

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko komunikacijske tehnologije

ROBERT ANIĆ

RAZVOJ MIKROPROCESORA

Završni rad

Pula, 29.01.2018. godine

Sveučilište Jurja Dobrile u Puli
Odjel za informacijsko komunikacijske tehnologije

ROBERT ANIĆ

RAZVOJ MIKROPROCESORA

Završni rad

JMBAG: 0303031801, izvanredni student

Studijski smjer: Informatika

Predmet: Osnove IKT

Mentor: prof. dr. sc. Vanja Bevanda

Pula, 29.01.2018. godine

Sadržaj

UVOD.....	1
1. RAČUNALNI SUSTAV	2
2. POVIJEST RAZVOJA MIKROPROCESORA.....	3
3. ARHITEKTURA MIKROPROCESORA	7
3.1 Općenito o mikroprocesorima.....	7
3.2 Sabirnice	7
3.3 Osnovni princip rada procesora.....	9
4. INSTRUKCIJSKI SKUP PROCESORA.....	10
4.1 Opći tipovi instrukcija.....	11
4.2 Arhitektura procesora s obzirom na instrukcijski skup.....	14
5. PROTOČNA STRUKTURA (PIPELINE)	16
6. CACHE MEMORIJA.....	18
7. POTPORA OPERACIJSKOG SUSTAVA.....	20
7.1 Prosesi	20
7.2 Dretve.....	21
7.3 Prekidi	23
8. PRIMJERI ARHITEKTURE INTELLOVIH MIKROPROCESORA.....	25
8.1 Intel Core Duo	25
8.2 Intel Core i7-990X.....	26
9. UNAPREĐENJA PERFORMANSI MIKROPROCESORA.....	28
ZAKLJUČAK	30
Literatura.....	31
Popis slika i tablica.....	32
Sažetak	33
Summary.....	34



IZJAVA O AKADEMSKOJ ČESTITOSTI

Ja, dolje potpisani Robert Anić, kandidat za prvostupnika informatike ovime izjavljujem da je ovaj Završni rad rezultat isključivo mogega vlastitog rada, da se temelji na mojim istraživanjima te da se oslanja na objavljenu literaturu kao što to pokazuju korištene bilješke i bibliografija. Izjavljujem da niti jedan dio Završnog rada nije napisan na nedozvoljen način, odnosno da je prepisan iz kojega necitiranog rada, te da ikoji dio rada krši bilo čija autorska prava. Izjavljujem, također, da nijedan dio rada nije iskorišten za koji drugi rad pri bilo kojoj drugoj visokoškolskoj, znanstvenoj ili radnoj ustanovi.

Student

U Puli, _____, _____ godine



IZJAVA o korištenju autorskog djela

Ja, Robert Anić dajem odobrenje Sveučilištu Jurja Dobrile u Puli, kao nositelju prava iskorištavanja, da moj završni rad pod nazivom „Razvoj mikroprocesora“ koristi na način da gore navedeno autorsko djelo, kao cjeloviti tekst trajno objavi u javnoj internetskoj bazi Sveučilišne knjižnice Sveučilišta Jurja Dobrile u Puli te kopira u javnu internetsku bazu završnih radova Nacionalne i sveučilišne knjižnice (stavljanje na raspolaganje javnosti), sve u skladu s Zakonom o autorskom pravu i drugim srodnim pravima i dobrom akademskom praksom, a radi promicanja otvorenoga, slobodnoga pristupa znanstvenim informacijama.

Za korištenje autorskog djela na gore navedeni način ne potražujem naknadu.

U Puli, _____ (datum)

Potpis

UVOD

Današnji proizvodi bazirani na mikroprocesorima uvelike su unaprijedili svako područje elektronike i imaju velik utjecaj na kvalitetu ljudskog života.

Napredak tehnologije rezultira eksponencijalnim rastom brzine takta i količine broja tranzistora koje je moguće postaviti na jedan čip.

Današnji mikroprocesori sadrže preko stotinu milijuna tranzistora ukomponiranih na silicijskoj pločici. Razvoj mikroprocesora se do danas nije zaustavio, no osnova njihovog rada je u principu ista: mikroprocesor kao ulaz prihvaća binarne podatke, obrađuje ih prema naredbama koje su spremljene u njegovoj memoriji i daje neki rezultat obrade tih podataka. Cilj ovog rada je opisati arhitekturu, način rada mikroprocesora i njihov razvoj kroz povijest.

U prvom poglavlju su opisane osnovne funkcije i glavne strukturalne komponente računalnog sustava.

U drugom poglavlju je opisana povijest razvoja mikroprocesora, od izuma mikroprocesora 1971. do suvremenih višejezgrenih mikroprocesora.

U trećem poglavlju je opisana arhitektura mikroprocesora, rad sabirnica i osnovni princip rada mikroprocesora.

U četvrtom poglavlju je dan opis instrukcijskog skupa procesora. Opisani su opći tipovi programskih instrukcija i arhitektura procesora s obzirom na skup instrukcija.

U petom poglavlju je opisana protočna arhitektura mikroprocesora.

U šestom poglavlju je opisana memorijska hijerarhija računala i objašnjena je funkcija priručnih (cache) memorija u računalu.

U sedmom poglavlju je opisana potpora operacijskog sustava kroz procese, dretve i prekide.

U osmom poglavlju su opisane arhitekture i načini rada višejezgrenih Intelovih arhitektura procesora Intel Core Duo i Intel Core i7-990X.

U devetom poglavlju je dan opis tehnika koje se koriste u svrhu unapređenja performansi mikroprocesora.

1. RAČUNALNI SUSTAV

Današnji računalni sustavi imaju brojne izvedbe i prisutni su u velikom dijelu našeg svakodnevnog života. Računala su složeni sustavi koji se sastoje od milijuna osnovnih elektroničkih dijelova. Računalni sustav promatramo kroz njegovu strukturu (način na koji su komponente međusobno povezane) i funkciju (radnje koje svaka komponenta obavlja).

Osnovne funkcije koje računalo može obavljati su:

Obrada podataka, pohrana podataka, prijenos podataka i upravljanje svim komponentama.

Najosnovnija funkcija računala je obrada podataka. Također je potrebno da se podaci sačuvaju. Čak i da računalo, dok radi, odmah šalje podatke, svejedno mu je potrebna kratkotrajna pohrana dijela podataka na kojima radi. Potrebna je i dugotrajna pohrana podataka kako bi se oni po potrebi mogli vraćati na obradu i ažurirati.

Računalo mora slati podatke između sebe i vanjskih uređaja. Takve uređaje zovemo ulazno/izlaznim uređajima i mogu služiti kao izvor ili kao odredište podataka.

Računalo mora raspolagati svojim resursima i upravljati ovim funkcijama.

Današnja računala, iako moderna, i dalje prate von Neumannovu arhitekturu.

Po njemu računalo ima četiri glavne strukturalne komponente:

CPU, glavna memorija, ulaz/izlaz i sustav koji povezuje navedene elemente.

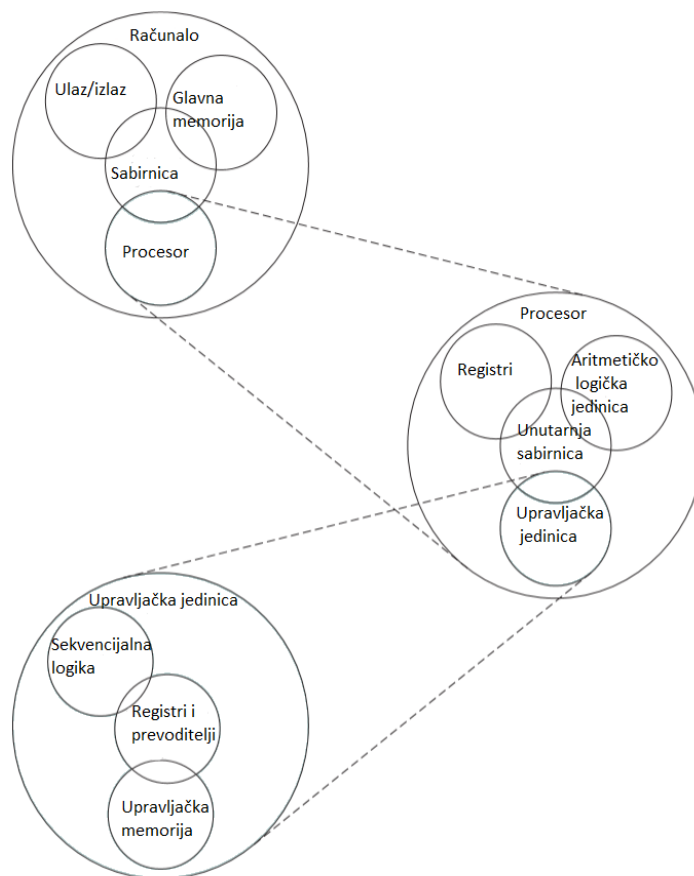
Najvažniji dio računala je procesor. Procesor je aktivni dio računala koji upravlja radom računala i obavlja različite operacije nad podacima.

Dio računalnog sustava u kojem se spremaju programi koji se izvode na procesoru te podaci potrebni za izvođenje tih programa naziva se memorija.

Ulaz je dio računalnog sustava namijenjen primanju podataka iz okoline u računalo, a izlaz je dio računalnog sustava namijenjen slanju podataka iz računala u okolinu.

Primjeri ulaznih jedinica su tipkovnica ili mikrofon, a izlaznih printer i zvučnik.

Mehanizam koji pruža komunikaciju između procesora, glavne memorije i ulazno/izlaznih uređaja. Primjer je sabirnica koja se sastoji od određenog broja žica na koje su sve ostale komponente spojene.



Slika 1: Računalo: top level struktura (Izvor: W. Stallings, 2012.)

Na slici 1 prikazan je računalni sustav. Na njoj vidimo da se računalo sastoji od procesora, glavne memorije, ulaza/izlaza i sabirnica koji međusobno povezuju sve navedene komponente. Procesor, kao najvažniji dio računala, sastoji se od upravljačke jedinice, aritmetičko logičke jedinice, registara i unutarnje sabirnice koja povezuje navedene komponente. Nadalje, upravljačka jedinica sadrži upravljačku memoriju, sekvencijalnu logiku, registre i prevoditelje.

