

Napredne arhitekture mikroprocesora

Žepina, Luka

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Maritime Studies, Rijeka / Sveučilište u Rijeci, Pomorski fakultet**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:187:773206>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-07-27**



Sveučilište u Rijeci, Pomorski fakultet
University of Rijeka, Faculty of Maritime Studies

Repository / Repozitorij:

[Repository of the University of Rijeka, Faculty of Maritime Studies - FMSRI Repository](#)



**SVEUČILIŠTE U RIJECI
POMORSKI FAKULTET**

LUKA ŽEPINA

NAPREDNE ARHITEKTURE MIKROPROCESORA

ZAVRŠNI RAD

Rijeka, 2024.

**SVEUČILIŠTE U RIJECI
POMORSKI FAKULTET**

**NAPREDNE ARHITEKTURE MIKROPROCESORA
ADVANCED MICROPROCESSOR ARCHITECTURE**

**ZAVRŠNI RAD
BACHELOR THESIS**

Kolegij: Mikro i osobna računala

Mentor: izv. prof. dr. sc. Jasmin Čelić

Student: Luka Žepina

Studijski smjer: Elektroničke i informatičke tehnologije u pomorstvu

JMBAG: 0112081081

Rijeka, srpanj 2024.

Student: Luka Žepina

Studijski program: Elektroničke i informatičke tehnologije u pomorstvu

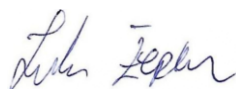
JMBAG: 0112081081

IZJAVA O SAMOSTALNOJ IZRADI ZAVRŠNOG RADA

Kojom izjavljujem da sam završni rad s naslovom „Napredne arhitekture mikroprocesora“ izradio samostalno pod mentorstvom izv. prof. dr. sc. Jasmina Čelića.

U radu sam primijenio metodologiju izrade stručnog/znanstvenog rada i koristio literaturu koja je navedena na kraju završnog rada. Tuđe spoznaje, stavove, zaključke, teorije i zakonitosti koje sam izravno ili parafrazirajući naveo u završnom radu na uobičajen, standardan način citirao sam i povezoao s fusnotama i korištenim bibliografskim jedinicama, te nijedan dio rada ne krši bilo čija autorska prava. Rad je pisan u duhu hrvatskoga jezika.

Student



Luka Žepina

Student: Luka Žepina

Studijski program: Elektroničke i informatičke tehnologije u pomorstvu

JMBAG: 0112081081

IZJAVA STUDENTA – AUTORA
O JAVNOJ OBJAVI OBRANJENOG ZAVRŠNOG RADA

Izjavljujem da kao student – autor završnog rada dozvoljavam Pomorskom fakultetu Sveučilišta u Rijeci da ga trajno javno objavi i besplatno učini dostupnim javnosti u cjelovitom tekstu u mrežnom digitalnom repozitoriju Pomorskog fakulteta.

U svrhu podržavanja otvorenog pristupa završnim radovima trajno objavljenim u javno dostupnom digitalnom repozitoriju Pomorskog fakulteta, ovom izjavom dajem neisključivo imovinsko pravo iskorištavanja bez sadržajnog, vremenskog i prostornog ograničenja mog završnog rada kao autorskog djela pod uvjetima *Creative Commons* licencije CC BY Imenovanje, prema opisu dostupnom na <http://creativecommons.org/licenses/>

Student - autor



SAŽETAK

Cilj ovog završnog rada je prikazati napredne arhitekture mikroprocesora. U prvom poglavlju opisano je što je integrirani krug i kako je nastao. Osim toga, navodi se kako se integrirani krugovi mogu klasificirati ovisno o vrsti signala, kućištu ili stupnju integracije. Također, u prvom poglavlju opisan je povijesni razvoj mikroprocesora koji je podijeljen na pet generacija. Za svaku generaciju opisano je razdoblje trajanja, koji modeli mikroprocesora su bili značajni za to razdoblje i koja su tehnološka postignuća ostvarena. U drugom poglavlju uz pomoć općeg modela mikroprocesora opisani su dijelovi mikroprocesora, a to su: aritmetičko-logička jedinica, upravljačka jedinica i registarski skup. Za kraj drugog poglavlja, opisan je princip rada mikroprocesora tijekom izvođenja programa. U trećem poglavlju opisane su arhitekture CISC i RISC koje se nalaze u današnjim mikroprocesorima te njihov princip rada. Osim toga, opisano je nekoliko tehnologija koje se koriste kod današnjih mikroprocesora poput: višejezgreni procesori, hiperdretvenost, turbo ubrzanje i sl.

Ključne riječi: mikroprocesor, napredne arhitekture, integrirani krugovi, hiperdretvenost, višejezgreni procesor, turbo ubrzanje.

SUMMARY

The goal of this undergraduate thesis is to present advanced architectures of microprocessors. The first chapter describes what an integrated circuit is, how it was created, and how integrated circuits can be classified depending on the type of signal, package or degree of integration. Also, the historical development of the microprocessor is described, which is divided into five generations. For each generation, the period of duration is described, as well as significant microprocessor models and technological achievements. In the second chapter, using the general microprocessor model, the parts of the microprocessor are described, namely: arithmetic-logic unit, control unit and register set. At the end of the chapter, the work principle of microprocessor during program execution is described. The third chapter describes the CISC and RISC architectures found in today's microprocessors and their operating principles, and several technologies used in today's microprocessors such as: multicore processors, hyperthreading, turbo boost and etc.

Keywords: microprocessor, advanced architectures, integrated circuits, hyperthreading, multicore processor, turbo boost.

SADRŽAJ

SAŽETAK	I
SUMMARY	I
SADRŽAJ	II
1. UVOD	1
2. POVIJESNI RAZVOJ MIKROPROCESORA	2
2.1. INTEGRIRANI KRUG	2
2.2.1. <i>Razvoj integriranog kruga</i>	2
2.2.2. <i>Vrste integriranih krugova</i>	3
2.2.2.1. <i>Podjela prema vrsti signala</i>	3
2.2.2.2. <i>Podjela prema vrsti kućišta</i>	3
2.2.2.2. <i>Podjela prema stupnju integracije</i>	5
2.2. GENERACIJE MIKROPROCESORA	6
2.2.1. <i>Prva generacija mikroprocesora (1971. – 1973.)</i>	6
2.2.2. <i>Druga generacija mikroprocesora (1974. – 1977.)</i>	6
2.2.1. <i>Treća generacija mikroprocesora (1978. – 1985.)</i>	7
2.2.2. <i>Četvrta generacija mikroprocesora (1985. – 1995.)</i>	7
2.2.1. <i>Peta generacija mikroprocesora (1995. – danas)</i>	8
3. MIKROPROCESOR	9
3.1. OPĆI MODEL MIKROPROCESORA	9
3.2.1. <i>Registarski skup</i>	10
3.2.2. <i>Aritmetičko-logička jedinica</i>	15
3.2.2. <i>Upravljačka jedinica</i>	19
3.2. PRIMJER IZVOĐENJA PROGRAMA	21
4. ARHITEKTURE DANAŠNJICE	23
4.1. CISC	23
4.2. RISC	24
4.3. TEHNOLOGIJE DANAŠNJICE	25
4.3.1. <i>Višejezgreni procesor</i>	25
4.3.2. <i>Hiperdretvenost</i>	28
4.3.3. <i>Tehnologija turbo ubrzanja</i>	29
4.3.4. <i>Umjetna inteligencija</i>	30

5. ZAKLJUČAK.....	31
LITERATURA.....	32
KAZALO KRATICA	33
POPIS TABLICA.....	34
POPIS SLIKA.....	34

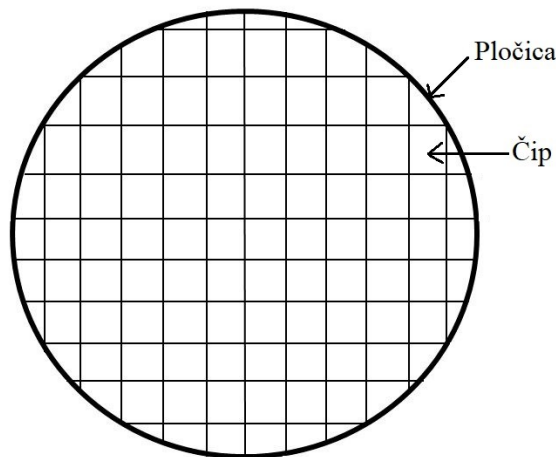
1. UVOD

Mikroprocesor je elektronički sklop koji je jedan od najbitnijih elemenata svakog elektroničkog uređaja. Zadaća mikroprocesora jest obrada podataka gdje se izvode razne aritmetičke i logičke operacije. Osim toga, mikroprocesor je povezan s ostalim dijelovima elektroničkog uređaja kao što su memorije, ulazno/izlazne jedinice i slično. Od nastanka prvog mikroprocesora 1971. godine koji je bio izrazito limitiran, pa do današnjih mikroprocesora koji mogu procesirati iznimno velik broj podataka, mikroprocesori su doživjeli izrazito velik napredak u svakom mogućem aspektu. Danas se mikroprocesori primjenjuju u računalima, automobilima, mobitelima, industrijskim postrojenjima i slično. Može se sa sigurnošću reći da su mikroprocesori velik dio svakodnevnog života, a njihov napredak i napredak tehnologija koje su vezane uz mikroprocesore je izuzetno fascinantno. Cilj ovog završnog rada je prikazati povijesni razvoj mikroprocesora, princip rada mikroprocesora i tehnologije koje se danas koriste za poboljšanje mikroprocesora. Pošto je temelj mikroprocesora integrirani krug, prvo poglavlje opisuje što je integrirani krug, kratku povijest razvoja integriranog kruga te nekoliko vrsta klasifikacija istih, a zatim povijesni razvoj mikroprocesora koji je podijeljen na pet generacija. Drugo poglavlje uz prikaz općeg modela mikroprocesora opisuje što sve čini mikroprocesor, koja je uloga svih komponenti i kratki primjer izvođenja programa. Treće poglavlje opisuje CISC i RISC arhitekture i njihovu logiku izvođenja instrukcija, a zatim se još opisuje nekoliko tehnologija koje se koriste kod današnjih mikroprocesora za poboljšanje performansi.

2. POVIJESNI RAZVOJ MIKROPROCESORA

2.1. INTEGRIRANI KRUG

Integrirani krug je elektronički sklop koji se sastoji od tanke poluvodičke pločice na kojoj se pomoću nekoliko tehnoloških procesa stvaraju željeni sklopovi. Materijal pločice je u današnje doba najčešće silicij dok je tipična površina od 1 do 100 mm², a debljina oko 0,25 mm¹. Površina silicijske pločice se dijeli na manje komadiće (čipove) na kojoj se uz pomoć procesa planarne tehnologije kao što su: oksidacija, difuzija, fotolitografija, jetkanje, ionska implementacija, epitaksijalni rast i metalizacija stvara željeni sklop. Termin „integrirani“ proizlazi iz toga što se cijeli željeni sklop, koji se sastoji od velikog broja otpornika, kondenzatora, tranzistora, dioda i slično, nalazi na jednom čipu.



Slika 1. Ilustracija silicijske pločice i čipa tijekom proizvodnje

Izvor: Peruško, U.: *Digitalna elektronika: logičko i električko projektiranje*, Školska knjiga, Zagreb, 1991.

2.2.1. Razvoj integriranih krugova

Sredinom 20. stoljeća tranzistori se su dokazali kao bolja alternativa naspram vakuumskih cijevi jer su tranzistori manjih dimenzija, boljih performansa i zahtjevi za snagom su manji. 1958. godine Jack Kilby je demonstrirao prvi funkcionalni model integriranog kruga, ali problem je bio u tome što su žice tog modela bile na vanjskoj strani što otežava masovnu proizvodnju. Nedugo nakon Kilbyja, Robert Noyce je unaprijedio

¹ Peruško, U.: *Digitalna elektronika: logičko i električko projektiranje*, Školska knjiga, Zagreb, 1991., str. 85

Kilbyjev model tako da je tijelo integriranog kruga bilo izrađeno od silicija te se pomoću planarne tehnologije omogućilo međusobno povezivanje elemenata na samoj pločici.

U početku na samoj pločici se nalazila nekolicina tranzistora i dioda, ali od tada se tehnologije vezane uz izradu integriranih krugova unaprjeđuju kako bi se smanjile dimenzije integriranih krugova, povećale performanse i broj elemenata na pločici koji može biti i preko milijardu. Zahvaljujući integriranim krugovima nastali su mikroprocesori koji se danas nalaze u svakom elektroničkom uređaju.

