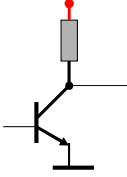


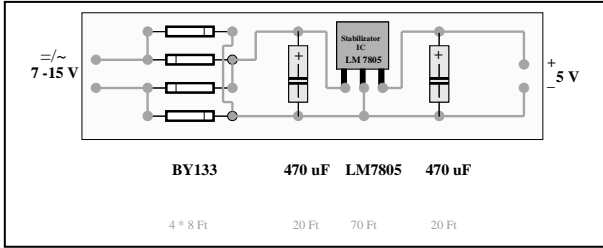
1. Milyen a nyitott kollektoros kimenet?

Amikor a **nyitott kollektoros kapuk kimeneteit** összekötjük, és a közös vezetékét a V_{CC} tápfeszültséghez csatlakoztatjuk egy ellenállással, akkor ez a közös vezeték csak abban az esetben lesz H szintű, ha valamennyi 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.

Azért jó, mert össze lehet őket kapcsolni, pl. busz, vagy „kérelem-vonal”. A normál TTL (totem pole) kimeneteket nem szabad összekötni.



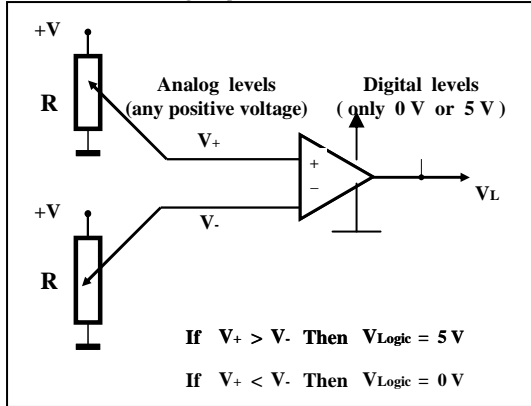
2. Hogyan készíthetünk stabilizált tápfeszültség forrást?



Kell: 1db feszültségstabilizátor IC: 78XX -> X=hány volt. 79XX -> negatív hány volt. (Más lábkiosztás)
2 kondenzátor, a bemenetre és a kimentre, a nagyobb frekvenciák kiszűrése és a gerjedés megakadályozása miatt.
4 dióda egyenirányításra, vagy 1db egyenirányító híd.

Ez az áramkör pontatlan váltakozó feszültségből (olesó trafó ☹) csinál NAGYON pontos (0.00 V) stabilizált feszültséget.

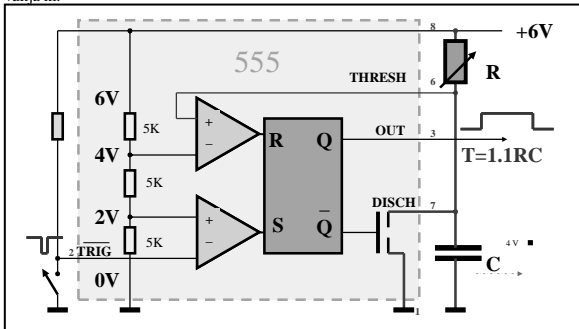
3. Ismertesse az analóg komparátor működését!



Tartalmaz 1 műveleti erősítőt, ami a „+” és „-” bemenetei között lévő feszültségkülönbséget erősíti, tehát $V_L = A * |V_+ - V_-|$, ahol A nagyon nagy lehet (több tíz milliós). Ezért ha a „+” bemenet egy picit a „-” fölé megy, akkor a kimenet azonnal +5V feszültségű lesz. Ha alá megy, akkor pedig 0.

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

*Felépítése digitális és analóg áramkörök keveréke, impulzusok előállításra. 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 állapota 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.

