

CPU, glavni pomnilnik, vhodno izhodne enote, mikro procesor, mikro pomnilnik

CPU usklajuje in skrbi za pravilno delovanje računalnika. Iz pomnilnika jemlje ukaze in jih izvršuje. Sestavljajo jo tri enote:-registri(služijo kot začasni pomnilnik podatkov),-aritmetično logična enota ALE (služi za izvajanje aritmetičnih in logičnih operacij),-kontrolna enota (nadzoruje in usklajuje pretok podatkov in ukazov znotraj CPU in med CPU in pomnilnikom). Glavni pomnilnik:v njem so shranjeni ukazi , ki jih CPU uporablja (RAM, EPROM, FLASHRAM...). Videti je kot enodimenzionalno zaporedje pomnilniških besed od katerih ima vsaka enolično določen naslov. Pomnilniška beseda je sestavljena iz določenega števila pomnilniških celic. Vsaka celica hrani en bit informacije. Št. Celic v besedi imenujemo dolžina pomnilniške besede. Iz pomnilnika ne moremo pisati/brati manj kot eno besedo. Standardne dolžine so(4,8,16,32,64,128,...). Vhodno izhodne enote:skrbi za pretvarjanje podatkov iz človeku razumljive oblike v računalniku razumljivo obliko. Je tudi najbolj dostopen del računalnika. Vsaka V/I enota je priključena na Rač. preko krmilne naprave. Te so lahko zelo enostavne ali kompleksne.Mikro procesor: mikroračunalnik na enem integriranem vezju. Mikro krmilnik: mikroračunalnik na enem integriranem vezju. Vodilo prisotne enote delimo v dve skupini: gospodar/suženj (master/slave). Vsa komunikacija med enotami poteka vedno prek CPU (gospodar). Vse ostale enote se ji morajo podrediti._

Način delovanja mikrokrmilnikov, značilnosti

MK lahko deluje v 4 načinih:-Zaprtem(samostojni)način (single chip),-razširjenem(expandet),-samo nalagalnem(samo zagonskem-boot start),-testnem. Samostojni način naslovno in podatkovno vodilo od zunaj nista dostopna. Primeren za veliko serijske aplikacije. Razširjen način omogoči dodajanje zunanjih enot. Na vratih B in C sta dostopna naslovno in podatkovno vodilo. Vrata B in C lahko nadomestimo s posebnim zunanjim vezjem 2xHC24. Samo zagonski način program je omejen na 512 bytov. MK deluje enako kot v samostojnem načinu. Po restartu se požene posebni nalagalni program, ki omogoča prenos programa preko serijskega vmesnika in njegov samodejni zagon po končanem nalaganju.

Programsko dostopni registri -vrste in značilnosti

Vrata A (dvosmerna paralelna vrata ali kot vhodi števec). -Vrata B (dvosmerna paralelna vrata ali kot del naslovnega vodila). -Vrata C(dvosmerna paralelna vrata ali kot multiplasirani naslovi in podatki. -Vrata E (kot digitalni vhodi ali kot vhodi za A/D pretvornik). -IRQ (prekinitveni vhod prožen nivojsko ali ob prehodu). -XIRQ (prekinitveni vhod nivojsko prožen). -Vreft in Vreft (vhoda za referenčno napetost A/D pretvornika). -STRA/AS STRB/R/W (v razširjanem načinu sta uporabljena kot izhoda. Kadar so na vratih C naslovi je AS=1. R/W pa predstavlja kontrolni signal za pisanje in branje)

Vrste in tipični signali mikrokrmilnika HC11

Vdd in Vss napajanje 3V-7V(1MHz) - MODA/LIR in MODB/V (stby-način delovanja) - EXTAL XTAL (Uporabimo lahko interni oscilator ali pa preko EXTAL vhoda pripeljemo urin signal iz zunanjega oscilatorja) - E (in signal za sinhronizacijo 1/4 frekvence oscilatorja) - RESET (signal mora biti prisoten tako dolgo, da napajalna napetost doseže maksimalno vrednost ter da oscilator zanesljivo zaniha)

Načini naslavljanja

MK 68HC11 pozna 6 različnih načinov naslavljanja pomnilnika: - takojšnje (immediate) - direktno (direct) - razširjeno (extended) - indeksno (indexed) z obema 16 bit-nima registroma in 8 bitnim odmikom - vsebovano (inbcrent) - relativno (relative).