2. POVIJEST RAZVOJA MIKROPROCESORA

Izraz „mikroprocesor“ se prvi put spominje 1971. godine, kada je tvrtka Intel proizvela prvi mikroprocesor INTEL 4004. Mikroprocesor je programabilna digitalna elektronička komponenta koja objedinjuje funkcije centralne upravljačke jedinice na jednom integriranom strujnom krugu. (D.K. Kaushik, 2010)

Najzaslužniji za to je Marcian E. Hoff. Po današnjim standardima 4004 je primitivan,

ali označava početak razvoja mikroprocesora. Sljedeći veći napredak se dogodio 1972. kada Intel proizvodi prvi 8-bitni mikroprocesor Intel 8008, a 1974. izlazi prvi multinamjenski mikroprocesor 8080. S dolaskom 16-bitnih mikroprocesora dolaze i neki novi koncepti poput pipelining-a i upravljanja virtualnom memorijom. Uvedene su naredbe pomnoži i podijeli, te poboljšani načini adresiranja. Intel 8086 je bio prvi od novije vrste 16-bitnih mikroprocesora visokih performansi. Pojavio se 1978. godine i sadržavao je oko 29 000 tranzistora, a pristup memoriji se ostvarivao za 100 ns. Postizao je ovu brzinu bez brzih memorijskih komponenti ili odvojenih podatkovnih i adresnih sabirnica. 20-bitna adresna sabirnica dopušta 1 MB memorije. Intel 80286 koji je predstavljen 1982. ima još bolje mogućnosti. Sadrži procesor radnog takta 10 MHz, memorijsku upravljačku jedinicu s 4 nivoa memorijske zaštite i virtualnu memoriju. Podržava 16 MB fizičke i 1 GB virtualne memorije.

Prvi u potpunosti 32-bitni mikroprocesor s 32-bitnom sabirnicom i 32-bitnim adresama je bio BELLMAC-32A tvrtke AT&T Bell Labs 1982. godine.

Prvi mikroprocesor koji je dosegao broj od milijun tranzistora je bio Intelov 32-bitni 80486 iz 1989. godine. Točnije, sadržavao ih je 1.18 milijuna. Dostizao je brzinu od 40 milijuna operacija u sekundi s taktom brzine od 50 megaherca, a koristio se u osobnim računalima i serverima.

Pentium je ponuđen tržištu u ožujku 1993., bio je to 32-bitni procesor, potpuno kompatibilan s prijašnjim programima, imao je takt brzine 60 megaherca i 3.1 milijun tranzistora. proizvedenih u 0,35 mikronskom procesu. Jedna od inovacija kojom se odlikovao je bilo paralelno izvođenje operacija, radi čega je dostizao veću brzinu izvođenja.

Intelov Pentium 4 je na tržištu izašao 2000. i uveo je neke nove osobine mikroarhitekture nazvane "NetBurst" koje su u velikom značaju utjecale na performanse. Uveo je implementaciju skrivene priručne memorije prve razine otklanjajući njome kašnjenje pridruženo s dekoderom instrukcija u glavnim petljama izvršenja, priručnu memoriju druge razine s naprednim prijenosom od 256 kilobajta koja osigurava snagu od 256 bita za prijenos podataka u svaki blok jezgri - omogućavajući tako veću propusnu moć kanala.

Tablica 1: Intelovi mikroprocesori 1970-te (W. Stallings, 2012)

	4004	8008	8080	8086	8088
Predstavljen	1971	1972	1974	1978	1979
Brzina takta	108 kHz	108 kHz	2 MHz	5 MHz, 8 MHz, 10 MHz	5 MHz, 8 MHz
Širina sabirnice	4 bits	8 bits	8 bits	16 bits	8 bits
Broj tranzistora	2300	3500	6000	29,000	29,000
Veličina elemenata (μm)	10		6	3	6
Adresabilna memorija	640 Bytes	16 kB	64 kB	1 MB	1 MB

Tablica 2: Intelovi mikroprocesori 1980-te (W. Stallings, 2012)

	80286	386TM DX	386TM SX	486TM DX CPU
Predstavljen	1982	1985	1988	1989
Brzina takta	6 MHz-12.5 MHz	16 MHz-33 MHz	16 MHz-33 MHz	25 MHz-50 MHz
Širina sabirnice	16 bits	32 bits	16 bits	32 bits
Broj tranzistora	134,000	275,000	275,000	1.2 milijuna
Veličina elemenata (μm)	1.5	1	1	0.8-1
Adresabilna memorija	16 MB	4 GB	16 MB	4 GB
Virtualna memorija	1 GB	64 TB	64 TB	64 TB
Cache	-	-	-	8 kB

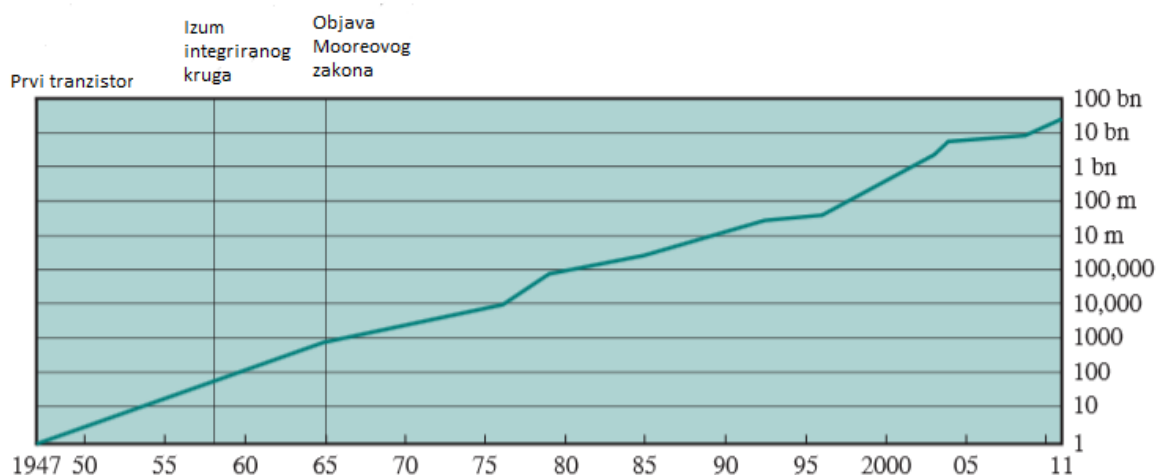
Tablica 3: Intelovi mikroprocesori 1990-te (W. Stallings, 2012)

	486TM SX	Pentium	Pentium Pro	Pentium II
Predstavljen	1991	1993	1995	1997
Brzina takta	16 MHz-33 MHz	60 MHz-166 MHz	150 MHz-200 MHz	200 MHz-300 MHz
Širina sabirnice	32 bits	32 bits	64 bits	64 bits
Broj tranzistora	1.185 milijuna	3.1 milijun	5.5 milijuna	7.5 milijuna
Veličina elemenata (μm)	1	0.8	0.6	0.35
Adresabilna memorija	4 GB	4 GB	64 GB	64 GB
Virtualna memorija	64 TB	64 TB	64 TB	64 TB
Cache	8 kB	8 kB	512 kB L1 i 1 MB L2	512 kB L2

Tablica 4: Intelovi mikroprocesori 2000-te (W. Stallings, 2012)

	Pentium III	Pentium 4	Core 2 Duo	Core i7 EE 990
Predstavljen	1999	2000	2006	2011
Brzina takta	450 MHz-660 MHz	1.3-1.8 GHz	1.06-1.2 GHz	3.5 GHz
Širina sabirnice	64 bits	64 bits	64 bits	64 bits
Broj tranzistora	9.5 milijuna	42 milijuna	167 milijuna	1170 milijuna
Veličina elemenata (nm)	250	180	65	32
Adresabilna memorija	64 GB	64 GB	64 GB	64 GB
Virtualna memorija	64 TB	64 TB	64 TB	64 TB
Cache	512 kB L2	256 kB L2	2 MB L2	1.5 MB L2/12 MB L3

1965. Gordon Moore primjećuje da se broj tranzistora koji se mogu postaviti na jedan čip udvostručuje svake godine i predviđa da će se ta brzina nastaviti. Predviđanje se ispostavilo točnim, ali je brzina udvostručavanja usporila na 18 mjeseci u 70-tim godinama, ali od tada se ta brzina nastavlja.



Slika 2: Mooreov zakon (W. Stallings, 2012)

Proizvođači mikroprocesorskih čipova se konstantno trude proizvoditi sve brže i brže procesore. Dok god evolucija prati Mooreov zakon, svake tri godine se može proizvesti nova generacija čipova s četiri puta više tranzistora. Mnogi stručnjaci predviđaju kolaps ovog zakona u sljedećih 10-ak godina.

Iako i dalje vrijedi Mooreov zakon (naravno, to nije zakon niti u matematičkom, niti u fizikalnom smislu, to je statistička prognoza), danas se kaže: "vrijeme jednostavnog

povećanja brzine programa je prošlo". Radni takt procesora praktički je prestao rasti oko 2005. godine. Razlozi za to su prije svega veliko povećanje potrošnje energije procesora na velikim brzinama i problem zagrijavanja. Zbog toga su se proizvođači okrenuli drugačijem načinu povećanja performansi procesora. Umjesto povećanja brzine, povećali su broj CPU-a na jednom mikroprocesorskom čipu, tj. počeli su proizvoditi višejezgrene procesore.

3. ARHITEKTURA MIKROPROCESORA

3.1 Općenito o mikroprocesorima

„Mikroprocesor je složeni programski upravljivi sklop koji pribavlja, dekodira i izvršava instrukcije. Mikroprocesor komunicira s memorijskim i ulazno-izlaznim modulima digitalnog sustava: pribavlja instrukcije ili naredbe te prima, obrađuje i šalje podatke. Mikroprocesor, kao centralna upravljačka jedinica, ima sklopove za rukovanje podacima i upravljačke sklopove.“ (S. Ribarić, 1992.)

Mikroprocesor (CPU) je najvažniji elektronički sklop računala, koji preuzima programske naredbe i na osnovu njih obrađuje podatke. Naziv "mikro" koristi se zbog malih dimenzija. Upravlja memorijom i ulazno/izlaznim uređajima putem naredbi koje su pohranjene u memoriji. Sadrži do nekoliko milijuna tranzistora ukomponiranih na silicijskoj pločici upakiranoj u plastično kućište. Mikroprocesori koji se koriste u računalnim sustavima opće namjene dizajnirani su za rad s podacima duljine 4, 8, 16, 32 i 64 bit-a.

Funkcije procesora su:

dohvat naredbi, prevođenje naredbe, dohvat podataka, obrada podataka i zapis podataka.

Procesor čita naredbu iz memorije. Naredba se dekodira da bi se ustanovilo koje se radnje trebaju izvršiti. Izvršavanje naredbe može uključivati učitavanje podataka iz memorije i ulazno/izlaznih modula. Izvršavanje naredbe također može zahtijevati neku aritmetičku ili logičku operaciju nad podacima. Rezultat izvršavanja može zahtijevati upis podataka u memoriju ili ulazno/izlazni modul.

