

1. UVOD

Računala su nas uvela u eru informatike. Stvorena jedva pred pedeset godina imaju ogroman utjecaj praktički na svaku pojedinu sferu čovjekova života. Što je priroda ovih uređaja? Kakav je princip njihova rada? Kako se interno programiraju? Kako se ostvaruje njihovo sučelje s vanjskim svijetom? Ovo su neka od pitanja na koja ćemo pokušati dati odgovore tijekom ovog kolegija. Vjerojatno najbolji način da se shvati kako neki uređaj radi je da se prođe kroz proces njegovog projektiranja. Iz tog razloga tijekom ovog kolegija razmatranja će se radije provoditi iz perspektive projektanta računala nego iz perspektive njegovog korisnika. Za savladavanje ovog kolegija može se preporučiti sljedeća literatura:

Literatura:

1. Vincent P. Heuring, Harry F. Jordan "Computer Systems Design and Architecture", Addison-Wesley, 1997. ISBN 0-8053-4330-X, cijena: 456 kn.

Ovo je knjiga profesora s University of Colorado prema kojoj su i pripremana ova predavanja

2. John L. Hennessy, David A. Patterson "Computer Architecture, a Quantitative Approach", Morgan Kaufmann Publishers INC., 1996, ISBN 1-55860-329-8.

Profesori sa Stanford i Berkeley University napisali su ovu knjigu u kojoj opisuju pristupe vrednovanja različitih računalnih arhitektura. Knjiga je namijenjena studentima koji već imaju dostatna predznanja iz problematike arhitekture računala te služi za produbljivanje stečenih znanja.

Digitalno računalo, obično nazvano računalo opće namjene, je vjerojatno jedan od najsloženijih uređaja koje je čovjek stvorio. Svako složeni sustav može se promatrati iz više perspektiva. Tako se npr. zgrada može promatrati s gledišta njene arhitekture, odnosno izgleda, ili s gledišta njene funkcionalnosti (veličine i iskoristivosti prostora). Raspored i veličina prostorija te raspored prozora i vrata zanimaju projektante sustava centralnog grijanja koji ima sasvim drugačiji pristup stanu ili zgradi od arhitekta ili korisnika stana.

Računalo se može promatrati s gledišta svoje sveukupne strukture i principa rada, što se obično smatra pod pojmom **arhitekture računala**. Računala se mogu i promatrati s gledišta korisnika ili s gledišta programera. Konačno mogu se promatrati na razini najniže logičke strukture, odnosno projektanta logičkih vrata koja tvore računalo.

Sva ova gledišta međusobno su vezana i isprepletena te su važna za razumijevanje cjelovite problematike vezane uz digitalna računala. Kroz ovaj kolegij naročita pažnja biti će posvećena sljedećim problemima:

- a) analiza s gledišta programera u strojnom (engl. *machine*), odnosno simboličkom (engl. *assembly*) jeziku,
- b) analiza s gledišta projektanta arhitekture računala,
- c) analiza s gledišta projektiranja logičkih struktura potrebnih za realizaciju funkcija procesora.

Ova tri značajna područja pomoći će nam da razumijemo složenu problematiku, projektiranja, analize rada i korištenja digitalnih računala.

U prvom poglavlju biti će objašnjeno što se podrazumijeva pod pojmovima: gledište programera u simboličkom jeziku, gledište projektanta arhitekture računala ili procesora, gledište projektanta logičkih sklopova. Nakon što budu objašnjeni ovi pojmovi opisati će se koji se dijelovi računala razmatraju sa pojedinog od navedenih gledišta, te koji su zadaci, ciljevi i alati koje koriste pojedinci koji rade na određenoj razini.

Kroz kratak povijesni pregled razvoja računala prikazati će se ono što je bilo najznačajnije tijekom razvoja računala i što je dovelo računala na današnju tehnološku i funkcionalnu razinu.

Navedeno će poslužiti kako bi se ustanovili trendovi u razvoju računala kao i trenutne znanstveno-stručne aktivnosti na ovom području.

Sljedeće poglavlje započinje analizom klasa računala i njihovih naredbi temeljenom na pristupu računalu s gledišta programera, tj. pogledu s gledišta arhitekture skupa naredbi (engl. *Instruction Set Architecture* skr. ISA). Ijedeće su razlike među računalima:

- ❑ prema vrsti i veličini spremnika unutar centralne procesne jedinice (engl. *Central Processing Unit* skr. CPU),
- ❑ struktura glavne memorije,
- ❑ način kako naredbe pristupaju operandima bilo u spremnicima bilo u glavnoj memoriji.

