1 Centralna procesna enota (CPE, CPU)

CPU usklajuje in skrbi za pravilno delovanje računalnika. Iz pomnilnika jemlje ukaze in jih izvršuje.Sestavljajo jo najmanj 3 enote:

- registri(služijo kot začasni pomnilnik podatkov)
- aritmetična logična enota (ALE) služi za izvajanje matematičnih in logičnih operacij (seštevanje,odštevanje, deljenje, množenje, komplement, IN, ALI, dvojiški komplement ipd.). Vsebuje tudi poseben register za shranjevanje zastavic (flag), ki se uporabljajo za prikaz statusa operacije (>,<, =,ipd.).
- kontrolna enota (nadzoruje in usklajuje pretok podatkov in ukazov znotraj CPE in med CPE ter pomnilnikom. Za prenos podatkov uporablja povezave -vodilo. To je sestavljeno iz signalov, ki jih delimo v tri skupine: naslovne, podatkovne in kontrolne.

2 Glavni pomnilnik (GP)

V njem so shranjeni ukazi in operandi, ki jih CPE uporablja (RAM, EPROM, ROM, EEPROM, flash ram, ..).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 hrani1 bit informacije. Število celic v besedi imenujemo dolžina pomnilniške besede. Iz pomnilnika ne moremo brati/pisati manj kot 1 besedo.Standardne dolžine so 4, 8 , 16, 32, 64 bitov.

3 Vhodno/izhodni sistem (V/I)

Skrbi za pretvarjanje podatkov iz človeku razumljive oblike v računalniku razumljivo obliko. Je tudi najbolj dostopen del računalnika. Vsaka V/I naprava je priključena na računalnik preko krmilnika naprave. Ti so lahko zelo enostavni ali pa kompleksni. Navzven je krmilnik videti kot določeno število registrov v katere lahko pišemo ali iz njih beremo.Obravnavamo jih lahko kot pomnilniške lokacije, razlika je le v tem, da vpis sproži operacijo v napravi oz. branje izraža stanje naprave.
mikroprocesor - mikroračunalnik na enem integriranem vezju
mikrokrmilnik - mikroračunalnik na enem integriranem vezju
Na vodilu prisotne enote delimo v 2 skupini: gospodar/suženj (master/slave). Vsa komunikacija med enotami v računalniku poteka vedno preko CPE (gospodar), vse ostale enote se ji morajo podrejati.

4 Načini delovanja mikrokrmilnikov – značilnosti

MK lahko deluje v 4 načinih:

- zaprtem (samostojni) načinu (single chip)
- razširjenem (expanded)
- samonalagalnem (samozagonskem) (bootstrap)

- testnem

4.1 1. Samostojni način

Naslovno in podatkovno vodilo od zunaj nista dostopna. Primeren za velikoserijske aplikacije.

4.2 Razširjeni način

Omogoča dodajanje zunanjih enot. Na vratih B in C sta dostopna naslovno in podatkovno vodilo.Vrat B in C lahko nadomestimo s posebnim zunanjim vezjem 28HC24.

4.3 Samozagonski način

Program je omejen na 512 bytov. MK deluje enako kot v samostojnem načinu. Po resetu se požene posebni nalagalni program, ki omogoča prenos programa preko serijskega vmesnika in njegov samodejni zagon po končanem nalaganju.

5 Programsko dostopni registri – vrste in značilnosti

- Vrata A (dvosmerna paralelna vrata ali kot vhodi števca)
- Vrata B (dvosmerna paralelna vrata ali kot del naslovnega vodila)
- Vrata C (dvosmerna paralelna vrata ali kot multipleksirani 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)
- Vrefl in Vrefh (vhoda za referenčno napetost A/D pretvornika)
- STRA/AS, STRB/R//W (V razširjenem načinu sta uporabljena kot izhoda. Kadar so na vratih C naslovi, je AS=1, R//W pa predstavlja kontrolni signal za pisanje in branje.)