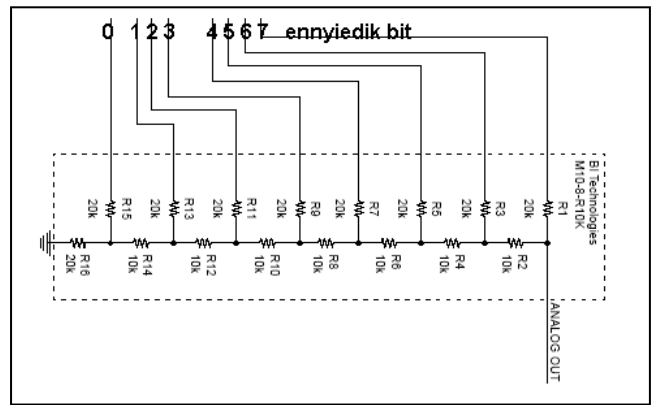
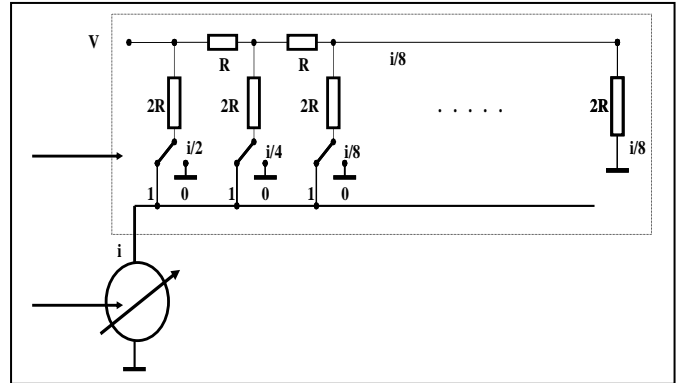
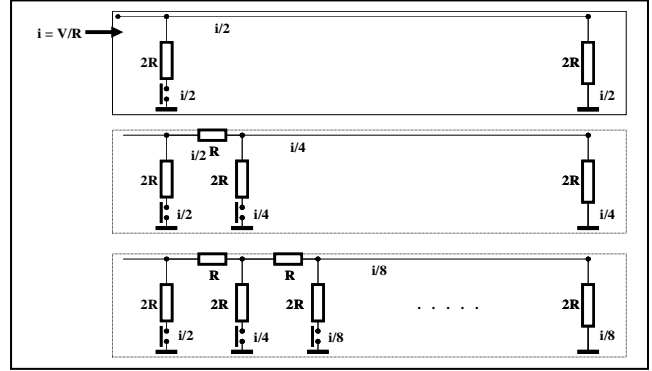
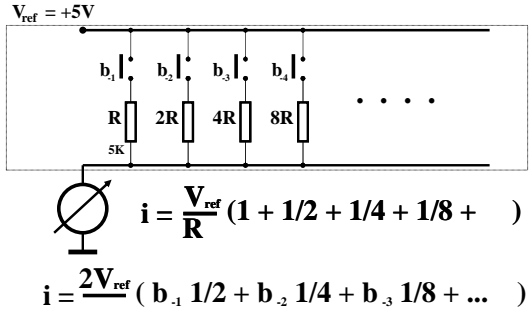


Ha megnyomják a kapcsolót, az a TRIG lábón át SET-eli a RSFF-et, amitől a DISCH láb kisíti a kondenzátort. Ha elengedik a kapcsolót, a DISCH láb nagy ellenállásba vált át (lezár a tranzisztor). Ezután a kondenzátor R ellenálláson keresztül feltöltődik. Mikor $T=1.1RC$ idő után eléri a 4 voltot, a TRESH láb reseteli az RSFF-et. A kimeneten az látszik, hogy a gomb megnyomásakor +6V jön ki, majd a gomb elengedése után 1.1RC ideig úgy is marad, majd visszavált 0V-ra.

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.

ELLENÁLLÁS LÉTRA



Ellenálláslétra: Egy digitális bitsorozatból állít elő analóg áramot.