2.2.2. Vrste integriranih krugova

2.2.2.1. Podjela prema vrsti signala

Integrirani krugovi mogu biti analogni ili digitalni, ali postoje i modeli čiji je ulaz analogan, a izlaz digitalan i obrnuto.

Analogni integrirani krugovi koriste kontinuirane signale koji se kroz vrijeme mogu mijenjati poput zvuka ili temperature stoga se primjenjuju u pojačalima zvuka, sensorima ili se mogu primjenjivati kao aktivni filtri. Nedostatak je taj što su kontinuirani signali podložni vanjskim smetnjama zbog čega je posljedica lošija kvaliteta signala. Vanjske smetnje se mogu smanjiti na razne načine, a neki od njih su: korištenje filtera, bolji odabir komponenti, udaljavanje od izvora smetnji i oklapanje sklopa.

Digitalni integrirani krugovi koriste diskretne signale. Za razliku od kontinuiranih signala koji u teoriji mogu imati beskonačno mnogo stanja, diskretni signali koriste određene točke u vremenu čija vrijednost može biti 0 ili 1 te ih digitalni integrirani krug obrađuje pomoću logičkih funkcija kao što su I, ILI, NI i slično. Imaju široku primjenu u elektronici gdje se među ostalom koriste i kod mikroprocesora. Prednost je visoka brzina obrade podataka, manja osjetljivost na smetnje i adaptivno ponašanje, dok su nedostaci veći zahtjevi snage i kompleksnija izvedba.

Ovisno o potrebi integrirani krugovi također mogu imati ulaz jedne vrste, a izlaz druge vrste signala. S takvom izvedbom integrirani krugovi se mogu koristiti u svrhu analogno-digitalnog ili digitalno-analognog konvertiranja.

2.2.2.2. Podjela prema vrsti kućišta

Pakiranja ili kućišta služe za lakše rukovanje i ugradnju integriranih krugova na tiskanu pločicu te još istovremeno štiti sklop od potencijalnih udaraca. Najčešća izvedba kućišta jest dvoredno kućište (eng. *dual-in-line package*) čiji broj nožica se kreće od 8 do

40 ovisno o funkciji sklopa, a sukladno broju nožica se mijenja i veličina kućišta. Instalacija s takvom vrstom kućišta se izvršava s tzv. *through-hole* tehnikom tako da se na tiskanoj pločici izbuše rupe kroz koje prolaze nožice i zatim zaleme. Bitno je napomenuti da ova tehnika, iako daje čvrstu povezanost sklopa i pločice, ima relativno visoku cijenu zbog potrebnog bušenja rupa na pločici pa se koristi uglavnom kod spajanja težih komponenti na pločicu kao što su kondenzatori ili transformatori.

Još jedna tehnika spajanja je *surface-mount* tehnika koja je pretežito zamijenila *through-hole* tehniku. Ova tehnika omogućuje lakšu proizvodnju, a pošto su *surface-mount* komponente manje (izvodi su jako kratki ili se nalaze na samom tijelu komponente) broj komponenti na pločici je veći. Jedna od bitnih kućišta koje koriste ovu tehniku su LGA, PGA i BGA. PGA (Pin-Grid-Array), LGA (Land-Grid-Array) i BGA (Ball-Grid-Array) su vrste kućišta koje se najčešće koriste kod današnjih procesora.

Kontakti PGA kućišta nalaze na donjoj strani, te se ugrađuje na matičnu ploču u tzv. *socket* pomoću *through-hole* tehnike. Iako je prvo PGA kućište proizvedeno 1980-ih godina, proizvođač AMD i dalje koristi PGA kućišta za svoje procesore. PGA karakterizira jednostavna ugradnja i popravak savijenih pinova u slučaju ako su pinovi savijeni te relativno niska cijena. Nedostatak je što su pinovi također podložni lomljenju, pa se u tom slučaju mora zamijeniti cijeli procesor.

LGA kontakti se ne nalaze na procesoru, već na matičnoj ploči čime se smanjuje mogućnost oštećenja procesora, a time je postupak popravljavanja ili zamjene jeftiniji. LGA zahvaljujući manjem volumenu i većem broju pinova ima bolje performanse, ali je samim time i skuplji. LGA tehnologija je stekla popularnost izlaskom Intelovog Pentium 4 procesora 2004. godine te se koristi i kod AMD i Intel procesora.

BGA kućišta imaju jednak raspored kontakata kao i PGA, ali umjesto žičanih pinova se koriste kuglice lema. Ima izuzetno mali volumen i veliki broj kontakata i zbog toga se najviše koristi kod pametnih mobitela. Nedostatak je skupa proizvodnja, a nakon što se ugradi u uređaj teško se provodi inspekcija i popravak u slučaju kvara. Nastali su ubrzo nakon PGA kućišta.

2.2.2.2. Podjela prema stupnju integracije

Integrirani krugovi se mogu još i klasificirati po stupnju integracije odnosno po broju osnovnih logičkih sklopova. Kroz povijest broj logičkih sklopova je rastao, a time se i mijenjala njihova primjena od osnovnih logičkih sklopova s jednostavnim funkcijama do kompleksnih mikroprocesora.

Tablica 1. Razvoj stupnja integracije integriranih krugova kroz povijest

Naziv	Razdoblje	Broj logičkih sklopova	Primjena
SSI (Small-scale integration)	1961. - 1965.	Do 10	Osnovni logički sklopovi i bistabil
MSI (Medium-scale integration)	1966. - 1970.	Od 10 do 100	Aritmetički sklopovi, multiplekser, brojač i sl.
LSI (Large-scale integration)	1971. - 1979.	Od 100 do 10000	Memorije i mikroprocesori
VLSI (very-large-scale integration)	1980. - 1984.	Od 10000 do 100000	Veće memorije i mikroprocesori 32-bitna
ULSI (ultralarge-scale integration)	1985. - danas	Preko 100000	Izuzetno velike memorije i mikroprocesori 64-bitna

Izvor: Pripremio student prema Maini, A. K.: *Digital Electronics: Principles And Integrated Circuits* 1. izdanje, Wiley India, 2007, str. 184

2.2. GENERACIJE MIKROPROCESORA

Uz razvoj integriranih krugova 1971. godine nastali su prvi mikroprocesori. Prve komercijalne izvedbe su imale samo jednu funkciju, te je zbog toga bilo nejasno kako će se mikroprocesori razvijati i primjenjivati. Od nastanka prvog mikroprocesora pa do danas je ostvaren veliki napredak koji se očituje kroz veći broj izvedivih funkcija, veću frekvenciju radnog takta, manje dimenzije, naprednije arhitekture te čak i mogućnost korištenja umjetne inteligencije. Povijesni razvoj mikroprocesora se može podijeliti u pet generacija.

2.2.1. Prva generacija mikroprocesora (1971. – 1973.)

Prva generacija mikroprocesora je nastala 1971. godine, kada je Intel proizveo prvi mikroprocesor za komercijalne svrhe pod nazivom Intel 4004. Prvu generaciju obilježavaju mikroprocesori sa serijskim izvođenjem instrukcija koje se može podijeliti u tri faze, a to su: dohvat instrukcije, dekodiranje instrukcije i izvršavanje operacije. Nakon izvršene operacije mikroprocesor čeka sljedeću naredbu i zatim ponovno izvršava prethodno navedene funkcije.

Nedugo nakon 4004, Intel je 1972. godine proizveo 8-bitni mikroprocesor pod nazivom Intel 8008. Osim Intela, na tržištu su u tom periodu bili i Rockwell international sa svojim PPS-4 4-bitnim mikroprocesorom i National Semiconductor s 16-bitnim mikroprocesorom pod nazivom IMP-16². Za izradu mikroprocesora prve generacije koristili su se P-kanalni MOSFET³-i zbog čega je iznos minimalnog napajanja bio relativno visok (oko -25V), brzina rada je bila relativno malena i zbog svoje strukture imali su veće dimenzije.

2.2.2. Druga generacija mikroprocesora (1974. – 1977.)

Zahvaljujući napretkom integriranih krugova i uvođenjem N-kanalnih MOSFET-a⁴ omogućila se proizvodnja 8-bitnih mikroprocesora s većom gustoćom elemenata, manjim iznosom napajanja i učinkovitijim i bržim radom od prošle generacije mikroprocesora. Umjesto sekvencijalnog izvođenja instrukcija, što obilježava mikroprocesore prve generacije, kod druge generacije se instrukcije izvode paralelno. Time se povećala brzina rada jer, kod paralelnog izvršavanja instrukcija, više se od jedne instrukcije izvršava

² Betker, M. R., Fernando, J. S., Whalen, J. P.: *The Birth, Evolution and Future of Microprocessor*, Bell Labs technical journal, Bell System Memorial, 1997, str. 34-37

³ Metal Oxide Semiconductor Field Effect Transistor

⁴ Betker, M. R., Fernando, J. S., Whalen, J. P., op. cit., str. 35

istovremeno npr. dok se jedna operacija izvršava, u isto vrijeme se dekodira jedna instrukcija i dohvaća druga.

Jedan od poznatijih modela je Intel 8080 koji se temeljio na 8008 modelu. Osim Intela tu su i Motorolin 6800 model i Z80 model koji je napravila kompanija Zilog koji su s preinakama uspjeli postati bolji izbor naspram 8080. Korištenjem Intelovog 8080 modela kod Altair 8800 mikoračunala, koji se smatra prvim komercijalno uspješnim osobnim računalom, utvrdilo se da postoji mogućnost korištenja mikroprocesora u više svrha⁵.

2.2.3. Treća generacija mikroprocesora (1978. – 1985.)

Treća generacija nastala je uvođenjem 16-bitnih mikroprocesora poput Intelovog 8086 i Zilogovog Z8000 modela koji se zasnivaju na CISC arhitekturi. Ideja CISC arhitekture jest da se aritmetičko-logička jedinica, odnosno instrukcijski skup, izvede na način da se na razini elektronike s jednom kompleksnijom instrukcijom obavi više operacija. Zbog kompleksnijih instrukcija vrijeme izvršavanja operacija može trajati više od jednog ciklusa. Osim skupe i složene proizvodnje, od svih dostupnih instrukcija veliki broj njih se rijetko koristi. Također, CISC arhitektura otežava razvoj novih generacija procesora, ali se i danas koristi⁶.

2.2.4. Četvrta generacija mikroprocesora (1985. – 1995.)

Četvrtu generaciju mikroprocesora obilježava nastanak 32-bitnih mikroprocesora. Korištenjem 32-bitnih mikroprocesora omogućilo se manipuliranje s većim brojem podataka, instrukcijski skup se povećao, te se i povećala količina fizičke memorije (RAM⁷) koju mikroprocesor može adresirati. U teoriji, za 32-bitne mikroprocesore iznos adresabilne fizičke memorije iznosi 4 GB, ali u praksi iznosi oko 3,5 GB pošto se dio registra ne koristi za memoriju.

Jedni od popularnih modela četvrte generacije su Motorola 68020 i Intel 80386 mikroprocesori koji su koristili CISC arhitekturu. Osim CISC arhitekture se pojavljuje i RISC arhitektura. Iako se RISC koncept razvijao od 1970-ih godina, tek se u kasnim 1980-

⁵ Betker, M. R., Fernando, J. S., Whalen, J. P., op. cit., str. 35-36

⁶ 'RISC and CISC in Computer Organization', 13.3.2024., online: <https://www.geeksforgeeks.org/computer-organization-risc-and-cisc/> (28.6.2024)

⁷ Random Access Memory

im i ranim 1990-ima pronašao komercijalni uspjeh⁸. Ideja RISC arhitekture je da se ubrza rad mikroprocesora na način da se umjesto jedne kompleksne instrukcije koriste više jednostavnih instrukcija koje se mogu brže izvoditi, te zbog toga RISC mikroprocesori imaju manji opseg instrukcija uz povećani broj registra. MIPS R2000 mikroprocesor je prvi komercijalno dostupan RISC mikroprocesor⁹, a jedna od poznatijih serija mikroprocesora su PowerPC modeli koji su nastali 1991. godine zajedničkim trudom kompanije Apple, IBM i Motorola. Primjenjivali su se najčešće kod Apple stolnih računala i igračih konzola.

