Az alsó RAM-terület szerkezete

A RAM alsó 128 bájtnyi területe - a külső RAM-mal ellentétben - nem egyenesen címezhető. Ez a RAM-terület tagolt. Három alapvető szegmensre van osztva

- 00H-1FH címtartomány
  Itt négy - egyenként 8 bájtos - regiszterbank helyezkedik el.

Egyidejűleg a négy bank közül csak egy használható.

A PSW.4 és a PSW.3 bittel kell kiválasztani az aktuális bankot (2.6.13. pont).

- 20H-2FH címtartomány
  Ezen a területen 128, egyenként címezhető bit van. A 20H című bájt legkisebb helyi értékű bitjénél kezdődik a címzés 00H-val. A 7FH bitcíme a 2FH című bájt legmagasabb helyi értékének van.

Bájtcím	Bitcímek							
2FH	7FH	7EH	7DH	7CH	7BH	7AH	79H	78H
2EH	77H	76H	75H	74H	73H	72H	71H	70H
.								
.								
.								
21H	0FH	0EH	0DH	0CH	0BH	0AH	09H	08H
20H	07H	06H	0SH	04H	03H	02H	01H	00H

- 30H-7FH címtartomány
  Ez a terület közönséges RAM-ként használható. Nincs hozzárendelve különleges funkció.

A RESET jel nem hat a RAM tartalmára, ezért a tárolócellák olvasásakor határozatlan értékeket kapunk.

2.13 A 8051-es és a 8052-es típusú mikrokontrollerek összehasonlítása

A következőkben felsoroljuk a 8051-es és a 8052-es típusú mikrokontrollerek jellemzőit.

8051
- 8 bites CPU;
-16 MHz oszcillátorfrekvenciáig használható (szabványos a 12 MHz);
- a tipikus utasítás-ciklusidő 1 μs;
- öt megszakítás kétszintű vezérlése;
- max. 64 Kbájt külső ROM;
- max. 64 Kbájt külső RAM;
- 4 Kbájt belső ROM;
- 128 bájt belső RAM;
- 21 speciális funkcióregiszter;
- változtatható sebességű soros vonal;
- két 16 bites időzítő/számláló;
- a 8085-össel kompatibilis buszkezelés;
- bitműveletek;
- Boole-műveletek;
- DIL 40 és PLCC44 tokozás;
- nagyobb hőmérséklet-tartomány, max. -40...+110 °C;
- rendelésre katonai kivitel.

8052
- 8 bites CPU;
- 16 MHz oszcillátorfrekvenciáig használható (szabványos a 12 MHz);
- a tipikus utasítás-ciklusidő 1 μs;
- hat megszakítás kétszintű vezérlése;
- max. 64 Kbájt külső ROM;
- max. 64 Kbájt külső RAM;
- 8 Kbájt belső ROM;
- 256 bájt belső RAM;
- 26 speciális funkcióregiszter;
- változtatható sebességű soros vonal;
- két 16 bites időzítő/számláló;
- T2 időzítő/számláló továbbfejlesztett jellemzőkkel;
- a 8085-ös típussal kompatíbilis buszkezelés;
- bitműveletek;
- Boole-műveletek;
- DIL 40 és PLCC44 tokozás;
- nagyobb hőmérséklet-tartomány, max. -40...+110 °C;
- rendelésre katonai kivitel.

3 A 8051-es típus utasításkészlete

3.1 A bit- és bájtszervezésű működés

A 8051-es mikrokontroller utasításkészlete nagyon hasonlít a 8080-as típusú mikroprocesszor utasításaihoz. Lényeges eltérés azonban, hogy a 8051-et elsősorban különböző vezérlési feladatokhoz fejlesztették ki. A mikrokontrollert olyan utasításokkal látták el, amelyekkel egyszerűen lehet bitműveleteket végezni. Ezt szolgálja a bitcímzés és a bitekre is értelmezett logikai műveletek (Boole-utasítások). A hagyományos mikroprocesszorokkal összehasonlítva még egy lényeges eltérésre kell felhívni a figyelmet. A mikroprocesszorok főleg bájtszervezésű számításokat végeznek. A vezérlési feladatoknál erre ritkábban van szükség. Többségében egyes kapcsolók állapotát kell lekérdezni, valamint relék ki-, ill. bekapcsolását vezérelni. Az ilyen jellegű működés bitszervezést igényel.

A mikroprocesszorok legtöbbször számokkal dolgoznak, ami bájtszervezésű. A mikrokontrollerek bitszervezésűek.

3.2 Az utasítások hossza

A 8051-es típusú mikrokontrollernek összesen 111 utasítása van. Az utasítások között egy, két és három bájt hosszúakat találunk. Az utasítás bájtszáma a műveleti kódot és - rendszerint közvetetten - az operandusokat is tartalmazza. Az utasítások között
- 49 egybájtos,
- 45 kétbájtos és
- 17 hárombájtos hosszúságú van.

Az utasításhossz alapján állapítható meg a memória-felhasználás és az ugrások címe. Befolyásolja a futási időt is.

3.3 Futási idő

Egy program futási idejének pontos meghatározásához ismerni kell, hogy az egyes utasítások feldolgozása hány gépi ciklust igényel. A 8051-es mikrokontroller utasításainak végrehajtása - a MUL és a DIV aritmetikai utasítások kivételével - egy vagy két gépi ciklus alatt befejeződik. Az előbb említett mindkét utasítás végrehajtásához négy gépi ciklus szükséges.
A 2. fejezetben írtuk le, hogy egy műveleti ciklus 12 oszcillátorperiódus hosszú. Az órajel frekvenciájának ismeretében a program végrehajtásához szükséges idő - a futási idő - a következők szerint számolható ki:

A program összes utasításaihoz tartozó műveleti ciklusok összegét szorozni kell a kvarc periódusidejének 12-szeresével.

A 8051-es utasításai - a végrehajtási ciklusszám alapján - a következők szerint oszlanak meg:
- 63 egyciklusú,
- 46 kétciklusú és
- 2 négyciklusú.

3.4 Címzési módok

3.4.1 A címzésről általában

A 8051-esnél a következő öt különböző címzési mód alkalmazható: - regisztercímzés,
- direkt címzés,
- indirekt címzés,
- közvetlen címzés (más néven bennfoglalt címzés),
- indirekt regisztercímzés.

A mikrokontroller különböző memóriaterületekkel kommunikál. A fizikailag vagy logikailag elkülönített területek más-más címzési móddal érhetők el. A 8051-es típus ötféle címzési módja alapvetően két nagy csoportot alkot, mégpedig a direkt és az indirekt címzésűeket.

Direkt címzés esetén az utasítás része a megfelelő tárolócella címe (hexadecimális szám). Indirekt címzés esetén a tárolócella címe egy regiszterben vagy egy másik tárolócellában van. Az utasítás ez utóbbi címét tartalmazza.

Az indirekt címzés kissé nehézkesebb, mivel a programozónak először arról kell gondoskodnia, hogy a szükséges cím a megfelelő tárolóegységbe kerüljön. Ugyanakkor rugalmasabb programozást tesz lehetővé. Az indirekt címzés egy másik lehetősége az ún. adatmutató, vagyis a Data-Pointer használata. Az adatmutatóba egy 16 bites szám írható, amely 64 Kbájt kapacitású adatmemória címzését teszi lehetővé. Az adott értékkel relatív címzés is megoldható.