Vrste ukazov MK HC11

HC11 pozna 329 ukazov. Ker je operacijska koda 8 bitna. To omogoča uporabo samo 56 ukazov. Pri določenih ukazih je uporabljen še dodaten bayt ki je lahko 518,521 SCD. Ukaze lahko razdelimo v več skupin: - Polnjenje in shranjevanje registrov. - pomikanje registrov v registre. - povečevanje in zmanjševanje vsebine registrov. - brisanje postavljanje registrov. - aritmetični ukazi. - premiki in rotacije. - testiranje podatkov. - pogojni skoki. - operacije s statusnim registrom (CCR). - prekinitve in ostali.

Sklad - pomen, uporaba

Sklad je poseben del ram pomnilnika, ki ga uporabljamo za začasno odlagališče. Mehanizem dela s sklado (shranjevanje in branje podatkov s sklada) je strukturno podprt. organizacija sklada je tipa LIFO. Dostop do sklada poteka preko posebnega 16-bitnega registra imenovanega kazalec sklada. Ta vedno kaže na naslednjo prosto lokacijo v skladu. Začetno vrednost moramo vedno določiti na začetku izvajanja programa. Povečevanje in zmanjševanje kazalca sklada poteka samodejno pri vsakem branju in pisanju. Nastajata 2 pristopa: pri vsakem vpisu se vrednost kazalca sklada zmanjša za 1 ali 2. Na začetku moramo kazalec sklada postaviti na vrh ram pomnilnika. Pri vsakem vpisu se kazalec sklada poveča za 1 ali 2. KS moramo v tem primeru postaviti na začetek RAM pomnilnika (INTEL SERVIS). Povečevanje in zmanjševanje kazalca sklada poteka samodejno brez posredovanja programerja, čeprav ta lahko direktno vpliva na vrednost kazalca sklada. Aparaturno pa prenos podatkov na sklad ter branje s sklada poteka enako kot vsi ostali dostopi do pomnilnika. Uporaba sklada: svojo pravo veljavo dobi sklad pri delu s podprogrami in uporabi prekinitev. Brez sklada je uporaba podprogramov in prekinitev nemogoča. Ob klicu podprograma JSR se na sklad najprej shrani vrednost programskega števca, ki kaže tedaj na naslednji ukaz za ukazom JSR, nato pa se začne izvajati podprogram. Ta se mora obvezno končati z ukazom RTS. Ta ukaz povzroči, da se vrednost prej shranjenega programskega števca prenese iz sklada v programski števec. Druga uporaba sklada je pri prekinitvah. V tem primeru shranijo vrednosti programske dostopnih registrov. Pred vsako aplikacijo moramo določiti globino sklada ter s tem rezervirati del pomnilnika. Pri izračunu moramo upoštevati naslednje: - vsak klic podprograma poglobi sklad za dve lokaciji, - vsak program lahko kliče tudi druge podprograme, - vsaka prekinitve poglobi sklad za 7 lokacij. Prekinitve pa so lahko tudi ugnezdene. Določeni MK (PIC) imajo v naprej določeno globino sklada, ki ne presega 8 lokacij. Pri uporabi določenega MK moramo te posebnosti upoštevati.

Podprogrami

Pri pisanju podprogramov večkrat naletimo na primer, ko moramo del programa napisati večkrat. Temu se izognemo tako, da napišemo samo enkrat kot podprogram, v glavnem programu pa ga s posebnim ukazom kličemo. Po izvedbi podprograma se izvajanje programa nadaljuje z naslednjim ukazom, ki sledi klicu podprograma. Posebno uporabnost dobijo podprogrami tudi pri delu z VHO/IZH enotami.

Npr. podprogrami za pisanje in branje podatka z V/I enote.