6. Műveleti erősítők alkalmazási lehetőségei TTL környezetben.

*TTL: tranzisztor-tranzisztor-logika
1)D/A 2)A/D 3)fázisfordító s+ 4)multivibrátor 5)feszültségmegosztás

A/D átalakításhoz használható. Mivel a műveleti erősítő komparátorként működtethető, feszültségzinteket lehet digitális 1/0 (+5V, 0V) jelekkel alakítani. Műveleti erősítő komparátort használnak például a szűkebbv approxmációs A/D átalakításnál.

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

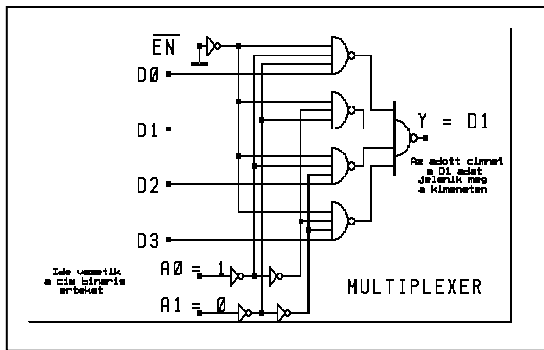
Multiplexerek: A multiplexer olyan áramkör, amely a címzésnek megfelelően a bemeneteire kapcsolódó vonalak valamelyikét a kimenetre kapcsolja, azaz a kimeneten megjelenik az adott bemenet logikai állapota. A D0...D3 adatbemenetek valamelyike a Q kimenetre kapcsolódik. Ezt az A0 és A1 címbeemenetek logikai állapota határozza meg. Pl.: ha A0=0, az A1=1 szintű, akkor a D2 bemenet kapcsolódik a kimenetre.

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

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

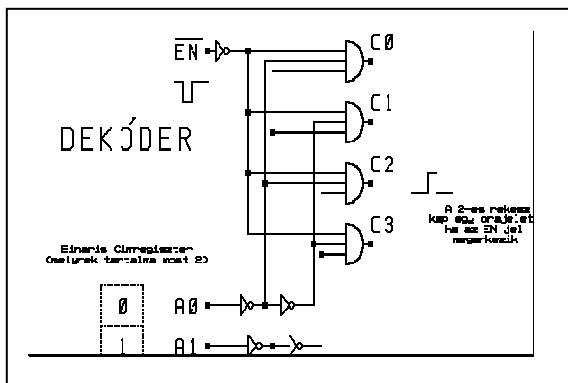
Dekódolók: A dekódoló áramkör az információnak az egyik ábrázolási formáról egy másik ábrázolási formára való átalakítására szolgál. Sok esetben van szükség olyan áramkörre, amelynek a bemenetére bináris értéket kapcsolva, annak csak egyik - adott bináris értékhez tartozó- kimentén jelenik meg jel. Egy

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 szegmens LED diódáit ki kell gyújtaniuk.



A multiplexer a „D” bemenetei közül egyet kiválaszt és az értékét kiírja a kimenetére. A választás „A” bemenetek szerint történik. Mégpedig úgy, hogy ha az „A”-n 1, 2, stb... szerepel, akkor az 1., 2., stb „D” bemenet értéke lesz kiírva az Y kimenetre. Az ábrán látható kapuk NEM-ÉS kapuk.

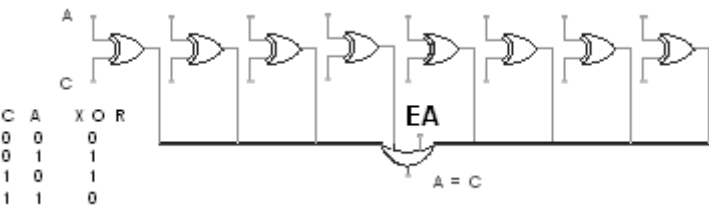
A demultiplexer ennek ellenkezőjét csinálja, tehát van 1 „D” bemenete, egy „A” bemenete, és „Y1” – „Yn” kimenete. Az „A” bemenet értéke szerint választ egy kimenetet és oda teszi a „D” bemenetét.