2.2.5. Peta generacija mikroprocesora (1995. – danas)

U petoj generaciji mikroprocesori se unaprjeđuju iz godine u godinu. Osim što se postepeno uvodi 64-bitna arhitektura nastaju tehnologije poput hiperdretvenosti, pojavljuju se višejezgreni mikroprocesori, tehnologija turbo ubrzanja¹⁰ koja proporcionalno opterećenju mijenja brzinu rada mikroprocesora i slično. Osobna računala su sve više i više počeli koristiti 64-bitne sustave, a 2003. godina se smatra prekretnicom kad je AMD proizveo Opteron model koji je, osim što je 64-bitan, kompatibilan i s 32-bitnim sustavima. Osim Opterona, modeli poput Pentium 4 i Celeron od Intela ili Opteron i Athlon modeli od AMD-a su bili popularan odabir kod osobnih računala. Danas se najčešće primjenjuju Intel Core procesori, koji su poznati s oznakama i3, i5, i7 i i9, AMD-ovi Ryzen procesori te razni mikroprocesori poput Snapdragon 8 za mobilne uređaje.

⁸ Betker, M. R., Fernando, J. S., Whalen, J. P., op. cit., str. 45

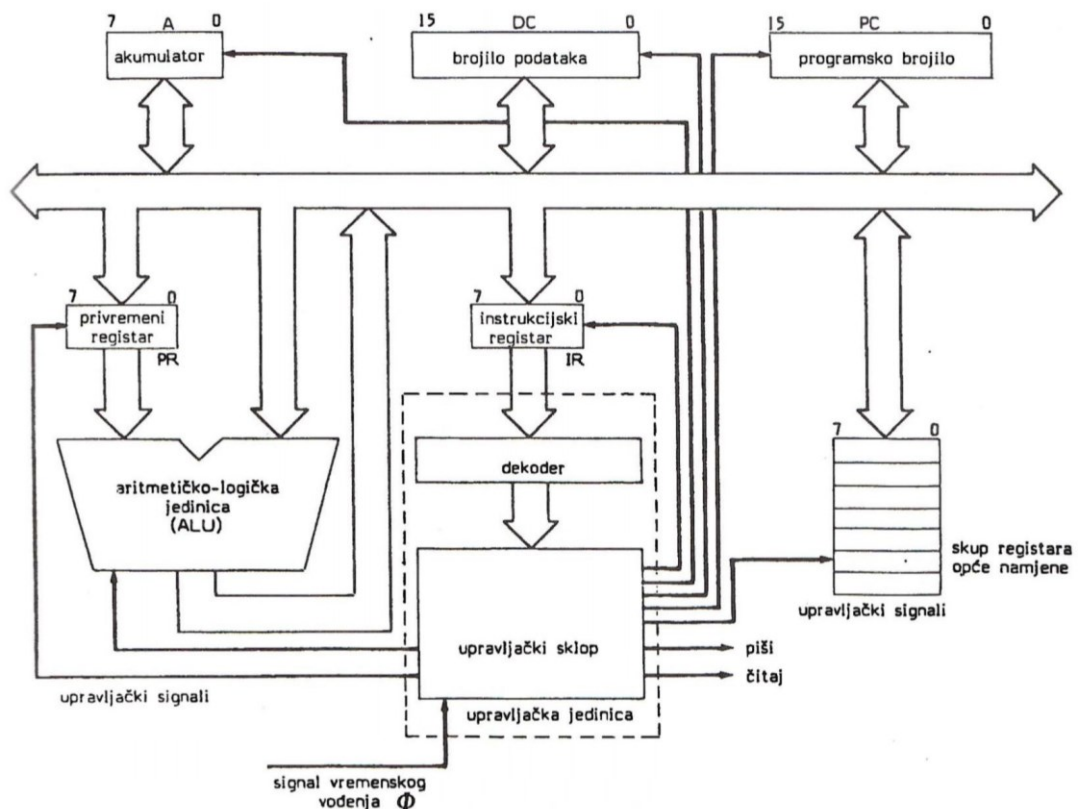
⁹ Ibidem, str. 46

¹⁰ Vidi poglavlje 4.3.3.

3. MIKROPROCESOR

3.1. Opći model mikroprocesora

Mikroprocesor je programibilna elektronička komponenta čiji se dijelovi koji su potrebni za izvršavanje instrukcija nalaze na jednom ili više integriranih krugova tj. sve nužne jedinice su objedinjene u jednu cjelinu. Tu cjelinu mikroprocesora čine: aritmetičko-logička jedinica (ALU), upravljačka jedinica (CPU) i registarski skup u kojem se nalazi veći broj registra gdje svaki ima svoju određenu funkciju. Moderni mikroprocesori osim ALU imaju i FPU¹¹ i SIMD¹² jedinice pomoću kojih se obavljaju kompleksnije aritmetičke funkcije. Djelovanje mikroprocesora se svodi na faze koje čine dohvat instrukcije koja je pohranjena u memoriji, dekodiranje instrukcije, izvršavanje instrukcije te ponavljanje istog ciklusa do kraja programa.



Slika 2. Pojednostavljeni model mikroprocesora

Izvor: Ribarić, S.: *Naprednije arhitekture mikroprocesora*, 2. Izdanje, Element, Zagreb, 2002., str. 25

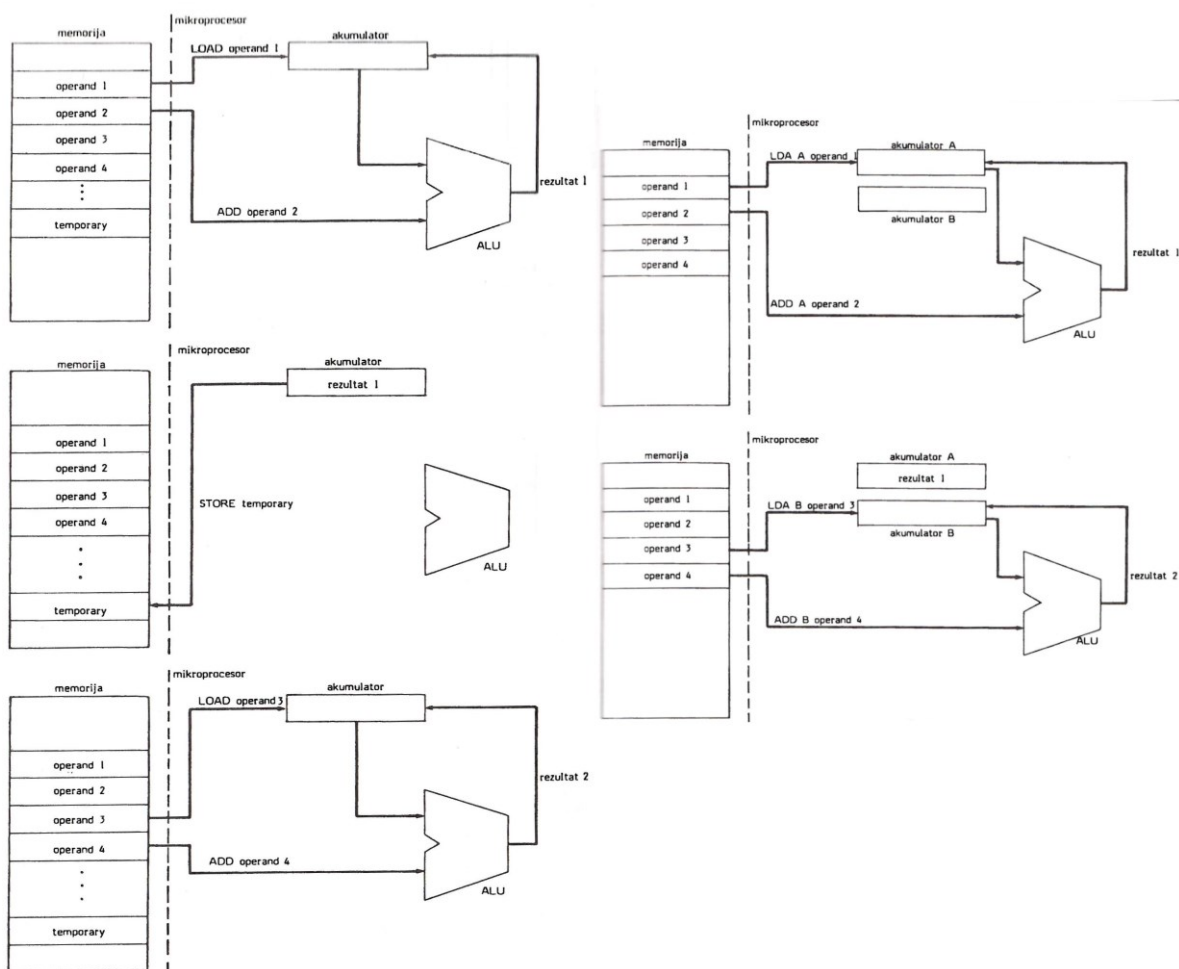
Slika 2 prikazuje pojednostavljeni model mikroprocesora, koji se temelji na 8-bitnom mikroprocesoru, pomoću kojeg će se opisati generalni princip rada mikroprocesora.

¹¹ Floating Point Unit

¹² Single Instruction Multiple Data

3.1.1. Registariski skup

Akumulator je registar koja služi za privremenu pohranu instrukcije koja se trenutno koristi za izvršavanje aritmetičkih ili logičkih funkcija i čiji se rezultat najčešće u njemu pohranjuje. Osim kod aritmetičkih i logičkih funkcija, akumulator služi i za prijenos podataka između mikroprocesora i ostalih komponenti nekakvog računala ili slično. Najčešće se postavlja na ulaz ALU u kombinaciji s privremenim registrom kako bi se odvojili ALU ulaz i izlaz, a tim se postupkom smanjuje mogućnost pojave greške pri obradi podataka. Problem prvih mikroprocesora je taj što su imali samo jedan akumulator pa bi znalo dolaziti do pojave *bottlenecka*.



Slika 3. Usporedba arhitekture s jednim akumulatorom i strukture s dva akumulatora

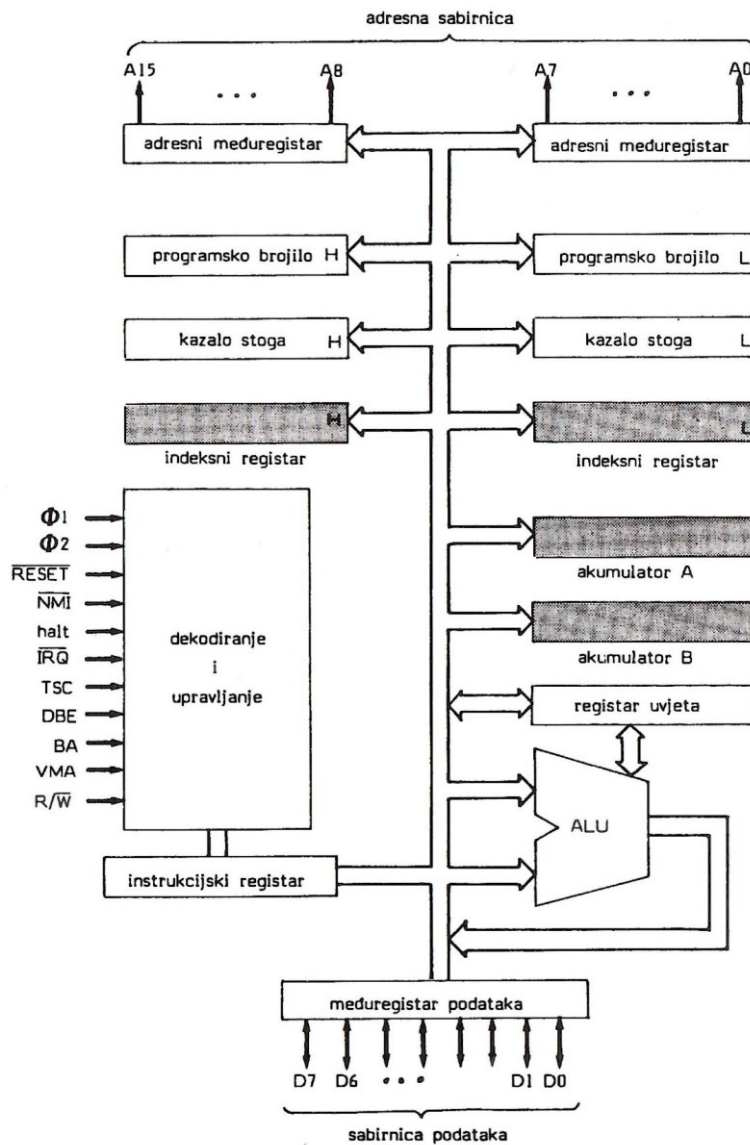
Izvor: Ribarić, S.: *Naprednije arhitekture mikroprocesora*, 2. Izdanje, Element, Zagreb, 2002., str. 42-43

Slika 3 s lijeve strane prikazuje arhitekturu mikroprocesora koji sadrži jedan akumulator, dok se s desne strane nalazi mikroprocesor koji ima dva akumulatora. Željena operacija je da se operandi 1 i 2 zbroje, a zatim da se zbroje operandi 3 i 4. Instrukcija tj.

operand 1 se dobavlja u akumulator čija će se vrijednost zbrojiti s operandom 2 u ALU. Kada se dobije rezultat (rezultat 1) on se privremeno pohranjuje u memoriju, a nakon pohrane akumulator je spreman za izvršavanje sljedeće instrukcije koja je u ovom slučaju zbrajanje operanda 3 i 4. Proces privremene pohrane rezultata u memoriju se obavlja isključivo kada mikroprocesor ima samo jedan akumulator. Iste operacije kod mikroprocesora s dva akumulatora će se odvijati na sličan način tako da će umjesto pohranjivanja rezultata u privremenu memoriju oni ostati pohranjeni u akumulatorima A i B. Uklanjanjem koraka, gdje se rezultat privremeno pohranjuje u memoriju, omogućuje se brže izvršavanje instrukcija, a i štedi se na memorijskom prostoru. Bez akumulatora rezultati bi se morali pohranjivati u glavnu memoriju ili cache čime bi se vrijeme pristupa podacima povećalo.