Slijedi klasifikacija naredbi: premještanje podataka, aritmetičko-logičke, upravljačke i grananja. Nadalje je opisana međusprega naredbi i sklopovlja, odnosno dan je prikaz različitih vrsta spremnika i memorijskih organizacija, te načina pristupa operandima.

Na osnovi provedenih razmatranja napraviti će se model jednostavnog RISC (*Reduced Instruction Set Computer*) računala tzv. SRC (*Simple RISC*) na razini ISA. Izbjegavanje složenih rješenja koja su prisutna kod današnjih RISC računala omogućava da se funkcionirane RISCa u potpunosti opiše neformalno odnosno tekstualno i pomoću slika. Za komercijalne RISCove za ovo bi bile potrebne čitave knjige.

Za formalan, odnosno precizan opis rada procesora koristi se različiti jezici, a mi ćemo za to koristiti RTN (engl. *Register Transfer Notation*). Svrha uvođenja formalnog jezika je jednostavan i jednoznačan opis funkcioniranja procesora. Ovakav pristup opravdan je iz sljedećih razloga:

- ❑ osnovno djelovanje CPUa je prijenos podataka (naredbi i operandima) među spremnicima,
- ❑ RTN omogućava jednostavan i jednoznačan opis strukture procesora i njegovo djelovanje,
- ❑ RTN omogućava direktno preslikavanje funkcija u sklopovlje (engl. *hardware*).

Analizirane metode adresiranja, odnosno pristupa podacima (naredbama i operandima) prikazane su pomoću RTNa.

Nakon što se razmotri gledište programera na arhitekturu računala, ovo poglavlje zaključiti će se projektiranjem logičkih sklopova za realizaciju potrebnih funkcija, odnosno pristupom s gledišta projektanta sklopovlja računala. Ova razmatranja objasniti će kako se ostvaruje prijenos podataka između spremnicima, te njihova obrada tijekom prolaska kroz ALU (*Arithmetic Logic Unit*), i to sve na razini elementarnih sklopova odnosno logičkih vrata. Prvi put u razmatranje se uzima i vremenski tijek ovih događaja (engl. *timing*). Prijenos podataka odvija se preko sabirnica (engl. *bus*) koji međusobno povezuju sklopove računala. Ovdje će se vidjeti različitost metoda projektiranja procesora od onih primijenjenih pri projektiranju klasičnih logičkih sklopova.

Svrha sljedećeg poglavlja je da se razumiju komercijalni procesori te da se shvati zašto su upravo na taj način projektirani. Razumijevanje kompromisa prihvaćenih tijekom postupaka projektiranja komercijalnih procesora važno je za razumijevanje tehnike projektiranja procesora koja se obrađuje u ovom kolegiju. Nakon što budu obrađeni konkretni procesori, studenti bi trebali biti u stanju bolje procijeniti važnosti pojedinih značajki procesora.

Slijedi usporedba cijene i performansi procesora (*price/performance*) što također važan faktor pri projektiranju. Posebna pažnja posvetiti će se razlikovanju dvaju osnovnih procesorskih arhitektura: procesori sa složenim skupom naredbi (engl. *Complex Instruction Set Computer* skr. CISC) i procesora sa smanjenim skupom naredbi (engl. *Reduced Instruction Set Computer* skr. RISC).

Slijedi neformalni opis ISA (*Instruction Set Architecture*) CISC procesora Motorola MC68000 uporabom teksta i slika, a potom i formalni opis uporabom RTN opisa osnovnih značajki. Ovim pristupom postiže se bolji opis procesora i studentima se pruža prilika da se bolje upoznaju s RTN opisom.

Poglavlje se završava opisom ISA RISC procesora SPARC.

Razumijevajući ISA analiziranih računala, te poznavajući alate koji opisuju arhitekturu procesora (RTN) u ovom poglavlju pristupa se postupku projektiranja procesora, projektiranju centralne procesne jedinice (engl. *Central Processing Unit* skr. CPU) i to s gledišta projektanta logičkih sklopova.

Na primjeru SRCa opisanog u drugom poglavlju opisati će se postupak projektiranja CPU-a. Cjelovit postupak projektiranja temeljiti će se na jedno-sabirničkom sustavu za povezivanje funkcionalnih cjelina, a spominjući pri tome i neka alternativna rješenja. Nadalje, postupak projektiranja, od mikroarhitekture postepeno će se proširiti preko projektiranja putova prijenosa podataka i njihovog logičkog rješenja, sve do projektiranja upravljačke jedinice. Postupak projektiranja potom će se proširiti skupom dodatnih rješenja. Tako da će se razmatrati alternativna rješenja temeljena na dvo- i trosabirničkim mikroarhitekturama procesora.