A dekódoló úgy működik, mint egy olyan demultiplexer, aminek a bemenetén folyamatosan „1” érték van jelen. Tehát bemenete egy bináris szám, kimenete pedig az, hogy a számnak megfelelő kimenet 1-be vált, tehát megmondhatjuk neki, hogy hányas lámpa égjen

8. Hogyan működik a digitális komparátor?

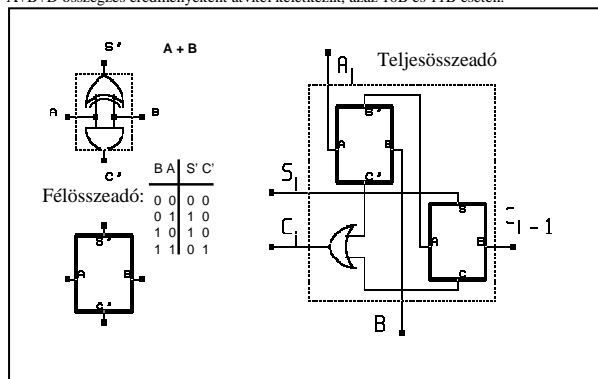
A digitális komparátorok olyan áramkörök, amelyek két bináris értéket hasonlítanak össze egymással. A 3 legfontosabb összehasonlítási feltétel: egyenlőség, a kisebb és nagyobb viszony. Két bináris érték egyenlőségének feltétele, hogy minden bitjük megegyezzen. A komparátornak csak akkor van a kimenetén 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.



A digitális komparátor működése az XOR kapun alapul. Ha az XOR kapu bemenetein különböző értékek vannak, akkor a kimenetén 1 szerepel, különben 0. Ha ezeket összevagyoljuk, akkor megkapjuk azt, hogy a két érték mikor különbözik egymástól (Ekkor 1 a kimenet.)

9. Hogyan működik a teljes összeadó?

Jelölje a két összeadandót A és B, az előző helyi értéktől származó átvitel 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-est, amikor az A+B+D összegzés eredményeként átvitel keletkezik, azaz 10B és 11B esetén.



Félfösszeadó: A és B bemenetéből generál egy S összeget és egy C átvitelt.

Teljes összeadó: Ai és Bi összeadandókból és az előző helyiértékből származó Ci-1 átviteltől generál egy Si összeget és egy Ci átvitelt.

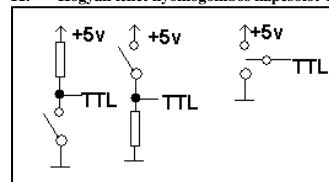
10. Milyen feladatokra használhatók a tároló regiszterek és osztók?

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 átrődik 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 bitű párhuzamosan tároló regisztereket. Ilyenkor az egyes tárolók órajel bemeneteit közösítik.

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

RSFF, DFF, JKFF

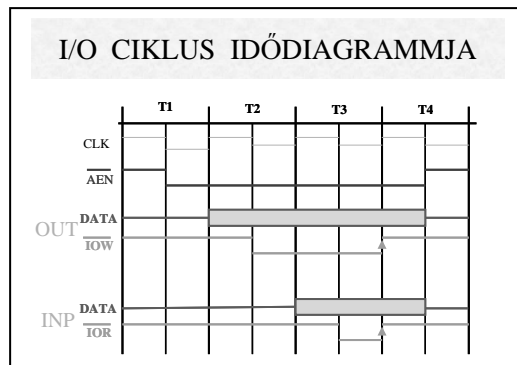
11. Hogyan lehet nyomógombos kapcsolót TTL bemenethez illeszteni?



