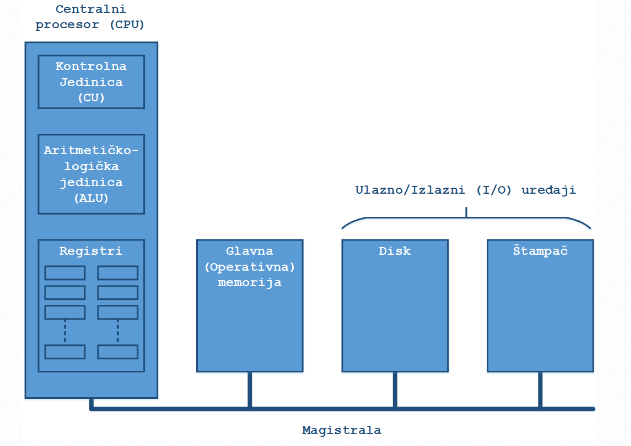
**L02 – Organizacija racunara**

***PROCESORI***

Centralna procesorka jedinica je mozak racunara. 

Funkcija CPU-a je da izvrsava programe uskladistene u operativnoj memoriji racunara tako sto preuzima njihove instrukcije onda ih deokdira I izvrsava jednu za drugom.

Komponente su povezana magistralom (Bus) koja je ustvari skup paralelnih provodinka.

Magistrale mogu prenositi adrese, podatke I kontrolne signale. Magistrale isto mogu da budu eksterne tako sto povezuju CPU sa drugim uredjajima a moze biti I interna koja se naalzi unutar CPU-a.

**DELOVI CENTRALNOG PROCESORA**

**Kontrolna jedinica** je odgovorna za preuzimanje instrukcija iz glavne memorije I odredjivanja tipa instrukcije. Ovo se zove dekodovanje instrukcija.

**Aritmeticko logicka jedinica** obavlja operacije kao sto su sabiranje, oduzimanje, pomeranje I logicke operacije AND ili OR.

CPU takodje sadrzi male memorijske jedinice velike brzine koje se koriste za privremeno skladistenje rezultata. Ove brze memorije cine niz **registara** od kojih svaki ima odredjenu velicinu tj kapacitet. Svaki registar moze sadrzati broj od 0 do odredjenog maksimuma koji je odredjen kapacitetom registra. Registri se nalaze na integrisanom kolu Procesora I kao takvi mogu citati I pisati velikom brzinom.

Najvazniji registar je **Programski Brojac** i on ima zadatak da ukaze na sledecu funkciju koja treba da se izvrsi. .

**Registar instrukcija** je isto vazan deo koji sadrzi instrukciju koja se trenutno izvrsava.

***INSTRUKCIJE PROCESORA***

Vecina instrukcija se mogu podeliti na:

-registar-memorija

-registar-registar

Istrukcije tipa registar memorija omogucavaju da se podaci iz memorije preuzimaju u registre ili iz registara u memoriju.

Druga vrsta instrukciaj je registar-registar. Primer instrukcije registar-registar - preuzima dva operanda iz registra, prebacuje u ALU ulazne registre I izvodi neku operaciju nad njima kao sto je sabiranje ili logicka AND, OR. Na kraju se rezultat vraca u neki od registara.

Proces pokretanja dva operanda kroz ALU I cuvanje rezultata se naziva **ciklus putanje podatka** (data path cycle) I predstavlja srce vecine CPU-a. Ovim ciklusom je definisano sta racunarska masina moze da uradi.

**CIKLUS IZVRSENJA INSTRUKCIJA**

CPU izvrsava instrukcije u nekoliko malih koraka:

-Nabavi sledecu instrukciju iz memorije I stavi je u instrukcijski registar

-Promeni programski brojac da pokazuje na sledecu instrukciju

-Odredi tip pribavljene instrukcije (sabiranje, oduzimanje)

-Ukoliko instrukcija koristi podatke iz memroije, pronadji taj deo memorije

-Pribavi deo memorije I stavi u CPU registar

-Izvrsi instrukciju

-Vrati se na korak 1 I kreni sa izvrsavanjem sledece instrukcije

Ovaj niz koraka se naziva **pribavi-dekoduj-izvrsi (fetch-decode-execute)**

**KOMPLEKSNE INSTRUKCIJE**

Primeri kompleksnih instrukcija:

-Instrukcija sa pokretnom zapetom

-Direktan pristup elementu liste/niza

Vrlo vesto kompleksnu funkciju moze zameniti niz jednostavnih funkcija.