Programsko brojilo je 16-bitni registar koji sadrži adresu one instrukcije koja će se u sljedećem ciklusu dohvatiti, dok je kôd instrukcije koja se trenutno izvršava pohranjen u instrukcijskom registru koji je u slučaju na slici 8-bitan. Adresa memorijske lokacije instrukcije se nalazi u brojilu podataka. Od registara specifične namjene još preostaje privremeni registar koji je povezan na ulaz ALU te služi za privremenu pohranu podataka koji su koriste prilikom izvođenja aritmetičkih ili logičkih operacija.

Skup registara opće namjene služi za privremenu pohranu instrukcija, međurezultata i rezultata. Pošto se nalaze u samom mikroprocesoru njihovo vrijeme pristupa je najmanje. Razlika između registara koji imaju specifičnu namjenu i registara opće namjene je ta što registri specifične namjene nisu dio programskog modela mikroprocesora dok registri opće namjene jesu. Pod programskim modelom misli se na to da je pristupačan programeru. Svaki od registara opće namjene može imati funkciju programskog brojila, kazala stoga, registra podataka ili akumulatora.

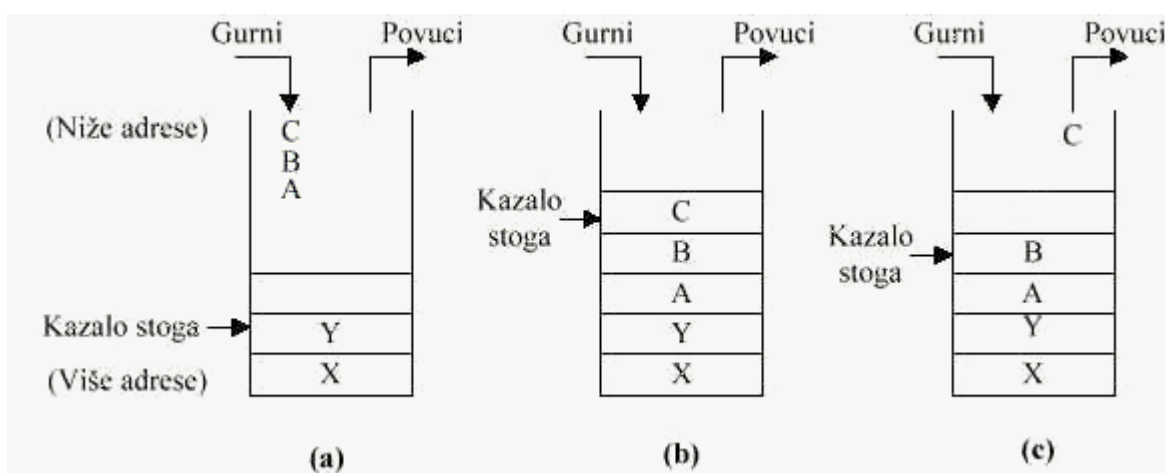


Slika 4. Raspored registara MC 6800 mikroprocesora

Izvor: Ribarić, S.: *Naprednije arhitekture mikroprocesora*, 2. Izdanje, Element, Zagreb, 2002., str. 46

Slika 4 prikazuje raspored registarskog skupa 8-bitnog mikroprocesora MC 6800. Osim već prije opisanih komponenti na mikroprocesoru se nalaze i kazalo stoga, indeksni registar, registar uvjeta i adresni međuregistar.

Stog je memorijska struktura poznata po LIFO (Last In First Out) principu rada. To znači da se najstariji podatak koji je pohranjen u stogu nalazi na dnu, a najnoviji se nalazi na vrhu stoga. Na stog se mogu primijeniti dvije funkcije: *push* koja pohranjuje podatke u stog i *pull* pomoću koje se dohvaća podatak u stogu. Može se ostvariti programski ili sklopovski, a svrha mu je privremena pohrana podataka. Kazalo stoga je registar koji sadrži adresu podatka koji je na vrhu stoga. Duljina registra ovisi o samoj veličini memorije, a svaka izmjena podataka u stogu mijenja sadržaj kazala stoga. Slika 5 vizualno prikazuje stogovni način pohrane podataka.



Slika 5. Prikaz stogovnog načina pohranjivanja podataka

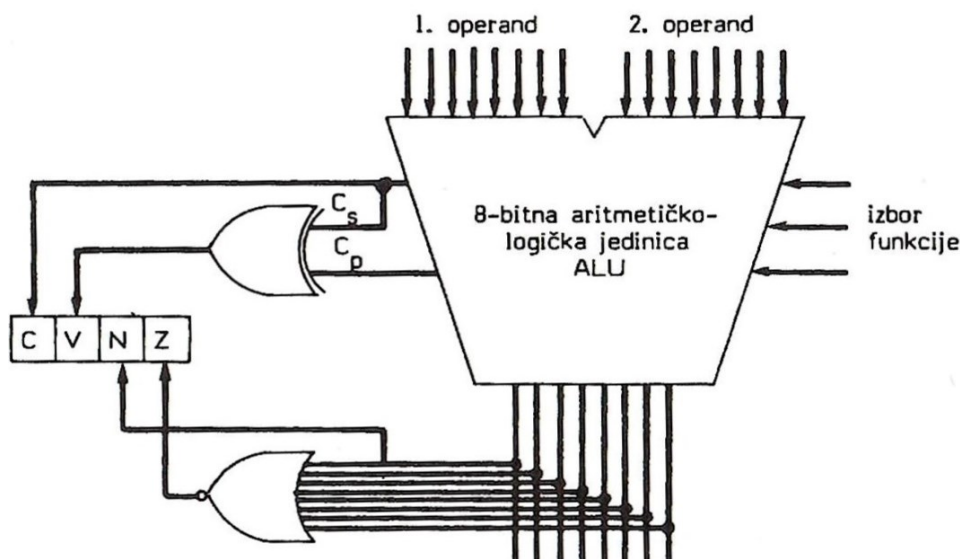
Izvor: *Rad sa složnikom i posluživanje zahtjeva za prekidom*, Fakultet prometnih znanosti, Zagreb,

<https://www.fpz.unizg.hr/hgold/es/de/zahtjev.htm> (10.5.2024.)

Slika (a) prikazuje početno stanje stoga. Korištenjem push operacije u stog se pohranjuju podaci A, B i C. Pošto stogovni način pohrane radi na LIFO principu na slici (b) vidimo da je podatak C na vrhu pošto je zadnji ušao u stog i kazalo stoga sadržava adresu podatka C. Korištenjem pull operacije uzima se C podatak, a kazalo stoga poprima vrijednost podatka B kao što je prikazano na slici (c).

Indeksni registar je registar koji se primjenjuje u indeksnom načinu adresiranja gdje se njegov sadržaj koristi za postupak određivanja adrese instrukcije. Mikroprocesori mogu imati jedan ili više indeksnih registara, a neki mikroprocesori koriste registre opće namjene kako bi postigli isti rezultat.

Registar uvjeta je registar posebne namjene koji se sastoji od bistabila koji se zove dozivni bistabil ili zastavica. Dozivni bistabili se mogu razvrstati na: bistabile koji služe za indikaciju stanja nakon izvršavanja aritmetičko-logičkih operacija, bistabile za rukovanje prekidom i označivanje prekidnih razina i bistabile za indikaciju glavnih stanja mikroprocesora. Aritmetičko-logičkoj jedinici je pridruženo u registru uvjeta polje zastavica. Za vrijeme izvođenja aritmetičko-logičkih operacija mikroprocesor postavlja ili briše odgovarajuće zastavice shodno rezultatima operacije. Stanje zastavica služi da bi se stvorilo uvjetno grananje. Uvjetnim grananjem se ispituje stanje zastavica tj. ispravnost operacija i dobivenih rezultata i prema rezultatima ispitivanja uvjeta se sadržaj programskog brojila može mijenjati ili ostaje isti. Zahvaljujući tome mikroprocesor „sam“ određuje daljni tijek odvijanja programa. Preostaje još adresni međuregistar koji služi za adresiranje memorije i ostalih dijelova računala.



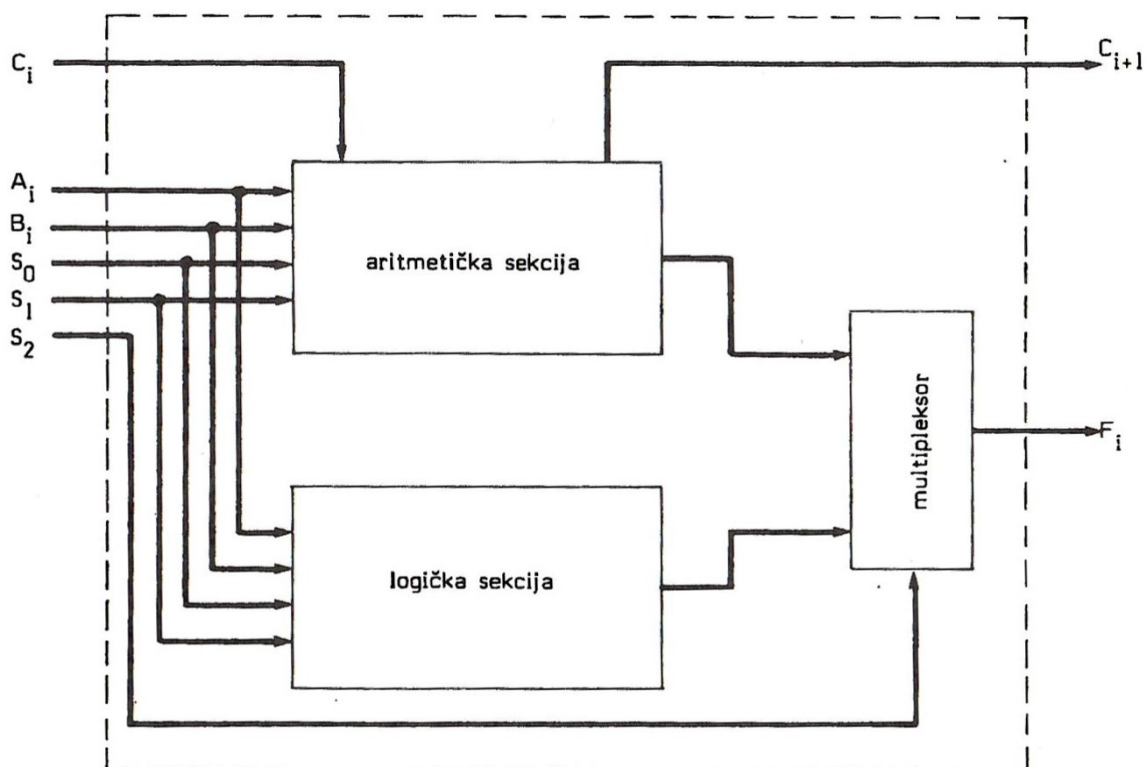
Slika 6. Sklop za postavljanje zastavica C,V,N i Z u registru uvjeta

Izvor: Ribarić, S.: *Naprednije arhitekture mikroprocesora*, 2. Izdanje, Element, Zagreb, 2002., str. 65

Slika 6 prikazuje sklop za postavljanje zastavica C, V, N i Z u registru uvjeta uz pomoć isključivo ILI i ILI logičkih sklopova. Svaka zastavica ima svoju specifičnu ulogu, a stanje zastavice ovisi o rezultatu operacije: Z zastavica označava kada je rezultat logičke ili aritmetičke operacije nula, N zastavica se postavlja kada je rezultat operacije negativan, C zastavica služi za označavanje ako se desio prijenos najznačajnijeg bita rezultata, a V zastavica (ponekad označena kao O ili W) služi kada je dobiveni rezultat prevelik za registar.

