

1. Milyen a nyitott kollektoros kimenet?

Ha dióda anódjára pozitív feszültséget kapcsolunk a katódjához képest, a dióda kinyit, míg ellentétes feszültség esetén nem. A dióda két vége között egyik irányban nagy ellenállást, felcserélve rövidzárt kell mérnünk. Ez nevezzük a dióda nyitóirányú feszültségének. A tranzisztor már 2 P-N átmenetet tartalmaz, amely 3 kivezetéshez kapcsolódik. (emitter, kollektor, bázis) A tranzisztor működése: a bázis-emitteren átfolyó nyitóirányú árammal arányos a kollektor-emitter közötti folyó áram. A bázis-emitter diódán átfolyó árammal szabályozni tudjuk a kollektor-emitter kivezetések között átfolyó áramot.

Amikor a **nyitott kollektoros kapuk kimeneteit** összekötjük, és a közös vezetékét a V_{CC} tápfeszültséghez csatlakoztatjuk, akkor ez a közös vezeték csak abban az esetben lesz H szintű, ha valamelyik H szinten van. Ha csak egyetlen kimenet is L szinten van, akkor a közös kimenet is L szinten lesz, azaz a kimenetekkel **ÉS** kapcsolatot valósíthatunk meg. Az ilyen összekötött kimeneteket szokták huzalozott **ÉS** kapcsolatnak is nevezni.

2. Hogyan készíthetünk stabilizált tápfeszültség forrást?

ábra

3. Ismertesse az analóg komparátor működését!

*Amikor az áramkör bemeneti feszültsége meghalad egy referencia feszültséget, akkor ad az áramkör jelet.

4. Az 555-ös időzítő áramkör felépítése és alkalmazásai.

*Felépítése digitális és analóg áramkörök keveréke, impulzusok előállítása. A következő kapcsolásokat lehet megvalósítani vele: 1)astabil multivibrátor: nincs stabil állapota, külső vezérlés nélkül változtatja a kimeneti állapotát. a két kimeneti szint között. 2)monostabil multivibrátor: Egy stabil állapota van a másik állapot csak meghatározott ideig marad fenn, miután a második állapot "lejár" a kapcsolás ismét az eredeti állapotába kerül. 3)bistabil multivibrátor: Két stabil állapot, az átbillenési folyamatot egy külső vezérlőjel váltja ki.

5. Ellenállás létra és R-2R felépítése és használata D/A konverterekben.

*A digitál-analóg átalakítók olyan funkcionális egységek, amelyek a bemeneteikre érkező kódolt számmal arányos feszültséget állítanak elő a kimeneteiken.

6. Műveleti erősítők alkalmazási lehetőségei TTL környezetben.

*TTL: tranzisztor-tranzisztor-logika

1)D/A 2)A/D 3)fázisfordító s+ 4)multivibrátor 5)feszültségmegosztás

7. Magyarozza meg, mik a multiplexerek, demultiplexerek, dekódolók?

Multiplexerek: A multiplexer olyan áramkör, amely a címzésnek megfelelően a bemeneteire kapcsolódó vonalak valamelyikét a kimenetre kapcsolja, azaz a kimenten megjelenik az adott bemenet logikai állapota.

A D0...D3 adatbemenetek valamelyike a Q kimenetre kapcsolódik. Ezt az A0 és A1 címbemenetek logikai állapota határozza meg. Pl.: ha A0=0, az A1=1 szintű, akkor a D2 bemenet kapcsolódik a kimenetre. (ábra 25 oldal)

Demultiplexer: Ennek a fordítottját végzi. Egy bemeneti vonalat a címzésnek megfelelően, kiválasztott kimeneti vonalra kapcsolja.

A D bemenet az A0 és A1 címbemenetek állapotának függvényében a Q0...Q3 adatkimenetre kapcsolódik. Pl.: ha A1=L szintű, akkor a D bemenet kapcsolódik a Q1 kimenetre.

Dekódolók: A dekódoló áramkör az információnak az egyik ábrázolási formáról egy másik ábrázolási formára való átalakítására szolgál. Sok esetben van szükség olyan áramkörre, amelynek a bemenetére bináris értéket kapcsolva, annak csak egyik - adott bináris értékhez tartozó- kimentén jelenik meg jel. Egy másik gyakori feladat a BCD számok megjelenítése. Ez is megoldható dekódoló áramkörrel. Pl.: ha a megjelenítendő szám a nulla (0000), akkor az 1-2-4-8 bemeneteken lévő nullák hatására a G kivételével az összes kimeneteknek a szegmensek LED diódáit ki kell gyújtaniuk.

