

Slika 4

**ALE, vodilo, Instrukcijski dekodler, nadzor, vrata v podatkovno in naslovno vodilo, PC, DP**

POMOŽNI AKUMULATOR (B). Razen tega bomo imeli še 8 splošnih registrov z oznakami RO do R7. Predpostavimo, da lahko ALE izvaja operacije nad vsemi registri (razen nad PC in DP) ter pomnilniškimi lokacijami. CPE ima svojo »uro«, to je vezje, ki daje napr. milijon impulzov na sekundo (frekvenca 1 MHz). Ura omogoča usklajeno delovanje vseh delov CPE. Ko že imamo zanesljiv časomerilec, si omislimo še register ČASOVNIK, katerega naloga bo, da se vrednost v njem vsako tisočinko sekunde poveča za ena. Dodajmo še en kazalec, register KAZALEC SKLADA (angl. Stack Pointer, SP), ki kaže na zadnji vnos v kopico podatkov nekje v pomnilniku. Potrebujemo ga za obravnavo posebnih podatkovnih struktur, ki jih bomo razlo-

žili kdaj drugič. Vpeljimo še indeksi register IND. Potrebovali ga bomo pri pregledovanju tabel in polj v pomnilniku. Pomemben je pri indeksnem naslavljanju, kjer se dejanski naslov generira kot vsota baznega naslova in odmika, shranjenega v indeksnem registru. Tako lahko ob stalnem baznem naslovu in spreminjanju odmika pregledamo polja v pomnilniku. Več o tem, ko bomo spregovorili o tem, kako naslavljamo operande.

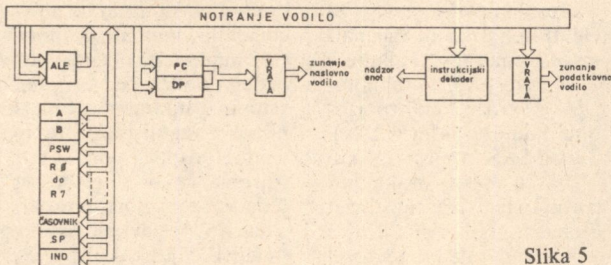
Še malenkosten dodatek. Uporabljamo še statusno besedo (angl. Program Status Word, PSW), to je register, ki hrani informacije o stanju zadnje operacije, iz-

peljane v ALE. Vsebuje zaznamke, ki povedo, ali je bil rezultat zadnje operacije nič, negativen, prevelik, da bi ga lahko zapisali v register, itd. Glede na te zaznamke bo programer določil, kaj naj se zgodi ob različnih prilikah.

S tem je naš preprosti CPE končan: celotna struktura je predstavljena na sliki 5.

V podrobnosti, kako posamezni deli delujejo, se ne bomo spuščali, zapomnimo si samo imena registrov, kajti ti so za uporabnika najvažnejši. Za uporabnika so pomembne samo funkcije

**ALE, vodilo, Instrukcijski dekodler, nadzor, vrata na podatkovno in naslovno vodilo, PC, DP, A, B, PSW, RO do R7, časovnik, SP in IND**



Slika 5

je, s katerimi preureja podatke, ti pa so shranjeni v registrih ali pomnilniku. Uporabniku sploh ni treba vedeti, da je ALE tista, ki bo seštelala dve vrednosti.

Dovolj je, da pozna inštrukcijo ADD A, B.

Ta sešteje vsebini registrov A in B.

**Prihodnjič: o naboru inštrukcij**

## RAČUNALNIŠKI SLOVARČEK

**VODILO** – Pot po kateri se pretakajo podatki med funkcionalnimi sklopi računalnika. Podatkovno vodilo je za podatke, naslovno vodilo pa prenaša naslove.

**VRATA** – (Vhodno/Izhodna). Imajo toliko bitov, kolikor je dolga računalniška beseda. Vrednost, ki pride na vrata, ostane zapisana na njih, dokler je ne prepíše nov podatek. Služi za prenosu podatkov iz naprave in v napravo.

**PROGRAMSKI ŠTEVNIK** – Register, ki kaže na naslednjo instrukcijo v pomnilniku, ki jo mora izvesti CPE. Ko je instrukcija dekodirana v instrukcijskem dekodlerju, se vrednost v programskem števniku avtomatsko poveča, tako da kaže zopet na naslednjo instrukcijo.

**PODATKOVNI ŠTEVNIKI** – Nekateri procesorji naslavljajo podatke v pomnilniku s pomočjo podatkovnega števnik, ki vsebuje kazalec na podatek.

**AKUMULATOR** – Register, v katerem se izvajajo aritmetične in logične operacije. Nekateri procesorji lahko izvajajo operacije tudi v drugih registrih.

**KAZALEC SKLADA** – Register, ki vsebuje kazalec na zadnji podatek v podatkovni strukturi SKLAD.

**SKLAD** – Podatkovna struktura. Podatki so zloženi kot opeka na opeki. Vsakič, ko je skladu dodan nov podatek, se ta zapiše na vrhu. Vsakič, ko podatek vzamemo s sklada, ga vzamemo z vrha. Temu principu pravijo LIFO (Last In First Out = Zadnji noter, prvi ven).