3.1.2. Aritmetičko-logička jedinica

Aritmetičko-logička jedinica, ili skraćeno ALU, je sklop koji služi za izvođenje osnovnih aritmetičkih i logičkih funkcija. Aritmetičko-logička funkcija ima strukturu koja se sastoji od identičnih stupnjeva koji su povezani u kaskadu. Svaki stupanj se može prikazati kao kombinacija sklopova aritmetičke i logičke sekcije.

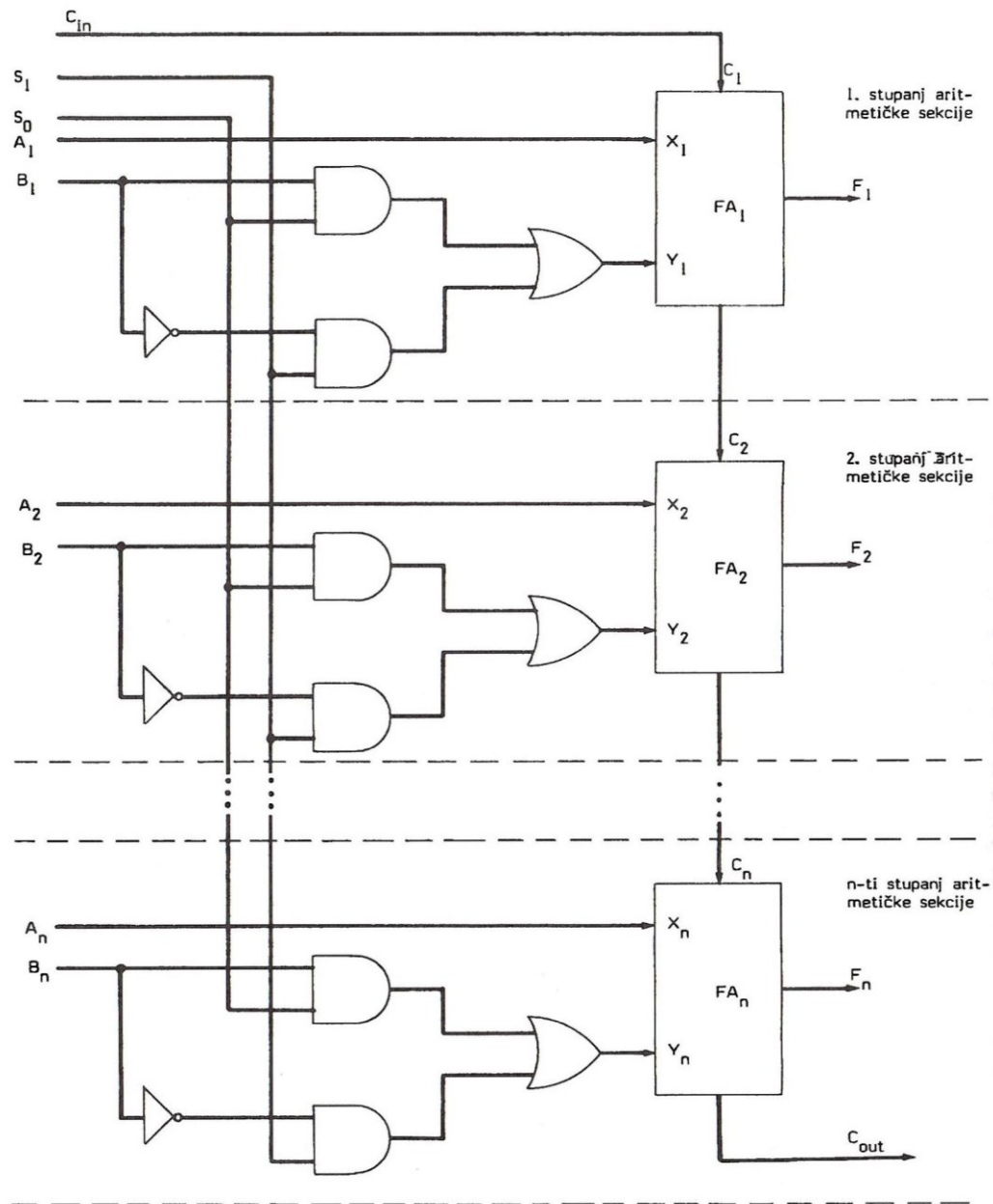


Slika 7. Prikaz jednog stupnja aritmetičko-logičke jedinice

Izvor: Ribarić, S.: *Naprednije arhitekture mikroprocesora*, 2. Izdanje, Element, Zagreb, 2002., str. 73

Slika 7 prikazuje jedan od stupnjeva aritmetičko-logičke jedinice koji se sastoji od aritmetičke i logičke sekcije. Linije C_i , koja predstavlja bit prijenosa iz prošlog stupnja i C_{i+1} , koja predstavlja bit prijenosa u sljedeći stupanj služe za povezivanje stupnjeva. A_i i B_i su ulazi za instrukcije. Na izlazu iz multipleksora se nalazi F_i linija preko koje se šalju jednobitni rezultati aritmetičke ili logičke operacije. Upravljački ulazi S_0 , S_1 i S_2 služe za odabir različitih aritmetičkih ili logičkih operacija.

Aritmetička sekcija služi za obavljanje aritmetičkih operacija. U početku zbog tehnoloških ograničenja mogle su se odvijati osnovne aritmetičke operacije poput zbrajanja i oduzimanja. Kasnije kako su mikroprocesori napredovali omogućilo se izvršavanje kompleksnijih aritmetičkih operacija poput množenja, dijeljenja te korištenje većih tipova podataka.

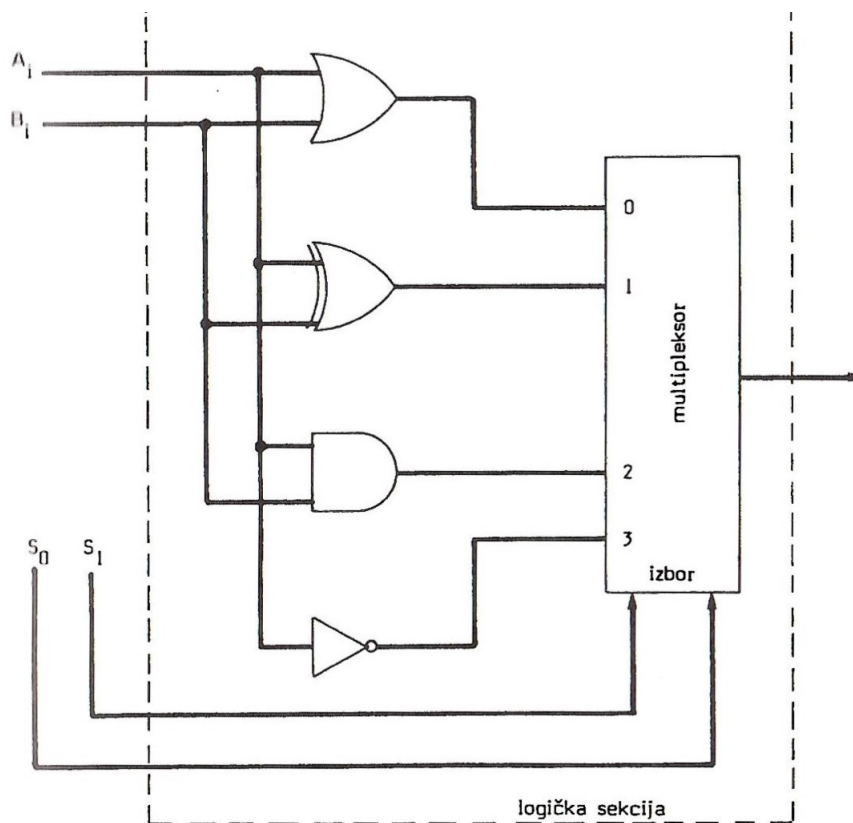


Slika 8. Prikaz aritmetičke sekcije

Izvor: Ribarić, S.: *Naprednije arhitekture mikroprocesora*, 2. Izdanje, Element, Zagreb, 2002., str. 76

Slika 8 prikazuje stupnje aritmetičkih sekcija koji su kaskadno povezani. X_n i Y_n predstavljaju bitove instrukcija A_n i B_n , a C_{in} ulazni bit prijenosa. Pomoću upravljačkih ulaza S_0 i S_1 i ulaza C_{in} se određuje koje će se operacije izvoditi unutar aritmetičke sekcije. Također, na slici se može uočiti izlaz F_n otkud se šalju dobiveni rezultati operacije i izlaz C_{out} gdje se nalazi bit prijenosa koji se upućuje u registar uvjeta.

Logička sekcija promatra bitove instrukcija kao logičku ili binarnu varijablu. Izvršava osnovne logičke funkcije poput I, ILI, XILI i NE kao što je prikazano na slici 9. Pošto se sve logičke operacije mogu izvršiti pomoću te četiri logičke funkcije, logička sekcija najčešće izvodi samo te funkcije.



Slika 9. Prikaz logičke sekcije

Izvor: Ribarić, S.: *Naprednije arhitekture mikroprocesora*, 2. Izdanje, Element, Zagreb, 2002., str. 81

Podjela stupnja aritmetičko-logičke jedinice na aritmetičku i logičku sekciju je funkcionalna jer se pri izradi ALU koristi pristup gdje se aritmetička sekcija oblikuje zasebno od logičke sekcije, a zatim se određuju logičke operacije koje se mogu izvesti sklopovima iz aritmetičke sekcije. Kasnije se još aritmetički sklopovi modificiraju na način da se omogući izvođenje i željenih logičkih operacija. Takav pristup smanjuje složenost i broj sklopova u svakom stupnju aritmetičko-logičke jedinice.

Generalni princip rada aritmetičko-logičke funkcije se svodi na to da aritmetičko-logička jedinica ima pristup glavnoj memoriji (RAM i ROM¹³) i ostalim uređajima (periferija). Podaci iz periferije i memorija dolaze na aritmetičko-logičku jedinicu u obliku instrukcija gdje se unutar instrukcije nalazi kôd koji određuje koje će se operacije izvršiti unutar aritmetičko-logičke jedinice. Zatim aritmetičko-logička jedinica obavlja potrebne operacije, a rezultat se pohranjuje u prikladan registar.