8. Hogyan működik a digitális komparátor?

A digitális komparátorok olyan áramkörök, amelyek két bináris értéket hasonlítanak össze egymással. A 3 legfontosabb összehasonlítási feltétel: egyenlőség, a kisebb és nagyobb viszony. Két bináris érték egyenlőségének feltétele, hogy minden bitjük megegyezzen. A komparátornak csak akkor van a kimenetén L szint, ha a bemenetén a két szám egyenlő. N bit esetén az N számú ekvivalencia kapu kimeneteit egy **ÉS** kapuhoz csatlakoztatva jutunk az N bites komparátor áramkörhöz.

9. Hogyan működik a teljes összeadó?

Jelölje a két összeadandót A és B, az előző helyi értéktől származó átvitelt D; az összeget S és a keletkező átvitelt C. Ekkor A, B, D összes lehetséges értékét figyelembe véve: (ABDSC) (00000) (00110) (01010) (01101) (10010) (10101) (11001) (11111). Tehát az S oszlop tartalmazza azokat az eseteket, amikor az A+B+D összeg értéke 1, vagy 11 lesz binárisan. C oszlopa azokban az esetekben tartalmaz 1-et, amikor az A+B+D összegzés eredményeként átvitel keletkezik, azaz 10B és 11B esetén.

10. Milyen feladatokra használhatók a tároló regiszterek és osztók?

Tároló regiszterek: Az egyik leggyakrabban használt integrált áramkörös tároló elem a D(ata) tároló. A D bemeneten lévő H vagy L szint a CLK órajel felfutó élére átíródik a Q kimenetre és a /Q negált kimenetén pedig az ellentettje jelenik meg. A PR (preset) bemenetre adott alacsony szint Q=H / Q=L állapot, míg a CL bemenetre adott L szint Q=L / Q=H állapot kialakulását okozza. Ilyen tárolókból állítják elő a több bitet párhuzamosan tároló regisztereket. Ilyenkor az egyes tárolók órajel bemeneteit közömbösítik. (ábra 28. oldal)

Számlálók: A jelek számlálása a tárolás és az összeadás műveleteire bontható. A számláló áramkörnek valamilyen kódban tárolnia kell a már megszámlált jelek számát, majd az újabb jel érkezésekor ehhez 1-et kell hozzáadni,

vagy 1-et kell kivonni. A számlálók egymással összekapcsolt flip-flopokból épülnek fel. A számlálók kivezetései funkcionálisan 4 csoportba sorolhatók: tápfeszültség kivezetés; órajel; a számláló állapotát tartalmazó kimenetek; egyéb, működést meghatározó jel. a működési mód alapján megkülönböztetünk szinkron és asszinkron számlálókat.

A számláló előnyös tulajdonsága hogy kimenetein a bemenőjel leosztott frekvenciájával jelenik meg az órajel.

11. Hogyan lehet nyomógombos kapcsolót TTL bemenethez illeszteni?

ábra

12. Hogyan lehet láthatóvá tenni egy kimenet logikai állapotát?

*LED, lámpa, 7 v. 16 bites kijelző

13. Ismertesse az I/O ciklus idődiagramját!

14. Az output port felépítése.

15. Az input port felépítése.

16. A PC printer portjának felépítése.

17. Ismertesse az ALU felépítését és működését!

Ha az utasításokat egymás után akarjuk végrehajtani, akkor ezeket a memóriából kell sorba egymás után kiolvasni, és az ALU segítségével feldolgozni. Ennek végrehajtását a vezérlőegység (CU) végzi.

Először be kell tölteni a PC-be az elsőnek végrehajtandó utasítás címét. Ez a cím kijutva a memóriát címző vonalakra, a címdekódoló segítségével kiválasztja az utasítást tartalmazó regisztert és a tartalmát a memória adatvezetékein keresztül az IR utasítás regiszterbe írja. Ezután már csak meg kell hívni. Ha az utasításban egy adat van operandusként, akkor a művelet azonnal végrehajtható az ALU segítségével. Ha az út. adatrésze a tényleges adatrészt tartalmazza, akkor megint a memóriához kell fordulni.