6 Tipični signali mikrokrmilnika HC11

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

7 Načini naslavljanja

Mikrokrmilnik 68HC11 pozna 6 različnih načinov naslavljanja pomnilnika:

a) takojšnje (immmediate),
b) direktno (direct),
c) razširjeno (extended),
d) indeksno (indexed) z obema 16 bitnima registroma in 8 bitnim odmikom
e) vsebovano (inherent)
f) relativno (relative)

8 Vrste ukazov mikrokrmilnika HC11

HC11 pozna 329 ukazov. Ker je operacijska koda 8 bitna, kar omogoča uporabo samo 256 ukazov, je pri
določenih ukazih uporabljen še dodatni byte, ki je lahko $18, $21 ali $CD.
Ukaze lahko razdelimo v več skupin:

- polnjenje in shranjevanje registrov
- pomikanje registrov v registre
- povečevanje in zmanjševanje vsebine registrov
- brisanje in postavljanje bitov
- aritmetični ukazi
- premiki in rotacije
- testiranje podatkov
- pogojni skoki
- operacije s statusnim registrom (CCR)
- prekinitve in ostali

9 Sklad

9.1 Pomen sklada

Sklad (Stack) je poseben del RAM pomnilnika, ki ga uporabljamo kot začasno odlagališče. Mehanizem dela s skladom (shranjevanje in branje podatkov s sklada) je aparaturno podprt. Organizacija sklada je tipa LIFO (Last In First Out).Dostop do sklada poteka preko posebnega 16 bitnega registra imenovanega kazalec sklada (SP- stack pointer). 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.
Obstajata dva pristopa:
1. Pri vsakem vpisu se vrednost kazalca sklada zmanjša za 1 ali 2. Na začetku moramo kazalec sklada postaviti na vrh RAM pomnilnika (Motorola).
2. Pri vsakem vpisu se kazalec sklada poveča za 1 ali 2. Kazalec sklada moramo v tem primeru postaviti na začetek RAM pomnilnika (Intel).
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.

9.2 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 prične izvajati podprogram. Ta se mora obvezno končati z ukazom RTS. Ta ukaz povzroči, da se vrednost preje shranjenega programskega števca prenese iz sklada v programski števec. Druga uporaba sklada je pri prekinitvah. V tem primeru se na sklad shranijo vrednosti programsko dostopnih registrov.
Pred vsako aplikacijo moramo določiti globino sklada ter s tem tudi rezervirati del pomnilnika. Pri izračunu moramo upoštevati naslednje:

- vsak klic podprograma poglobi sklad za 2 lokaciji. Vsak podprogram lahko kliče tudi druge podprograme.
- vsaka prekinitev poglobi sklad za 7 lokacij. Prekinitve pa so lahko tudi vgnezdene. Določeni mikrokrmilniki (PIC) imajo vnaprej določeno globino sklada, ki ne presega 8 lokacij. Pri uporabi določenega krmilnika moramo te posebnosti upoštevati.