A relatív címzés a strukturált programozást teszi könnyebbé. Sok esetben előnyös ez a módszer.

3.4.2 Regisztercímzés

Az aktuális bank R0. . .R7 regiszterei címzésre is használhatók. Segítségükkel a belső RAM egyes memóriaelemei - így az A, B a CY és DPTR is - közötti kommunikáció megvalósítható. Az utasításkód három legalacsonyabb helyi értékű bitje határozza meg az aktuális RO...R7 regisztert.

3.4.3 Direkt címzés

Direkt címzés esetén az utasítás része az elérendő memória címe. A cím az utasítás műveleti kódja utáni hexadecimális szám.
Pl. a MOV A,32H alakú utasítás a 32H című belső memória tartalmát az Akkumulátorba (A) viszi.

A speciális funkcióregiszterek (SFR-ek) csak direkt címzéssel érhetők el.

Az SFR regisztereken kívül a belső RAM alsó 128 bájtja is címezhető direkt módon.

3.4.4 Indirekt címzés

Indirekt címzés esetén nem az utasítás, hanem az aktuális regiszterbank R0 vagy R1 regisztere tartalmazza az elérendő memóriacella címét. A 8 bites tartalommal 256 bájt széles RAM-terület címezhető. E terület lehet a belső vagy a külső RAM-ban is. Természetesen a 8051-es típusnál a belső RAM csak 128 bájtos.

Indirekt címzés esetén egy regiszterben van az a 8 bites cím, amellyel a 256 bájt valamelyikébe írni vagy olvasni akarunk.

Az indirekt címzésre példa a MOVX A, @Ri utasítás, amely az akkumulátorba viszi a külső RAM egy cellájának tartalmát. A cella címe az Ri regiszterben van. A magasabb helyi értékű 8 címbit ezalatt nem változik meg. Az utasítással relatív címzés is megvalósítható. Először a P2 SFR-be kell beírni a cím magasabb értékű bájtját, az Ri-be pedig az alacsonyabb értékű bájtot.
Az indirekt címzést megvalósító utasítások egybájtosak; a teljes 64 Kbájtnyi külső RAM egy kisebb területe érhető el velük. A teljes memóriaterület a DPTR adatmutató regiszterrel címezhető (@ DPTR).
Amikor a stackterület kisebb 256 bájtnál, akkor a PUSH és POP utasításoknál is indirekt a címzés. A cím itt a stackmutatóban (SP) van.

3.4.5 Közvetlen címzés

Közvetlen címzés esetén az adat, amellyel műveletet akarunk végezni, nem valamelyik RAM egyik cellájában, hanem az utasításban van.

Egy állandó beírása esetén a bevitelt előíró műveleti kód és az állandó értéke is az utasításban van.

Ez azt jelenti, hogy az értéket az utasítás részeként a programmemóriába (ROM vagy EPROM) kell beírni.
Pl. a MOV A, #23H utasítás 23 hexadecimális értéket ír az akkumulátorba. A közvetlen címzést leggyakrabban a matematikai vagy a fizikai képleteket megoldó programok során használjuk.

3.4.6 Indirekt regisztercímzés

Indirekt regisztercímzés esetén a tényleges fizikai címet két regiszter tartalmának az összege adja. A cím tehát egy bázis- és egy eltolási (ofszet) címrészből áll. A báziscím vagy az adatmutatóban (DPTR), vagy a programszámlálóban (PC) van. Ehhez a címhez adja hozzá egy belső regiszter pl. @A tartalmát.

Az indirekt regisztercímzési formát rendszerint a programtárolóban lévő táblázat kezeléséhez használjuk.

3.5 A különböző utasításfajták

A 8051-esnek 111 különböző utasítása van. Ezek a következő négy csoportba sorolhatók:
- adatátviteli utasítások,
- aritmetikai utasítások,
- logikai, ill. bitműveleti utasítások,
- vezérlőutasítások.

3.5.1 Adatátviteli utasítások: általános, akkumulátorral kapcsolatos és a Data-Pointeres adatmozgató utasítások

Az adatátviteli - az ún. MOV - utasítások a regiszter vagy a RAM-cella tartalmát viszik át másikba.

A PUSH és POP utasítások is a MOV csoportba tartoznak.

Ez az utasítástípus három csoportra osztható.

Általános adatátviteli utasítások
A csoportot alkotó utasítások:
MOV Egy bitet vagy bájtot visz át regiszterből regiszterbe vagy RAM-ba, ill. fordítva.
PUSH Inkrementálja a Stack-Pointer tartalmát, majd a vonatkozó adatot az SP által címzett memóriába írja.
POP Az SP által címzett memória tartalmát átírja a vonatkozó helyre, majd dekrementálja az SP tartalmát.
A MOV utasítások a címzett helyről átmásolják az adatot a célhelyre, eközben azonban a forrás tartalmát nem változtatják meg.

Akkumulátorral kapcsolatos utasítások
Ezeknél az utasításoknál az ACC regiszter az adatátvitel célja vagy forrása. Az akkumulátor önmaga lehet a cél és a forrás is.
XCH (Exchange) Az akkumulátor és a címzett memória tartalmát cseréli fel.
XCHD Hasonló az XCH utasításhoz, de csak az akkumulátor és a címzett memória tartalmának alsó négy bitjét cseréli meg.
MOVX A külső adatmemória és az akkumulátor között végez adatátvitelt.
MOVC A programtárolóból visz egy bájtot az akkumulátorba. A címzésnél a DPTR-ben vagy PC-ben van a báziscím.

Data-Pointer utasítás
MOV DPTR A megadott 16 bites állandóval tölti fel a Data-Pointer DPH és DPL regisztereit.

3.5.2 Aritmetikai utasítások: összeadás, kivonás, szorzás és osztás

A 8051-es típusú mikrokontroller matematikai műveletei korlátozottak. Csupán a 8 bites, előjel nélküli számokkal lehet műveleteket végezni. Az Overflow-flag (OV) segíti a felhasználót az előjeles számok összeadásakor és kivonásakor. A 8051-es aritmetikai utasításai hasonlítanak a 8080-as és a 8085-ös mikroprocesszorok azonos utasításaihoz.

Összeadó utasítások
ADD A címzett memória és az akkumulátor tartalmát adja össze. Az eredmény az akkumulátorba kerül.
ADDC Mint az ADD utasítás, de még a CY flag értékét is az eredményhez adja.
DA A BCD számok összeadásakor végez korrekciót.
INC Az adott memória tartalmát inkrementálja (1-gyel növeli).

Kivonóutasítások
DEC Az INC utasítás ellentettje. Az adott memória tartalmát dekrementálja (1-gyel csökkenti).
SUBB Az akkumulátor tartalmából levonja az adott bájtot. Ha a CY=1, akkor az eredményből még 1-et levon. A művelet eredménye az akkumulátorba kerül.

Szorzó- és osztóutasítások
A B regisztert (az SFR-ben) kizárólag ezen utasítások esetén használja a kontroller.
MUL Két, előjel nélküli 8 bites számot szoroz össze. A szorzandókat az ACC és a B regiszterekbe kell tölteni. A szorzat kétbájtos lesz. Az eredmény alacsonyabb helyi értékű bájtja az ACC-be, a magasabb helyi értékű pedig a B-be kerül. A 256-nál nagyobb eredmény esetén az OV 1-be íródik.
DIV Az ACC tartalmát osztja a B tartalmával. Előjelet nem vesz figyelembe! Az osztás egész részét az ACC fogja tartalmazni. A maradék kerül a B regiszterbe. Ha a hányados 0, akkor az OV flag 1-be íródik.