3.2 Sabirnice

Sabirnica je zajednički snop vodiča koji međusobno povezuju komponente unutar računala. Osim vodiča, sabirnica ima i svoj sabirnički elektronički sklop koji pomaže

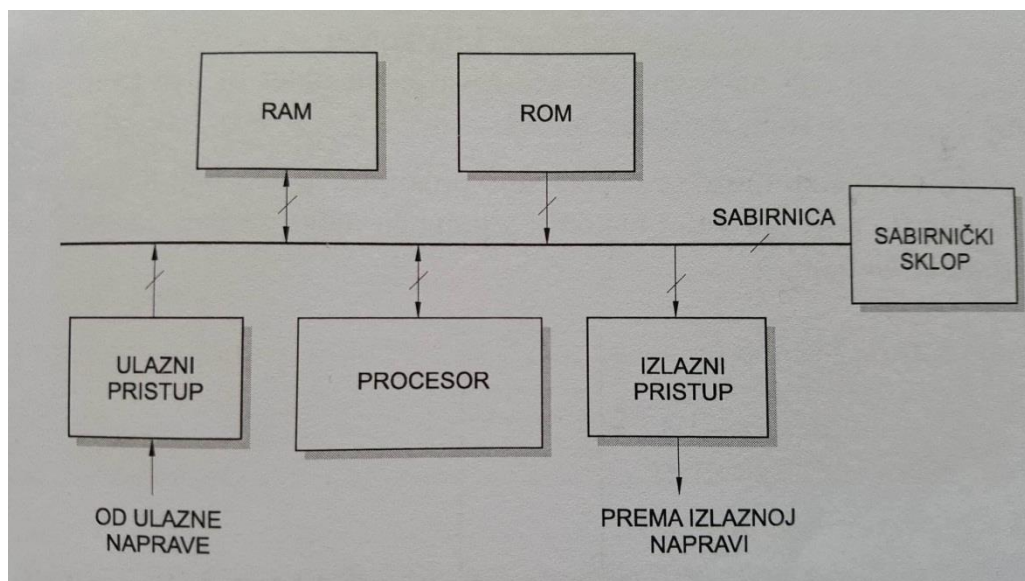
pri ostvarivanju veza. Sabirnice prenose adrese, podatke i upravljačke informacije između mikroprocesora, memorije i ulazno/izlaznih uređaja, te ih stoga dijelimo na tri osnovna tipa: adresna, podatkovna i upravljačka sabirnica.

Adresna sabirnica zahtijeva memorijsku lokaciju iz memorije ili ulazno/izlaznu lokaciju od ulazno/izlaznih uređaja. Ako je adresiran ulazno/izlazni uređaj adresna sabirnica sadrži 16-bitnu adresu od 0000H do FFFFH adresnog prostora. Ako je adresirana memorija adresna sabirnica sadrži memorijsku adresu koja varira s obzirom na to koja verzija mikroprocesora je u sustavu.

Podatkovna sabirnica prenosi informacije između mikroprocesora, njegove memorije i ulazno/izlaznog adresnog prostora. Preneseni podatci variraju po veličini, od 8-bitnih do 64-bitnih.

Upravljačka sabirnica upravlja prijenosom podataka i radom sustava. (Barry B. Brey , 2008.)

Glavne prednosti upotrebe sabirnica u sustavu su jednostavnost, fleksibilnost i pouzdanost. Glavna mana upotrebe sabirnica je što svi podatci koji idu iz memorije ili u memoriju moraju proći kroz sabirnicu, te stoga njezin vremenski ciklus ograničuje brzinu cijelog sustava, ali ta mana se uvelike smanjuje upotrebom cache memorija. (W. Stallings, 2012.)



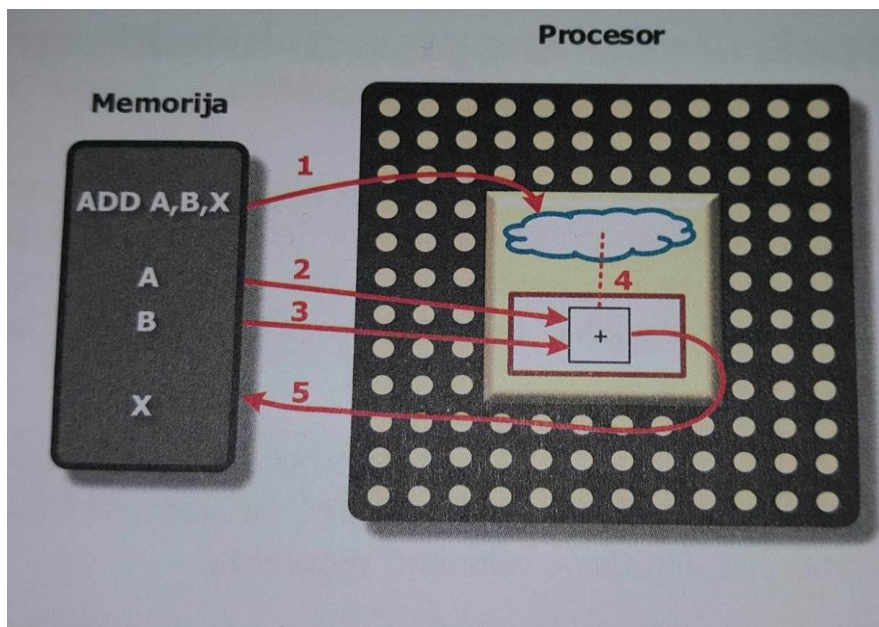
Slika 3: Sklopovlje računala povezano sabirnicom (Izvor: L. Budin, 2011.)

Na slici 3 je prikazano računalo sa sabirničkim povezivanjem dijelova. Preko sabirnica se obavljaju sve razmjene podataka, instrukcija i upravljačkih signala. Kosa

crta preko crte koja predstavlja sabirnicu označava da je riječ o snopu vodiča. Preko zajedničkih vodiča se ne može uspostavljati više od jedne istovremene veze, već se mogu obavljati samo pojedinačni prijenosi. Sabirnica se stoga mora, razmjenom vremena, koristiti za ostvarenje potrebnih veza između dijelova računala. Vrijeme se na sabirnici dijeli na sabirničke cikluse.

3.3 Osnovni princip rada procesora

Osnovna funkcija procesora je izvođenje programa. Program se nalazi u memoriji i sastoji se od niza naredbi (instrukcija). Naredba je najmanja cjelina koju procesor može razumjeti i sastoji se od niza bitova po kojima procesor zna koje će aktivnosti obaviti. Za privremeno čuvanje podataka procesor koristi registre. Oni služe za pohranu onih podataka koji ulaze u aritmetičko logičku jedinicu i rezultata obrade aritmetičko logičke jedinice. Rad procesora se odvija u tri osnovna koraka koji se ponavljaju, a to su: čitanje naredbe iz memorije, prepoznavanje pojedine kombinacije bitova naredbe i obavljanje aktivnosti koje su zadane unutar naredbe.



Slika 4: Osnovni koraci izvođenja naredbe (Izvor: M. Kovač, 2015)

Na slici 2 prikazani su osnovni koraci rada procesora tijekom izvođenja operacije zbrajanja. Brojevi na slici označavaju redoslijed izvođenja operacija. Naredba za izvođenje operacije zbrajanja ADD se nalazi u memoriji računala. U ovom primjeru naredba opisuje da procesor treba obaviti operaciju zbrajanja dvaju operanda A i B te rezultat X spremi natrag u memoriju. Za vrijeme dohvata upravljačka jedinica

generira sve potrebne signale kojima se naredba čita iz memorije i dovodi do upravljačke jedinice procesora. Zatim slijedi dekodiranje gdje upravljačka jedinica prepoznaje što treba napraviti, u ovom slučaju treba zbrojiti dva operanda. Tijekom koraka izvođenja, upravljačka jedinica generira sve potrebne signale da se iz memorije učita operand A i dovede ga do sklopa za obradu podataka. Na isti način dovodi i operand B na obradu. Nakon što su dohvaćeni svi potrebni podaci, upravljačka jedinica pokreće operaciju zbrajanja i rezultat pohranjuje u memoriju.

4. INSTRUKCIJSKI SKUP PROCESORA

Osnovna svojstva i ponašanje procesora određeni su skupom registara i skupom instrukcija koje procesor može obaviti. Računalna instrukcija je binarni kod koji određuje slijed mikrooperacija koje računalo treba izvršiti. Instrukcije su , zajedno s podacima, spremljene u memoriji računala. Da bi procesor izvršio zadani program i dobio rezultate mora izvršiti određen broj naredbi. Procesor razumije samo strojne naredbe, a one su određene prema arhitekturi instrukcijskog seta procesora (ISA). Broj strojnih naredbi kod izvođenja naredbi se razlikuje od sustava do sustava zbog razlika u arhitekturi instrukcijskog seta i prevoditeljskih alata.

Zadatak koji će procesor izvršiti ovisi o instrukciji koju izvršava. Sve instrukcije koje procesor može izvršiti nazivaju se instrukcijskim skupom procesora. U najjednostavnijem obliku, izvršavanje instrukcija sastoji se od dva koraka: dohvat instrukcije iz memorije i izvršavanje instrukcije. Izvršavanje programa se sastoji od ponavljanja ova dva koraka. Upravljačka jedinica mora dekodirati instrukcije i na temelju njihova sadržaja narediti aritmetičko logičkoj jedinici koju operaciju treba obaviti te povezati potrebne registre na sabirnice. Izvršavanje koje je potrebno da bi se izvršila instrukcija nazivamo instrukcijskim ciklusom. Na početku svakog instrukcijskog ciklusa procesor dohvaća instrukciju iz memorije. Programski brojač sadrži adresu sljedeće instrukcije koju treba dohvatiti. Nakon što procesor dohvati sljedeću instrukciju, programski brojač, ako mu se ne naredi drukčije, dohvaća sljedeću instrukciju sekvencijalno. Dohvaćena instrukcija se sprema u instrukcijski registar. Procesor prevodi bitove u instrukciji koji mu dodjeljuju koju radnju treba izvršiti. Te radnje dijelimo u četiri kategorije:

Procesor-memorija, procesor-ulaz/izlaz, rad na podacima i upravljanje.

Podatci se premještaju iz procesora u memoriju ili iz memorije u procesor.

Podatci se mogu premještati na ili s vanjskog uređaja razmjenom između procesora i ulazno/izlaznog modula. Procesor može obavljati neku aritmetičku ili logičku radnju nad podacima. Instrukcija može narediti da se redoslijed izvršavanja promijeni.

Memorijska jedinica ima kapacitet 4096 riječi i svaka riječ sadrži 16 bitova. Dvanaest bitova instrukcijske riječi je potrebno da bi se odredila adresa operanda. To ostavlja tri bita za operacijski dio instrukcije i jedan bit da bi se odredila adresa. Podatkovni registar (PDR) sadrži operand koji je dohvaćen iz memorije. Akumulatorski registar (AK) je registar opće namjene. U instrukcijski registar (IR) se učitava instrukcija iz memorije. Privremeni registar (PRR) sadrži privremene podatke tijekom izvršavanja radnji. Registar memorijske adrese (RMA) ima 12 bitova koliko je velika i adresa. Programski brojač (PB) također ima 12 bitova i sadrži adresu instrukcije koja se treba učitati iz memorije. Ulazni registar (ULR) i izlazni registar (IZR) imaju veličinu 8 bitova.