***PROEJKTOVANJE SAVREMENIH PROCESORA***

Savremeni x68/x64 CISC arhitekture uz neke primene RICS-a.

Kasnih 70tih se eksperimentisalo sa kompleksnim instrukcijama da bi se smanjila razlika u tome sta mogu masine da zavrse I sta mogu da naprave jezici visokog nivoa.

Ovakvi procesori su nazvani **CISC (Complex Instruction Set Computer)**

**RICS** koncept je pokrenut 80tih godina I zasnovao se na brzom izvrsavanju jednostavnih instrukcija. (**Reduced Instruction Set Compputer**)

**POTREBA ZA PARALELNIM IZVRSENJEM INSTRUKCIJA**

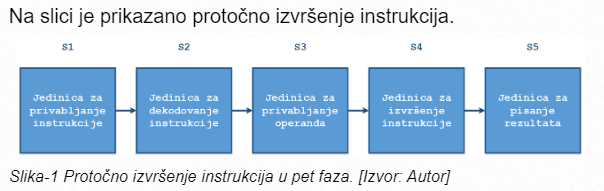
Danas se ubrzanje procesora ogleda u paralelizmu, tj izvrsenje vise instrukcija odjendom.

**Paralelizam** se moze ostvariti na nivou instrukcije I na nivou procesora

* U prvom slucaju paralelizam se ostvaruje izvrsavanjem vise instrukcija u sekundi.
* U drugom slucaju se postize kada vise CPU-a radi na resavanju jednog problema.

**PROTOCNO IZVRSENJE INSTRUKCIJA**

Protocno izvrsenje instrukcija radi na principu podele instrukcijskog ciklusa na vise delova gde je svaka faza (**stage**) dodeljena jedinstvenom hardveru od kojih svaki moze raditi paralelno.



S1 – Preuzima instrukciju iz memoriej I smesta je u bafer dok ne bude potrebna

S2 – Dekoduje instrukciju odredjujuci njen tip I operande koji su potrebni toj instrukciji

S3- Pronalazi I preuzima operande iz registara ili iz operativne memorije

S4 – Izvrsava instrukcije

S5 – upisuje podatke

**OPERATIVNA MEMORIJA**

**Memorija** je deo racunara gde se cuvaju podaci. Osnovna jednicia memorije je binarna cifra nazvana **bitom.**

Binarni brojni sistemima samo dve vrednosti koje treba razlikovati I zbog toga se smatra najpouzdanijom metodom kodiranja digitalnih informacija.

**MEMORIJSKA CELIJA**

Bitovi se grupisu u bajtove a bajtovi u memorijske reci.

Memorija se sastoji od celija gde svaka moze da sadrzi neku informaciju. Svaka celija sadrzi broj koji se naziva adresa memorijske lokacije kojoj pristupaju programi.

Ukoliko memorija ima n celija ona moze sadrzati od 0 do n na 1 adresa.

Ukoliko se celija sastoji od k bitova onda moze da sadrzi bilo koju informaciju od 2 na k kombinacija.

Racunar sa 32-bitnom recju ima 4 bajta po reci, dok onaj sa 64-bita ima 8 bajta po reci.

**UBRZANJE OPERATIVNE MEMORIJE**

Kada memoriju malog kapaciteta postavimo na CPU, I ako je kombinujemo sa memorijom velikog kapaciteta koja je sporija, dobicemo performanse koje su bolje nego kao kad bi celu operativnu memorijji stavili na CPU.

Ovo je dovelo do kes memorije. Najcesce koriscene reci se smestaju u kes memoriji. CPU trazi neku rec u kes memoriji a ako se ona ne nalazi tamo trazi je od glavne memorije.

**SISTEM NA CIPU**

Savremeni mobilni telefoni iamju koncept koji se naziva sistem na cipu. To je kada je sve smesteno na CPU kao sto je Apple M1 ili M2.