10 Podprogrami – pomen, uporaba

10.1 Podprogrami

Pri pisanju programov večkrat naletimo na primer, ko moramo del programa napisati večkrat. Temu se izognemo tako, ga 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 sledu klicu podprograma.
Posebno uporabnost dobijo podprogrami tudi pri delu z vhodno/izhodnimi 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 pispar
bra
…...
* podprogram
Pisipar STAA $8003 na lokacijo $8003 shrani vrednost v registru A.
RTS vrnitev iz podprograma
```

11 Prekinitve – pomen, uporaba, maskirane, nemaskirane, uporaba sklada, postopek ob prekinitvi

Sistem prekinitev omogoča CPE, da prekine trenutno izvajani program ter prične izvajati t.i. prekinitveni strežni program. Te zahteve lahko sprožijo vhodno izhodne naprave s tem, da aktivirajo posebne signale.To se običajno zgodi ob kateremkoli trenutku kar pomeni, da je zahteva po prekinitvi naključen dogodek, ki ga ne moremo vnaprej napovedati. Akcija, ki se ob prekinitveni zahtevi izvede je aparaturno realizirana in se izvaja samodejno brez posega programerja.V/I naprava poda

zahtevo za prekinitev kadar potrebuje poseg CPE npr. branje sprejetega znaka, konec A/D pretvorbe ipd. Po končanem prekinitvenem strežnem programu se mora program nadaljevati na mestu, kjer je bil prekinjen. Prekinitev ne sme vplivati na izvajanje programa kar pomeni, da se mora stanje programsko 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 ponovno prenesejo v registre.
Sam **postopek prekinitve** zajema 3 faze:

- prepoznavanje naprave, ki je sprožila prekinitev.
- določanje prioritete prekinitve v primeru, kadar je naprav več.
- potrjevanje prekinitve. Napravo moramo obvestiti, da je njeni zahtevi ugodeno.

Prepoznavanje naprave je običajno izvedeno programsko. Vsaka V/I naprava ima poseben register, v katerem označi prekinitveno zahtevo. Z branjem in testiranjem registra ugotovimo, katera naprava je zahtevala prekinitev. Določanje prioritete je lahko izvedeno programsko ali aparaturno. Prvo je povezano že kar s prepoznavanjem naprave.Potrjevanje prekinitve pomeni, da V/I naprava umakne svojo zahtevo za prekinitev.Seveda lahko komunikacijo z V/I napravo izvedemo tudi brez uporabe prekinitev s t.i. programskim izpraševanjem (polling). Ta način zahteva veliko večje posredovanje CPE pri komunikaciji kot uporaba prekinitev.
Prekinitve so podobne podprogramom, bistvena razlika med njima pa je v načinu klicanja. Podprogrami so klicani iz glavnega programa na točno določenih mestih, prekinitveni programi pa se izvajajo naključno ter samodejno. Prekinitve 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 fiksna ali pa programsko določena. Pri tem velja pravilo, da lahko prekinitev z večjo prioriteto prekine izvajanje prekinitve z nižjo prioriteto.Prekinitev delimo tudi na **maskiran**e in **nemaskirane.**
Maskirane prekinitev lahko programsko 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 prekinitve s posebnim ukazom onemogočimo, na koncu pa jih ponovno omogočimo.**Nemaskiranih** ne moremo programsko onemogočiti. Število zunanjih prekinitvenih vhodov je običajno 2 (1 maskirani, 1 nemaskirani).Čeprav je pristop do obravnave prekinitev podoben pri različnih mikrokrmilnikih, pa je pred uporabo konkretnega mikrokrmilnika potrebno dodobra preveriti način dela.

12 Prekinitveni vektor – uporaba

12.1 Prekinitveni vektor

Ker se prekinitvi izvajanje 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 t.i. prekinitvene vektorje. To so posebne lokacije v pomnilniku, iz katerih CPE prenese vsebino v programski števec. Te lokacije so točno določene in prirejene posameznim V/I 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.

13 Priključevanje vhodno/izhodnih naprav – vrste registrov za dostop do krmilnikov oz. V/I 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: tipke, LED diode in prikazovalniku, LCD prikazovalniki, A/D in D/A pretvorniki, mali tiskalniki ipd.Seveda pa direktno teh naprav ne moremo priključiti na vodilo mikrokrmilnika. To opravimo s posebnimi vezji ,ki jim pravimo krmilniki (vmesniki, adapterji). Ti morajo omogočati na eni strani povezavo na vodilo, na drugi pa čimbolj enostavno priključitev V/I naprave. Krmilniki se glede na zgradbo med sabo zelo razlikujejo. Tako srečamo na eni strani zelo enostavne, na drugi pa zelo kompleksne. Po namenu jih ločimo na univerzalne (npr. paralelni) ter namenske (komunikacijski). Mikrokrmilniki vsebujejo nekaj krmilnikov V/I naprav. Tipične krmilniki so paralelni, serijski, časovnik, A/D pretvornik ali pa tudi posebne krmilnike npr. za priključitev plinskega prikazovalnika, PLL sintetizatorja ipd.V zaprtih aplikacijah (mikrokrmilnik deluje v zaprtem načinu) lahko 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 in kontrolnih). Lahko pa jih priključimo preko posebnega vmesnika vodila tudi serijsko (IIC vodilo). Dostop do V/I naprave poteka torej preko vmesnika. Vsak vmesnik je programsko dostopen preko ene ali več pomnilniških lokacij (registrov krmilnika V/I naprav). 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 tri osnovne registre (pri enostavnih je lahko samo 1 register pri kompleksnejših jih je lahko nekaj 10):