Tablica 5: Popis registara za osnovno računalo (M. Mano, 2009)

Simbol registra	Broj bitova	Naziv registra	Funkcija
PDR	16	Podatkovni registar	Sadrži operand
AK	16	Akumulatorski registar	Registar procesora
IR	16	Instrukcijski registar	Sadrži instrukcijski kod
PRR	16	Privremeni registar	Sadrži privremene podatke
RMA	12	Adresni registar	Sadrži adresu za memoriju
PB	12	Programski brojač	Sadrži adresu sljedeće instrukcije
ULR	8	Ulazni registar	Sadrži ulazne podatke
IZR	8	Izlazni registar	Sadrži izlazne podatke

4.1 Opći tipovi instrukcija

Opći tipovi instrukcija dijele se na više skupina: Instrukcije za prijenos podataka, aritmetičke instrukcije, logičke instrukcije, pretvaračke instrukcije, instrukcije za prijenos kontrole i ulazno/izlazne instrukcije.

Najosnovniji tipovi instrukcija su oni za prijenos podataka. Instrukcija prvo mora

odrediti lokaciju izvor i odredište operanada. Zatim mora naglasiti veličinu podataka koji se trebaju prenijeti i mora odrediti način adresiranja za svaki operand.

Tablica 6: Osnovne instrukcije za prijenos podataka(W. Stallings, 2012)

Naziv radnje	Opis
Premjestiti	Premještaj riječi iz izvora do odredišta
Pohraniti	Premještaj riječi iz procesora u memoriju
Dohvatiti	Premještaj riječi iz memorije u procesor
Razmijeniti	Zamjena sadržaja izvora i odredišta
Očistiti	Premještaj vrijednosti 0 u odredište
Postaviti	Premještaj vrijednosti 1 u odredište
Gurnuti	Premještaj riječi iz izvora na vrh stoga
Izbaciti	Premještaj riječi iz vrha stoga na izvor

Aritmetičke instrukcije u većini osnovnih sustava pružaju osnovne operacije zbrajanja, oduzimanja, množenja i dijeljenja. Obrada aritmetičke instrukcije zahtijeva prijenos operanada u aritmetičko logičku jedinicu i rezultata obrade iz nje.

Tablica 7: Osnovne aritmetičke instrukcije (W. Stallings, 2012)

Naziv radnje	Opis
Dodaj	Računanje sume dva operanda
Oduzmi	Računanje razlike dva operanda
Pomnoži	Računanje produkta dva operanda
Podijeli	Računanje kvocijenta dva operanda
Apsolutna vrijednost	Mijenjanje operanda njegovom apsolutnom vrijednosti
Negiraj	Mijenjanje predznaka operanda
Priraštaj	Uvećanje operanda za 1
Smanjiti	Smanjenje operanda za 1

Logičke instrukcije se baziraju na Booleovoj algebri. Mogu pomicati bitove u riječi i tako ih preslagivati.

Tablica 8: Osnovne logičke instrukcije (W. Stallings, 2012)

Naziv radnje	Opis
I	Izvođenje logičkog I
ILI	Izvođenje logičkog ILI
NE	Izvođenje logičkog NE
Isključujući ILI	Izvođenje logičkog XOR
Provjera	Provjera određenog uvjeta

Usporedba	Logička ili aritmetička usporedba dvaju ili više operanada
Postavljanje upravljačkih varijabli	Klasa instrukcija za postavljanje upravljanja iz sigurnosnih razloga
Premještaj	Operand za premještaj lijevo ili desno sa konstantama na kraju
Rotacija	Operand za premještaj lijevo ili desno sa sveobuhvatnim krajem

Pretvaračke instrukcije mijenjaju format podataka. Primjer je mijenjanje podataka iz decimalnog oblika u binarni.

Tablica 9: Osnovne pretvaračke instrukcije (W. Stallings, 2012)

Naziv radnje	Opis
Prijevod	Prevodi vrijednosti u dijelu memorije prema tablici odgovaranja
Pretvaranje	Pretvara sadržaj riječi iz jednog oblika u drugi (npr. Iz decimalnog u binarni)

Postoji više razloga zbog kojih su potrebne instrukcije prijenosa kontrole.

U radu s računalom od velike je važnosti mogućnost da se svaka instrukcija može izvršiti više puta, ponekad i tisućama puta. Nezamislivo bi bilo kada bi za implementiranje neke aplikacije bili potrebni milijuni instrukcija i da se svaka od njih mora zasebno ispisati. Ako se treba obraditi tablica ili više stavki potrebna je programska petlja koja poziva jednu sekvencu instrukcija da se iznova ponavlja dok se ne obrade svi podatci.

Gotovo svi programi zahtijevaju donošenje odluka. Računalu dajemo neki uvjet, ako se taj uvjet ispuni računalo treba reagirati na jedan način, a ako se uvjet ne ispuni treba reagirati na drugi način.

Izraditi računalni program za velika računala ili računala srednje veličine je iznimno težak zadatak. Da bi olakšali taj zadatak koristimo mehanizme koji razbijaju zadatak u više manjih dijelova da se može raditi na njima na svakom zasebno.

Tablica 10: Osnovne upravljačke instrukcije (W. Stallings, 2012)

Naziv radnje	Opis
Skok (grananje)	Bezuvjetni prijenos; Napuni programsko brojilo određenom adresom
Uvjetni skok	Provjeri određeni uvjet; ovisno o uvjetu,

	napuni programsko brojilo određenom adresom ili ne radi ništa
Skok na određeno mjesto	Postavlja trenutnu informaciju o kontroli programa na znanu lokaciju; skok na određenu adresu
Povratak	Mijenja sadržaj programskog brojila i drugih registara sa znane lokacije
Izvršenje	Dohvati operand iz određene lokacije i izvrši instrukciju
Preskok	Dodaj priraštaj programskom brojilu i preskoči sljedeću instrukciju
Uvjetni preskok	Provjeri određeni uvjet; ovisno o uvjetu, preskoči ili ne radi ništa
Zastoj	Zaustavi izvršenje programa
Čekanje	Zaustavi izvršenje programa; kontinuirano provjeravaj određeni uvjet; nastavi s izvršenjem kada je uvjet zadovoljen
Bez radnje	Nikakva radnja se ne događa, ali program nastavlja s izvršavanjem

Da bi se izvršila ulazno/izlazna instrukcija procesor izdaje adresu i tako određuje ulazno/izlazni modul i vanjski uređaj i ulazno/izlazni zapovijed. Postoje četiri tipa ulazno/izlaznih zapovijedi, to su: kontrolne, testne, zapovijedi za čitanje i zapovijedi za pisanje.

Tablica 11: Osnovne ulazno/izlazne instrukcije (W. Stallings, 2012)

Naziv radnje	Opis
Ulaz (čitaj)	Premjesti podatke iz određenog ulazno/izlaznog utora ili uređaja na određeno mjesto
Izlaz (piši)	Premjesti podatke iz određenog izvora na ulazno/izlazni utor ili uređaj
Započni ulaz/izlaz	Premjesti instrukcije u ulazno/izlazni procesor da bi započela ulazno/izlazna radnja
Provjeri ulaz/izlaz	Premjesti informacije o stanju iz ulazno/izlaznog sustava na određenu lokaciju

4.2 Arhitektura procesora s obzirom na instrukcijski skup

Postoje dvije dominantne arhitekture procesora, a to su: CISC (engl. Complex Instruction Set Computer) i RISC (engl. Reduced Instruction Set Computer).

CISC arhitektura pokušava koristiti što manje instrukcija po programu, ali zato instrukcije zahtijevaju više vremena za izvršenje. Osnovne karakteristike CISC arhitekture su: baziran je na sklopovlju računala, sadrži složene multiciklusne instrukcije, LOAD i STORE funkcije su uključene u instrukcije, koristi manje tranzistora, upotrebljava Harvardski model memorije.

RISC arhitektura koristi visoko optimizirani set instrukcija, koristi se na raznim platformama, od mobilnih uređaja i tableta do superračunala. Osnovne karakteristike RISC arhitekture su: jednostavne instrukcije, instrukcije imaju fiksnu duljinu i stoga je pogodan za pipeline, registri nemaju samo jednu namjenu, sadrži velik broj registara, LOAD i STORE funkcije su odvojene.

Ako želimo pomnožiti dva broja, prvi broj je spremljen na memorijskoj lokaciji 2:3, a drugi na memorijskoj lokaciji 5:2, a rezultat želimo spremiti na lokaciju 2:3. CISC procesor će pripremiti instrukciju koju ćemo nazvati „MULT“. Pri izvršavanju, ova instrukcija učitava dvije vrijednosti u odvojene registre, množi brojeve u izvršnoj jedinici i zatim pohranjuje rezultat u registar. Dakle, CISC procesor će ovaj zadatak izvršiti sa samo jednom instrukcijom:

```
MULT 2:3,5:2
```

Instrukcija MULT je složena instrukcija kojoj nije potrebno da programer zove funkcije učitaj i pohrani.

RISC procesori koriste samo jednostavne naredbe koje se mogu izvršiti u jednom vremenskom ciklusu. Mult naredba će u RISC procesoru biti podijeljena na 3 naredbe: LOAD, koja pomiče podatke iz glavne memorije u registre, PROD, koja pronalazi rezultat množenja unutar registara i STORE, koja pomiče podatke iz registara u glavnu memoriju. Da bi pomnožio dva broja programer sada mora ispisati 4 linije koda:

```
LOAD A, 2:3
```

```
LOAD B, 5:2
```

```
PROD A, B
```

```
STORE 2:3, A
```

Pošto svaka instrukcija u RISC arhitekturi traje samo jedan vremenski ciklus, cijeli program će se izvršiti u otprilike isto vremena koliko traje MULT naredba u CISC arhitekturi. Ovakve jednostavnije naredbe zahtijevaju manje tranzistora i pošto naredbe traju otprilike jednako moguć je pipeline.