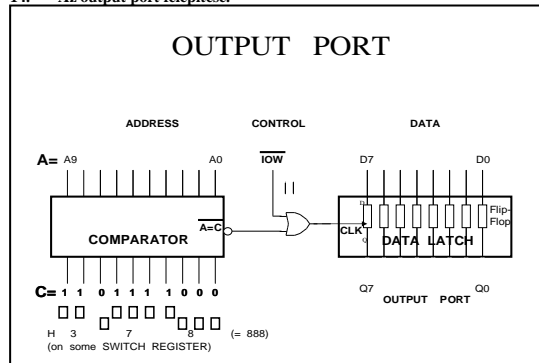
12. Hogyan lehet láthatóvá tenni egy kimenet logikai állapotát?

Lámpa,LED, vagy több kimenet esetén 7 szegmenses LED kijelzővel. Ebben az esetben szükség van egy bináris->BCD->7szegmens átalakítóra.

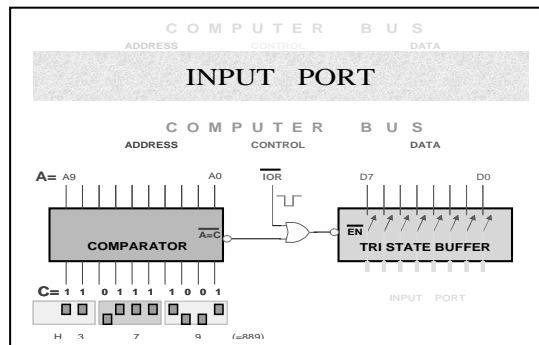
13. Ismertesse az I/O ciklus idődiagramját!



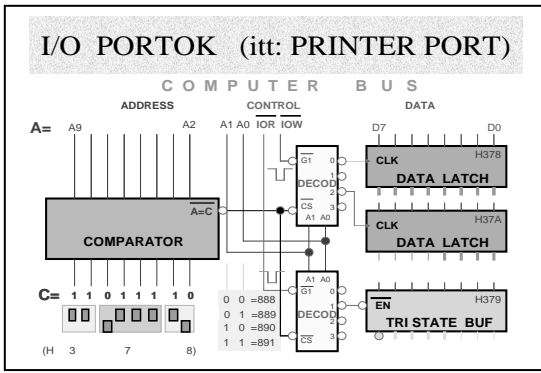
14. Az output port felépítése.



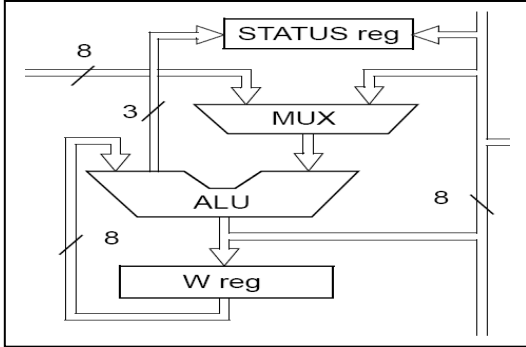
15. Az input port felépítése.



16. A PC printer portjának felépítése.

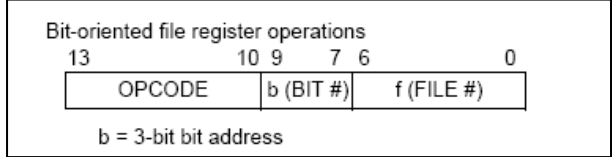
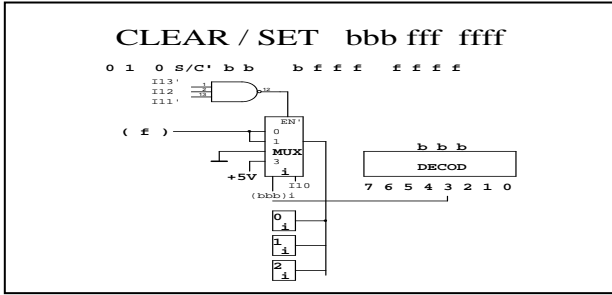