Primer: podprogram za vpis podatka v register paralelnega vmesnika

- glavni program

LDAA#\$55

JSR PIS PAR

BRA

- podprogram

PISPAR STAA\$8003 na lokacijo \$8003 shrani vrednost v registru A

RTS vrnitev iz podprograma

Prekinitve-pomen, uporaba, maskirane, nemaskirane, uporaba sklada, postopek ob prekinitvi

Sistem prekinitvev omogoča CPE enota da prekine trenutno izvajani program ter prične izvajati prekinitveni strežni program. Te zahteve lahko sprožijo VH/IZ naprave s tem da aktivirajo posebne signale. To se običajno zgodi ob katerem koli trenutku kar pomeni da je zahteva po prekinitvi naključen dogodek, ki ga ne moremo v naprej napovedati. Akcija ki se ob prekinitveni zahtevi izvede je aparaturno realizirana in se izvaja samodejno. V/I naprava poda zahtevo za prekinitvev kadar potrebuje poseg CPE.

Po končanem prekinitvenim strežnem programu se mora program nadaljevati na mestu kjer je bil prekinjen.

Prekinitvev ne sme vplivati na izvajanje programa kar pomeni , da se mora stanje programske dostopnih registrov pred izvajanjem prekinitvenega programa shraniti. Mikrokrmilnik v ta namen uporablja **sklad**. Vanj se ob začetku izvajanja prekinitvenega programa shranijo registri, katerih vrednosti se med izvajanjem spremenijo, na koncu pred vrnitvijo v glavni program pa se shranjene vrednosti prenesejo v registre.

Sam postopek prekinitve zajema 3 faze;

- Prepoznavanje naprave, ki je sprožila prekinitvev
- Določanje prioritete prekinitve v primeru kadar je naprav več
- Potrjevanje prekinitve, napravo moramo obvestiti da je njeni zahtevi ugodeno

Prepoznavanje naprave je običajno izvedeno programske. Vsaka VH/IZ naprava ima poseben register, v katerem označi prekinitveno zahtevo. Z branjem in testiranjem registra ugotovimo katera naprava je zahtevala prekinitvev. Določanje prioritete je lahko izvedeno programske ali aparaturno. Prvo je povezano že kar s prepoznavanjem naprave. Potrjevanje prekinitve pomeni, da VH/IZ naprava umakne svojo zahtevo za prekinitvev. Seveda da lahko komunikacijo z VH/IZ napravo izvedemo tudi brez uporabe prekinitvev. Ta način zahteva veliko težje posredovanje CPE pri komunikaciji kot uporaba prekinitvev. **Prekinitvev** so podobne podprogramom, bistvena razlika med njima je v načinu klicanja. Podprogrami so klicani iz glavnega programa na točno določenih mestih, prekinitveni programi pa se izvajajo naključno ter samodejno. Prekinitvev delimo na zunanje (prožijo jih zunanje naprave preko posebnih prekinitvenih linij) in notranje (te prožijo v mikrokrmilnik vgrajene enote).

Prioriteta posameznih prekinitvenih vhodov je lahko fiksno ali pa programske določeno. Pri tem velja pravilo, da lahko prekinitvev z večjo prioriteto prekine izvajanje prekinitvev z nižjo prioriteto. Prekinitvev delimo na maskirane in nemaskirane.

Maskirane prekinitvev lahko programske onemogočimo s postavitvijo posebnega bita v CCR registru. Smisel onemogočitve je v tem, da lahko CPE izvaja nek del programa, ki ne sme biti prekinjen. Pred začetkom izvajanja tega dela programa prekinitvev s posebnim ukazom onemogočimo, na koncu pa jih ponovno omogočimo. **Nemaskiranih** ne moremo programske onemogočiti. Število zunanjih prekinitvenih vhodov je običajno 2 (EN MASKIRANI, EN NEMASKIRANI). Čeprav je prestop do obravnave prekinitvev podoben pri različnih

mikrokrmilnikih, pa je pred uporabo konkretnega MK potrebno do dobra preveriti način dela.

Prekinitveni vektor uporaba

Prekinitveni vektor

Ker se prekinitvev izvajanja prekinitvenega programa izvede avtomatsko, se postavi vprašanje na kakšen način CPE dobi podatek o tem kje v pomnilniku se nahaja prekinitveni strežni program mikrokrmilniki uporabljajo v ta namen prekinitvene vektorje. To so posebne lokacije v pomnilniku iz katerih CPE prenese vsebino v programski števec. Te lokacije so točno določene in prirejene posameznim VH/IZ enotam. Vrednosti na posameznih lokacijah prekinitvenega vektorja moramo določiti pred uporabo prekinitve. Nedefinirana vrednost lahko ob prekinitveni zahtevi pomeni popolno blokado izvajanja programa. Prekinitveni vektorji so pri različnih mikrokrmilnikih postavljeni na različnih delih pomnilnika. Običajno so postavljeni na začetek (INTEL) ali konec (MOTOROLA) pomnilniškega področja.