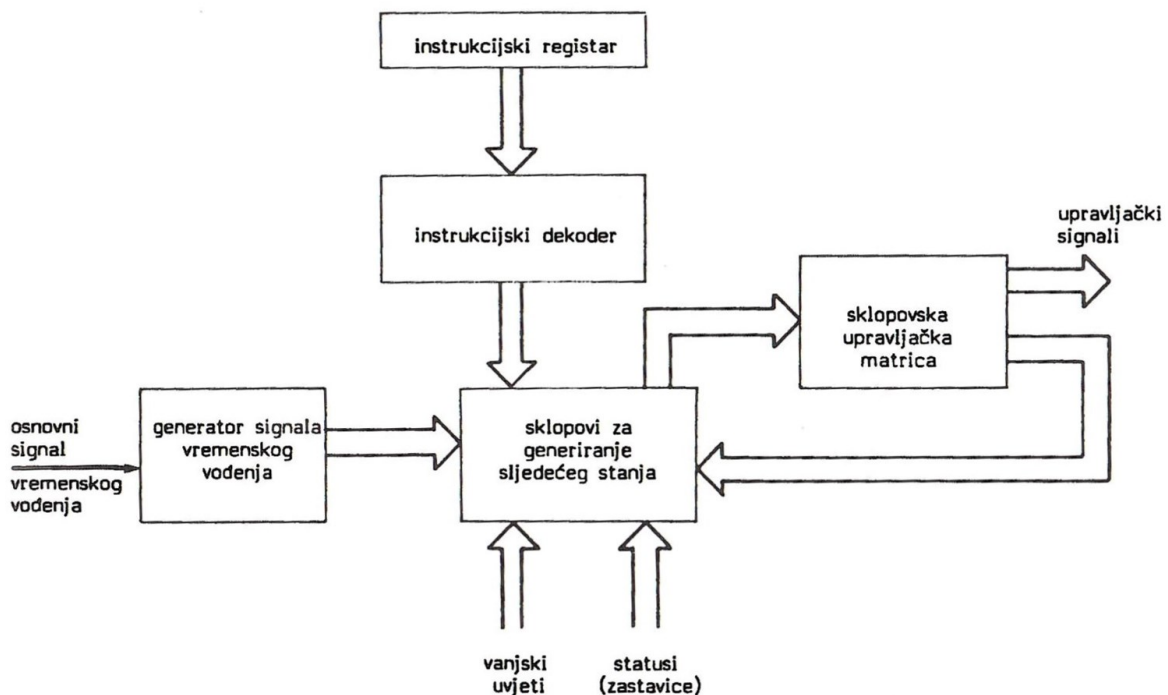
¹³ Read Only Memory

3.1.3. Upravljačka jedinica

Upravljačka jedinica je komponenta mikroprocesora pomoću koje se upravlja cijelim radom mikroprocesora. Ona generira upravljačke signale, sinkronizira rad svih ostalih komponenti te pribavlja, dekodira i upravlja izvođenjem instrukcija. Osim što omogućava komunikaciju između ostalih komponenti mikroprocesora, pomoću sabirnica je povezana s memorijom i „vanjskim svijetom“ tj. U/I sučeljem te upravlja s njihovim signalima. U prvoj fazi rada mikroprocesora pomoću upravljačke jedinice se dobavlja instrukcija i operacijski kôd se dekodira pomoću instrukcijskog dekodera. Shodno operacijskom kodu upravljačka jedinica šalje signale ostalim komponentama mikroprocesora pomoću kojih se izvode instrukcije koje su dobavljene iz memorije. Kako upravljačka jedinica šalje signale nastaju mikrooperacije. Mikrooperacije su osnovne operacije poput prijenosa podataka između registara, aktiviranje sklopova u aritmetičko-logičkoj jedinici i slično. One se izvode u već utvrđenim diskretnim vremenskim trenutcima na način da su upravljački signali u sinkronizaciji sa signalom vremenskog vođenja.

Upravljačka jedinica mora podržavati određivanje stanja mikroprocesora tijekom svakog strojnog ciklusa, odrediti sljedeće stanje na temelju trenutnog stanja i zastavica, pohraniti podatke koji opisuju trenutno stanje te ovisno o stanju generirati prikladne upravljačke signale.

Organizacija upravljačke jedinice općeg modela 8-bitnog mikroprocesora se izvodi primjenom logičkih sklopova, brojila, registra, dekodera, multipleksora i ostalih elemenata, ali bitno je napomenuti da, iako većina 8-bitnih mikroprocesora ima izvedenu upravljačku jedinicu kao što je prikazano na slici 10, radi neprilagodljivosti i relativno složene strukture nije prikladno koristiti ovakvu organizaciju kod mikroprocesora koji imaju veći stupanj integracije. Instrukcijski dekodier dekodira operacijski kôd instrukcije i pobuđuje odgovarajuću izlaznu liniju. Na temelju pobuđene linije signala koji odgovara vanjskim uvjetima, stanju zastavica i povratnih signala iz sklopovske upravljačke matrice sklopovi za generiranje sljedećeg stanja generiraju signale koji odgovaraju vremenskom slijedu izvođenja mikrooperacija. Sklopovska upravljačka matrica na temelju izlaznih signala i operacijskog kôda oblikuje upravljačke signale koji pobuđuju odgovarajuće mikrooperacije.



Slika 10. Prikaz organizacije upravljačke jedinice

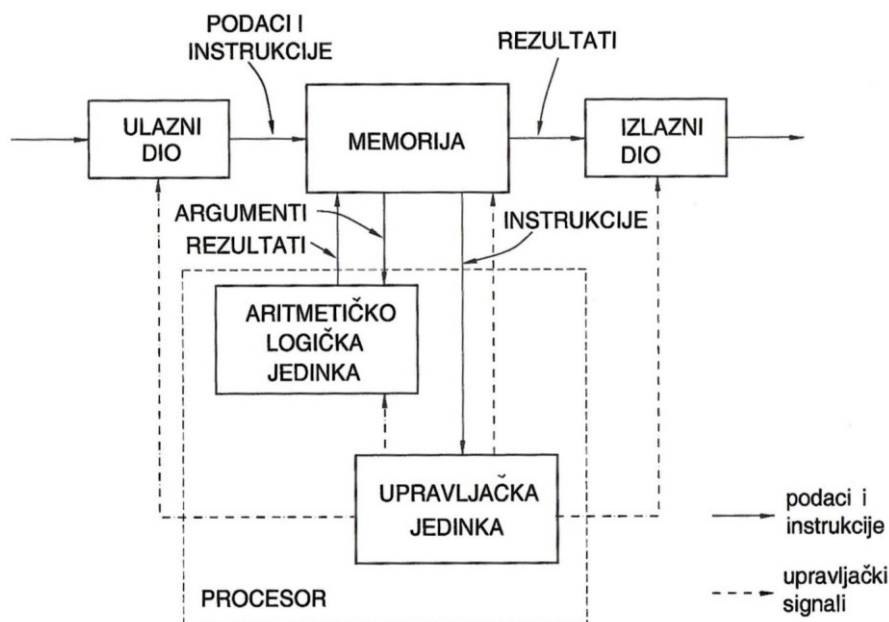
Izvor: Ribarić, S.: *Naprednije arhitekture mikroprocesora*, 2. Izdanje, Element, Zagreb, 2002., str. 107

Prijašnje navedene mikrooperacije, koje su zapravo instrukcije koje se izvode na razini strojnog jezika, mogu se opisati mikroinstrukcijom. Mikroinstrukcije se pohranjuju u mikroprogramskoj memoriji, koja je najčešće ROM ili PLA¹⁴ memorija, i definira mikrooperaciju koja se pobuđuje, a nakon toga se mikroinstrukcija adresira i pročita iz memorije. Niz takvih mikroinstrukcija čine mikroprogram. Mikroprogramiranje predstavlja sustavnu metodu izvedbe upravljačke jedinice i omogućava proizvođaču mikroprocesora jednostavnu promjenu skupa instrukcija na razini zbirnog jezika. Za 16 i 32-bitne mikroprocesore mikroprogramiranje je standardan pristup oblikovanja upravljačke jedinice. Korištenjem dvije razine ove vrste programiranja, mikroprogramiranje i nanoprogramiranje, koje se izvode s bržim sklopovima uz manju memoriju, moguće je smanjiti veličinu same upravljačke jedinice čime se oslobađa prostor na čipu. Koristeći ove dvije razine povećava se ukupno vrijeme pristupa, ali korištenjem *pipeline* arhitekture se vrijeme pristupa i obrade može smanjiti.

¹⁴ Programmable Logic Array – uređaj koji se sastoji od programibilnih logičkih sklopova

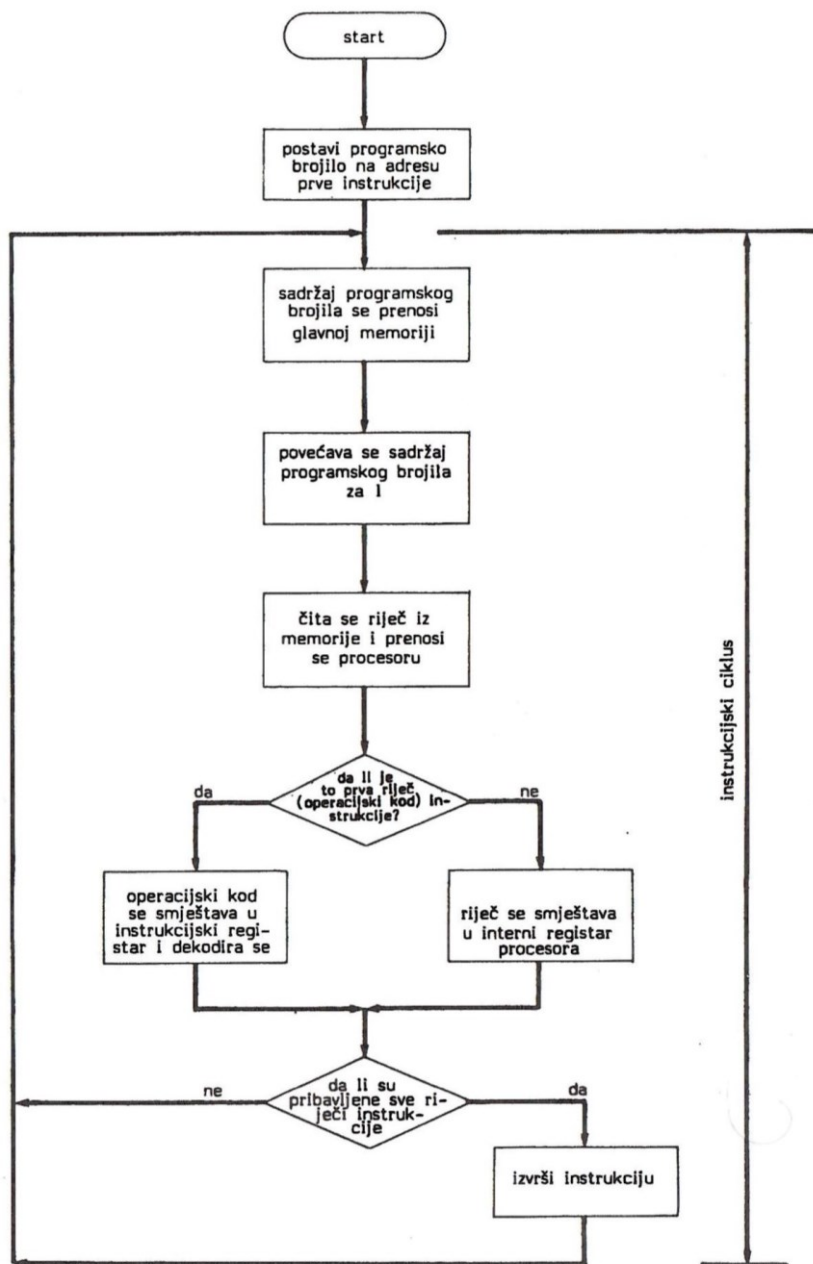
3.2. Primjer izvođenja programa

Uz pomoć slike 2, koja prikazuje opći model mikroprocesora, može se opisati izvođenje nekakvog programa. Program je zapravo slijed instrukcija koje će mikroprocesor izvršavati, a pohranjen je u memoriji te ima svoje prikladne adrese. U fazi dohvata instrukcije programsko brojilo mikroprocesora ima vrijednost adrese prve instrukcije koje će se izvoditi. Sadržaj programskog brojila se prenosi preko sabirnice na upravljački sklop te se generira upravljački signal koji dohvaća tu instrukciju. Nakon što je instrukcija dobavljena, sadržaj programskog brojila se povećava tj. prikazuje adresu sljedeće instrukcije. Instrukcijski registar pohranjuje operacijski kôd instrukcije, a mikroprocesor prelazi u fazu dekodiranja gdje se ovisno o operacijskom kodu utvrđuje koje će se operacije provesti. Istovremeno se adresa pribavljene instrukcije nalazi u brojilu podataka. Ovisno o veličini riječi instrukcije i specifikacijama mikroprocesora faza pribavljanja instrukcija se može izvršiti u jednom ili više ciklusa. U fazi izvršavanja, pomoću informacija iz dekodirane instrukcije, upravljačka jedinica generira odgovarajuće upravljačke signale koji se šalju na ulaz aritmetičko-logičke jedinice, koja zatim obavlja potrebne operacije i pohranjuje rezultat u registar čija je adresa navedena u adresnom dijelu instrukcije, a shodno rezultatu se postavljaju zastavice u registru uvjeta.