17. Ismertesse az ALU felépítését és működését!



Az ALU (Aritmetikai, logika egység) matematikai és logikai műveletek végrehajtására alkalmas. Bemenetei az operandusok és egy művelet kiválasztó kód. Kimenete (és/vagy egyik operandusa) gyakran egy fix, ún. akkumulátor regiszter

18. Ismertesse a CLEAR/SET utasítás működését.



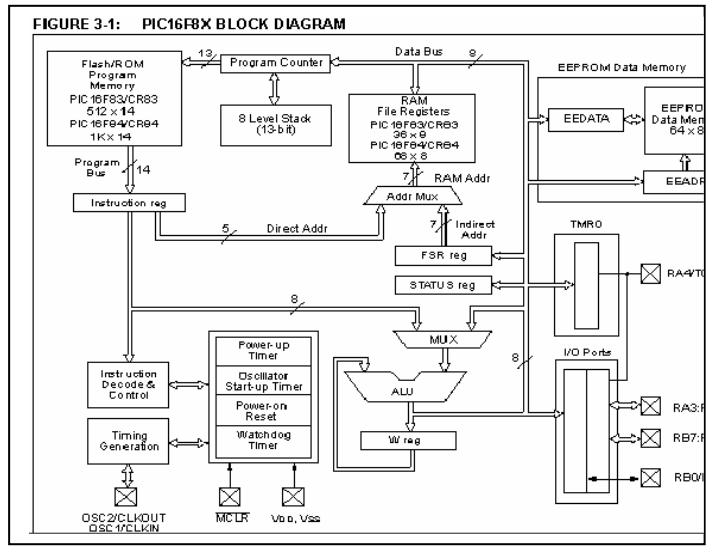
19. Milyen alapvető digitális áramkörökből épül fel a számítógép?
 Kapuáramkörökből, amik további magasabb rendű működést megvalósító egységeket alkotnak: flip-flopok, regiszterek; információátvitel, memória, cache; multiplexerek, komparátorok, dekóderek; címzés, utasításkódolás; fél-, egész összeadók, kivonók), szorzók; aritmetikai áramkörök, ALU.

20. Magyarázza 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 bájttal hosszúságú. A PIC mikrovezérlők két megoldással is gyorsítják a működését: A megnövelt utasításhosszban elérhető 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.

Harvard: utasítás és adatmemória külön helyezkedik el, míg a Neumann-nál ez egység.

21. A blokkvázlat alapján mutassa be a PIC mikrovezérlők működését! (PIC16C54-es)
 A 12 bites utasításhosszú álló programot az 512*12 bites szervezésű EPROM memóriába tartalmazza. Ezt a memóriát a PC címzi meg. A PC által megcímezett memóriahelyről kiolvassuk az aktuális utasítást az utasításregiszterbe, majd onnan ez a tartalom az utasításkódolóba kerül, és ott megtörténik az utasítás dekódolása, majd a dekódolás alapján a végrehajtás. 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 regiszter tartalma az ALU-ba kerül, és megtörténik a művelet végrehajtása. Ilyenkor az eredmény vagy a w regiszterben marad, vagy visszairódnak a megcímezett regiszterbe.



A bal felső sarokban látható a program memória. Ezt címzi a „Program Counter” nevű regiszter (kicsit jobbra). A kiválasztott utasítás beletyol az „Instruction Register” nevű dologba (lejjebb). Ez rákérül az utasításkódolóra. A cím illetve címrésze (ha van) a cím, illetve adatmultiplexeren keresztül megcímezheti a regiszter file-okat, illetve beletyolhat az ALU-ba. Az ALU ezzel az operandussal illetve a W regiszter tartalmával végez műveletet. 8 regiszterek címzése. Indirekt: FSR regiszterrel címzünk. Direkt címzés: Az utasításban lévő 5 bit + a STATUS regiszter RP0 és RP1 bitjei.