- podatkovni - preko njega poteka prenos podatkov v ali iz V/I naprave.
- statusni - preko njega lahko CPE dobi podatke o stanju V/I naprave.
- kontrolni - služi za nadzor nad V/I napravo ter krmilnikom.

Način delovanja krmilnika določimo s t.i. inicializacijo krmilnika. Ta 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. Poglejmo primer zgradbe in uporabe paralelnih vrat (vmesnika) na mikrokrmilniku HC11. Mikrokrmilnik 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 pripadojači register vrat.

14 Paralelna vrata – enosmerna, dvosmerna, programska nastavitev smeri, smerni register

Mikrokrmilnik 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 pripadojači register vrat.Dvosmernim vratom, ki imajo možnost izbire vhod/izhod pripadata dva registra - podatkovni in smerni,enosmernim vratom pa samo en register - podatkovni.Registri se nahajajo na točno določenih lokacijah v t.i. registerskem bloku. Privzeta vrednost začetne lokacije je $1000, dolžina pa 96 bytov. Registerski blok lahko programsko prestavimo na drugo lokacijo.
Vse V/I enote so dostopne preko registrov v registerskem bloku

15 A/D pretvornik – načini delovanja, registri, postopek pretvorbe

A/D sistem sestavlja 8 kanalni, 8 bitni pretvornik z multipleksiranimi vhodi. Referenčna napetost je pripeljana na vhod Vrh, vhod Vrl pa je vezan na maso. A/D pretvornik ne potrebuje zunanjega 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 750 KHz.

- **multiplekser** izbere za pretvorbo enega od 16 vhodov. Vhod programsko izberemo z vpisom vrednosti v kontrolni register A/D pretvornika (register ADCTL). Analogni signali so pripeljani na multiplekser preko 8 priključkov vrat E.
- **analogni pretvornik** pretvori napetost na analognem vhodu izbranim z multiplekserjem. Sam pretvornik sestavlja D/A (DAC) kondenzatorsko polje, komparator ter sukcesivno aproksimativni register(SAR). Vsaka pretvorba je sestavljena iz osmih operacij primerjave, začenši z bitom z največjo utežjo (MSB). Rezultat vsake primerjave se shrani v SAR register. DAC polje deluje kot vezje za vzorčenje in držanje med celotno proceduro pretvorbe. Po končani pretvorbi se vrednost SAR registra prenese v ustrezni podatkovni register.
- **digitalni nadzor** delovanje A/D pretvornika je nadzorovano preko bitov v kontrolnem registru A/D pretvornika (ADCTL).Poleg izbire analognega vhoda vsebuje ADCTL še bite, ki označujejo stanje pretvorbe ter način delovanja A/D pretvornika.
- **shranjevanje rezultatov pretvorbe** V štiri 8 bitne registre (ADR1- ADR4) se po končani pretvorbi prenesejo vrednosti iz SAR registra. Vsak register posebej je programsko dostopen.
- **postopek pretvorb** A/D pretvorbo sestavljajo 4 zaporedne pretvorbe. Pretvorba se lahko izvaja zaporedoma ali pa enkratno. Po končani 4 pretvorbi, se v ADCTL registru postavi bit CCF, ki označuje končano pretvorbo.

15.1 Vklop A/D pretvornika

A/D pretvorba se prične s postavitvijo bita ADPU v OPTION registru na visok nivo. Po omogočitvi pretvorbe je poredba zakasnitev vsaj 100 us zaradi stabilizacije.

16 Časovniki – uporaba, sistem in vrste časovnikov, vhodni zajem, izhodna primerjava, uporaba prekinitev, števci

Časovniki predstavljajo pomembno enoto mikrokrmilnika. Kakšna je sploh razlika med njimi?

- v funkciji števca se vrednost registra poveča za 1 s prehodom signala iz 0 na 1 ali obratno na enem od zunanjih priključkov.
- 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 kompleksno vhodno/izhodno enoto v mikrokrmilniku. Nadzor je izveden preko registrov ter V/I priključkov vrat A. Uporabljena sta 2 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) - TCNT. Funkcije časovnikov seveda ne smemo enačiti z uro realnega časa, čeprav jo lahko z njimi realiziramo.
Osnovo sistem predstavlja veriga, ki jo sestavljata programirljivi delilnik osnovnega takta mikrokrmilnika ter prosto tekoči števec.

17 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 s 16 resoulucija 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

18 Primeri izračunov programskih zakasnitev, uporabe A/D pretvornika in časovnika

Primer :
Ugašanje in prižiganje led diode z zakasnitvijo

```
        ORG $4100          ( začetek programa )
ZAC     CLRA               ( resetiraj register A )
        STAA $8003         ( shrani na led diode )
        LDX #5000          ( ponovi cikel 5000 – 1 cikel   0,5  mikro sekunde )
ZAK1    DEX
        BNE ZAK1
        LDAA #$FF
        STAA $8003
        LDX #5000
ZAK2    DEX
        BNE ZAK2           ( skoči če ni enako )
        BRA ZAC            ( skoči na začetek
```