3.5.3 Logikai és Boole-utasítások

A logikai utasítások formailag nagyon hasonlóak a mikroprocesszorok utasításaihoz. Ez a csoport viszont - az aritmetikai műveletekkel ellentétben - sokkal bővebb alkalmazási lehetőséget nyújt. A 8051-es mikrokontrollert elsődlegesen vezérlésekhez fejlesztették ki, ezért bitekkel, ill. bájtokkal is képes logikai műveleteket végezni.
ANL Logikai ÉS művelet az operandusok azonos helyi értékű bitjei között. Az eredmény az első operandus helyére íródik, míg a második nem változik meg.
CLR Törli a direkt címzett bitet.
ORL Az ANL-hez hasonlóan végez logikai VAGY műveletet. RL Az akkumulátor tartalmát egy hellyel balra forgatja.
RLC Az akkumulátor tartalmát a CY közbeiktatásával forgatja egy hellyel balra.
RR Az akkumulátor tartalmát egy hellyel jobbra forgatja.
RRC Az akkumulátor tartalmát a CY közbeiktatásával forgatja egy hellyel jobbra.
Az RLC és RRC utasításoknál a CY az akkumulátor kilencedik bitjének tekinthető.
SETB A CLR utasítás ellentettje. A direkt címzett bitet 1-be írja.
SWAP Felcseréli az akkumulátor felső és alsó négy bitjét.

XRL KIZÁRÓ VAGY művelet két operandus azonos helyi értékű bitjei között. Az eredmény - az ANL és ORL műveletekhez hasonlóan - az első operandus helyére kerül.

Az ANL, ORL műveletek egyes bitek között is alkalmazhatók. A hagyományos diszkrét logikai hálózatok kiválthatók a 8051-es bázisú rendszerrel (pl. a különböző tárolt programú vezérlések).

3.5.4 Vezérlőutasítások: feltétel nélküli CALL és JUMP, feltételes JUMP

A vezérlőutasítások a programokon belüli különböző ugrások végrehajtására alkalmazhatók. Ilyen lehet egy szubrutin hívása vagy egy feltételtől függő programelágazás. A vezérlőutasítások a következő három osztályba sorolhatók:
- feltétel nélküli CALL és JUMP,
- feltételes JUMP,
- megszakítások.

Ezen utasítások közös jellemzője, hogy a programszámláló tartalmát változtatják meg. A PC határozza meg, hogy a kontroller mely címről hív be utasítást. Ennek megváltoztatásával vezérelhető egy programelágazás.

A JUMP és a CALL utasítások megtörik a program tiszta sorrendi (lineáris) lefutását.

Először meg kell ismerni a JUMP, a CALL és a megszakításhívás közötti alapvető különbségeket. - - A JUMP-ok (esetleg egy meghatározott feltételtől függően) a program meghatározott helyére való ugrást vezérlik. A program végrehajtása erről a helyről fog folytatódni. A JUMP utasítás a BASIC GOTO utasításával egyezik meg.
- A CALL utasítás egy szubrutint hív. A rutin feldolgozása egy RET utasításig tart. A RET hatására a főprogram a CALL utasítást követő helyről fut tovább. Ugyanaz a szubrutin a főprogram tetszőleges helyéről és többször is hívható. Elsődlegesen a különböző, összetettebb matematikai műveletekhez használjuk a szubrutinokat.

Egy szubrutin CALL utasítással való hívása kizárólag szoftverből lehetséges.

A megszakítási elágazás - a CALL hoz hasonlóan - egy szubrutin hívását jelenti. Lényeges eltérés, hogy ezt az ugrást egy hardveresemény váltja ki. A szubrutin hívása a megszakítás után azonnal lezajlik. A CALL-lal való szubrutinhívásra addig kell várni, amíg a program az adott CALL-hoz nem ér.

Feltétel nélküli CALL és JUMP

A feltétel nélküli ugrás, mint ahogy erre a neve is utal, nem függvénye valamilyen eredménynek. Az ugrás minden esetben bekövetkezik, amikor a program egy ilyen utasításhoz ér. Szigorúan véve a RETURN utasítás is ebbe a csoportba tartozik. A CALL végrehajtása előtt a főprogram következő utasításának címe a stackbe íródik. Ezután a Stack-Pointer értéke kétszer inkrementálódik (16 bites cím kerül be a stackbe). A RET utasítás hatására ez a cím visszaíródik a PC-be. Végül az SP tartalma kettővel csökken.

ACALL Kétbájtos szubrutinhívó utasítás. 2 Kbájtos szegmensen belüli programugrást hajt végre. Az ACALL-hoz 11 bites cím tartozik. A PC-ben lévő legnagyobb helyi értékű öt bit érvényes marad (együtt adják a 16 bites címet). Az ACALL hívásakor a PC tartalma inkrementálódik. Ha az ACALL egy 256 bájtos szegmens utolsó két bájtja, akkor a PC inkrementálása miatt az a következő szegmensbe kerül.

LCALL Hárombájtos szubrutinhívó utasítás. Hasonló az ACALL hoz, de alkalmas a teljes 64 Kbájton belüli, tetszőleges című rutin hívására.

AJMP/LJMP Lényegében az ACALL és LCALL utasításokhoz hasonlóan használhatók.

SJMP Rövid ugrásutasítás. A SHORT JUMP használatával csak 256 bájtos területen belüli ugrás oldható meg.

JMP@ A+DPTR Az ugrás címét a DPTR és az akkumulátor tartalmának összege adja. A 8 bites akkumulátortartalom egy lapot fog át. A DPTR a teljes 64 Kbájtos programmemória tetszőleges helyére mutathat.

RET Az ACALL vagy az LCALL utasításokkal meghívott szubrutinból való visszatérést vezérli. Hatására a PC-be íródik a szubrutinhívást követő utasítás címe.

Feltételes JUMP

Az előzőekkel ellentétben az ugrás csak akkor következik, ha meghatározott feltétel teljesül.

A feltételes ugrás mindig relatív. Az éppen aktuális helytől számítottan 8 bites címtávolságon belül lehet a célhely.

Ez azt jelenti, hogy a feltételes ugrásoknál az utasítás helyétől számítva -128 vagy +127 területen belül lehet a célcím. A 8051-nek a következő feltételes ugróutasításai vannak:

JZ Akkor következik az ugrás, ha az akkumulátor tartalma 0.

JNZ Akkor következik az ugrás, ha az akkumulátor tartalma nem 0. JC Akkor következik az ugrás, ha a Carry-Flag 1.

JNC Akkor következik az ugrás, ha a Carry Flag 0.

JB Akkor következik az ugrás, ha egy tetszőleges, direkt címzett bit 1.

JNB Akkor következik az ugrás, ha egy tetszőleges, direkt címzett bit 0.

JBC Akkor következik az ugrás, ha egy tetszőleges, direkt címzett bit 1. Utána törli a bitet.

CJNE Összehasonlít két 8 bites értéket. Akkor következik az ugrás, ha a két tartalom nem egyforma. Amikor az elsőnek adott regiszter tartalma a kisebb, akkor a CY is 1-be íródik. Ellenkező esetben törlődik.