22. Csoportosítsa a PIC mikrovezérlők létezőit!
 R10-RA3 RB0-RB7 jelű portkivezetések; RTCC 8 bites számláló; MCLR alaphelyzetbe állító; OSC1 oszc. bemenet OSC2/CLKOUT oszc. kimenet; VDD tápfeszültség
 Perifériáknak az RA0-RA3, valamint az RB0-RB7 jelű portkivezetések szolgálnak. A port biteinek irányát (hogy melyik a bemenet és melyik a kimenet), a TRISA és TRISB jelű regiszterekbe írt 0-k -kimenet- és 1-ek -bemenet- határozzák meg. Önálló periféria még az RTCC jelű 8 bites számláló. Ennek létezőjét 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 működtetéséhez mindössze csak 5 kivezetés, a tápfeszültség két pontja és a tok belső regiszterei alaphelyzetbe hozó MCLR (Master Clear) pont és két oszcillátor kivezetés szükséges. A tok többi kivezetése már az I/O funkciókat valósítja meg.

A-E portokhoz tartozó RA0-RA7, ..., RE0-RE7. Általában RAX analog bemenetként programozható lábak. Megszakítást okozó láb: RB0/INT, USART: soros kommunikáció, USB lábak: D+, D-, OSC1, OSC2: órajel lábak pl. kvarcnak, RTCC

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 visszairódnak a fájlregiszterbe. Ez elmaradhat. Az utasítás-végrehajtás gyorsítása érdekében a PIC átlapol 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 ugrott. 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, BTFSZ, BTFSB
 adatmozgató: MOV
 programvezérlő: CALL, GOTO, NOP, RETURN, RETFIE, RETLW
 rendszervezélő: SLEEP, CLRWDWT
 Logikai utasítások: a 8 bites adatok bitenkénti logikai kapcsolatát végzik. Ezekkel az utasításokkal lehet egy 8 bites tárolt adat bizonyos biteinek a többitől való leválasztását elvégezni. Ez a művelet a maszkolás. Aritmetikai és SHIFT utasítások: regisztertartalom 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 adatcsere tesznek lehetővé. Programvezérlő utasítások: Ezek közül a fontosak a szubrutinokat kezelő utasítások. Pl.: CALL, RETURN Rendszervezélő utasítások: A processzort alacsony fogyasztású állapotba küldő SLEEP és a WATCHDOG-ot törölő CLRWDWT utasítás.

25. Milyen feladatot lát el a veremtar?
 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 helyre 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 többi címével hivatkozunk rá. Mivel az eredmény képződésének 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 státuszbit jelentését!
 IPR regiszter indirekt címzésnél; IRP0-IPR1 regiszter választó bitek; TO Time Out bit (1 bekapcsolás; 0 WDT túlsoroldás történt). PD Power Down bit (1 bekapcs; 0 SLEEP utasításakor). 1 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 túlsoroldul, DC=1, ha minden 4. biten van túlsoroldulás.

28. Miért van szükség memórialapok kialakítására, és mi az a lapváltás?
 A korlátozott utasításhosszából adódó korlátozott programmemória-címzés memórialapok (memóriaszegmensek) bevezetését igényli. 12 bites utasításhosszú PIC-ek esetén az utasításkódoló 11 bites: ez elvileg 2048 utasításhosszú megcímezést teszi lehetővé. Azonban az ugró utasításoknál 9 bit áll rendelkezésre a programmemória címzésére, amely 512 memóriahely elérését biztosítja. A megoldás a lapozóbit bevezetése: ez azt jelenti, hogy egy adott, az utasításban szereplő 9 bites című helynek a memóriában való tényleges elhelyezkedését a 9 bites cím lapozóbittel kiegészített értéke határozza meg.