5. PROTOČNA STRUKTURA (PIPELINE)

Pipelining je tehnika kojom procesor može istovremeno raditi na više programskih instrukcija i tako maksimalno iskoristiti svoje resurse da bi ubrzao svoj rad. Svaka instrukcija koja se istovremeno izvršava je u drukčijoj fazi obrade, npr. ako se jedna instrukcija izvršava druga se dekodira. Izvršavanje instrukcija se dijeli na faze:

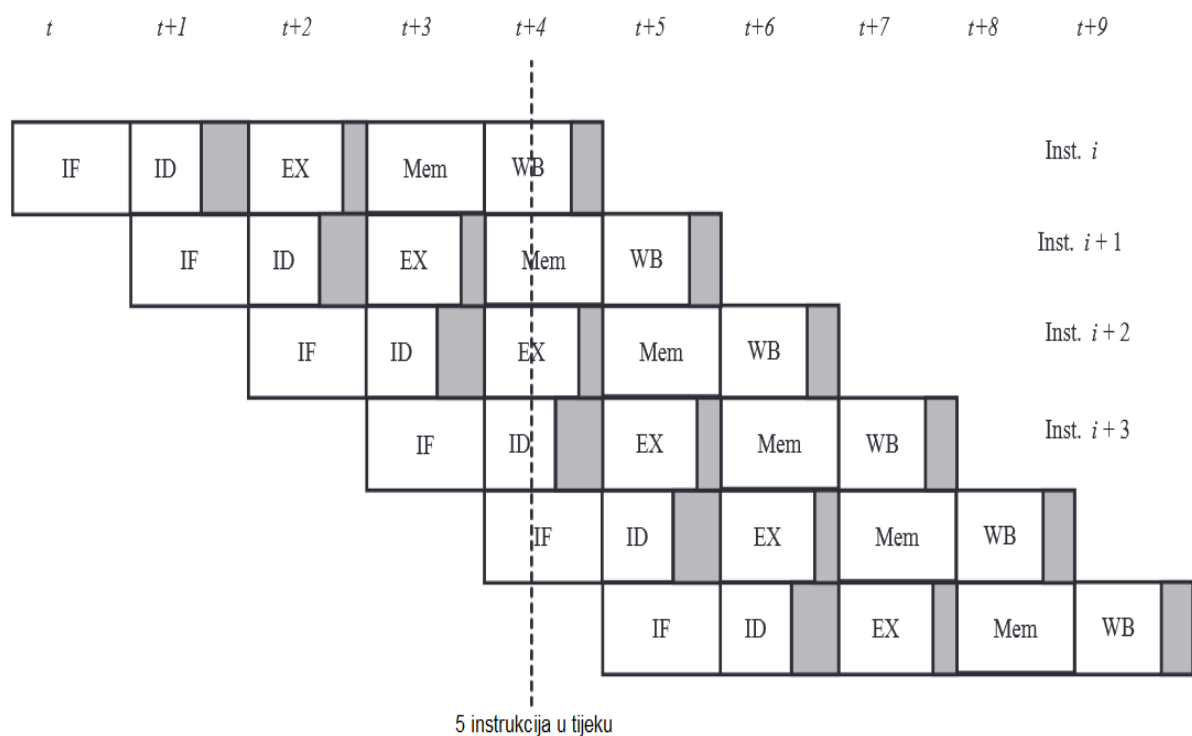
IF (Instruction fetch) – Instrukcija se dohvaća iz memorije.

ID (Instruction decode) – Instrukcija se dekodira i određuje se njezin tip.

EX (Execution) – Izvršavanje instrukcije

Mem (Memory access) – Pristup memoriji

WB (Writeback) – Pohrana rezultata u registar



Slika 5: Protočna struktura (pipeline) (Izvor: J. L. Baer, 2010)

Pipeline se sastoji od proizvoljnog broja razina a svaka razina izvodi jednu fazu. U pipeline-u sve faze zauzimaju otprilike isto, ali ne i točno isto vremena. Buffer (registar) između faza postoji zbog tih malih vremenskih razlika. Svaki posao mora proći kroz sve faze obrade čak i ako tom poslu nije potrebna ta faza.

Na slici je prikazan primjer izvršavanja sekvencijalnog programa u pipeline-u.

Pipeline je pun u trenutku $t+4$ kada se 5 naredbi izvršava istovremeno. Sivi dio znači da su faze ID, EX i WB završile svoje izvršavanje prije faza IF i Mem. (J.L.Baer 2010.) Opasnosti koje ometaju rad pipeline-a su strukturalni, podatkovni i upravljački hazardi. Oni izazivaju zastoje u radu, a duljina tih zastoja ovisi o duljini pipeline-a, što je pipeline duži, duži je i zastoj.

Strukturalni hazardi se događaju kada faze međusobno konkuriraju za resurs ili kada registri između faza preplave, odnosno nemaju više mjesta u memoriji. (kovač)- npr. ako koristimo kod naredbe „store“ tijekom izvođenja, instrukcija zahtijeva korištenje sabirnice za spremanje podataka, ali u isto vrijeme u pipeline-u se treba dohvaćati kod iduće naredbe iz memorije koji također zahtijeva korištenje sabirnice, a zamislimo da u našem slučaju arhitektura pristupa memoriji ne može omogućiti dva pristupa memoriji istovremeno. U tom slučaju dolazi do strukturalnog hazarda. Da bi procesor nastavio normalan rad strukturalni hazard se prvo mora riješiti. U ovom slučaju procesor će odgoditi dohvat sljedeće instrukcije dok se instrukcija „store“ ne izvede i tada će nastaviti s normalnim radom.

Podatkovni hazardi se događaju kada instrukcija zavisi o rezultatu obrade prijašnje instrukcije. Kontrolne opasnosti se događaju kada kontrolni tok nije sekvencijalan. Rješavanje ovih hazarda je vrlo zahtjevno. Neki procesori rješavaju ovaj problem ubacivanjem stanja čekanja, ali time mnogo gube na efikasnosti procesora. Napredniji procesori koriste razne tehnike kao „result forwarding“ što znači da su u procesoru interno povezani izlaz iz aritmetičko logičke jedinice na ulaz u aritmetičko logičku jedinicu čime se dobiveni rezultati mogu koristiti već u sljedećoj naredbi. Ovaj problem se, osim na razini procesora, može riješiti i na razini alata za prevođenje. Napredni prevoditelji koji znaju način funkcioniranja pipeline-a za koji se generira kod mogu uočiti potencijalnu međusobnu ovisnost naredbi. Kada to uoče prevoditelji pokušavaju, ako je to moguće, promijeniti redoslijed naredbi kako bi se izbjegla ovisnost.

Upravljački hazardi se događaju kada naredba koja se nalazi u pipeline-u i spremna je za izvršavanje, nije naredba koja se treba izvesti. To se događa kod naredbi grananja u slučajevima kada je procesor već učitao sljedeću naredbu, ali zbog grananja program mora nastaviti svoje izvođenje na nekoj drugoj adresi.

6. CACHE MEMORIJA

U računalu razlikujemo više vrsta memorije: RAM (koji se dijeli na statički i dinamički), ROM, cache ili priručne memorije i registri.

Memorije se razlikuju i po metodama pristupa podacima.

Po sekvencijalnom pristupu memorija je organizirana u jedinice podataka i pristup se mora odvijati sekvencijalno. Pohranjene informacije o adresama služe za odvajanje jedinica podataka i za vraćanje podataka. Upotrebljava dijeljeni čitaj-piši mehanizam. Magnetske trake koriste ovaj pristup.

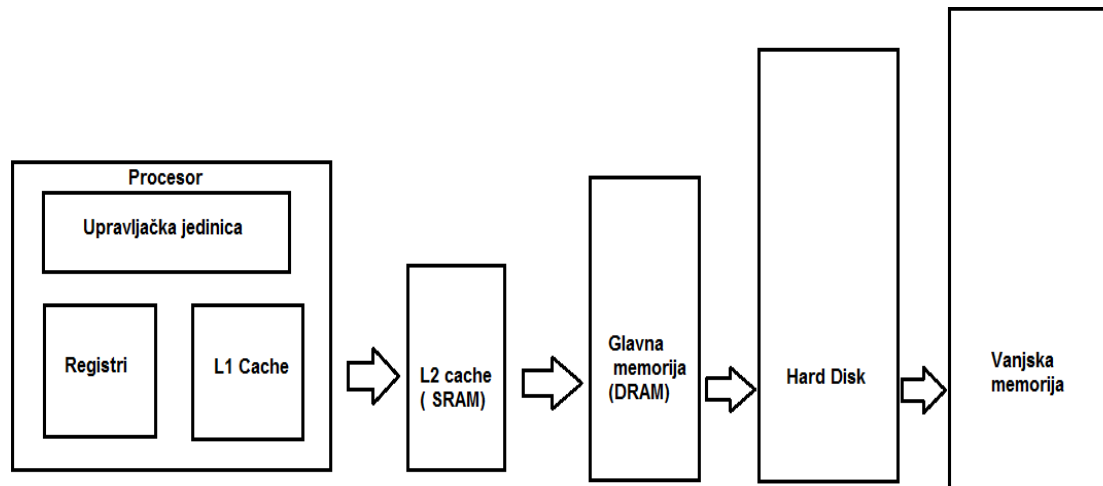
Direktni pristup također koristi čitaj-piši mehanizam, ali jedinice podataka imaju jedinstvenu adresu na osnovi svoje fizičke lokacije. Pristup podacima se vrši direktno. Diskovi koriste direktni pristup.

Kod nasumičnog pristupa svaka lokacija u memoriji ima svoj jedinstveni mehanizam za adresiranje. Vrijeme potrebno za pristup nekoj lokaciji je neovisno o prijašnjim podacima i konstantno je. Dakle, svakoj lokaciji se može nasumično pristupiti i direktno ju adresirati. Glavna memorija i neke cache memorije koriste nasumični pristup.

Asocijativni pristup je tip nasumične memorije koja ne pretražuje memorijsku lokaciju putem adrese već po sadržaju. Cache memorije mogu upotrebljavati ovaj pristup. (D.S. Martin 2012.)

Mikroprocesori su danas mnogo sofisticiraniji nego što su bili prije 30 godina. Tada je pristup memoriji bio tek malo sporiji od pristupa registrima. Dogodio se nesrazmjer u povećanju brzine rada procesora i memorije, te je danas frekvencija jezgre mikroprocesora mnogo veća od memorijske. Naravno da je moguće povećati brzinu dohvata podataka iz memorije, ali je to preskupo i neisplativo. Zato je osmišljena hijerarhijska organizacija memorije koja se pokazala kao idealno rješenje ovog problema. Cache memorija je memorija male veličine, obično tisućiti dio glavne memorije. Npr., ako je glavna memorija 4GB cache će biti 4MB, koja uvelike ubrzava rad procesora. Cache memorija je u današnjim mikroprocesorima podijeljena na tri razine: L1, L2 i L3 tako da je L1 najmanja memorija, obično oko 32KB, a L3 najveća. L1 cache je podijeljen na cache za spremanje programskih instrukcija i na cache za spremanje podataka. Intelov procesor i7 prve generacije imao je L1 podatkovni

cache veličine 32KB + 32KB instrukcijski, L2 cache veličine 256KB i L3 cache veličine 8MB.



Slika 6: Memorijska hijerarhija računala

Slika prikazuje hijerarhiju memorije u računalu. Najsporije memorije, koje su ujedno i najjeftinije su vanjske memorije, zatim hard diskovi, RAM memorija, Cache i na kraju najbrža i najskuplja memorija su registri procesora.

„Procesoru treba u prosjeku oko 40 ns za dobiti podatke iz RAM-a, odnosno 100 ciklusa za procesor od 2.5 GHz a procesor radi u ciklusima od 2 ns i stoga gubi jako puno vremena na dohvat podataka iz RAM-a.“ (J.L. Baer 2010.)