RTN koji se koristio za opis principa rada i ponašanja procesora koristan je i u postupku projektiranja putova prijenosa podataka. Pri tome treba uzeti u razmatranje i utjecaj samih značajki sklopova koji se koriste u realizaciji. Naime prilikom prijenosa podataka između spremnika koji imaju različita djelovanja, potrebno je i generirati upravljačke signale koji upravljaju prijenosom podataka.

Upravljačke signale generira upravljačke jedinice sa ciljem ispravnog prijenosa podataka između spremnika.

Kao proširenje upravljačkih funkcija obrađuju se i dvije dodatne značajne funkcije upravljačke jedinice a to su:

- dovođenje sustava u početno stanje (engl. *reset*),
- obrada iznimnih stanja (engl. *exception*) i prekida (engl. *interrupt*).

Sljedeće poglavlje posvećeno je tehnikama koje dovode do ubrzanja rada procesora. Cjevovodom (engl. *pipelining*) se uzima u obradu nova naredba iako prethodne nisu završene. Ova tehnika primijenjena je praktički kod svakog suvremenijeg procesora. Naravno ovakav pristup izvođenju naredbi postavlja i neke posebne zahtjeve na skup naredbi. U tu svrhu biti će analizirano kako pojedine klase naredbi utječu na projektiranje procesora koji koristi cjevovode. Prilikom implementacije složenijih naredbi, odnosno određeni redoslijedi izvođenja naredbi, može izazvati probleme u obradi. Zato u opisani i postupci za otkrivanje i otklanjanje mogućih problema. Ova problematika obrađivati će se na primjeru projektiranja cjevovoda za RISC procesor.

Osim cjevovoda postoje i druge mogućnosti ubrzanja izvođenja programa. Jedno rješenje je tzv. superskalarni procesor. Postoje i arhitekture koje istovremeno izvode više naredbi (engl. *multiple-instruction-issue architecture*). Ova računala imaju više cjelovitih jedinica za izvođenje što omogućava istovremenu obradu više naredbi. Operacije nad vektorima, za koje su ovi procesori posebno projektirani, omogućavaju istovremenu obradu većeg broja operanada.

Procesor s vrlo velikom dužinom naredbe (engl. *Very Long Instruction Word* skr. VLIW) također posjeduju više funkcionalnih jedinica u okviru jednog CPUa, ali su njihove naredbe značajno duže od onih kod konvencionalnih procesora. Ovim pristupom specificira se više različitih operacija u okviru jedne naredbe.

Ovo poglavlje zaključiti će se razmatranjima o projektiranju mikrokodirane upravljačke jedinice (engl. *microcoded control unit*) čime će se proširiti razmatranja iz prethodnog poglavlja gdje su obrađene metode povezivanja upravljačke jedinice i ostalih funkcionalnih jedinica procesora. Ovo je najpopularnija metoda realizacije upravljačke jedinice temeljena na pristupu programera njenom projektiranju. Upravljački signali se u obliku riječi pohranjuju u memoriji mikrokodirane upravljačke jedinice

Sljedeće poglavlje posvećeno je dijelu procesora koji provodi proračune i obradu. Aritmetičko-logička jedinica je srce računala u smislu da izvodi obradu na temelju koje je računalo i dobilo svoje ime. Prema svojoj funkciji ima veliki utjecaj na performanse računala (ne samo brzina izvođenja proračuna nego i točnost).

Kao što je za projektiranje upravljačke jedinice bio potreban formalan opis procesora pomoću RTNa, tako je i računalna aritmetika temelj projektiranja aritmetičko-logičke jedinice. U tom smislu biti će obrađeni različiti brojevi sustavi i pretvorbe baze (baza 10 u bazu 2 odnosno 16 koja se koristi u računalnoj aritmetici). Binarni brojevi i binarna aritmetika neophodni su prilikom definiranja transformacija potrebnih kako bi se učinkovito projektirala aritmetičko-logička jedinica. U ovim razmatranjima dostatna je osnovna aritmetika uz neke osnovne ideje konačne matematike. Posebna pažnja posvećena je sklopovskoj realizaciji

pojedinih aritmetičkih operacija s naglaskom na što učinkovitije (brže) izvođenje uz minimalnu sklopovsku realizaciju.

Aritmetika nad cjelobrojnim veličinama proširiti će se nadalje na identičan način i na aritmetiku nad realnim brojevima s konačnim ciljem projektiranja sklopova koji će izvoditi takve proračune.