Az utasításban fixen csak 5 bitnyi cím fér el. Címezni viszont csak 7 bittel lehet. A fennmaradó 2 bitet a STATUS regiszterek RP0 és RP1-es bitjei adják. Ezen bitek által 4 memóriaterület különül el. Ezek a „bank”-ok.

29. Mik az RP0, RP1, IRP bitek?

A 14 bites családoknál 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 elemű 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.

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

A tápfeszültség bekapcsolásakor a mikrovezérlő belső áramkörét, 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 oszcillátorral működő 10 bites számláló. Ennek szerepe az esetleges lassú tápfeszültség növekedésének a kompenzálása. Amikor ez túlszorodul, elindul egy, az oszcillátor esetleges lassú berezgése miatt 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 kimenetén megjelenik egy TTL szintű, a belső órajelből származtatott órajel, amely a tok áramkörti környezetében felhasználható.

32. Hogyan valósítható meg indirekt címzés?

Lehetséges úgy is műveletet végezni, hogy nem az utasításban lévő cím határozza meg az operandus, 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. Technikaileg 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 regisztereire 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 (=órákutyá). 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 elköszül, azaz valamilyen elektromos zaj miatt, rossz címre lépve hibásan működik, akkor a törlés elmaradása miatti túlszorodulás 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 módszerrel a külső esemény létrejöttét egy bemeneti kapu bitváltozásának 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 vagy 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 közi.

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

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

(PICSTART Plus egy jó programozó)

37. Ismertesse röviden a PIC perifériákat.

Párhuzamos portok: RAX-REX: A megfelelő regiszterekbe írt értékek megjelennek a kimeneteken (0 = 0V, 1 = Tápfeszültség (Vcc)).

USART: Universal Synchronous-Asynchronous Receiver-Transmitter: szinkron-aszinkron soros kommunikációt valósít meg 2 lábon. (Aszinkron: RX/TX (receive/transmit, vesz/ad), Szinkron TX/CK, (Transmit/clock, átvitel/óra))

TIMER0-TIMER2: Időzítést szolgáló perifériák. Tulajdonképpen különféle bitszélességű számlálók (8-16). Órajelvet vehetik külön forrásból, vagy a proc órajeléből. Rendelkez(het)nek előosztóval.

CCPWM (Capture-Compare/PWM): Külső események figyelése illetve számlálása (CC), Impulzusszélesség-moduláció: négyzögjelet gyárt, változtatható a frekvencia és a kitöltési tényező (PWM) A/D átalakító: Külső feszültségértéket számértékké alakít.

38. PIC I/O portok működése.

A leggyakrabban használt perifériaelemek az I/O portok. Lényegében a kontroller lábkievezéseinek 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 portozó tartozó I/O vezérlő latch írását jelenti.

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

A számlálóáramköröket perifériaként használva a processzort tudjuk tehermentesíteni. A számláló bemenetére egy külső esemény esetén megjelenő jélváltást 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 számlálóbemenetre, akkor a számláló tartalma és az órajel periódusidősszára a szorzata az eltelt idővel arányos. A számláló által átfogható számlálási tartomány növelése miatt a számlálót alkotó flip-flopokból álló számlálólánc hosszát növelik meg.

TMR0: Minden PIC tartalmazza.

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 (T1CKI) vagy: egy kisfrekvenciás (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 PR² regiszterbe írt értékekkel hasonlítja össze a TMR2 számlálót, és egyezés esetén okozhat megszakítást. A megszakítás engedélyezése a TMR2IE bit 1-be állításával lehetséges, a túlszorodulást 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 kapcsolgató feszültség szint Ub, akkor a PWM modulátor kimenetén a feszültség középvértéke: $U_k = U_b \cdot W/T$ Mivel ez a négyzögjel az Uk középvértékű DC jel és szinuszhullámok összegére bontható, ezért megfelelő aluláteresztő szűrővel kiszűrhetjük a DC komponenset 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.