18. Ismertesse a CLEAR/SET utasítás működését.

19. Milyen alapvető digitális áramkörökből épül fel a számítógép?

1)NAND 2)kapuáramkörök, címdekóder, dekóder, multiplexerek, demultiplexerek, elemi tárolók 3)flip-flop-ok

20. Magyarozza el, mi a Harvard-felépítés és hasonlítsa össze a Neumann-felépítésű számítógéppel.

A PIC mikrovezérlők RISC jellegű, viszonylag kevés és egyszerű utasítást végrehajtó mikroprocesszor magra épülnek. (ábra 52 oldal)

A memória szervezése Harvard architektúrájú: ez a felépítés két külön tárolót: egy adat- és egy programtárolót használ., megszűnik a sorban állás a memóriáért. Ez eltérő bitszélességű adat- és programbusz használatát engedi meg.

Eddig ez volt: egy utasítás = műveleti kód + operandus. Adott hossz esetén az utasítás műveleti kód részét és operandus részét is tárolnunk kell az utasításban. A közös program- és adatmemóriát használó processzoroknál a műveleti kód + operandus mindig egész bájt hosszúságú.

A PIC mikrovezérlők két megoldással is gyorsítják a működését:

A megnövelt utasításhosszban elérhet mind a műveleti kód, mind az operandus (ha konstans), mind a címe, ha egy regiszter tartalmáról van szó. Mivel ezt az operandusként használt regiszterek csoportját a CPU-ban helyezik el, ezért az utasítás dekódolása után már nem kell külső memóriához fordulni a művelet végrehajtásához.

21. A blokkvázlat alapján mutassa be a PIC mikrovezérlők működését! (PIC16C54- es)

A 12 bites utasításvázlatból álló programot az 512*12 bites szervezésű EPROM memória tartalmazza. Ezt a memóriát a PC címzi meg. A PC által megcímezett memóriahelyről kiolvassuk az aktuális utasításvázlatot az utasításregiszterbe, majd onnan ez a tartalom az utasításdekódolóba kerül., és ott megtörténik az utasítás dekódolása, majd a dekódolás alapján a végrehajtása.

Aritmetikai-logikai utasításoknál az egyik operandus a w regiszterben van. A művelethez szükség van egy másik operandusra, amelynek a címét az utasítás operandus mezője tartalmazza. Az utasítás dekódolásakor az Általános célú regisztertömböt fogja megcímezni ez a mezőtartalom, onnan az adatbuszon keresztül a megcímezett fájlregiszter tartalma az ALU-ba kerül, és megtörténik a művelet végrehajtása. Ilyenkor az eredmény vagy a w regiszterben marad, vagy visszaíródik a megcímezett regiszterbe.

22. Csoportosítsa a PIC mikrovezérlők lábkivezetéseit!

RA0-RA3 RB0-RB7 jelű portkivezetések; RTCC 8 bites számláló; MCLR alaphelyzetbe állító; OSC1 oszc. bemenet OSC2/CKLOUT oszc. kimenet; VDD tápfeszültség

Perifériaként az RA0-RA3, valamint az RB0-RB7 jelű portkivezetések szolgálnak. A port bitjeinek irányát (hogy melyik a bemenet és melyik a kimenet, a TRISA és TRISB jelű regiszterekben írt 0-k - kimenetek- és 1-ek - bemenetek- határozzák meg.

Önálló periféria még az RTCC jelű 8 bites számláló. Ennek léptetőjét vagy a tokot működtető órajel leosztott jele, vagy egy külső impulzus (a tok RTCC kivezetésén keresztül) szolgáltatja. (A többi pl.: watchdog, EPROM, oszcillátor)

Az ábrán látható, hogy a tok működéséhez mindössze csak 5 kivezetés, a tápfeszültség két pontja és a tok belső regisztereit alaphelyzetbe hozó MCLR (Master Clear) pont és két oszcillátor kivezetés szükséges. A tok többi kivezetése már az I/O funkciókat valósítja meg.