Priključevanje vh/iz naprav-vrste registrov za dostop oz. vh/iz naprav, vgrajeni in zunanji krmilniki.

Mikroprocesorski sistemi dobijo svojo uporabno vrednost šele takrat, ko jim dodamo vhodno/izhodne naprave preko katerih komunicira z zunanjim svetom. Tipične naprave ki jih srečamo pri mikrokrmilnikih so: LED diode in prikazovalnik, LCD prikazovalnik, A/D in D/A pretvorniki, mali tiskalniki in podobno. Seveda teh naprav ne moremo priključiti direkt na vodilo mikrokrmilnik. To opravimo s posebnimi vezji, ki jim pravimo krmilniki (vmesniki, adapterji). Ti morajo omogočati na eni strani povezavo na vodilo na drugi strani pa čim bolj enostavno priključitev VH/IZ naprave. Krmilniki se glede na zgradbo med seboj zelo razlikujejo. Tako srečamo na eni strani zelo enostavne na drugi pa kompleksne. namenske. Mikrokrmilniki vsebujejo nekaj krmilnikov V/I naprav. Tipični krmilniki so serijski, paralelni, časovnik, A/D pretvornik ali pa tudi posebne krmilnike. V zaprtih aplikacijah uporabljamo samo vmesnike vgrajene v mikrokrmilnik običajno pa se pojavi potreba po dodajanju zunanjih vmesnikov. V tem primeru pa mora mikrokrmilnik delovati v razširjenem načinu saj moramo imeti dostop do vodila. Vmesnike priključujemo na vodilo paralelno preko večjega števila signalnih linij (podatkovnih, naslovnih, kontrolnih) Lahko pa jih priključimo preko posebnega vmesnika vodila tudi serijsko. Dostop do V/I naprave poteka torej preko vmesnika. Vsak vmesnik je programsko dostopen preko ene ali več pomnilniških lokacij. Glede na dostop do teh lokacij ločimo pomnilniško preslikan V/I ali posebni V/I. Pri prvem je dostop programsko enak dostopu do pomnilnika, v drugem pa za dostop uporabljamo posebne signale ter posebne ukaze. V splošnem vsak krmilnik vsebuje 3 osnovne registre:

- Podatkovni (preko njega poteka prenos podatkov v ali iz V/I naprave)
- Statusni (preko njega lahko CPE dobi podatke o stanju V/I naprav)
- Kontrolni (služi za nadzor nad V/I napravo ter krmilnikom)

Način delovanja krmilnika določimo z inicializacijo krmilnika. To pomeni vpis točno določenih vrednosti v registre krmilnika. Krmilnik je šele zatem pripravljen za delo z V/I napravo. Sama komunikacija z V/I napravo zajema vpis in branje podatkovnega in statusnega registra. Primer zgradbe in uporabe paralelnih vrat vmesnika na mikrokrmilniku HC11 vsebuje do 54 vhodno izhodnih linij odvisno od načina delovanja. Linije so razdeljene na 7 vrat, vse pa imajo dvojni pomen. Način delovanja posameznih vrat se izbira programsko z vpisom ustreznih vrednosti v pripadajoči register vrat.

Paralelna vrata MK 68HC11F1

Vsebuje do 54 vhodno izhodnih linij odvisno od načina delovanja. Linije so razdeljene na 7 vrat (port A do port G), vse pa imajo dvojni pomen. Način delovanja posameznih vrat se izbira programsko z vpisom ustreznih vrednosti v pripadajoči register vrat. Dvosmernim vratom, ki imajo možnost izbire vhod / izhod pripadata 2 registra: podatkovni in smerni . enosmernim vratom pa smo en register: podatkovni. Registri se nahajajo natančno določenih lokacijah v T. I. registrskem bloku. Privzeta vrednost začetne lokacije je \$1000, dolžina pa 56 bytov. Registrski blok lahko programsko prestavimo na drugo lokacijo. Vse V / I enote so dostopne preko registrov v registrskem bloku.