Podatci između cache memorije i procesora idu mnogo brže od protoka podataka između glavne memorije i procesora. Cache memorije su skupe za izradu i zbog toga nisu veće. One su napravljene da bi kombinirali pristup memoriji koji je vrlo brz i skup s glavnom memorijom koja je velika, sporija i jeftinija. Cache memorija sadrži dijelove podataka iz glavne memorije. Kada procesor želi dohvatiti neku jedinicu podataka prvo se provjerava dali je tražena jedinica podataka spremljena u cache memoriji. Ako je spremljena u cache memoriji brzo se dostavlja procesoru na obradu. Ako tražena jedinica podataka nije u cache memoriji iz glavne memorije se učitava blok podataka u cache memoriju i procesoru se onda dostavlja tražena jedinica podataka. (W. Stallings, 2012.)

„Kada procesor traži podatak koji je već spremljen u cache dolazi do cache hit-a, odnosno cache pogotka. Kada traženi podatak nije u cache memoriji dolazi do cache miss-a, odnosno cache promašaja. Performanse cache memorije ovise o omjeru

cache promašaja i pogodaka.“ (S. D. Burd,2015.)

7. POTPORA OPERACIJSKOG SUSTAVA

Kako su se s vremenom razvijale hardverske komponente računala softver je taj trend morao pratiti dodavanjem novih odlika, rješavanjem većih problema s većom točnošću i rješavanjem prije nerješivih zadataka. Visoko sofisticirani programski jezici omogućuju programerima brže razvijanje kompleksnih programa koji rezultiraju bržim radom računala. Moderni softverski sustavi se oslanjaju na više slojeva softvera za izvršenje programa. Ti slojevi uključuju operacijske sustave, virtualne strojeve i prevoditelje. U ovom poglavlju ću objasniti potporu operacijskog sustava radu računala.

Operacijski sustav možemo definirati kao skup programa koji upravlja kontrolom korištenja resursa računalnog sustava kao što su procesori, glavna memorija, vanjske memorije i ulazno/izlazna jedinica. On djeluje kao posrednik između korisničkih programa i sklopovlja računala. Na korisnički zahtjev operacijski sustav obavlja upisivanje programa u unutarnju memoriju pomoću ulaznih jedinica, rezervira unutrašnji memorijski prostor, kontrolira rad programa te rješava probleme vezane uz prekide programa. Operacijski sustav rješava i probleme multiprogramiranja i multiprocesiranja.

7.1 Procesi

Osnovni zadatak operacijskog sustava je upravljanje procesima i stvaranje uvjeta za njegov rad. Između ostalog, operacijski sustav mora osigurati spremnički prostor i procesor u skladu s modelom računala. Unutar procesa mora postojati barem jedna instrukcijska dretva. Operacijski sustav mora dodijeliti resurse procesima, omogućiti procesima razmjenu informacija, štititi resurse jednog procesa od drugog i omogućiti sinkronizaciju između procesa. Proces je instanca programa u izvršavanju. Proces možemo promatrati kao entitet koji se sastoji od niza dijelova. Dva najbitnija elementa procesa su programski kod i set podataka povezanih s tim kodom. Još neki elementi procesa su identifikator, stanje, prioritet, programski brojač, pokazivač memorije, informacije o ulazu/izlazu.

Stvaranje procesa

Do stvaranja procesa vode dva najučestalija razloga: novi serijski (batch) posao, interaktivni login, stvara ga operacijski sustav da bi pružio uslugu i poziva ga postojeći proces. U ova dva slučaja operacijski sustav je odgovoran za stvaranje novog procesa. Operacijski sustav može stvoriti proces umjesto aplikacije, npr. ako korisnik želi printati neku datoteku, operacijski sustav može stvoriti proces koji će izvršiti taj zahtjev. Proces također može stvarati nove procese. Proces koji je stvorio novi proces nazivamo proces roditelj, a novi proces nazivamo proces dijete.

Uništavanje procesa

Procesi mogu završiti na više različitih načina. Normalno završava kada poziva i obavještava operacijski sustav da je proces gotov s izvršavanjem. Mogući su i sljedeći razlozi za uništenje procesa: ako se proces predugo izvršava, traži više memorijskog prostora nego mu sustav može dodijeliti, pokušava pristupiti memorijskoj lokaciji za koju nema dopuštenje, pokušava upotrijebiti resurs za koji nema dopuštenje, pokušava zabranjen izračun, kao što je dijeljenje s nulom, ili pokušava spremiti veće brojeve nego što sklopovlje dopušta, ako predugo čeka da se dogodi neki događaj da bi nastavio s izvršavanjem, ako se dogodi greška u ulazno/izlaznim operacijama, netočna naredba, ako je dio podataka pogrešnog tipa, ako se uništi proces roditelj operacijski sustav može uništiti i sve procese potekle od tog procesa, proces roditelj može podnijeti zahtjev za uništenje svojih procesa djece.

7.2 Dretve

Dretva predstavlja oblik paralelizacije unutar procesa. Svaki proces sadržava jednu ili više dretvi koje se izvršavaju, kao i procesi, prividno paralelno.

Višedretvenost se odnosi na sposobnost operacijskog sustava da podržava višestruka i istovremena izvršavanja unutar jednog procesa. Neki operacijski sustavi, kao npr. MS-DOS, koriste jedno dretveni pristup, a drugi, kao neke izvedbe UNIX-a, podržavaju više korisničkih procesa, ali podržavaju samo jednu dretvu po procesu. Windows, Solaris i mnoge novije verzije UNIX-a koriste višestruke procese, od kojih svaki proces podržava više dretvi. Unutar procesa može biti jedna ili više dretvi koja se sastoji od stanja izvršavanja, objašnjenja dretve, izvršnog stoga, statičke memorije za varijable i pristupa memoriji i resursima svog procesa koji se dijeli sa svim ostalim dretvama u tom procesu. Prednost upotrebe dretvi je ta da treba mnogo

manje vremena za stvaranje nove dretve u postojećem procesu nego za stvaranje cijelog novog procesa. Također, potrebno je manje vremena za uništenje dretve od procesa. Dretve poboljšavaju komunikaciju između različitih programa u izvršavanju. Za komunikaciju između neovisnih procesa u većini operacijskih sustava je potrebna intervencija jezgre. Pošto dretve unutar istog procesa dijele memoriju i podatke, mogu komunicirati bez uplitanja jezgre. Još jedna prednost dretvi je brže mijenjanje između dretvi nego između procesa. Iz ovih prednosti možemo zaključiti da kad bi trebali implementirati neku aplikaciju ili funkciju kao set povezanih jedinica, mnogo je efikasnije to napraviti kao skupinu dretvi nego kao skupinu procesa. Kao primjer aplikacije koja bi imala koristi od upotrebe dretvi možemo navesti podatkovni server. Kako dolaze zahtjevi za podacima, stvara se nova dretva. Pošto serveru dolazi mnogo zahtjeva za podacima, mnogo dretvi će se stvoriti i uništiti u kratkom vremenu. Ako je server izveden na višeprosorskom računalu, tada se više dretvi unutar jednog procesa mogu izvršavati istovremeno na različitim procesorima. Također pošto procesi ili dretve u podatkovnom serveru moraju dijeliti podatke i stoga koordinirati svoje djelatnosti, brža je upotreba dretvi i zajedničke memorije od procesa i slanja signala za tu koordinaciju.

Osnovna stanja dretve su: izvršavanje, spreman i blokiran. Za promjenu tih stanja koriste se četiri operacije: stvaranje, blokiranje, deblokiranje i završetak.

Kod stvaranja se stvori novi proces, a s njim se stvori i dretva za taj proces. Stvorena dretva tada može stvoriti novu dretvu unutar istog procesa. Operacija blokiranja se koristi kada dretva za nastavak rada treba čekati neki događaj. Procesor tada kreće s izvršavanjem druge spremne dretve u istom ili različitom procesu. Kada se dogodi događaj zbog kojeg je dretva blokirana, dretva se iz blokiranog stanja mijenja u spremno stanje. Kada se dretva izvrši, njezino stanje i stog se dodjeljuju drugoj dretvi.

Dretve možemo podijeliti na korisničke dretve i dretve unutar jezgre. Korisničke dretve se stvaraju i upravljaju pomoću dretvene knjižnice koja je pokrenuta u korisničkom prostoru procesa. Korisničke dretve su vrlo učinkovite jer im nije potrebna sklopka za mijenjanje stanja da bi se mijenjale između dretvi. U istom trenutku može se izvršavati samo jedna korisnička dretva unutar procesa i ako je jedna dretva u blokiranom stanju cijeli proces je blokiran. Dretve unutar jezgre su dretve unutar procesa kojima upravlja jezgra operacijskog sustava. Više dretvi unutar

istog procesa se mogu izvršavati paralelno na mikroprocesoru i blokiranje dretve ne blokira cijeli proces. Da bi se mijenjalo s jedna dretve na drugu, potrebna je sklopka za mijenjanje stanja. (W. Stallings, 2008)

7.3 Prekidi

Prekidi služe da prekinu normalno sekvencijalno izvršavanje procesora u svrhu povećanja iskoristivosti procesora. Razlikujemo četiri klase prekida: programski prekid, prekid brojača, ulazno/izlazni prekid i prekid izazvan kvarom sklopovlja.

Programske prekide stvara neko stanje koje se dogodilo kao rezultat izvršavanja instrukcije, kao npr. dijeljenje s nulom, pokušaj izvršavanja nedozvoljene instrukcije ili referenca na lokaciju izvan dozvoljenog memorijskog prostora.

Prekid brojača stvara brojač unutar procesora. Ovo dozvoljava operacijskom sustavu izvršavanje određenih funkcija redovno.

Ulazno/izlazni prekid stvara ulazno/izlazni upravljač da bi javio normalno izvršenje operacije ili grešku.

Kvar sklopovlja kao npr. kvar napajanja ili greška u memoriji, također može stvoriti prekid.