DJNZ Először dekrementálja az adott címen lévő értéket, majd ellenőrzi, hogy a csökkentett érték 0, vagy pedig nem, és az utóbbi esetben hajtja végre az előírt ugrást.

A továbbiakban az utasításokat részletesebben ismertetjük. Egyúttal megadjuk a pontos Assembly jelölésüket (mnemonik) is. Az INTEL cég processzoraira és kontrollerjeire érvényes a következő írási sorrend: első a művelet, második az adat célterülete és végül következik a forrásterület.
A felhasználói könyvek, katalógusok a következő rövidítéseket használják:

Rn	Az R0-R7 munkaregiszterek valamelyikét jelöli. Az utasítás e regiszterek tartalmára vonatkozik.
Direct	Egy cím, a belső RAM-ban I/O port vagy státuszregiszter. Az utasításban hexadecimálisan kell megadni.
@Ri	Az R0 vagy az R1 regiszterek tartalma, amely az elérendő bájt címe.
#data	Az utasításban megadott 8 bites adat.
#data 16	Az utasításban megadott 16 bites adat (a 2. és a 3. bájt).
rel	Egy relatív cím. A következő utasítás címéhez viszonyítottan -128 és +127 területen belülre mutathat.
bit	Jelentheti a 128 direkt bit (szoftver-flag) valamelyikét, egy I/O bitet, ill. vezérlő- vagy státuszbitet. Az összes mnemonikjelölést az Intel Corporation 1980-as leírása alapján adjuk meg.

3.6 Adatátviteli utasítások

3.6.1 MOV

A MOV utasítások egy memóriacella adatát másolják egy másikba. A forráscímű bájt értéke nem változik. A MOV utasítások felépítése a következő:
MOV (cél), (forrás)
A MOV utasításokkal bit, bájt vagy dupla bájt (szó) átvitele történhet.
- MOV utasítások bájtátviteléhez
MOV (célbájt), (forrásbájt)

Mind a (célbájt), mind pedig a (forrásbájt) egymástól függetlenül, tetszőlegesen címezhető.

A MOV utasítások tehát nagyon rugalmasan használhatók.

Mnemonikjelölések és leírások

MOV A, Rn	Az Rn regiszter tartalmát az akkumulátorba írja.
MOV A, direct	A címzett (hexadecimális szám) memóriacella tartalmát írja az akkumulátorba.
MOV A, @Ri	Az Ri regiszterrel címzett memóriacella tartalmát írja az akkumulátorba.
MOV A, #data	Az utasításban adott bájtot az akkumulátorba írja.
MOV Rn, A	Az akkumulátor tartalmát az Rn regiszterbe írja.
MOV Rn, direct	A címzett (hexadecimális szám) memóriacella tartalmát írja az Rn regiszterbe.
MOV Rn, #data	Az utasításban adott bájtot az Rn regiszterbe írja.
MOV direct, A	Az akkumulátor tartalmát a címzett memóriacellába írja.
MOV direct, Rn	Az Rn tartalmát a címzett memóriacellába írja.
MOV direct, direct	Egy memóriacella tartalmát egy másikba másolja. Mindkét cím az utasításban van.
MOV direct, @Ri	Az Ri regiszterrel címzett memóriacella tartalmát másolja át a direkt címzett cellába.
MOV direct, #data	Az utasításban megadott bájtot másolja a direkt címzett memóriacellába.
MOV @Ri, A	Az Ri regiszterrel címzett memóriacellába másolja az akkumulátor tartalmát.
MOV @Ri, direct	Az Ri regiszterrel címzett memóriacellába másolja a direkt címzett memóriacella tartalmát.
MOV @Ri, #data	Az Ri regiszterrel címzett memóriacellába másolja az utasításban megadott bájtot.

- MOV utasítás dupla bájt átviteléhez

Csak egy két bájtot mozgató adatátviteli utasítás van.

MOV DPTR, #datal6	A Data-Pointert az utasításban adott 16 bites értékkel tölti fel. Ez egy olyan MOV utasítás, amely két bájtot mozgat. A DPH-ba, ill. a DPL-be kerül az utasítás második és harmadik bájtja.

- MOV utasítások bitátvitelhez

Ezek az utasítások csak egy bitet mozgatnak. Megkönnyítik a 8051-es vezérléstechnikai alkalmazását. Felépítésük megegyezik a többi MOV utasításéval.
- MOV (célbit), (forrásbit)
Ez az utasításcsoport nem olyan rugalmas, mint a bájtátviteli MOV utasítások. A korlátozás a következő:

A bitmozgató utasítások esetében az egyik hely (a cél vagy a forrás) csak a Carry-Flag lehet.

Közvetlenül nem lehet egy bitet, pl. egyik portlatch bitjét egy másikba átmásolni.

MOV P1.2, C	A CY értékét másolja a P1 port 2-es bitjébe (a port tárolójába).

MOV C, P1.2 A P1 port 2-es bitjét a CY-ba írja.

3.6.2 MOVC

A MOVC utasítások egy bájtot vagy egy állandót írnak az akkumulátorba.

A MOVC utasításokkal csak a programtároló érhető el! Az adattároló kezeléséhez nem használható!

MOVC utasítások esetén a kiolvasandó memóriacella címét az akkumulátor 8 bites tartalma és egy bázisregiszter tartalma adja. Bázisregiszterként a Data-Pointer DPTR vagy a programszámláló PC használható. Az ily módon címzett memóriacella tartalmát másolja át az akkumulátorba. Az utasítás nem változtatja a flageket.

MOVC A, @A+DPTR Az ACC és a DPTR tartalmának összege adja a forrás címét. E memóriacella tartalmát másolja át az akkumulátorba.

MOVC A, @A+PC Az ACC és a PC tartalmának összege adja a forrás címét. E memóriacella tartalmát másolja át az akkumulátorba.

3.6.3 MOVX

A MOVX utasítások a MOV utasítások kibővítései.

A MOVX utasításokkal csak a külső adatmemória érhető el!

Az utasítás végére ki kell írni az X-et!
A MOVX utasításoknak két változatuk van. Az egyik típus a 8 bites, a másik 16 bites címmel éri el a memóriát.

- 8 bites címzés

A 8 bites címet a multiplexelt P0 port szolgáltatja. Ugyanekkor a P2 port tartalma nem változik. Ez lehetőséget ad arra, hogy a P2-be egy adott értéket írjunk. Ez lesz a cím magasabb helyi értékű bájtja. A címzésnek ezt a módját csak viszonylag kis terület (256 bájt) elérése esetén használják korlátozott méretű memória vagy I/O szegmens alkalmazásakor. A memóriaszegmens a P2 megfelelő beírásával a RAM tetszőleges helyére kijelölhető.

MOVX A, @Ri Az Ri regiszter tartalmával címzett külső memóriacella tartalmát az ACC-be másolja.

MOVX @Ri, A Az ACC tartalmát az Ri regiszter tartalmával címzett külső memóriacellába másolja.

- 16 bites címzés

Ebben az esetben a DPTR-t használjuk a címzéshez. A DPH tartalma a P2, a DPL tartalma a PO portra kerül. A P2 SFR tároló értéke nem változik. A címzés mindig a portlábakon keresztül megy végbe.