23. Milyen lépésekből áll egy utasítás végrehajtása, és mi a pipe-line?

Az utasítás végrehajtás lépései: 1. utasítás beolvasás, 2. dekódolás, (operandus kiolvasás), 3. végrehajtás, 4. eredmény visszairása a fájlregiszterbe. Ez elmaradhat. Az utasítás-végrehajtás gyorsítása érdekében a PIC átlapolt utasításciklust használ. Ez az ún. pipe-line. Lényege: Míg az egyik utasítást végrehajtjuk, addig lehetséges a következő utasítás memóriából történő lehívása és dekódolása. Ez a módszer egy esetben nem jó: akkor, ha az éppen végrehajtandó utasítás ugró. Ilyenkor ugyanis a következő utasítást az ugrás helyéről kell betölteni, a másik lehívott utasítást pedig el kell dobni.

24. Csoportosítsa, és röviden mutassa be a PIC mikrovezérlők utasításkészletét!

logikai: AND, OR, XOR, COM

aritmetikai+SHIFT: ADD,SUB,CLR,DEC,DECSZ,INC,INCSZ,RL,RP,SWAP

bit: BSF,BCF,BTFSC,BTFSS

adatmozgató: MOV

programvezérlő: CALL, GOTO,NOP,RETURN,RETFIE,RETLW

rendszervezérlő: SLEEP, CLRWDT

Logikai utasítások: a 8 bites adatok bitenkénti logikai kapcsolatát végzik. Ezekkel az utasításokkal lehet egy 8 bitesen tárolt adat bizonyos bitjeinek a többtől való leválasztását elvégezni. Ez a művelet a maszkolás.

Aritmetikai és SHIFT utasítás: regisztertartalmak összeadását, kivonását, eggyel való növelését , eggyel való csökkentését , carryn keresztül történő jobbra ill. balra történő forgatását végzik.

Bites utasítások: adott bit törlése, egybe állítása, a következő utasítás átlépése, ha a vizsgált bit nulla vagy egy.

Adatmozgató utasítások: fájlregiszterek és a W regiszter közötti adatszerét tesznek lehetővé.

Programvezérlő utasítások: Ezek közül a fontosak a szubrutinokat kezelő utasítások. Pl.: CALL, RETURN

Rendszervezérlő utasítások: A processzort alacsony fogyasztású állapotba küldő SLEEP és a WATCHDOG-ot törölő CLRWDT utasítás.

25. Milyen feladatot lát el a veremtár?

A többször ismétlődő programrészleteket nem szoktuk minden előfordulási helyére írni, hanem egy közös ún. szubrutinba foglaljuk. Ennek végrehajtásakor egy ugró utasítással elugrunk a szubrutin elejére, majd a végrehajtása után ismét vezérlésátadással térünk vissza a hívást követő helyen lévő utasításra. Ez csak úgy lehetséges, ha ezt a visszatérési címet valahogy tároljuk. Ez a tároló a verem. Ez egy olyan memória, amelynek mindig csak a teteje érhető el. (58 oldal)

26. Két operandust igénylő utasítások esetén hol képződik az eredmény?

A két operandust igénylő műveleteknél az egyik operandus W regiszterben, a másik operandus a fájlregiszter (fr) tömbben van, az utasításban a tömbbeli címével hivatkozunk rá. Mivel az eredmény keletkezésének a helyét a d (destination) bit határozza meg, ezért ezt az utasításban jelezni kell.

Ha az eredményt a W-ben akarjuk: ADDWF REG,W vagy ADDWF REG,0

Ha az eredményt a REG-ben akarjuk: ADDWF REG,F, ADDWF REG,1 vagy ADDWF REG

27. Magyarázza el a státuszbitek jelentését!

IPR regiszter indirekt címzésnél; IRP0-IPR1 regiszter választó bitek; TO Time Out bit (1 bekapcsolás; 0 WOT túlsordulás történt). PD Power Down bit (1 bekapcs; 0 SLO?P utasításkor). t zéró bit (1 eredmény, 0 különben). DC Digit átvitel bit (1 4.bitre átvitel, 0 4.bitre nem volt átvitel). C (1 8.bitre átvitel, 0 8.bitre nem volt átvitel). Az utasítások végrehajtása során a STATUS regiszterben lévő Z, DC, és C bitek eredményétől függően állítódnak. Z=1, ha a művelet eredménye 0, C=1, ha az eredmény túlsordul, DC=1, ha minden 4. biten van túlsordulás.

