

PROCESOR

[CPU]

[central procesing unit]

ZAČETKI:

Pri izdelavi procesorja so si inženirji, za zgled vzeli človeške možgane in njihovo delovanje preizkusili uporabiti pri izdelavi procesorja. Ampak tu obstaja problem, kajti ljudje še sedaj ne vemo kako pravzaprav človeški možgani delujejo. Zato inženirji niso mogli kopirati narave ampak so se morali spomniti kako se bodo lotili te zahtevne naloge.

Procesor lahko torej primerjamo s človeškimi možgani, saj je računalnik brez procesorja popolnoma neuporaben, kot tudi človek brez možganov. Procesor je zapleteno integrirano vezje zato, ker je prav od njega odvisna sleherni komponenta v računalniku. Prav zaradi zapletenosti izdelave računalniškega procesorja pa je na svetu zelo malo izdelovalcev teh čipov (Intel, AMD ter VIA). Največkrat je hitrostno procesor označen v megahercih (MHz), ki nam pove samo hitrost ne pa tudi zmogljivosti. Zmogljivost je dovisna od tega kako so strokovnjaki zasnovali procesor. Prvi začetki idej za izdelavo procesorja se gibljejo zelo v preteklost, ko so inženirji imeli v mislih procesor, kot skupek mehaničnih delov. Današnji procesorji pa so sestavljeni iz tranzistorjev. Tranzistorje so izumili v podjetju Bell labs. Podjetje Texas instruments pa je tranzistor prvič preneslo na silicijev kristal in tako je nastalo integrirano vezje.

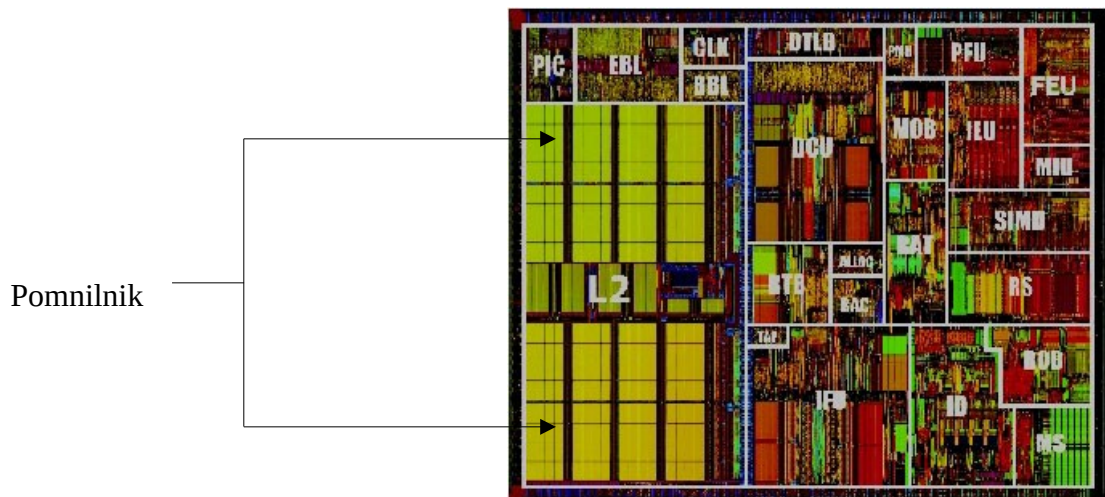
ARHITEKTURA:

Procesor je namenjen obdelavi informacij. Informacije so lahko podatki ali ukazi. Vse informacije so sestavljene v binarni kodi iz enic in ničel (barve, črke, številke,...).toda sami podatki procesorju ne povedo ničesar o tem, kaj mora z njimi storiti. Potrebne informacije za obdelavo podatkov mu nudijo ukazi, ki procesorju povedo, kaj naj s podatki naredi.

PREDPOMNILNIK:

Procesor si podatke dobiva iz osrednjega pomnilnika toda ta pomnilnik je za procesorske potrebe prepočasen. Zaradi te lastnosti so inženirji izumili predpomnilnik, ki je vgrajen v samo jedro procesorja, razdelimo ga na primarnega in sekundarnega. Včasih, ko tehnologija ni bila tako napredna, kot danes se je predpomnilnik delil na notranjega in zunanjega. Zunanji se je nahajal na matični plošči, notranji pa je bil vgrajen v jedro procesorja. Toda kmalu so inženirji ugotovili, da procesor rabi veliko več lasa da pride do zunanjega predpomnilnika, kot do notranjega. Zato so od takrat naprej začeli oba predpomnilnika vgrajevati v jedro procesorja. Toda to takrat ni bil lahek postopek, ker tehnologija ni bila tako napredna, kot danes. Tako so dobili ves predpomnilnik vgrajen v jedro šele prvi Celeroni s končnico A. Čeprav je bila hitrost vodila

Celeronov le 66MHz so se zaradi tega posega lahko primerjali z enako hitrimi Pentiumi 3. predpomnilnik mora biti hiter, zato so se inženirji odločili da bodo uporabili SRAM (static RAM) in ne DRAM (dynamic RAM), ker je SRAM hitrejši. Poleg tega pa SRAM ne potrebuje osveževalnega vezja (ta skrbi za osveževanje podatkov pri DRAM-u). Brez tega vezja bi bil sistemski pomnilnik SDRAM popolnoma neuporaben, kajti podatki bi se hitro izgubili. To pomeni da se podatki zapisani v SRAM v pomnilniku tudi obdržijo, da pa se podatki ne izbrišejo mora biti SRAM vedno pod električno napetostjo. Toda SRAM ima tudi slabost in to je njegova cena, poleg tega pa zasede tudi več prostora od DRAM-a.