Slika 11. Povezanost komponenti mikroprocesora

Izvor: Budin, L.: *Mikroračunala i mikroupravljači*, 4. izdanje, Element, Zagreb, 2015., str. 2



Slika 12. Dijagram toka instrukcije

Izvor: Ribarić, S.: *Naprednije arhitekture mikroprocesora*, 2. Izdanje, Element, Zagreb, 2002., str. 118

4. ARHITEKTURE DANAŠNJICE

4.1. CISC

CISC ili Complex Instruction Set Computing arhitektura, još poznata pod imenom x86, se danas može pronaći u Intelovim i AMD-ovim mikroprocesorima koji se primjenjuju kod različitih poslužitelja i stolnih računala. Ideja CISC mikroprocesora jest izvršavanje programa u što manjem broju instrukcija. Radi toga su same instrukcije kompleksnije te se zbog toga broj ciklusa povećava. Primjer kompleksne instrukcije:

ADD 1200, 1201

Željena operacija je zbrajanje dvaju brojeva. U CISC arhitekturi programer nije primoran koristiti instrukcije koje dobavljaju, učitavaju ili pohranjuju podatke jer se te instrukcije već nalaze unutar ADD instrukcije, a operacija zbrajanja se izvršava direktno u memoriji. Na taj način instrukcija ADD dobavlja brojeve koji su na memorijskim lokacijama 1200 i 1201, zbroji ih te ih pohrani na memorijsku lokaciju 1200. Također, s ovim principom izvođenja instrukcija se smanjuje opterećenje kompilatora, a osim toga se i oslobađa prostor na nasumičnoj memoriji zahvaljujući tome što se veći dio rada odvija u sklopovlju. Zbog kompleksnih instrukcija koje se izvršavaju u većini elektroničkih sklopova zahtijeva se veći broj tranzistora što za posljedicu stvara skuplju i složeniju proizvodnju, a i s većim brojem sklopova raste i disipacija snage. Unatoč nedostacima, CISC arhitektura je prisutna i danas, a s razvojem mikroprocesora gdje brzine rada iznose nekoliko GHz i uvođenjem hiperdretvenosti, veći broj jezgri, turbo ubrzanja i ostalih tehnologija itekako su dobar odabir za generalni rad i izvršavanje kompleksnih programa. Primjer jednog CISC procesora može biti AMD-ov Ryzen 5 5600x.

4.2. RISC

RISC ili Reduced Instruction Set Computing arhitektura je nastala nakon CISC arhitekture s ciljem da se izrade mikroprocesori koji će brže i uspješnije izvoditi instrukcije. Umjesto velikog skupa kompleksnih instrukcija, u RISC se nalazi manji broj jednostavnih instrukcija. Neka je željena operacija zbrajanje dvaju brojeva kao što je bilo u prošlom poglavlju:

LOAD A, 1200

LOAD B, 1201

ADD A, B

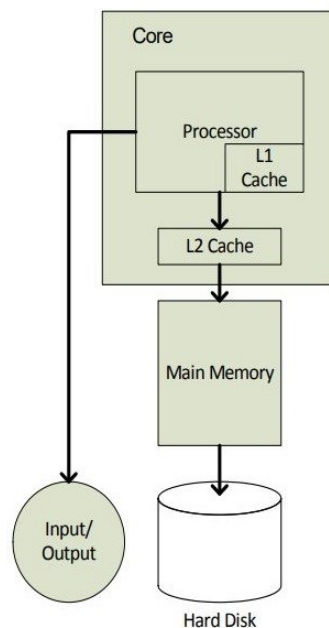
STORE 1200, A

U CISC arhitekturi operacija zbrajanja dvaju brojeva se izvršila pomoću samo jedne instrukcije. Kod RISC arhitekture to nije slučaj već se koristi više instrukcija kako bi se postigao isti rezultat. LOAD instrukcija učitava podatke koji se nalaze na memorijskoj lokaciji 1200 u registar A te ponavlja isti postupak za registar B. Zatim se s ADD instrukcijom podaci registara A i B zbroje, a rezultat se pohranjuje sa STORE instrukcijom u memorijsku lokaciju 1200. Izvršavanje instrukcije je u ovom kontekstu uspješnije jer se jednostavne instrukcije brže izvode čime se smanjuje broj ciklusa. Problem korištenja jednostavnih instrukcija je ukoliko neki program ima kompleksne instrukcije one se moraju rastaviti na veći broj jednostavnijih instrukcija. Pošto se proces rastavljanja kompleksnih instrukcija obavlja pomoću kompilatora njegovo opterećenje je veće nego kod CISC arhitekture. Zbog toga se veći naglasak stavlja na programska rješenja pri obradi podataka čime se smanjuje broj tranzistora tj. elektroničkih sklopova, prema tome se pojednostavljuje proizvodnja i smanjuje se disipacija snage. To čini RISC arhitekturu učinkovitijom od CISC u izradi i energetske smislu, ali se zbog većeg naglaska na programska rješenja koristi više nasumične memorije. Zbog energetske učinkovitosti RISC mikroprocesori se najčešće koriste kod sustava gdje je energetska učinkovitost prioritet. Jedan od RISC procesora koji se danas često koristi je Snapdragon 8 Gen 2.

4.3. Tehnologije današnjice

4.3.1. Višejezgreni procesor

Višejezgreni procesor (eng. *multicore processor*) je mikroprocesor koji u sebi sadrži dva ili više procesora, tj. jezgri koje služe za obradu podataka. U početku svi su mikroprocesori imali samo jednu jezgru, a brzina takta se poboljšavala iz godine u godinu smanjenjem dimenzija komponenti i ugradnjom većeg broja istih. Problem nastaje kod povećanja brzine takta gdje povećanjem brzine raste i potrošnja energije (povećanjem takta za 400 MHz potrošnja energije može porasti i za 60%¹⁵). Zbog toga je nastala ideja višejezgrenog procesora koji bi imao veći broj jezgri te su se u teoriji očekivale dvostruko veće performanse i smanjenje potrošnje energije, ali u praksi se pokazalo da veći broj jezgri radi s manjim brzinama nego jedna jezgra.



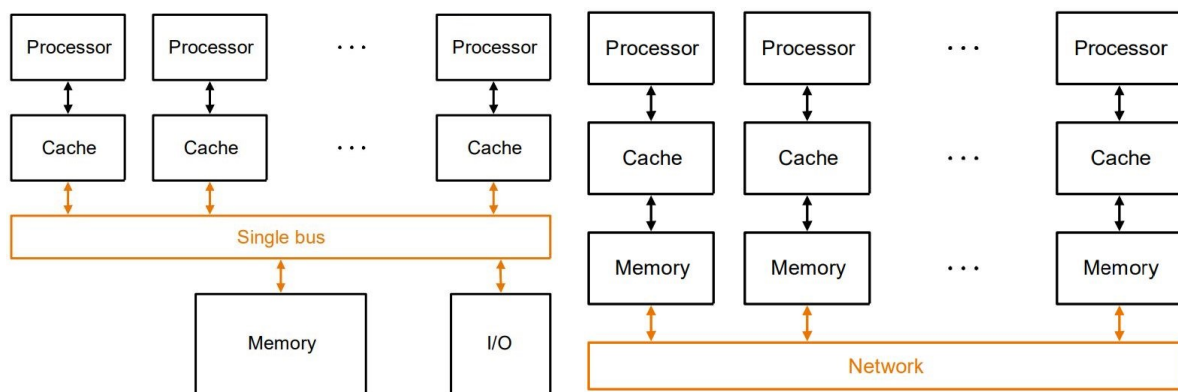
Slika 13. Grafički prikaz jezgre

Izvor: Schauer, B.: *Multicore processors – a necessity*, ProQuest discovery guides, ProQuest, 59, 2008, str. 4

Slika 13 grafički prikazuje dijelove jezgre i povezanost s ostalim dijelovima sustava. Jezgra se sastoji od procesora koji služi za izvršavanje operacija i priručne (eng. cache) memorije. Priručna memorija je memorija malenog kapaciteta i velike brzine koja je smještena na samom mikroprocesorskom čipu i nalazi se na spojnom putu između procesora i glavne memorije. Ona služi za pohranu podataka koje procesor često koristi. U

¹⁵ Schauer, B.: *Multicore processors – a necessity*, ProQuest discovery guides, ProQuest, 59, 2008, str. 4

usporedbi s ostalim memorijama priručna memorija je najbrža te se ona koristi kako bi se eliminirao problem *bottlenecka* u sustavu koji nastaje zbog sporog dohvata podataka iz glavne memorije. Oznake L1 i L2 priručne memorije označava razinu memorije. L1 je najmanja i najbrža te je tipično integrirana s procesorom. L2 je veća i sporija u usporedbi s L1 i može se nalaziti ili na mikroprocesorskom čipu ili izvan njega, ali u svakom slučaju je uvijek bliža procesoru nego glavna memorija. Iako nije prikazana na slici postoji i L3 cache memorija koja je veća i sporija od L1 i L2, a nalazi se izvan mikroprocesorskog čipa i zajedno ju koriste sve jezgre.



Slika 14. Međusobno povezivanje jezgri

Izvor: Schauer, B.: *Multicore processors – a necessity*, ProQuest discovery guides, ProQuest, 59, 2008, str. 5

Osim komunikacije jezgre s glavnom memorijom potrebno je ostvariti i komunikaciju između jezgri. To se najčešće postiže korištenjem konfiguracija koje prikazuje slika 14. S lijeve strane je prikazana konfiguracija s dijeljenom memorijom. Naziv proizlazi iz toga što sve jezgre koriste istu memoriju, a komunikacija je postignuta pomoću sabirnice. Na desnoj strani prikazana je konfiguracija gdje svaka jezgra ima svoju vlastitu memoriju, a komunikacija je ostvarena putem mreže.

Tablica 2. Usporedba jednojezgrenog i višejezgrenog procesora

	Jednojezgreni procesor (45 nm)	Višejezgreni procesor (45 nm)
V_{dd}	1 V	1 V
U/I pinovi (ukupno)	1280 (ITRS)	3000 (aproksimirano)
Radni takt	7.8 GHz	4 GHz
Brzina prijenosa čip-kućište	7.8 Gb/s	4 Gb/s
Propusnost	125 GiB/s	1 TB/s
Snaga	429.78 W	107.39 W
Ukupan broj pinova na čipu	3840	9000 (aproksimirano)
Broj pinova na kućištu	2480	4500 (aproksimirano)

Izvor: Schauer, B.: *Multicore processors – a necessity*, ProQuest discovery guides, ProQuest, 59, 2008, str. 5

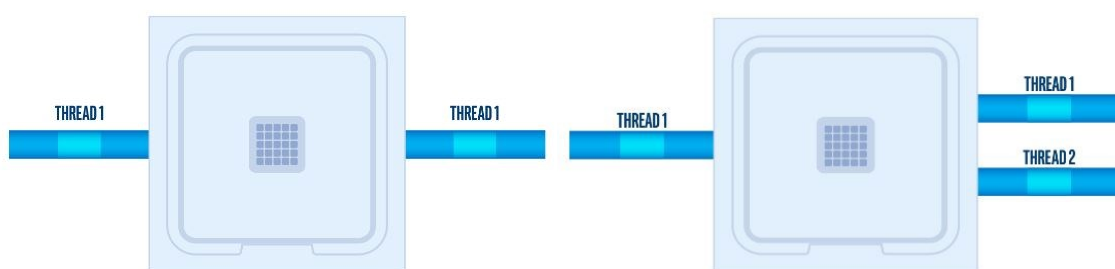
Razlika između jednojezgrenog i višejezgrenog procesora lako je uočljiva. Tablica 2 prikazuje rezultate ispitivanja jednojezgrenog i 8-jezgrenog procesora. Oba procesora imaju iste dimenzije komponenti u iznosu od 45 nm te su tijekom ispitivanja napajani s istim iznosom napona napajanja V_{dd} . Kao što je i prije navedeno, veći broj jezgri će najčešće raditi s manjim radnim taktom nego jedna jezgra, a tu tvrdnju potvrđuju i rezultati iz tablice gdje radni takt jednojezgrenog procesora iznosi 7.8 GHz, a višejezgrenog 4 GHz. Iako se naizgled gubi na brzini, kod višejezgrenog procesora se može uočiti skoro deset puta veća propusnost te još bitnije manja snaga tj. potrošnja energije. Osim performansi, prednost višejezgrenog mikroprocesora se još nalazi u mogućnosti multitaskinga i otpornosti na greške (programska greška na jednoj jezgri ne utječe na rad ostalih jezgri).

Višejezgreni mikroprocesori ovise o operativnom sustavu ili nekom drugom programu kako bi se na što učinkovitiji način iskoristile sve njegove jezgre. Neki programi ne iskorištavaju maksimalni potencijal svih jezgri, a u nekim slučajevima je povećanje performansi zanemarivo malo. Također, u usporedbi s jednojezgrenim procesorima, višejezgreni procesori prilikom mirovanja imaju veću potrošnju energije.