28. Miért van szükség memórialapok kialakítására, és mi az a lapváltás?

A korlátozott utasításhosszból adódó korlátozott programmemória-címzés memórialapok (memóriaszegmensek) bevezetését igényli. 12 bites utasításhosszúságú PIC-ek esetén az utasításszámláló 11 bites: ez elvileg 2048 utasításszó megcímzését teszi lehetővé. Azonban az ugró utasításoknál 9 bit áll rendelkezésre a programmemória címzésére, amely 512 memórialap elérését biztosítja. A megoldás a lapozóbitek bevezetése: ez azt jelenti, hogy egy adott, az utasításban szereplő 9 bites című helynek a memóriában való tényleges elhelyezkedését a 9 bites cím lapozóbitekkel kiegészített értéke határozza meg.

29. Mik az RP0, RP1, IRP bitek?

A 14 bites családon belül a bankon belüli 7 bites címet az operandus tartalmazza. A max. 4 bankot választó bitek a STATUS regiszter 5. és 6. bitje. Az ábrán max. 4, egyenként 128 elmű regisztertömb van definiálva, amelyeket a RP0, RP1 bitek szelektálnak. A 00 érték a nullás bankot választja ki, a 01 az egyes bankot jelenti és így tovább. A PIC-ek adatlapjai pontosan tartalmazzák azt, hogy milyen regiszterek vannak. Természetesen nem kötelező minden bankot kialakítani, az egyszerűbb 14 bites PIC vezérlőkben csak nullás és az egyes bank van megvalósítva. (ábra 61 oldal)

30. Mi a RESET feladata, és hogyan működik ez a PIC-eknél?

A tápfeszültség bekapcsolásakor a mikrovezérlő belső áramköreit, regisztereit a megfelelő működés miatt jól meghatározott alaphelyzetbe kell állítani. Ez a RESET folyamat. Ehhez a belső órajel stabil működése szükséges. Bekapcsoláskor elindul egy független belső RC oszcillátorról működő 10 bites számláló. Ennek szerepe az esetleges lassú tápfeszültség növekedésének a kompenzálása. Amikor ez túlsordul, elindul egy, az oszcillátor esetleges lassú berezgése miatti hibás működés kivédését célzó, második késleltetést biztosító számláló.

31. Órajel generálása PIC-eknél.

A PIC kontrollerek áramköreinek működését ütemező órajel előállítására több lehetőség van, amelyek közül a pontosság és az ár alapján választhatunk:

LP: Alacsony frekvenciájú kristály (tehát kis fogyasztású)

XT: Kristály vagy kerámia rezonátor

HS: nagyfrekvenciás kristály

RC: Külső ellenállás-kondenzátor oszcillátor áramkör CLKOUT kimenettel

A CLKOUT kimenetein megjelenik egy TTL szintű, a belső órajelből származtatott órajel, amely a tok áramköri környezetében felhasználható.

32. Hogyan valósítható meg indirekt címzés?

Lehetséges úgy is műveletet végezni, hogy nem az utasításban lévő cím határozza meg az operandust, hanem a 4-es címen lévő FSR tartalmát tekintjük címnek, és az ilyen módon megcímezett regiszter tartalmával végezzük a műveletet. Ezt indirekt (közvetett) címzésnek hívjuk. Technikailag úgy oldották meg, hogy ha az utasításban szereplő operandus címe 0, akkor az FSR-ben lévő tartalmat tekintjük címnek.

A bankok indirekt címzésénél egyszerű a helyzet. Mivel a cím a 8 bites FSR regiszterben van, ezért az indirekt címtartomány 256 regiszter címzését teszi lehetővé. ez azt jelenti, hogy míg a 0-s és 1-es regiszterbankot direkt címzéssel csak bankváltások segítségével kezelhetjük, addig ezen két bank regiszterei indirekt címzéssel közvetlenül elérhetők.

Ha négy regiszterbankot alakítunk ki, akkor az indirekt címzésnél csak egy bankszelektáló bittel válthatunk két, egyenként 256 elemű bank között.