Pojava prekidnog signala prebacuje procesor u sustavski način rada koji se naziva i jezgreni način rada. Dretva koja je pojavom prekidnog signala prekinuta najčešće izvodi neki korisnički posao i zato kažemo da se ona obavlja u korisničkom načinu rada. Procesor pri prelasku u jezgreni način rada privremeno onemogućava daljnje prekidanje, djeluje na adresni dio sabirnice tako da adresira odvojeni dio spremnika i aktivira drugi sustavski registar kazaljke stoga čime se omogućava ostvarenje sustavskog stoga. Procesor zatim pohranjuje trenutačni sadržaj programskog brojača na sustavski stog i u programski brojač učitava adresu na kojoj počinje potprogram za obradu prekida. Dakle, prekidni sustav djeluje kao sklopovski izazvan poziv potprograma. Procesor u svojim registrima ima sačuvane podatke o kontekstu dretve koja je u izvođenju. Kontekst dretve se čuva da bi se, nakon prekida, dretva mogla nastaviti izvršavati. Taj kontekst se pohranjuje na sustavski stog na koji je već procesor automatski pohranio adresu programskog brojila. U instrukcijskom skupu procesora postoje posebne instrukcije za pohranjivanje registara na stog. Nakon što je potprogram obavio svoj posao treba ponovno pokrenuti prekinutu dretvu

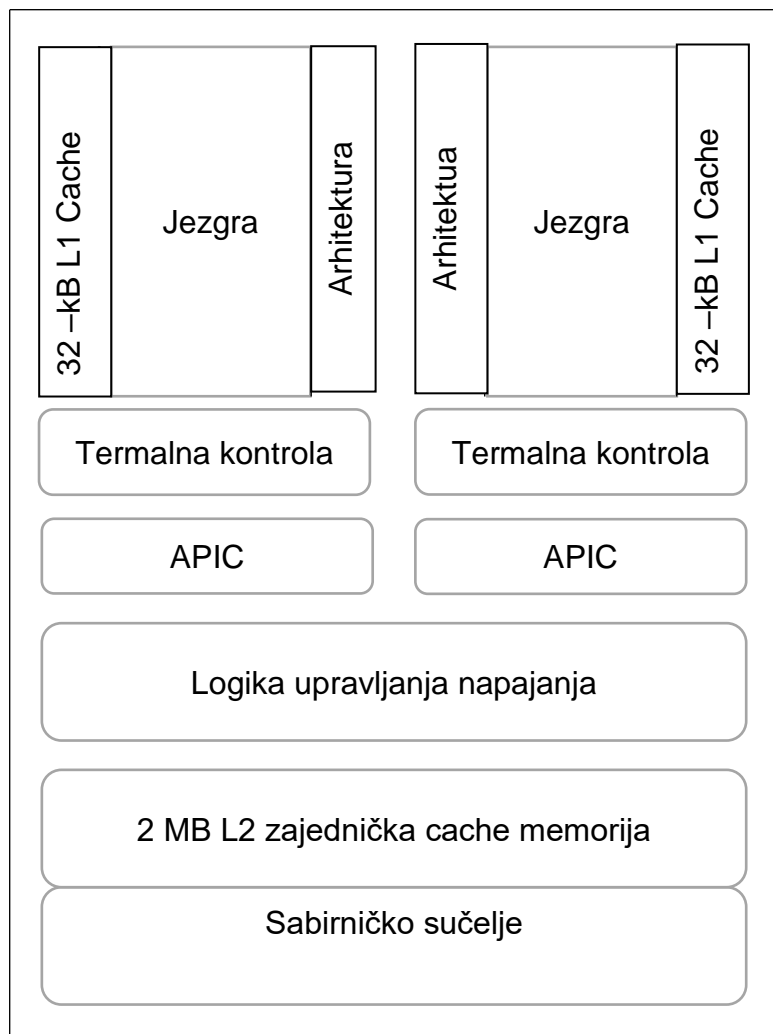
korisničkog programa. To se obavlja tako da se sa sustavskog stoga instrukcijama vraća sadržaje registara. Nakon toga na stogu ostaje samo sadržaj programskog brojača. Prije vraćanja tog sadržaja u procesor posebnom instrukcijom se procesoru omogućuje prekidanje. Ta instrukcija djeluje tako da će prekidanje biti moguće tek nakon što se u procesor vrati sadržaj programskog brojača. Ta instrukcija mora prebaciti adresiranje u korisnički adresni prostor i aktivirati korisnički registar kazaljke stoga. Procesor će u ovom načinu rada izvoditi dvije dretve, jednu dretvu u korisničkom načinu rada i drugu dretvu u jezgrenom načinu rada procesora. Te dvije dretve će se naizmjenice izvoditi.

Nedostatak ovakvog rješenja podsustava za prihvatanje prekida je da je tijekom cijele obrade prekida zabranjeno svako drugo prekidanje. To znači da neki drugi prekid koji zahtijeva trenutnu obradu ne može biti prihvaćen ako traje obrada nekog drugog manje važnog prekida koji se ranije pojavio. Taj nedostatak se može otkloniti programskim rješenjem gdje su prekidi razvrstani po prioritetu. Prekid s većim prioritetom ima prednost pri obradi.

8. PRIMJERI ARHITEKTURE INTELLOVIH MIKROPROCESORA

8.1 Intel Core Duo

Intel Core Duo predstavljen je 2006. godine. U svoje vrijeme je značio velik napredak i nasljednik je Pentium generacije procesora.



Slika 7: Intel Core Duo procesor (W. Stallings, 2012)

Na slici 7 predstavljena je osnovna struktura mikroprocesora Intel Core Duo. Najvažniji dio su dva x86 superskalarna procesora. Svaka jezgra ima svoju L1 cache memoriju za instrukcije i za podatke veličine 32-kB. Svaka jezgra ima svoju neovisnu upravljačku jedinicu za toplinu. Uloga kontrolne jedinice za toplinu je da kontrolira toplinu i drži je unutar zadanih granica kako bi se maksimizirale performanse procesora i kako bi se smanjila mogućnost kvara. Softver preko dodijeljenih registara

javlja kolika je temperatura u jezgri. Ako je temperatura previsoka kontrolna jedinica smanjuje takt rada te jezgre radi dovođenja temperature u optimalni stupanj.

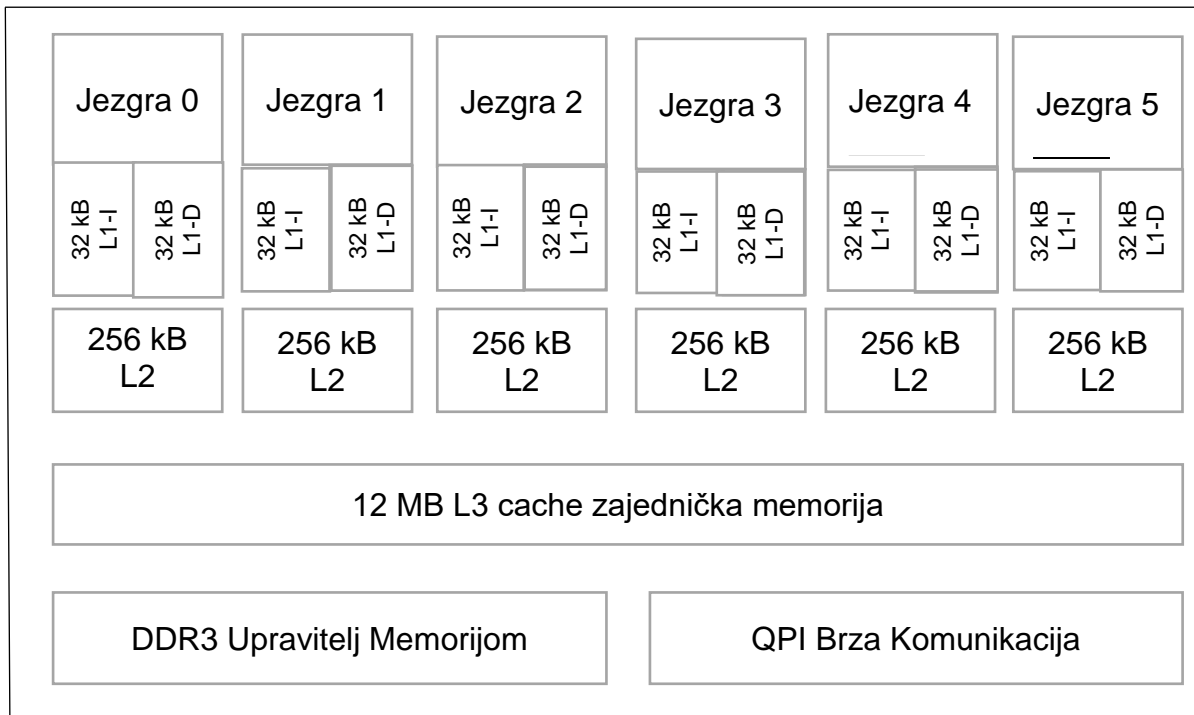
Sljedeći ključan element je napredni programabilni upravljač prekida (APIC). APIC može izazvati međuprocesorske prekide koji omogućuju procesu da prekine bilo koji drugi procesor. APIC prihvaća ulazno/izlazne prekide i upućuje ih na odgovarajuću jezgru. APIC sadrži brojač kojeg operacijski ustav može podesiti da stvori prekid u lokalnoj jezgri. Logika upravljanja energijom je odgovorna za, kad je to moguće, smanjivanje potrošnje električne energije. Prati se aktivnost procesora i temperatura i u skladu s tim se podešava voltaža i potrošnja energije.

Core Duo čip ima dijeljeni L2 cache veličine 2-MB. Cache druge razine se može dinamično dodjeljivati, to znači da u nekom trenutku jedna jezgra može imati na raspolaganju 100 % prostora na L2 cache.

Sabirničko sučelje upravlja protokom podataka između procesora i memorije.

8.2 Intel Core i7-990X

Intel Core i7-990X je procesor tvrtke Intel predstavljen 2008. godine. Sadrži 4 SMT procesora arhitekture x86 i 6 jezgri. Frekvencija rada procesora je 3.46 GHz i može se podesiti na 3.73 GHz. Da bi se povećala učinkovitost cache memorije Intel koristi tehniku predviđanja koja proučava pristup memoriji i pokušava ispuniti cache s podacima koji će vjerojatno biti zatraženi. Intel Core i7-990X podržava dva tipa vanjske komunikacije s ostalim čipovima. Maksimalna dozvoljena temperatura u procesoru je 67.9 °C. Veličina najmanjih komponenti na čipu je veličine 32 nanometra, a procesor ima snagu 130 W. Ovaj procesor koristi DDR3 tip memorije. Najveća brzina učitavanja ili čitanja podataka je 25.6 GB/s.



Slika 8: Intel Core i7-990X procesor (W. Stallings, 2012)

Na slici 8 prikazana je osnovna struktura mikroprocesora Intel Core i7-990X. Procesor se sastoji od 6 jezgri. Svaka jezgra ima dodijeljenu svoju L1 cache za instrukcije i L1 cache za podatke veličine 32 kB, L2 cache veličine 256 kB i dijeljenu L3 cache memoriju veličine 12-MB. Upravljač memorijom DDR3 se nalazi na čipu i podržava tri kanala širine 8 bitova, a cijela sabirnica je široka 192 bita za združen protok podataka do 32 GB/s. Intel Core i7-990X je jedan od prvih procesora koji koristi QPI način prijenosa podataka koji je zamijenio prednju sabirnicu u prijašnjim procesorima. QPI je razvila tvrtka Intel kako bi se povećala skalabilnost i širina protoka podataka između procesora i ulazno/izlaznih uređaja. Brzina protoka podataka je 6.4 GT/s.