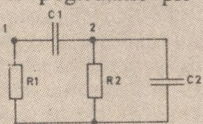
## Govor Vaše vprašanje

SPICE in ECAP). Potrebno je poudariti, da so taki programi zelo komplicirani in jih je skoraj nemogoče privediti za hišne računalnike.

Da bi analizirali elektronsko vezje, ga moramo računalniku najprej opisati, določiti pa moramo tudi vhodne signale. Računalnik nam nato izračuna napetosti in tokove, ki nas zanimajo, in jih izpiše v tabelah ali nariše grafe.

Elektronsko vezje je sestavljeno iz uporov, kondenzatorjev, (te elemente lahko imenujemo linearne) je enostavne. Ravno nasprotno pa velja za polprevodniške elemente, ki jih zato predstavljamo z nadomestnimi vezji. Pri opisu vezij uporabljamo teorijo grafov. To pomeni, da vozju določimo vozlišča in veje. Vozlišče je točka, kjer se stikata vsaj dva elementa, veja pa je povezava med dvema vozliščema, torej je to določen element vezja. Računalniku moramo opisati vse veje vezja in za vsako navesti, kateri vozlišči povezuje in kakšen element

predstavlja. Teorija bo jasnejša, če pogledamo primer:

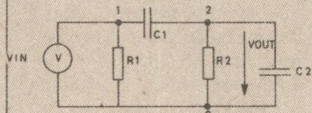


Enostavno vezje na sliki ima tri vozlišča (označena z 0, 1 in 2) in štiri veje (R1, R2, C1 in C2).

Vhodni signal elektronskega vezja lahko opišemo kot električno napetost ali električni tok. Signal je lahko konstanten (enosmeren), sinusni (izmenični) ali pa je sestavljen iz določenih impulzov. Računalniku vhodne signale opišemo tako, da vezju dodamo napetostne ali tokovne izvore, ki prav tako predstavljajo veje vezja.

Opis vezja je s tem zaključen. Računalniku moramo povedati, še, kaj želimo, da izpiše. Tudi tu smo »omejeni« na električno napetost med dvema vozliščema ali na električni tok po določeni veji vezja. Vsak računalniški program za analizo vezij ima določen vhodni jezik, ki ga lahko

primerjamo s programskimi jeziki, kot so FORTRAN ali PASCAL. Vsak jezik ima določena pravila (ali sintakso), ki jih moramo upoštevati, če hočemo, da bo računalnik opis vezja pravilno razumel. Še enkrat pogledajmo prejšnji primer, ki ga bomo zdaj razširili z vhodnim signalom in opisom računalniku:



R1 1,0 10K  
R2 q2,0 1K  
C1 1,2 100N  
O2 2,0 10N  
VIN 1,0 10N  
VIN 1,0 5 SIN  
PRINT VOUT 2,0  
END

Kot vidimo na sliki, je med vozlišči 1 in 0 priključen upor R1 z vrednostjo 10 kilo ohmov. Natančno to pove prva vrstica opisa. Podobno lahko analiziramo naslednje tri. Peta vrstica opiše vhodni signal, ki je v našem primeru sinusna napetost z amplitudo 5 voltov.

Naslednja vrstica zahteva, naj računalnik izračuna in napiše vrednost napetosti med vozliščema 2 in 0. Zadnja vrstica pa označuje konec podatkov.

Z računalnikom ponavadi analiziramo mnogo bolj zapletena vezja, ki vsebujejo tudi polprevodniške elemente in imajo lahko nekaj sto vozlišč. Kljub veliki hitrosti računanja lahko traja nekaj ur, preden dobimo rezultate. O računskih metodah, ki so za tak izračun potrebne, ne bomo govorili, saj to daleč presega naš nivo. Povejmo le to, da vse te metode temeljijo na treh osnovah elektronike, to so Ohmov in dva Kirchofova zakona, in na zahtevni numerični matematiki. O tem, da so programi za analizo

elektronskih vezij zares zahtevni, vas bo morda prepričal podatek, da je program SPICE dolg več kot tisoč fortranskih vrstic.

Prednosti računalniškega razvijanja vezij so vsekakor očitne. Ko vezje razvijemo »na papirju«, ga najprej preizkusimo z računalnikom. Spreminjamo lahko vrednosti posameznih elementov in njihov vpliv na delovanje celotnega vezja. Šele ko smo zadovoljni, pa vezje zares naredimo. Tako preizkušanje je hitrejšo, kot če bi bilo potrebno zares spreminjati pravo vezje. Izognemo pa se tudi temu, da zaradi napake pri načrtovanju uničimo pravkar narejeno vezje, takoj ko ga vklopimo.

Oglase za tekočo številko sprejemamo do vključno ponedeljka. Cena enega oglasa: 200 din, ne glede na dolžino.

Male oglase lahko bralci oddajo vsak dan med 7. in 19. uro, ob sobotah pa od 7. do 11. ure na blagajni za sprejem malih oglasov, v Ljubljani, Titova 35 (črna stolpnica) ali v Šubičevci 1. Lahko pa jih oddajo tudi kar po telefonu, v istem času, na številko 223-311 (klicna številka 061 za bralce izven Ljubljane). Male oglase lahko pošljete seveda tudi pismeno, in sicer na naslov: ČGP Delo Stik, Oglasno trženje, Titova 35.