33. Milyen célt szolgál a Watch Dog Timer?

A watch dog timer (=órákutya). A kontrollereknél használt WDT lényegében egy saját szabadon futó RC oszcillátor-órajellel léptetett 8 bites számláló, amelyhez egy szintén 8 bites utóosztó kapcsolódik. A programban elhelyezett, periodikusan végrehajtott CLRWDWT utasítással töröljük a számlálót és az utóosztót. Ha program elkészül, azaz valamilyen elektromos zaj miatt, rossz címre lépve hibásan működik, akkor a törlés elmaradása miatti túlsordulás nullázza és újraindítja a kontrollert.

34. Mi a polling? Mi a megszakítás (interrupt-IT)?

Ha egy számítógépes rendszerben valamilyen esemény létrejöttét kívánjuk érzékelni, ezt szokásos módon kétféleképpen tehetjük meg. Az első módszernél a külső esemény létrejöttét egy bemeneti kapu bitváltásának figyelésével érzékelhetjük.

Pl.: Bármelyik bill. megnyomásakor a bill. kimenetén lévő "adat érvényes" jel szintet vált. Ha ezt egy bemeneti portra kötjük, akkor az állapotának a programból való figyelése lehetővé teszi a billentyű megnyomásának érzékelését, majd a kód beolvasását. Ezt a módszert általánosan "polling"-nak, azaz programozott átvitelnek hívják. Alkalmazása azonban lelassítja a rendszer tényleges működési sebességét, hiszen a mikroprocesszor idejének nagy részét azzal tölti, hogy ciklikusan megvizsgálja a kijelölt bemeneti bit állapotát.

Sokkal szerencsésebb, ha az esemény maga jelzi a processzor számára állapotának megváltozását. Ez a megoldás a megszakítás v. interrupt. A megszakítás az eredetileg futó program utasításainak végrehajtását leállítja, és a processzor egy ún. megszakítási alprogramot hajt végre, ami az esemény kezelését végzi, majd ennek befejeztével a processzor visszatér a megszakított program végrehajtására. Lényegében a programunk "elágazik", majd visszatér folytatni az eredeti programot.

A megszakítás olyan speciális szubrutinhívás, amelynél a hívás bekövetkezésének időpontját nem tudjuk. Ha a processzor több megszakítási vonallal is rendelkezik, akkor a megszakítások sorrendje függ a prioritási sorrendtől. A program futása nem minden esetben szakítható meg károsodás nélkül.

A PIC-ek esetén a megszakítás egyszintű.

35. Milyen időbeli viszonyokra kell ügyelnünk a megszakítások használatánál?

Ha a processzor több megszakítási vonallal is rendelkezik, akkor a megszakítások sorrendje függ a prioritási sorrendtől. A legtöbb rendszer biztosítja, hogy a megszakítások a programból tilthatók ill. engedélyezhetőek legyenek. Több megszakítás esetén a megszakításokat egyenként kell engedélyezni ill. tiltani.

A PIC-ek esetén a megszakítás egyszintű, azaz egyszerre csak egy megszakítás kiszolgálása történik, az egy időben bekövetkező megszakítás esetén a prioritást az ISR-ben való lekérdezési sorrend határozza meg. Több belső és külső megszakításforrás lehetséges, de a megszakításoknak csak egy vektorcíme van.

A megszakítás létrejöttékor - mivel egy megszakítási cím van - még nem tudjuk, hogy melyik forrás okozta a megszakítást. Ezért a megszakítási alprogram elején meg kell vizsgálni az egyes megszakításokhoz tartozó IRQ bitek állapotát. Amelyik H állapotú, az a forrás okozta a megszakítást. Több forrás esetén a bitek lekérdezési sorrendje a prioritást is megadja, hiszen több IT egyidejű bekövetkezése esetén a lekérdezés sorrendje alapján az

első IT-t okozó forrást szolgáljuk ki. a megszakítás kiszolgálása végén töröljük a D tárolót, amely az IT kérését tárolta a kiszolgálás alatt.

Több megszakítás a processzort a szundi (SLEEP) módból ébreszti, a megszakítás felismerése 3 utasításciklus, ill. 4 ciklus a külső megszakításoknál. A megszakítás kiszolgálásakor fontos megőrizni a STATUS, a W regiszter tartalmát.

36. Hogyan lehet programozni a PIC-eket?

A gyorsabb párhuzamos, és a soros módszerrel. Soros programozásnál a memóriába kerülő szavakat bitenként írjuk két vezeték felhasználásával. A programozáshoz a Vdd tápfeszültségnél nagyobb feszültség szükséges. A programozás idődiagrammját a gyártó a mikrokontrollerek alapján közli.