9. UNAPREĐENJA PERFORMANSI MIKROPROCESORA

Postoji više tehnika za unapređenje performansi mikroprocesora. U nastavku teksta će biti navedene i objašnjene neke od njih.

Paralelizam na razini instrukcija je jedna od tehnika koja se koristi za poboljšanje performansi procesora. Osim toga, može služiti i za poboljšanje energetske učinkovitosti sustava. On omogućuje izvršavanje više od jedne instrukcije u isto vrijeme. Današnji procesor ima superskalarnu arhitekturu, što znači da koristi više paralelnih izvršavanja instrukcija u isto vrijeme. To se naziva „pipelining“ ili protočna struktura.

Paralelizam na razini dretvi paralelno obrađuje instrukcije različitih dretvi. Dretve koriste iste memorijske registre i cache memoriju i zato procesor koji podržava dvije dretve ne može imati duplo bolje performanse od procesora jednake snage koji podržava jednu dretvu, ali može ih poboljšati do 30%.

Iako broj tranzistora na čipu i dalje prati Mooreov zakon, postaje sve teže poboljšavati performanse sekvencijalnih procesora. Povećanje radnog takta stvara poteškoće zbog eksponencijalnog povećanja potrošnje energije i zagrijavanja procesora. Odgovor industrije na stagniranje napretka je predstavljanje višejezgrenih procesora. Na čip se postavlja dva ili više procesora i spaja ih se putem dijeljene memorije. Dretve jedne jezgre tada dijele priručne (cache) memorije za instrukcije L1 i za podatke L2 a jezgre dijele cache memorije druge razine L2 i treće razine L3. Procesori djeluju neovisno, ali koriste istu memoriju. Takve arhitekture dijeljene memorije se dijele u dvije skupine: UMA i NUMA.

Uniform Memory Access (UMA) koristi Symmetric Multiprocessor (SMP) sustave. Koriste se identični procesori i brzina pristupa memoriji je jednaka. Cache memorije su povezane, što znači da ako jedan procesor promijeni lokaciju nekog podatka u memoriji, svim ostalim procesorima je ta promjena odmah dostupna.

Non-Uniform Memory Access (NUMA) je često izveden kao dva ili više međusobno povezana SMP procesora. U NUMA arhitekturi jedan procesor može direktno pristupiti memoriji drugog procesora, svi procesori nemaju jednaku brzinu pristupa memoriji i memorijski pristup preko poveznice je sporiji.

Ova paralelna arhitektura nudi potencijalno rješenje stagniranja napretka performansi. Slaba točka ove arhitekture su problemi s kojima se susreću programeri. Paralelne programe je mnogo teže dizajnirati i kodirati od sekvencijalnih.

Transakcija je sekvenca koraka koje izvršava jedna dretva. Transakcije moraju biti serijalne, to znači da moraju prividno izgledati kao da se izvršavaju sekvencijalno. Jedan od najvećih izazova paralelnog programiranja je koordiniranje pristupa memoriji koju dijeli više dretvi, a transakcija memorija pomaže u tom izazovu. Realizacija transakcijske memorije(TM) može biti softverska, hardverska ili hibridna.

TM sustavi se oslanjaju na cache hijerarhiju računala kod prepoznavanja konflikata. TM pruža model za izoliranje međusobno konkurentnih računskih operacija jedne od druge i pomaže izbjeći brojne pogreške u paralelnom programiranju. Međutim, mnogi aspekti TM su u istraživanju, a ako se ti problemi ubrzo riješe TM će vjerojatno postati središnji stup paralelnog programiranja.

U nastavku je opisana mikroarhitektura tvrtke Intel iz 2017. godine.

Kaby Lake je poboljšana Intelova mikroarhitektura koja koristi elemente veličine 14 nm. Kaby Lake je razvijen u Izraelu i namijenjen je sedmoj generaciji Intelovih mikroprocesora. Posljednja verzija je Kaby Lake Refresh predstavljena u kolovozu 2018. godine.

Memorijska hijerarhija je strukturirana s 5 razina cache memorija i glavnom memorijom. Cache nulte razina L0 služi za pohranjivanje mikrooperacija, svaka jezgra ima svoj i sastoji se od 1536 mikrooperacija. Cache prve razine L1 je veličine 32 kB, svaka jezgra ima svoj i odvojene su memorije za instrukcije i podatke(L1 I i L1 D). Cache druge razine je veličine 256 kB i sadrži 1024 seta podataka. Cache treće razine L3 je veličine 2 MB i dijeli se na sve jezgre. Postoje i pričuvne cache memorije veličine 64 MB i 128 MB s brzinom čitanja i pisanja 32 B po ciklusu.

Kaby Lake mikroarhitekturu koriste moderni procesori predstavljeni u kolovozu 2017. Neki od njih su: i5-8250U, i5-8350U, i7-8550U i i7-8650U.

ZAKLJUČAK

Od 2003. godine do danas brzina radnog takta procesora se nije puno promijenila. Jedan od razloga je rasipanje energije koja ograničava povećanje frekvencije rada. Današnji pristup dobivanju boljih performansi je uporaba višejezgrenih procesora. Postoje mnogi izazovi u daljnjem razvoju višejezgrenih mikroprocesora. Tehnološki problemi su, kao što je već navedeno, rasipanje energije i brzina ulazno/izlaznih operacija zbog sve većeg broja zahtjeva. Problem mikroarhitekture je proizvodnja novih drevi velikom brzinom i sa što manje resursa, te da li je potrebno uvoditi promjene u instrukcijski skup procesora da bi se to napravilo. Treba razmotriti promjene u memorijskoj hijerarhiji i nova sklopovska rješenja. Tražit će se nova rješenja i u pitanju programskih jezika, ispravljanja greški i prevoditelja.

Napredak mikroprocesora ovisi i o drugim područjima znanosti. Pokretač napretka mikroprocesorske tehnologije u 21. stoljeću su novi oblici minijaturizacije i nove legure koje omogućuju veći broj tranzistora na čipu. Mooreov zakon još vrijedi, ali hoće li opstati je pitanje na koje se ne zna odgovor, ali je sigurno da su današnji mikroprocesori mnogo moćniji nego što je to bilo zamislivo u trenutku stvaranja računala prije pola stoljeća. Današnje globalizirano društvo zahtijeva sve veća i snažnija računala i sigurno je da nas očekuje još mnogo sjajnih tehnoloških napredaka.

Literatura

Knjige:

1. Baer, J. L. (2010.) *Microprocessor architecture: from simple pipelines to chip multiprocessors*, Cambridge: Cambridge University Press.
2. Budin, L. Golub M. Jakobović D. I Jelenković L. (2011.) *Operacijski sustavi*
3. Burd, S. D. (2015.) *Systems Architecture* [6 izd.], New Mexico: Cengage Learning.
4. Hennessy, J. Patterson, D. (2007.) *Computer Architecture A Quantitative Approach* [4izd.], San Francisco: Morgan Kaufmann Publishers.
5. Kovač, M. (2015.) *Arhitektura računala*, Zagreb : Fakultet elektrotehnike i računarstva.
6. Martin, D. S. (2012.) *Hardware and software techniques for scalable thousand-core systems*, Stanford: Stanford University.
7. Stallings, W. (2012.) *Computer Organization and Architecture* [9 izd.], New Jersey: Pearson Prentice Hall.
8. Stallings, W. (2008.) *Operating systems* [6 izd.], New Jersey: Pearson Prentice Hall.
9. Ribarić, S. (1999.) *Naprednije arhitekture mikroprocesora*, Zagreb : Element. [2 izd.], Zagreb: Element.

Internet:

1. www.intel.com (26.11.2017)
2. <http://www.znanje.org/knjige/computer/hardware/02/procesori.pdf> (20.11.2017)
3. <https://informatika.buzdo.com/s310-mikroprocesor-sabirnice.htm> (22.11.2017)
4. http://www.istrattech.hr/wp-content/uploads/2012/11/case2012_1.pdf (14.12.2017)
5. https://mavdisk.mnsu.edu/alleng/courses/EE%20613/Reading/DrDobbs_1.pdf (15.12.2017)
6. <https://cs.stanford.edu/people/eroberts/courses/soco/projects/risc/riscisc/> (27.11.2017)

Popis slika i tablica

Slike

Slika 1: Računalo: top level struktura	2
Slika 2: Mooreov zakon	6
Slika 3: Sklopovlje računala povezano sabirnicom	8
Slika 4: Osnovni koraci izvođenja naredbe	9
Slika 5: Protočna arhitektura (pipeline).....	16
Slika 6: Memorijska hijerarhija računala	19
Slika 7: Intel Core Duo procesor.....	25
Slika 8: Intel Core i7-990X procesor	27

Tablice

Tablica 1: Intelovi mikroprocesori 1970-te	5
Tablica 2: Intelovi mikroprocesori 1980-te	5
Tablica 3: Intelovi mikroprocesori 1990-te	5
Tablica 4: Intelovi mikroprocesori 2000-te	6
Tablica 5: Popis registara za osnovno računalo	11
Tablica 6: Osnovne instrukcije za prijenos podataka.....	12
Tablica 7: Osnovne aritmetičke instrukcije.....	12
Tablica 8: Osnovne logičke instrukcije.....	12
Tablica 9: Osnovne pretvaračke instrukcije	13
Tablica 10: Osnovne upravljačke instrukcije.....	13
Tablica 11: Osnovne ulazno/izlazne instrukcije	14

Sažetak

Gotovo sva današnja računala imaju više mikroprocesora, manja računala u obliku višejezgrenih procesora, a veća računala obično sadrže više procesora. Ciljevi ovog rada su: prikazati razvoj mikroprocesora od njihovog izuma 1971. do danas, objasniti njihovu arhitekturu i način rada, objasniti instrukcijski skup podataka i navesti primjere, objasniti memorijsku hijerarhiju računalnih sustava i predstaviti buduće izazove koje treba savladati u budućnosti razvoja mikroprocesora. U radu su navedene i tehnike koje se koriste u svrhu povećanja performansi mikroprocesora, što sve utječe na njih i kako se mjere. Objasnjena je potpora operacijskog sustava kroz procese, dretve i prekide. Na kraju su prikazane arhitekture novije generacije tvrtke Intel: Intel Core Duo i Intel Core i7-990X.

Ključne riječi: povijest, razvoj, mikroprocesor, tehnologija

Summary

Nowadays almost every computer is made of more than one microprocessor, smaller computers in the form of multicore processors, and the bigger ones usually contain several processors. In this paper my aim is to: describe the development of microprocessors since their invention in 1971 up to present day, explain the architecture and the way they operate, explain the memory hierarchy of operating systems and present future challenges to overcome in the further development of microprocessors. Also listed are the techniques used to enhance microprocessor performances, ways they can be influenced and how they are measured. The paper further explains the support for the operating system through processes, threads and interruptions. The end discusses new generation architectures developed by Intel corporation: Intel Core Duo and Intel Core i7-990X.

Key words: history, development, microprocessor, technology