MOVX A, @DPTR A DPTR tartalmával címzett külső memóriacella tartalmát másolja az ACC-be.

MOVX @DPTR, A Az ACC tartalmát másolja a DPTR tartalmával címzett külső memóriacellába.

A 16 bites címzésű MOVX utasításokat a nagyobb terület gyors elérésekor használjuk. A 16 bites cím előállításához nem kell más utasítás. A MOVX utasítást 8 bites címmel a kisebb memóriaterülettel való kommunikációhoz használjuk. Ilyenkor a P2 port szabadon felhasználható.

3.6.4 PUSH

A PUSH utasítás egy bájtot másol a stack tetejére. A pontos működés a következő:

PUSH direct Először inkrementálja a Stack-Pointer tartalmát. Ezután az SP által mutatott címre másolja az utasításban megadott című memóriacella tartalmát.

3.6.5 POP

POP direct A POP utasítás a PUSH ellentettje. Az SP által címzett memóriacella tartalmát kiolvassa és az utasításban megadott című cellába írja. Végül dekrementálja az SP tartalmát.

3.6.6 XCH

Az EXCHANGE utasítások az akkumulátor és a címzett bájt tartalmát cserélik fel. Különböző címzések alkalmazhatók. Az XCH utasítás a következő felépítésű:

XCH A, (bájt) A művelet elvégzése után az ACC tartalma a címzett memóriacellába, annak eredeti értéke pedig az ACC-be íródik. A flageket nem befolyásolja.

XCH, A, Rn Az ACC és az Rn regiszter tartalma felcserélődik.

XCH A, direct Az ACC és a direkt címzésű memória tartalma felcserélődik.

XCH A, @Ri Az ACC és az Ri regiszterrel címzett memória tartalma felcserélődik.

3.6.7 XCHD

Az EXCHANGE-DIGIT utasítás az XCH utasítások alváltozata.

Az utasítás a forrás és a cél alsó négy bitjét cseréli fel.

A magasabb helyi értékű 4 bit és a flagek változatlanok. Az XCHD utasítás csak indirekt címzéssel használható.

XCHD A, @Ri Az ACC és az Ri tartalmával címzett memóriacella alsó négy bitjét felcseréli.

3.7 Aritmetikai utasítások

A 8051-es típusú mikrokontrollerrel összeadni, kivonni, inkrementálni, dekrementálni, valamint - korlátozott formában - szorozni és osztani lehet. Az utolsó két utasítás kivételével az aritmetikai műveletek azonosak a többi Intel, ill. Z80 processzor utasításaival.
Mind a szorzás, mind pedig az osztás 8 bites operandusokkal történik. Ez elég erős korlátozás. A 8051-es fejlesztésének időpontjában még ez a korlátozott képességű, de egy utasítással hívható szorzás és osztás is jelentős előrelépés volt. A 8051-es család továbbfejlesztésekor az aritmetikai lehetőségeket is bővítették. Példaként a 80517-es mikrokontroller szorzás- és osztásműveleteit említjük.

3.7.1 ADD

Az ADD utasítással két bájtot lehet összeadni.
Az összeadásnál a CY bitet nem veszi figyelembe az ADD utasítás.
Az utasításcsoport általában a következő felépítésű:

ADD A, (forrás)	A forrás az adatmemória egy bájtja, amelyet különbözőképpen címezhetünk. Az utasítás ezt a bájtot adja az akkumulátor tartalmához, és az összeg az akkumulátorba kerül. Az ADD művelet állítja a Carry-, ill. az Auxiliary-Carry flageket. Az első a 7. bitnél, az utóbbi a 3. bitnél keletkező túlcsordulást jelzi. Az aritmetikai túlcsordulás esetében az Overflow-flag íródik 1-be. Az ADD műveletek során négyféle címzési módot alkalmazhatunk: a regiszter-, a direkt, a közvetlen és az indirekt regisztercímzéseket.
ADD A, Rn	Az akkumulátor és az Rn regiszterek tartalmát adja össze.
ADD A, direct	Hozzáadja az akkumulátor tartalmához a direkt címzésű bájtot.
ADD A, @Ri	Hozzáadja az akkumulátor tartalmához az Ri-vel címzett bájtot.
ADD A, #data	Hozzáadja az akkumulátor tartalmához az utasításban megadott számot.

3.7.2 ADDC

Az ADD-WITH-CARRY típusú utasítások az ADD utasításokhoz hasonlóak.

Az ADDC utasításoknál a CY flag is az összeadás tényezője.

Az ADDC utasítás végrehajtásakor a következő tényezők összege kerül az ACC-be:

Az eredeti ACC tartalom + a megadott RAM-cella tartalma + a CY értéke.

ADDC A, Rn	Az akkumulátor és az Rn regiszter tartalmát, valamint a CY-t adja össze.
ADDC A, direct	Az akkumulátor tartalmát, a direkt címzésű bájtot és a CY-t adja össze.
ADDC A, @Ri	Az akkumulátor tartalmát, az Ri-vel címzett bájtot és a CY-t adja össze.
ADDC A, #data	Az akkumulátor tartalmát, az utasításban megadott számot és a CY-t adja össze.

3.7.3 SUBB

A SUBB utasítások az akkumulátor tartalmából vonják ki a címzett bájtot. Az eredmény az akkumulátorba kerül.

A kivonás mindig a Borrow-val (áthozatbittel) együtt történik. Ezt az utasítás mnemonikjában a második B jelzi.

A Borrow a CY flag helyén keletkezik.
A SUBB utasításokhoz a regiszter-, a direkt, a közvetlen és az indirekt regisztercímzéseket lehet használni.

SUBB A, Rn	Kivonja az akkumulátor tartalmából az Rn tartalmát és a CY (Borrow) értékét.
SUBB A, direct	Kivonja az akkumulátor tartalmából a direkt címzésű bájtot és a CY (Borrow) értékét.
SUBB A, @Ri	Kivonja az akkumulátor tartalmából az Ri-vel címzett bájtot és a CY (Borrow) értékét.
SUBB A, #data	Kivonja az akkumulátor tartalmából az utasításban adott számot és a CY (Borrow) értékét.

3.7.4 INC

Az INCREMENT utasítások 1-gyel megnövelik az adott memóriacella tartalmát. Ha pl. az adott tartalom OFFH, akkor az INC művelet után az új érték 00H lesz. A következőre kell figyelni:

Amikor az INC utasítás végrehajtása túlcsordulást okoz, akkor OV és CY flagek nem változnak!

Az INC utasításoknál a regiszter-, a direkt és az indirekt regisztercímzések használhatók.
Az INC utasítás READ-MODIFY-WRITE típusú. Amikor egy portra használjuk, akkor nem a port kivezetésein lévő értékeket, hanem a latch tartalmát olvassa be és változtatja meg.

INC A	Inkrementálja az akkumulátort.
INC Rn	Inkrementálja az Rn regisztert.
INC direct	Inkrementálja a direkt címzésű bájtot.
INC @Ri	Inkrementálja az Ri-vel címzett bájtot.
INC DPTR	Inkrementálja a Data-Pointer-t.

3.7.5 DEC

A DECREMENT utasítások az INC utasításokhoz hasonlóak, azzal a különbséggel, hogy a címzett memóriacella tartalmát 1-gyel csökkentik. A Data-Pointer-t nem lehet dekrementálni. Ha egy memóriacella 00H értékét dekrementáljuk, akkor az új érték 0FFH lesz.