A programozás történhet a tok több lábát felhasználva párhuzamosan, ill. SHIFT regiszteres elven bitenként, sorosan.

Programozói állapotba kerülünk, ha a MCLR lábat VPP feszültségre (12V) emeljük. Ezután a programozó berendezés a CLOCK lábon órajellekkel érvényesítve elküld egy parancsot, majd a tok az órajellel ütemezve az RB7 lábán keresztül válaszolva végrehajtja a parancsot.

(PICSTART Plus egy jó programozó)

(74 oldal)

37. Ismertesse röviden a PIC perifériákat.

38. PIC I/O portok működése.

A leggyakrabban használt perifériaelemek az I/O portok. Lényegében a kollektor lábkivezetéseinek állapotát tudjuk a programba beolvasni, ill. a programból a 0 és 1 állapotnak megfelelően, a lábak feszültségét befolyásolni. A portokhoz rendelt I/O regisztereket pontosan úgy kezeljük, mint az egyéb fájlregisztereket, és helyileg is azok között helyezkednek el.

A portok olvasásakor mindig a láb állapotának beolvasása történik meg, függetlenül attól, hogy bemenetnek vagy kimenetnek van konfigurálva. A portok írása ténylegesen egy D tárolóba történő írás. RESET hatására minden port bemenet lesz, és a TRIS regiszter tartalmával tudjuk a port irányát beállítani. Ez a porthoz tartozó I/O vezérlő latch írását jelenti.

39. Számlálók és időzítők a PIC-ben. TMR0, TMR1, TMR2

A számlálóáramköröket perifériaként használva a processzort tudjuk tehermentesíteni. A számláló bemenetére egy külső esemény esetén megjelenő jelváltást kapcsolva, a számláló önállóan képes a külső események számlálására, és a processzornak csak le kell kérdezni a számláló tartalmát.

Ha ellenben a kontrollert működtető leosztott órajel juttatunk erre a számláló bemenetre, akkor a számláló tartalma és az órajel periódusidejének a szorzata az eltelt idővel arányos. A számláló által átfogható számlálási tartomány növelése miatt a számlálót alkotó flip-flopokból álló számláló lánc hosszát növelik meg.

TMR0: Minden PIC tartalmazza. (ábra 118 oldal)

TMR1: A számláló a nagyobb felbontás miatt 16 bites, de még van egy további, felbontást növelő elosztó.

Bemenete vagy a belső oszcillátor negyed frekvenciájú jele, vagy egy külső jel (TICKI) vagy: egy kisméretű (LP) kvarc oszcillátor.

TMR2: A 8 bites TMR2 időzítőt a soros adatátviteli egység ütemadójaként használjuk. Mind előosztója, mind utóosztója van. A TMR2 regiszter tárolja a számláló állapotát, a számláló beállítása a T2CON regiszter biteinek beállításával lehetséges.

A PR" regiszterbe írt értékekkel hasonlítja össze a TMR2 számlálót, és egyezés esetén okozhat megszakítást. A megszakítás engedélyezése a TMR2IE bit 1-be állításával lehetséges, a túlcsoportulást a TMR2IF bit 1 állapota jelzi.

40. PWM modulátor működési elve és felhasználási lehetőségei.

A PWM modulátor segítségével impulzusszélesség-modulációt tudunk létrehozni. Ez azt jelenti, hogy állandó T periódus idő mellett a bekapcsolás W idejét változtatva, a kimenő jel közértékét tudjuk változtatni. Ha a kapcsolgatott feszültség szint U_b , akkor a PWM modulátor kimenetén a feszültség közértéke: $U_k = U_b \cdot W / T$ Mivel ez a négyszögjel az U_k közértékű DC jel és szinuszhullámok összegére bontható, ezért megfelelő aluláteresztő szűrővel kiszűrhetjük a DC komponenst a váltakozó áramú szinuszos összetevőktől. Az aluláteresztő szűrő egy RC tag. A PWM jelet két adat jellemzi: a frekvencia és a kitöltési tényező.

Lényegében a PWM modulátor egy 1 bitet felhasználó digitál-analóg átalakító. (+132 oldal)