A/D pretvornik

Sestavlja 8 kanalni, 8 bitni pretvornik z multipleksnimi vhodi. Referenčna napetost je pripeljana na vhod Vrh, vhod Vrl pa je vezan na maso. A/D pretvornik ne vsebuje zunanega vezja za vzorčenje in zadrževanje signala, saj zato poskrbi sama izvedba A/D pretvornika (porazdelitev naboja). Časovno vodenje A/D pretvorbe je skladno s sistemskim E signalom, lahko pa izberemo notranji RC oscilator.ta način je uporaben kadar je frekvenca E signala manjša kot 750kHz. Multiplekser izbere za pretvorbo enega od 16-ih vhodov. Vhod programsko izberemo z vpisom vrednosti v kontrolni register A/D pretvornika. Analogni signali so pripeljani mna multiplekser preko 8 priključkov vrat E. Analogni pretvornik pretvori napetost na analognem vhodu izbranim z multiplekserjem. Sam pretvornik sestavlja D/A kondenzatorsko polje, komparator ter sukcesivno aproksimativni register. Vsaka pretvorba je sestavljena iz 8 operacij primerjave začeniš z bitom z največjo utežjo. Rezultat vsake primerjave se shrani v SAR register.Digitalni nadzor delovanja A/D pretvornika je nadzorovana preko bitov v kontrolnem registru A/D pretvornika. Poleg izbire analognega vhoda vsebuje z ADCTL še bite ki označujejo stanje pretvorbe ter način delovanja A/D pretvornika. -Shranjevanje rezultatov pretvorbe (ADR1-ADR4) se po končani pretvorbi prenesejo vrednosti iz SAR registra. Vsak register je posebej programsko dostopen.-postopek pretvorb A/D pretvorbo sestavljajo 4 zaporedne pretvorbe. Pretvorba se lahko izvaja zaporedno ali pa enkratno. Po končani 4. pretvorbi, se v ADCTL registru postavi bit CCF, ki označuje končano pretvorbo. - Vklop A/D pretvornika A/D pretvorba se začne s postavitvijo bita ADPU v OPTION registru na visok nivo. Po omogočitvi pretvorbe je potrebna zakasnitev saj 100us zaradi stabilizacije.

Števci in časovniki

Predstavljajo pomembno enoto MK razlika med njimi: - v funkciji števca se vrednost poveča za 1 s prehodom signala iz 0-1 ali obratno na enem od zunanjih priključku. - v funkciji časovnika se vrednost registra poveča v taktu notranjega generatorja, ki je običajno izveden iz osrednjega oscilatorja. - **uporaba** :- meritev časovnih intervalov - meritev frekvence - generiranje časovnih zakasnitev. - časovna nadzorna vezja. - štetje dogodkov. Časovniki in števci predstavljajo najbolj komplekдно vhodno/izhodno enoto MK. Nadzor je izveden preko registrov ter V/I priključkov vrat A. Uporabljen sta dva osnovna principa uporabe: 1. vhodni zajem(input capture) 2. izhodna primerjava(output compare) 1. funkcija vhodnega zajema omogoča beleženja časa, kdaj se je zgodil kakšen zunanji dogodek (sprememba stanja na V/I priključkih).2. funkcija služi za programiranje dogodkov, ki naj se zgodijo ob točno določenem času določenem z vrednostjo števca (registra) - TNCT. Funkcije časovnikov seveda ne smemo enačiti z uro realnega časa, čeprav jo lahko z njim realizirajo. Osnovo sistema predstavlja veriga, ki jo sestavljata programirjljivi delolnik osnovneka takta MK ter "prosto tekoči" števec. -

Programirljivi delilnik: Njegova naloga je ustrezno deljenje sistemske ure (E). Deljenje programsko nastavimo glede na pričakovane periode signalov, ki jih bomo merili. Primer : - deljenje z 1 resolucija časovnika 500 ns - 32,77 ms. - deljenje z 16 resolucija časovnika 8 us - 524,3 ms. Pravilna izbira faktorja deljenja je pomembna iz dveh razlogov: - obravnava daljših zakasnitev pri katerih pride do "preliva" časovnika je programsko bolj zahtevna zato se jo, če je le možno poskušamo izogniti. - poraba CMOS vezja, ki s frekvenco narašča.