DEC A	dekrementálja az akkumulátort;
DEC Rn	dekrementálja az Rn-t;
DEC direct	dekrementálja a direkt címzett bájtot;
DEC @ Ri	dekrementálja az Ri-vel címzett bájtot;

3.7.6 MUL és DIV

Mindkét utasítás két 8 bites regiszter tartalmával végez előjel nélküli műveletet (szorzást, ill. osztást). Segítségükkel egyszerű szorzások vagy osztások végezhetők. Több-bájtos számokat ezen alaputasítások felhasználásával lehet szorozni vagy osztani.

MUL AB	Az ACC és a B regiszterek tartalmát szorozza össze. A szorzat 16 bites lesz, amelynek a magasabb helyi értékei a B-be, az alacsonyabbak pedig az ACC-be kerülnek. Ha a szorzat nagyobb, mint 0FFH, és ezért a B-be is került értékes szám, azt az OV flag 1-be írásával jelzi. A művelet törli a CY-t.
DIV AB	Az akkumulátor tartalmát osztja a B regiszterben lévő értékkel. A hányados egész része kerül az ACC-be, a maradék pedig a B-be íródik. A CY és az OV flaget törli. Kivétel az az eset, amikor a hányados 0. Ezt az OV=1 jelzi.

3.7.7 DA

A DECIMAL ADJUST utasítás az akkumulátor tartalmának BCD korrekciójához használható. A matematikai műveletek bináris végrehajtása után a DA utasítással korrigálható BCD formátumúra az eredmény. A mikroszámítógépek rendszeresen használnak ilyen korrekciót (pl. a Z80 DAA utasítása). Részletes leírását mellőzzük.

3.8 Logikai és Boole-utasítások

Ez az utasításcsoportok két bit vagy két bájt között végeznek különböző logikai műveleteket.

A logikai utasítások bájtos operandusokkal is bitenkénti műveletet hajtanak végre.

3.8.1 Logikai utasítások

Ez a csoport az ún. "klasszikus" logikai műveletek, mint pl. az AND vagy az OR, ill. az akkumulátorforgatás utasításait tartalmazza.

ANL A, Rn	Logikai ÉS műveletet végez az akkumulátor és az Rn regiszter tartalma között. Az eredmény az akkumulátorba kerül.
ANL A, direct	Logikai ÉS műveletet végez az akkumulátor és a direkt címzésű bájt között. Az eredmény az akkumulátorba kerül.
ANL A, @Ri	Logikai ÉS műveletet végez az akkumulátor és az Ri-vel címzett bájt között. Az eredmény az akkumulátorba kerül.

ANL A. #data	Logikai ÉS műveletet végez az akkumulátor és az utasításban megadott szám között. Az eredmény az akkumulátorba kerül.
ANL direct, A	Logikai ÉS műveletet végez az akkumulátor és a direkt címzésű bájt között. Az eredmény a direkt címzésű memóriacellába kerül.
ANL direct, #data	Logikai ÉS műveletet végez a direkt címzésű bájt és az utasításban megadott szám között. Az eredmény a direkt címzésű memóriacellába kerül.
ORL A, Rn	Logikai VAGY műveletet végez az akkumulátor és az Rn regiszter tartalma között. Az eredmény az akkumulátorba kerül.
ORL A, direct	Logikai VAGY műveletet végez az akkumulátor és a direkt címzésű bájt között. Az eredmény az akkumulátorba kerül.
ORL A, @Ri	Logikai VAGY műveletet végez az akkumulátor és az Ri-vel címzett bájt között. Az eredmény az akkumulátorba kerül.
ORL A. #data	Logikai VAGY műveletet végez az akkumulátor és az utasításban megadott szám között. Az eredmény az akkumulátorba kerül.
ORL direct, A	Logikai VAGY műveletet végez az akkumulátor és a direkt címzett bájt között. Az eredmény a direkt címzésű memóriacellába kerül.
ORL direct, #data	Logikai VAGY műveletet végez a direkt címzésű bájt és az utasításban megadott szám között. Az eredmény a direkt címzésű memóriacellába kerül.
XRL A, Rn	KIZÁRÓ VAGY műveletet végez az akkumulátor és az Rn regiszter tartalma között. Az eredmény az akkumulátorba kerül.
XRL A, direct	KIZÁRÓ VAGY műveletet végez az akkumulátor és a direkt címzésű bájt között. Az eredmény az akkumulátorba kerül.
XRL A, @Ri	KIZÁRÓ VAGY műveletet végez az akkumulátor és az Ri-vel címzett bájt között. Az eredmény az akkumulátorba kerül.
XRL A, #data	KIZÁRÓ VAGY műveletet végez az akkumulátor és az utasításban megadott szám között. Az eredmény az akkumulátorba kerül.
XRL direct, A	KIZÁRÓ VAGY műveletet végez az akkumulátor és a direkt címzésű bájt között. Az eredmény a direkt címzésű memóriacellába kerül.
XRL direct, #data	KIZÁRÓ VAGY műveletet végez a direkt címzésű bájt és az utasításban megadott szám között. Az eredmény a direkt címzésű memóriacellába kerül.
CLR A	Az ACC törlése. Az akkumulátor minden bitje 0 lesz.
CPL A	Az ACC komplementálása. Az akkumulátor minden bitjét külön-külön invertálja.
RL A	Az ACC tartalmát balra forgatja. Az akkumulátor minden bitjét 1-gyel balra lépteti. A 7-es bit a 0-ás bit helyére lép. A flageket nem változtatja meg.
RR A	Az ACC tartalmát jobbra forgatja. Az akkumulátor minden bitjét 1-gyel jobbra lépteti. A 0-ás bit a 7-es bit helyére lép. A flageket nem változtatja meg.
RLC A	Az ACC tartalmát a CY-on keresztül balra forgatja. Az akkumulátor minden bitjét 1-gyel balra lépteti a CY közbeiktatásával. A 7-es bit lép a CY-ba, a CY pedig a 0-ás bit helyére. A flageket nem változtatja meg.
RRC A	Az ACC tartalmát a CY-on keresztül jobbra forgatja. Az akkumulátor minden bitjét 1-gyel jobbra lépteti a CY közbeiktatásával. A 0-ás bit lép a CY-ba, a CY pedig a 7-es bit helyére. A flageket nem változtatja meg.
SWAP A	Az akkumulátor alsó és felső 4 bitjét felcseréli. A 7-es bit helyére a 3-as bit, a 4-es bit helyére pedig a 0-ás bit kerül, ill. fordítva.

3.8.2 Boole-utasítások

A Boole-utasítások mindig bitekkel végeznek műveleteket. Vagy megváltoztatnak egy bitet, vagy egy bit aktuális értékétől függően hajtanak végre műveletet, pl. programugrást.

E csoportnál - a logikai utasításokkal ellentétben - a címzési lehetőség erősen redukált.

CLR C	A Carry törlése. A CY flag értéke 0 lesz.
CLR bit	Törli a direkt címzésű bitet. A 8 bites címzés miatt csak a belső RAM, ill. az SFR bitjeire vonatkozhat a művelet.
SETB C	A Carry beírása. A CY flag értéke 1 lesz.