Još jedna značajna funkcija ALU je proračuni uvjetnih grananja, odnosno skokova u izvođenju programa koji su posljedica rezultata aritmetičkih i logičkih operacija.

Ovo poglavlje zaključiti će se sklopovskim rješenjima ALUa.

U poglavljima koja obrađuju projektiranje procesora memorija se tretira kao jedinstveni blok za skladištenje podataka i naredbi ograničenog kapaciteta sa sposobnošću upisa i očitavanja sadržaja svake pojedine lokacije. U stvarnosti veličina, brzina, cijena, potrošnja i neki drugi čimbenici dodatno usložnjavaju pogled na memoriju računala. Navedeni čimbenici biti će detaljno analizirani u okviru ovog poglavlja.

Kao prvi korak objasniti će se koncept ispisne (engl. *Read Only Memory* skr. ROM) i ispisno/upisne memorije (engl. *Random Access Memory* skr. RAM) kao pojmovi koji opisuju memoriju kao što je kapacitet, vrijeme pristupa i sl. Nakon ovih uvodnih napomena pristupiti će se sklopovskom projektiranju memorije, polazeći prvo od jednostavne jednobitovne memorijske ćelije da bi se naknadno prešlo na memorijske integrirane sklopove, module pa i čitave ploče. Ova razmatranja biti će vođena s aspekta projektanta logičkih sklopova.

Slijedi prijelaz na pogled s aspekta projektanta arhitekture računala gdje se razmatra hijerarhijski koncept memorije. Hijerarhijski koncept memorije je kombinacija različitih komponenata sa svojstvom skladištenja podataka i naredbi koja čini sveukupni memorijski sustav računala. Različite memorijske komponente obično su poredane prema vremenu pristupa odnosno brzini od bržih prema sporijima. Danas je uobičajeno da se memorijski sustav računala sastoji od vrlo brze memorije (tzv. *Cache*) relativno manjeg kapaciteta 64kB –512kB, glavne memorije (engl. *main memory*) koju čini poluvodički RAM kapaciteta 16MB-256MB, te sekundarne koju čine magnetski diskovi, optički diskovi, magnetske trake vrlo velikog kapaciteta reda GB.

Slijedi uvođenje pojma virtualne memorije, odnosno memorijskog modela koji omogućava da procesor pristupa podacima i naredbama bilo gdje u memorijskom sustavu kao da se oni nalaze u samoj glavnoj memoriji.

Kako procesor pristupa memoriji u svakom ciklusu da dohvati naredbu i podatak te da ponovno uskladišti rezultat naredbe za brzinu rada računala značajno je da je vrijeme pristupa memoriji što kraće. Na nesreću vrijeme pristupa je obrnuto proporcionalno cijeni memorije pa jako brze memorije imaju relativno veliku cijenu. Zato brza memorije je manjeg kapaciteta i u njoj su uskladišteni dio naredbi i podataka koje će procesor najvjerojatnije koristiti. One koje nisu u brznoj memoriji već u radnoj memoriji procesor dohvata iz radne memorije ili prebacuje u brzu memoriju. Problemi koji se pri tome javljaju kao i njihovo rješavanje problematika je dijela ovog poglavlja.

Konačno razmatranja će se proširiti na cjeloviti memorijski sustav.

Iako je problematika povezivanja računala s vanjskim svijetom tzv. ulazno/izlazne operacije (engl. *Input/Output*) dotaknuta u prethodnim poglavljima u ovom dijelu predavanja pažnja će biti usredotočena na detaljniju analizu i projektiranje sklopovlja i programske podrške koji omogućavaju ostvarivanje i ove funkcije računala.

Procesor razmjenjuje podatke s vanjskim svijetom preko sustava linija tzv. sabirnice (engl. *bus*). Signali koji se aktiviraju na tim linijama moraju imati točno definiran vremenski slijed pojavljivanja, tzv. *timing*. Podaci se mogu prenositi paralelno (npr. 8 bita preko 8 linija) ili serijski (samo jedna linija prenosi podatke od procesora prema uređaju). Danas su uglavnom ovakve veze (tzv. *interface*) bilo serijske ili paralelne standardizirane (npr. serijski RS 232).

Kao najjednostavniji prijenos podataka između procesora i ulazno/izlaznog uređaja je programski upravljani prijenos koji često nije učinkovit. Njegovo poboljšanje je prekidno obavljanje ulazno/izlaznih operacija (engl. *interrupt driven I/O*) koje zahtjeva dodatne preinake u sklopovlju računala kao i izmjene u programskoj podršci.