Unatoč nedostacima, višejezgreni procesori u današnje doba prevladavaju tržištem te su se pokazali kao bolja opcija u usporedbi s jednojezgrenim procesorima.

4.3.2. Hiperdretvenost

Hiperdretvenost (eng. *hyperthreading*) tehnologija, koju je proizveo Intel, je tehnologija koja je usko povezana s višejezgrenim procesorima. *Thread* ili nit je naziv za niz instrukcija koje su zadane od strane nekakvog programa. Jednojezgreni procesor može u bilo kojem trenutku procesirati samo jednu nit, a višejezgreni može više niti procesirati istovremeno zahvaljujući većem broju jezgri. Umjesto velikog opterećenja jedne jezgre, programi dijele posao na više softverskih niti. Ove niti paralelno obrađuju različite jezgre radi uštede vremena. Ovakav način obrade podataka stoga je i dobilo naziv multithreading.



Slika 15. Grafički prikaz hiperdretvenosti

Izvor: Intel: *What is hyperthreading?* <https://www.intel.com/content/www/us/en/gaming/resources/hyper-threading.html> (27.6.2024.)

Cilj hiperdretvenosti je da se od svake fizičke jezgre naprave dvije logičke ili virtualne jezgre. Na taj način se omogućuje da jedna jezgra istovremeno obrađuje veći broj niti. Objе logičke jezgre dijele resurse od pripadajuće fizičke jezgre, a u slučaju ako je jedna logička jezgra pod većim opterećenjem od druge, tehnologija hiperdretvenosti će te resurse raspodijeliti na način da se učinkovito iskoriste resursi i jezgre.

Povećanjem broja instrukcija koje se mogu istovremeno izvršavati, hiperdretvenost povećava propusnost mikroprocesora kao što je vidljivo na slici 15. Osim učinkovitog iskorištavanja resursa fizičke jezgre, očitavaju se bolje performanse kod kompleksnih programa (simulacije, moderne video igrice, 3D grafički programi i sl.) koji su dizajnirani na način da koriste više niti. Pošto se resursi fizičke jezgre dijele, u nekim slučajevima kod opterećenja logičkih jezgri može doći do nestašice resursa što uzrokuje smanjenje performansi. Iako hiperdretvenost povećava performanse za samo otprilike 30%¹⁶ i dalje je itekako korisna kod prije navedenih kompleksnih programa.

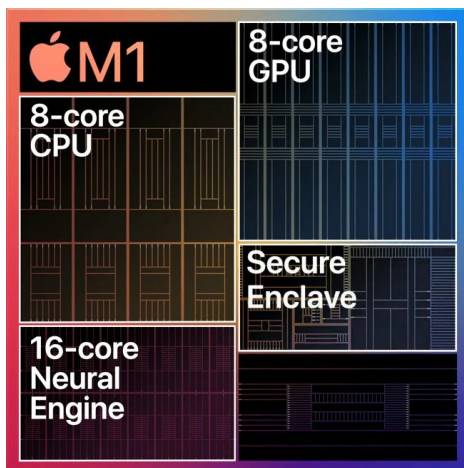
¹⁶ Magro, W., Petersen, P.: *Hyperthreading technology*, *Intel Technology Journal*, Intel, 6, 1, 2002, str. 14

4.3.3. Tehnologija turbo ubrzanja

Tehnologija turbo ubrzanja (eng. *Turbo Boost*), također razvijena od strane Intela, je tehnologija koja prati korištenje Intel Core procesora kako bi odredila koliko je procesor blizu maksimalne toplinske snage tj. dinamički mijenja brzinu takta procesora ovisno o opterećenju. Maksimalna toplinska snaga procesora je najveća snaga koju procesor treba iskoristiti. Ako tehnologija turbo ubrzanja otkrije da procesor dobro radi unutar svojih ograničenja, turbo ubrzanje se može aktivirati. Ova tehnologija je najviše korisna kod zahtjevnih programa gdje je brzina rada bitna te programa koji nisu optimizirani za korištenje više jezgri mikroprocesora. Tehnologija turbo ubrzanja je limitirana snagom i temperaturom mikroprocesora. Primjer: mikroprocesor čiji je nazivni radni takt 3 GHz će pri opterećenju kompleksnog programa podići svoj radni takt preko nazivne vrijednosti do maksimalne moguće frekvencije turbo ubrzanja od 3,6 GHz. Pri tome se uzimaju u obzir snaga i temperatura mikroprocesora, a maksimalna frekvencija turbo ubrzanja varira između modela mikroprocesora. „Intel Turbo Boost“ je dinamička značajka koja je dostupna za procesore Core i3, Core i5, Core i7 i Xeon. Ne sadrže svi Intel procesori značajku turbo ubrzanja, no to je uobičajena značajka za većinu procesora proizvedenih od 2008. godine.

4.3.4. Umjetna inteligencija

Razvojem umjetne inteligencije (skraćeno UI) veliki proizvođači mikroprocesora poput Intel, AMD i Apple-a su unazad nekoliko godina počeli integrirati umjetnu inteligenciju u svoje mikroprocesore s ciljem da se poboljšaju performanse mikroprocesora, pogotovo kod analize podataka u stvarnom vremenu i strojnog učenja. Integracija umjetne inteligencije i mikroprocesora se može izvesti u obliku uvođenja instrukcijskog skupa koji je specifično dizajniran za umjetnu inteligenciju, dodavanjem UI jezgre na čip mikroprocesora i slično. Današnje arhitekture Intel (VNNI¹⁷) i AMD (Zen 3) mikroprocesora opremljene su s instrukcijskim skupovima namijenjene za UI, dok primjerice Apple M1 mikroprocesor ima na sebi 16-jezgreni Neural Engine koji se koristi kod primjerice prepoznavanja ljudskog govora, analiza slika i slično. Izazovi koji se susreću integracijom UI su veća potrošnja energije i potrebna optimizacija softvera.



Slika 16. Pojednostavljena shema Apple M1 mikroprocesora

Izvor: *Neural Engine*, Apple Wiki, online: https://apple.fandom.com/wiki/Neural_Engine (27.6.2024.)

¹⁷ Vector Neural Network Instructions

5. ZAKLJUČAK

Budućnost mikroprocesora teško je predvidjeti. U zadnjih nekoliko godina napredak po pitanju brzine rada, veličine i gustoće tranzistora na čipu je usporio, a time se donosi pitanje vrijedi li i dalje Mooreov zakon, koji glasi da će se svake dvije godine broj tranzistora na čipu udvostručiti uz minimalno poskupljenje izrade mikroprocesora. Unatoč tome proizvođači i dan danas uspijevaju povećavati gustoću tranzistora na čipu. Istraživanja se i dalje odvijaju ne bi li se našao način za daljnje smanjenje tranzistora, ali to nije jedino područje gdje se analizira potencijalni napredak. Iako je silicij najčešći materijal koji će se koristiti i dalje u budućnosti, razmatra se uz silicij korištenje drugih materijala kako bi se poboljšala učinkovitost. Umjetna inteligencija se sve češće koristi i integrira u rad mikroprocesora. Mikroprocesori su također našli svoje mjesto u kvantnim računalima koji su tek u svojim počecima. U svim navedenim granama, a postoje ih i još više, mikroprocesori se nastavljaju i dalje razvijati. Pri tome treba uzeti u obzir potrošnju energije, cijenu izrade i ostale limitirajuće faktore. Unatoč svemu, mikroprocesori i ostale tehnologije vezane za njih će se i dalje razvijati dugi niz godina, gdje će i dalje biti bitan dio svakodnevnog života.

LITERATURA

1. Budin, L.: *Mikroročunala i mikroupravljači*, 4. izdanje, Element, Zagreb, 2015.
2. Ribarić, S.: *Naprednije arhitekture mikroprocesora*, 2. Izdanje, Element, Zagreb, 2002.
3. Magro, W., Petersen, P.: *Hyperthreading technology*, *Intel Technology Journal*, Intel, 6, 1, 2002.
4. Peruško, U.: *Digitalna elektronika: logičko i električko projektiranje*, Školska knjiga, Zagreb, 1991
5. Schauer, B.: *Multicore processors – a necessity*, *ProQuest discovery guides*, ProQuest, 59, 2008.
6. Betker, M. R., Fernando, J. S., Whalen, J. P.: *The Birth, Evolution and Future of Microprocessor*, *Bell Labs technical journal*, Bell System Memorial, 1997.
7. Maini, A. K.: *Digital Electronics: Principles And Integrated Circuits* 1. izdanje, Wiley India, 2007
8. *Rad sa složnikom i posluživanje zahtjeva za prekidom*, Fakultet prometnih znanosti, Zagreb, <https://www.fpz.unizg.hr/hgold/es/de/zahtjev.htm> (10.5.2024.)
9. *RISC and CISC in Computer Organization*, 13.3.2024., online: <https://www.geeksforgeeks.org/computer-organization-risc-and-cisc/> (28.6.2024)
10. Intel: *What is hyperthreading?* <https://www.intel.com/content/www/us/en/gaming/resources/hyper-threading.html> (27.6.2024.)
11. *Neural Engine*, Apple Wiki, online: https://apple.fandom.com/wiki/Neural_Engine (27.6.2024.)

KAZALO KRATICA

Kratika	Puni naziv	Tumačenje
ALU	eng. Arithmetic Logic Unit	Aritmetičko-logička jedinica
BGA	eng. Ball-Grid-Array	Vrsta kućišta mikroprocesora
CISC	eng. Complex Instruction Set Computing	Arhitektura mikroprocesora
CPU	eng. Central Processing Unit	Mikroprocesor
FPU	eng. Float Point Unit	Jedinica za kompleksne aritmetičke operacije
LGA	eng. Land-Grid-Array	Vrsta kućišta mikroprocesora
LIFO	eng. Last In, First Out	Vrsta memorijske strukture
LSI	eng. Large-Scale Integration	Stupanj integracije
MOSFET	eng. Metal Oxide Semiconductor Field Effect Transistor	Poluvodički ventil
MSI	eng. Middle-Scale Integration	Stupanj integracije
PGA	eng. Pin-Grid-Array	Vrsta kućišta mikroprocesora
PLA	eng. Programmable Logic Array	Uređaj s programabilnim logičkim sklopovima
RAM	eng. Random Access Memory	Memorija s nasumičnim pristupom
RISC	eng. Reduced Instruction Set Computing	Arhitektura mikroprocesora
ROM	eng. Read Only Memory	Vrsta memorije koja može samo čitati podatke
SIMD	eng. Single Instruction Multiple Data	Jedinica za kompleksne aritmetičke operacije
SSI	eng. Small-Scale Integration	Stupanj integracije
U/I	hrv. Ulaz/Izlaz	Periferija poput tipkovnice, miša, monitora i sl.
UI	hrv. Umjetna Inteligencija	Grana digitalne tehnologije
ULSI	eng. Ultra-Large Scale Integration	Stupanj integracije
VLSI	eng. Very-Large Scale Integration	Stupanj integracije
VNNI	eng. Vector Neural Network Instructions	Intel arhitektura

POPIS TABLICA

Tablica 1. Razvoj stupnja integracije integriranih krugova kroz povijest.....	5
Tablica 2. Usporedba jednojezgrenog i višejezgrenog procesora	27

POPIS SLIKA

Slika 1. Ilustracija silicijske pločice i čipa tijekom proizvodnje	2
Slika 2. Pojednostavljeni model mikroprocesora	9
Slika 3. Usporedba arhitekture s jednim akumulatorom i strukture s dva akumulatora	10
Slika 4. Raspored registara MC 6800 mikroprocesora	12
Slika 5. Prikaz stogovnog načina pohranjivanja podataka	13
Slika 6. Sklop za postavljanje zastavica C,V,N i Z u registru uvjeta	14
Slika 7. Prikaz jednog stupnja aritmetičko-logičke jedinice.....	15
Slika 8. Prikaz aritmetičke sekcije	16
Slika 9. Prikaz logičke sekcije	17
Slika 10. Prikaz organizacije upravljačke jedinice	20
Slika 11. Povezanost komponenti mikroprocesora.....	21
Slika 12. Dijagram toka instrukcije	22
Slika 13. Grafički prikaz jezgre	25
Slika 14. Međusobno povezivanje jezgri	26
Slika 15. Grafički prikaz hiperdretvenosti	28
Slika 16. Pojednostavljena shema Apple M1 mikroprocesora	30