SETB bit	1-be írja a direkt címzésű bitet. A 8 bites címzés miatt csak a belső RAM, ill. az SFR bitjeire vonatkozhat a művelet.
CPL C	Invertálja a CY flaget.
CPL bit	Tetszőleges, direkt címzésű bitet invertál.
ANL C, bit	A CY és a direkt címzésű bit között logikai ÉS műveletet végez. Az eredmény a CY-ba kerül.
ANL C, /bit	A CY és a direkt címzésű bit komplemense között logikai ÉS műveletet végez. Az eredmény a CY-ba kerül.
ORL C, bit	A CY és a direkt címzésű bit között logikai VAGY műveletet végez. Az eredmény a CY-ba kerül.
ORL C, /bit	A CY és a direkt címzésű bit komplemense között logikai VAGY műveletet végez. Az eredmény a CY-ba kerül.
MOV C, bit	Egy direkt címzésű bit értékét a CY-ba másolja.
MOV bit, C	A CY értékét a direkt címzésű bitbe másolja.

Megjegyzés.
A két utóbbi utasítás csak nevében MOV, de ténylegesen nem adatmozgató utasítás. Nem bájtot mozgatnak, csak bitértéket változtatnak.

JC rel	Ugrás, ha CY=1. Egy relatív ugrást (-128...+127 bájt) hajt végre, ha a CY=1.
JNC rel	Egy relatív ugrást hajt végre, ha a CY=0.
JB bit, rel	Egy relatív ugrást (-128...+127 bájt) hajt végre, ha a direkt címzésű bit értéke 1.
JNB bit, rel	Egy relatív ugrást hajt végre, ha a direkt címzésű bit értéke 0.
JBC bit, rel	Egy relatív ugrást hajt végre, ha a direkt címzésű bit értéke 1. Végül törli ezt a bitet.

3.9 Vezérlőutasítások

A vezérlőutasítások a főprogram lineáris (sorrendi) futását megtörik, és a program vagy a főprogram egy meghatározott helyén, vagy egy szubrutin elején folytatódik. Gyakran az ugrás valamilyen feltétel függvénye is.

3.9.1 CALL

A CALL utasításokkal lehet az olyan alprogramokat, ún. szubrutinokat hívni, amelyekből a RET utasítással megy végbe a visszatérés.
A 8051-es a CALL utasítás két fajtáját használja: az ACALL-t és az LCALL t. E kettő egyrészt az utasítás bájtszámában, másrészt a lehetséges ugrás hosszában különbözik egymástól. Működésük egyforma. Először a PC tartalmát növelik annyival, ahány bájtos az adott utasítás (kettő az ACALL és három az LCALL-nál). Ekkor a PC a CALL után következő utasításra mutat. Ez az érték bájtonként a verembe másolódik. Minden bájt átvitelekor növekszik az SP tartalma. A művelet végén az SP a verem újabb szabad helyére mutat. Befejezésül a PC-be másolja az utasításban megadott címet. Ezzel a szubrutin futtatása elkezdődik.
A RET utasítás vezérli a visszatérést a szubrutinból. Dekrementálja az SP-t, majd az így kapott címről másol a PC-be egy bájtot. Még egyszer megismétli a folyamatot. Ekkor a CALL t követő utasítás címe kerül a PC-be. Ezután folytatódik a főprogram futtatása.
A következőkre kell figyelni:

A CALL utasítás inkrementálja az SP-t, amely akkor a RAM következő címére mutat. A felhasználónak kell biztosítania, hogy hasznos adatot ne írjon felül.

ACALL addr11	Abszolút CALL. Minden esetben hív szubrutint. Nincs feltételhez kötve. A 2 bájtos utasításban van a 11 bites cím. Ez a 11 bit felülírja a PC alsó 11 helyi értékét. Az efeletti bitek változatlanok. Az utasítással csak 2 Kbájt területen belüli szubrutin hívható.
LCALL addr16	Hosszú CALL. Működése megegyezik az előzőével. Az utasítás 3 bájtos. Ez lehetőséget ad két teljes bájtos (16 bites) címzéshez. A szubrutin tehát a programmemória tetszőleges helyén lehet.
RET	Return. A szubrutinból való visszatérést vezérli. A PC-be helyezi a szubrutinhívást követő utasítás címét.

3.9.2 JUMP

A JUMP utasítások segítségével - a CALL utasításokhoz hasonlóan vezérelhető a futtatás a programtároló adott címére. Ezen a címen viszont nem szubrutin kezdődik, hanem a főprogram egy része.

A JUMP utasításoknál nem használható a RET.

A JUMP utasítások nem változtatják meg a stack tartalmát, vagyis nem helyeznek le visszatérési címet. A JUMP-pal vezérelt ugrás lehet feltételhez kötött is. A feltételes ugrás akkor következik be, ha az utasításban megadott feltétel teljesül.

A JUMP utasítások a főprogramon belüli elágazásokat - adott esetben a feltétel teljesülésekor - vezérlik. A CALL utasítások szubrutinhívást hajtanak végre.

Az utasítások egyetlen regisztert sem használnak. - Feltétel nélküli ugrások:

AJMP addr11	Abszolút ugrás. Működése hasonló az ACALL utasításéhoz, de nem használható a RET.
LJMP addrl6	Hosszú ugrás. Működése hasonló az LCALL utasításéhoz, de nem használható a RET.
SJMP rel	Rövid ugrás. Az AJMP és az LJMP utasításokhoz hasonlóan feltétel nélküli ugrás. Az ugrás címe a PC tartalma + 2 + az utasításban adott rel érték. Csak a -128, +127 relatív címtartományú területen belül használható.
JMP @A+DPTR	Az akkumulátor és a Data-Pointer tartalmának összege lesz az ugrás címe.

- Feltételes ugrások

JZ rel	Amikor az ACC=0, akkor lesz ugrás. Az ugrás címe a PC tartalma + 2 + az utasításban adott rel érték. Az összegnek a -128 és +127 közötti tartományban kell lennie.
JNZ rel	Amikor az ACC < > 0, akkor lesz ugrás. Egyébként úgy működik, mint a JZ utasítás.
CJNE A, direct, rel	Összehasonlítja a direkt címzésű bájtot az ACC tartalmával. Egyenlőtlenség esetén következik be az ugrás. Az ugrás címe a PC tartalma + 2 + az utasításban adott rel érték. Az összegnek a -128 és +127 közötti tartományban kell lennie.
CJNE A, #data, rel	Összehasonlítja az utasításban megadott hexadecimális számot az ACC tartalmával. Egyenlőtlenség esetén következik be az ugrás. Az ugrás címe a PC tartalma + 2 + az utasításban adott rel érték. Az összegnek a -128 és +127 közötti tartományban kell lennie.
CJNE Rn. #data, rel	Összehasonlítja az utasításban megadott hexadecimális számot az Rn tartalmával. Egyenlőtlenségnél következik be az ugrás. Az ugrás címe a PC tartalma + 2 + az utasításban adott rel érték. Az összegnek a -128 és +127 közötti tartományban kell lennie. CJNE @Ri, #data, rel Összehasonlítja az utasításban megadott hexadecimális számot az Ri-vel címzett memóriacella tartalmával. Egyenlőtlenség esetén következik be az ugrás. Az ugrás címe a PC tartalma + 2 + az utasításban adott rel érték. Az összegnek a -128 és +127 közötti tartományban kell lennie.
DJNZ RN, rel	Dekrementálja az Rn regisztert, és ugrik, ha a tartalom nem 0. Az ugrás címe a PC tartalma + 2 + az utasításban adott rel érték. Az összegnek a -128 és +127 közötti tartományban kell lennie.
DJNZ direct, rel	Dekrementálja a direkt címzésű bájtot, és ugrik, ha a tartalom nem 0. Az ugrás címe a PC tartalma + 2 + az utasításban adott rel érték. Az összegnek a -128 és +127 közötti tartományban kell lennie.
NOP	Nincs művelet. Az utasítás egyszer inkrementálja a PC-t, de egyéb műveletet nem hajt végre. A NOP utasítást leggyakrabban ciklusidejű késleltetés beiktatásához használják.
RETI	Visszatérés a megszakításból. Egy megszakítást kiszolgáló rutint mindig ezzel az utasítással kell lezárni. Dekrementálja az SP-t, és az ezzel címzett bájtot a PC-be másolja. Ezt a folyamatot még egyszer megismétli. Ekkor a PC a rutin hívását követő utasításra mutat, és innen folytatódik a program futása. A RETI ugyanakkor szabaddá teszi a megszakítási szintet.