CEVOVOD:

Cevovod je glavni segment obdelave ukazov, njemu sta podrejena tako CPU, kot tudi FPU. Cevovod je razdeljen na več delov, katerih število se lahko razlikuje. Cevovod zelo pripomore k zmogljivosti procesorja. Deluje v principu tekočega traku, to pa pomeni da lahko naenkrat obdeluje toliko ukazov, kolikor stopenj ima. 10 ukazov na enkrat se lahko obdeluje v cevovodu z desetimi stopnjami.

Za primer bom navedel cevovod s štirimi segmenti (fetch, decode, execute ter write back) ti segmenti pa si sledijo po vrsti. Fetch določen ukaz uvede v postopek obdelave. Zaradi počasnosti osrednjega pomnilnika, ukaze naloži posebno vezje iz sistemaškega pomnilnika v primarni predpomnilnik, od tam potem fetch naloži ukaze v cevovod. Potem je na vrsti decode, ki ukaze x86, ki so pravzaprav ukazi CISC in so dokaj zapleteni, prevede v ukaze RISC. Hitrost prevoda ukazov pa je odvisna od kompleksnosti ukazov x86. Pri enostavnejših ukazih se lahko prevede več ukazov na en cikel, pri zapletenejših ukazih pa prevod lahko traja več kot en cikel. Po segmentu decode je na vrsti segment execute. V tem koraku cevovod ugotovi, kje se bo ukaz izvršil (ukaz lahko tudi pošlje v ALU enoto ali enoto FPU ta je zadolžena za računanje s koreni, števili PI itd.). za executom sledi segment write back. Ta segment prejme rezultate izračunane v Fpu in integer enotah. Ta segment lahko rezultate shrani v registre ali pomnilnik. Zaradi

počasnosti systemskega pomnilnika rezultata ne zapiše direktno ampak ga najprej pošlje v tako imenovani Write Buffer. Od tu pa se rezultat zapiše v pomnilnik ali predpomnilnik. Več kot ima procesor Write bufferjev, več rezultatov lahko drži. Te štiri opisane segmente imajo vsi procesorji vendar imajo novejši procesorji te segmente še razdeljene. Nekateri imajo dva fetcha itd.

Cycles	Fetch	Decode	Execute	Save
1	Inst 1			
2	Inst 2	Inst 1		
3	Inst 3	Inst 2	Inst 1	
4	Inst 4	Inst 3	Inst 2	Inst 1
5		Inst 4	Inst 3	Inst 2
6			Inst 4	Inst 3
7				Inst 4

SLIKOVNI PRIKAZ CEVOVODA

To je zelo poenostavljen opis delovanja cevovoda saj imajo cevovodi v najnovejših procesorjih (Pentium 4 in novejši modeli podjetja AMD) še veliko drugih funkcij. S številom segmentov cevovoda lahko vplivamo tudi na frekvenco procesorja. Frekvenco lahko dvignemo zaradi manjših korakov, ki ukaze obdelujejo hitreje. Teoretično bi lahko segmente cevovoda razdelili tudi na 500 segmentov, vendar je tudi to število omejeno. Če bi naredili procesor s 500 stopnjami, bi bil ta procesor zelo velik in tudi drag. Poleg tega bi verjeno trpela tudi njegova učinkovitost. Problem je v samih ukazih, ki se obdelujejo v cevovodu, ker bi hitro naleteli na ukaze, ki vsebujejo pogoje (stavki if, else, then). Problem je v tem, da procesor ne more izvesti pogoja dokler pogojni ukaz ne pride do executa. Dokler ukaz ne pride do executa procesor ne ve katere ukaze naj pošlje v cevovod. Rešitev se imenuje *branch prediction* s pomočjo te tehnologije procesor predvidi, katere ukaze bo poslal v cevovod. Včasih pa se tudi procesor zmoti in napačno predvidi pogoj. V tem primeru mora sprazniti cevovod in ga ponovno napolniti z novimi ukazi.

ARITMETIČNO LOGIČNA ENOTA [ALU]

Ta logična enota je namenjena računanju s celimi števili. Novejši procesorji imajo več teh enot, medtem ko imajo starejši procesorji samo eno tako enoto. Več kot je aritmetičnih enot boljša je zmogljivost, ker se lahko vrši več ukazov naenkrat. Procesorji z več takimi enotami se imenujejo superskalarni. Naprednejši procesorji imajo aritmetično logične enote, ki so primerne za določene tipe ukazov.

FLOATING POINT UNIT [FPU]

Ta enota je namenjena računanju z realnimi števili in tistimi števili, ki niso cela. Prve Fpu enote niso bile vgrajene v procesor, zato so prvi procesorji uporabljali matematične koprocesorje, ki so bili ločeni od centralne procesne enote [CPU]. Prve FPU enote so bile vgrajene v jedra procesorjev 486 z oznako DX.

REGISTRI:

Registri so neke vrste hitri pomnilniki v jedru procesorja, ki vsebujejo vrednosti, ki jih procesor rabi pri računanju. Koliko biten je procesor nam pove širina registrov v bitih. Registri delujejo tako: če recimo hočemo zmnožiti dve števili mora procesor najprej naložiti ti dve števili v registre. Po končanem množenju procesor pošlje rezultat v pomnilnik ali predpomnilnik, lahko pa tudi v kak drug register.

Več kot ima procesor registrov, več lahko shrani podatkov, ki jih je potrebno obdelati. Toda več, kot ima procesor registrov zapletenejša je njegova izdelava.

CONTROL UNIT:

Kontrolna enota poskrbi za to, da vse enote v procesorju delujejo usklajeno. To je posebno vezje, ki ima pod nadzorom pretok podatkov skozi procesor (aritmetični logični enoti pove, kdaj začne prevajati ukaz v mikro ukaze itd....)