Daljnje ubrzanje prijenosa podataka između U/I uređaja i računala je uvođenje direktnog pristupa memoriji (engl. *Direct Memory Access* skr. DMA). Procesor tada ne sudjeluje u prijenosu nego se podaci direktno prenose iz U/I uređaja ili međusklopa u memoriju računala i obratno. To zahtjeva opet dodatna sklopovska rješenja kao i programsku podršku.

Konačno svaki od U/I uređaja povezuje se s računalom preko određenih sklopova tzv. U/I međusklopova (*interface*) čija izvedba je ovisna o odabranom načinu prijenosa podataka kao i odgovarajućom programskom podrškom (engl. *device driver*).

Ovo poglavlje veza je između prethodnih poglavlja koje tretiraju tematiku rada i projektiranja dijela računala koje obavlja obradu (procesor + memorija) i uređaja koji povezuju računalu s vanjskim svijetom.

Periferni uređaji ili skraćeno periferije su veza računala s vanjskim svijetom. U ovom poglavlju obraditi će se uobičajeniji periferni uređaji kao što su magnetski diskovi, video ekran, pisač, tipkovnica, miš. Prvenstveno ćemo se usredotočiti na logičko ponašanje ovih uređaja ali opisati će se i njihov princip rada.

Iako naziv periferni može dovesti do pogrešnog shvaćanja da se radi o relativno nevažnim komponentama računala, u osnovi su oni vrlo važne komponente računarskog sustava. Danas je vjerojatno više inženjera zaposleno na projektiranju sofisticiranih perifernih uređaja nego na projektiranju računala opće namjene.

Problematika računalnih mreža obrađuje se kao posljednje poglavlje ovog kolegija. Prvo će biti definirani komunikacijski protokoli. Naime da bi dvije ili više jedinki mogli međusobno komunicirati moraju govoriti i razumjeti isti jezik. Isto vrijedi i za računala. Računala mogu komunicirati samo ako koriste isti komunikacijski protokol. Komunikacijski protokol ima više razina. Najniža fizička razine gdje spadaju spojni vodovi odnosno mediji preko koji se signali prenose, sklopovi koji generiraju i primaju signale, i najviša razine na kojoj korisnik specificira podatke koje želi prenositi i kome ih želi slati odnosno od koga primati.

Primjer je višeslojni otvoreni sustav međuveza OSI koji ima sedam različitih razina. Ovaj model određuje arhitekturu komunikacijske mreže, a ne predstavlja explicitni komunikacijski protokol.

RS-232 protokol predstavlja realizaciju dvoslojnog komunikacijskog sustava koji obuhvaća razinu pripreme podataka za prijenos i razinu fizičkog serijskog prijenosa između dva korisnika ili dvije točke. Podaci se prenose serijskom komunikacijom znak po znak. Znakovi su obično ASCII (*American Standard Code for Information Interchange*) kodirani.

Komunikaciju između dva računala potrebno je proširiti na više računala. Tako se formira mreža računala koja ukoliko je manjeg obima i odnosi se na manje područje (jedna organizacija u jednoj ili više bližih zgrada) se naziva lokalna mreža (engl. *Local Area Network* skr. LAN). Npr. na FESBu LAN obuhvaća oko 400 spojnih mjesta.

Ethernet razvijen od Boba Metcalfa iz Xeroxa 1970 predstavlja najpopularniji LAN. Ethernet omogućava spajanje do 1024 računala na jednu mrežu s brzinama prijenosa 10Mbps do 100Mbps.

Povezivanje računala ili LANova u širu mrežu (engl. *Wide Area Network* skr. WAN) je daljnje unapređenje komunikacija među računalima. Primjer je Internet, najveća i najsloženija komunikacijska mreža u svijetu s preko 4 miliona "host" računala do 1995. Računala na Internetu komuniciraju pomoću TCP/IP protokola. Svako računalo na mreži ima svoje ime odnosno jedinstvenu IP adresu preko koje joj pristupaju druga računala s mreže. Podatke koje jedno računalo (npr. iz Splita) šalje drugom (npr. u New York) mora proći složen put. Prijenos se ostvaruje preko usmjerivača (engl. *ruter*) koji proslijeđuju podatke od izvorišta prema odredištu. Pristup računalu u nekoj lokalnoj mreži ostvaruje se pristupom usmjerivaču te mreže koji ima svoj dio IP adrese, a potom samom računalu u mreži kojem pripada dodatni dio IP adrese.

Ova razmatranja okončati ćemo diskusijom o primjeni Interneta kao i trendovima u njegovom razvoju i istraživanjima koje se u tom smjeru provode.