3.10 Az utasítások hossza és végrehajtási idejük

3.1. táblázat. Adatátviteli utasítások

Utasítás	Hossz, bájt	Órajel-periódus
MOV A, Rn	1	12
MOV A, direct	2	12
MOV A, @ Ri	1	12
MOV A, #data	2	12
MOV Rn, A	1	12
MOV Rn, direct	1	24
MOV Rn, #data	2	12
MOV direct, A	2	12
MOV direct, Rn	2	24
MOV direct, direct	3	24
MOV direct, @Ri	2	24
MOV direct, #data	3	24
MOV @Ri, A	1	12
MOV @Ri, direct	2	24
MOV @Ri, #data	2	12
MOV DPTR, #datal6	3	24
MOVC A, @A+DPTR	1	24
MOVC A, @A+PC	1	24
MOVX A, @Ri	1	24
MOVX A, @DPTR	1	24
MOVX @Ri, A	1	24
MOVX @DPTR, A	1	24
PUSH direct	2	24
POP direct	2	24
XCH A. Rn	1	12
XCH A, direct	2	12
XCH A, @Ri	1	12
XCHD A,@Ri	1	12

3.2. táblázat. Aritmetikai utasítások

Utasítás	Hossz, bájt	Órajel-periódus
ADD A, Rn	1	12
ADD A, direct	2	12
ADD A, @Ri	1	12
ADD A, #data	2	12
ADDC A, Rn	1	12
ADDC A, direct	2	12
ADDC A, @Ri	1	12
ADDC A, #data	2	12
SUBB A, Rn	1	12
SUBB A, direct	2	12
SUBB A, @Ri	1	12
SUBB A, #data	2	12
INC A	1	12
INC Rn	1	12
INC direct	2	12
INC @Ri	1	12
INC DPTR	1	24
DEC A	1	12
DEC Rn	1	12
DEC direct	2	12
DEC @Ri	1	12
MUL AB	1	48
DIV AB	1	48
DA	1	12

A 3.1.-3.4. táblázatban az egyes utasítások hosszát és az oszcillátor periódusában megadott végrehajtási időt adtuk meg. Ezek alapján a felhasználó kiszámíthatja a szükséges memóriaterületet és a futási időt. A 2. fejezetben már említettük, hogy egy gépi ciklus hossza 12 oszcillátorperiódus.

3.3. táblázat. Logikai és Boole-utasítások

Utasítás	Hossz, bájt	Órajel-periódus
ANL A, Rn	1	12
ANL A, direct	2	12
ANL A, @Ri	1	12
ANL A, #data	2	12
ANL direct, A	2	12
ANL direct, #data	3	24
ORL A, Rn	1	12
ORL A, direct	2	12
ORL A, @Ri	1	12
ORL A, #data	2	12
ORL direct, A	2	12
ORL direct, #data	3	24
XRL A, Rn	1	12
XRL A, direct	2	12
XRL A, @Ri	1	12
XRL A, #data	2	12
XRL direct, A	2	12
XRL direct, #data	3	24
CLR A	1	12
CPL A	1	12
RL A	1	12
RLC A	1	12
RR A	1	12
RRC A	1	12
SWAP A	1	12
CLR C	1	12
CLR bit	2	12
SETB C	1	12
SETB bit	2	12
CPL C	1	12
CPL bit	2	12
ANL C, bit	2	24
ANL C, bit	2	24
ORL C, bit	2	24
ORL C, bit	2	24
MOV C, bit	2	12
MOV bit, C	2	24
JC rel	2	24
JNC rel	2	24
JB bit, rel	3	24
JNB bit, rel	3	24
JBC bit, rel	3	24

3.4. táblázat. Vezérlőutasítások

Utasítás	Hossz, bájt	Órajel-periódus
ACALL addr11	2	24
LCALL addr16	3	24
RET	1	24
AJMP addr11l	2	24
LJMP addr16	3	24
SJMP rel	2	24
JMP	1	24
JZ rel	2	24
JNZ rel	2	24
CJNE A, direct, rel	3	24
CJNE A, #data, rel	3	24
CJNE Rn, #data, rel	3	24
CJNE @Ri, #data, rel	3	24
DJNZ Rn, rel	3	24
DJNZ direct, rel	3	24
NOP	1	12
RETI	1	2

3.11 A jelzőbiteket befolyásoló utasítások

A 8051-es utasításai közül csak nagyon kevés változtatja meg a mikrokontroller flagjeit. A 3.5. táblázatban foglaltuk össze, hogy az egyes flagekre melyik utasítás hat. Az X azt jelenti, hogy az adott flaget az utasítás

3.5. táblázat. A flageket állító utasítások

Utasítás	CY	OV	AC
ADD	X	X	X
ADDC	X	X	X
SUBB	X	X	X
MUL	0	X	
DIV	0	X	
DA	X		
RRC	X		
RLC	X		
SETB C	X		
CLR C	X		
CPL C	X		
ANL C, bit	X		
ANL C, bit	X		
ORL C, bit	X		
ORL C, bit	X		
MOV C, bit	X		
CJNE			

megváltoztatja. A 0, ill. az 1 jelöli azt a konkrét értéket, amelyre a flag mindig beáll a művelet hatására.

READ-MODIFY-WRITE utasítások
A READ-MODIFY-WRITE utasítások egy port tartalmát kiolvassák, a kívánt értékre változtatják és azonnal visszaírják a port-latch-be. A READ-MODIFY-WRITE utasításokhoz mindig egy port címe tartozik. A csoportba a következő utasítások tartoznak:

Utasítás	Példa
ANL	ANL P2,A
ORL	ORL P1,A
XRL	XRL P1,A
JBC	JBC P2.2, rel
CPL	CPL P1.1
INC	INC P1
DEC	DEC P1
DJNZ	DJNZ P1, rel
MOV	MOV P2.l, C
CLR	CLR P1.0
SETB	SETB P1.0

A lista első utasításairól azonnal látható, hogy READ-MODIFY-WRITE utasítások. Az utolsó három esetében első látásra kétséges, hogy milyen utasításokról van szó. Ezekre is érvényes, hogy először a port kiolvasása, az érték módosítása és a visszaírás követi egymást.