



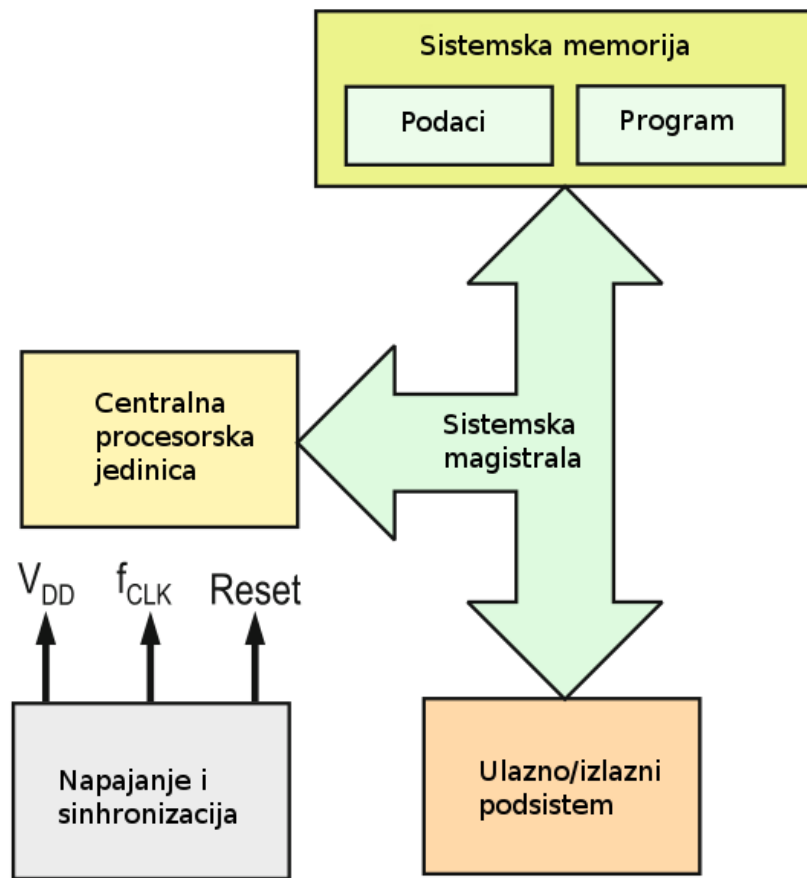
Arhitektura mikrokontrolera

Katedra za elektroniku

Osnovna arhitektura mikroračunarskog sistema

Tipičan mikroračunarski sistem sadrži sledeće fundamentalne komponente:

- **Centralna procesorska jedinica (CPU)** predstavlja ključni deo svakog mikroračunarskog sistema. CPU čita instrukcije iz programske memorije, dekoduje ih i izvršava ih. Osim toga, CPU je zadužen i za adekvatno korišćenje perifernih jedinica koje sačinjavaju ulazno/izlazni podsistem.
- **Sistemska memorija** zadužena je za smeštanje programa i podataka, koji se koriste od strane centralne procesorske jedinice. U okviru mikrokontrolera, obično su prisutna dva tipa memorije: programska memorija i memorija podataka.
 - U **Programsku memoriju** smešta se program u formi sekvence mašinskih instrukcija koju procesor izvršava. Program određuje funkcionalnost celokupnog sistema.
 - **Memorija za podatke** ima ulogu čuvanja podataka (promenljivih, varijabli) koji se obrađuju tokom izvršavanja programa.
- **Ulazno/Izlazni podsistem** uključuje sve komponente, tj. perifernijske uređaje koji omogućavaju centralnom procesoru da razmenjuje informacije sa drugim komponentama sistema i vrši interakciju sa spoljašnjim okruženjem.
- **Sistemske magistrale** predstavljaju skup linija koje povezuju CPU, memoriju i ulazno/izlazni podsistem. Postoje grupe linija koje obavljaju različite funkcije unutar sistema: **adresna magistrala**, **magistrala podataka** i **kontrolna magistrala**.



Arhitektura tipičnog mikroračunarskog sistema

Mikroprocesori opšte namene

- U slučaju sistema baziranih na mikroprocesorima opšte namene, osnovne komponente sistema (CPU, memorija, ulazno/izlazni uređaji) su odvojena integrisana kola. Ova kola se montiraju na jednu, ili više štampanih ploča (PCB) u okviru kojih su povezana provodnim vezama.
- Mikroprocesori opšte namene obično se odlikuju i nekim od naprednijih karakteristika kao što su:

- Optimizovana arhitektura u cilju preuzimanja programa i podataka iz eksterne memorije (skrivena memorija, engl. *cache*);
- Mogućnost protočne obrade više instrukcija istovremeno (engl. *pipelining*);
- Predviđanje grananja (engl. *branch prediction*);
- Postojanje numeričkih ko-procesora, itd.

- Vodeći proizvođači mikroprocesora opšte namene:

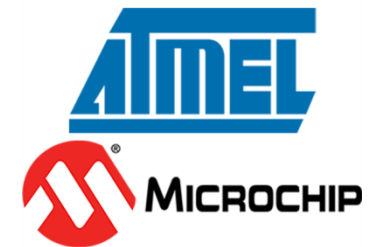


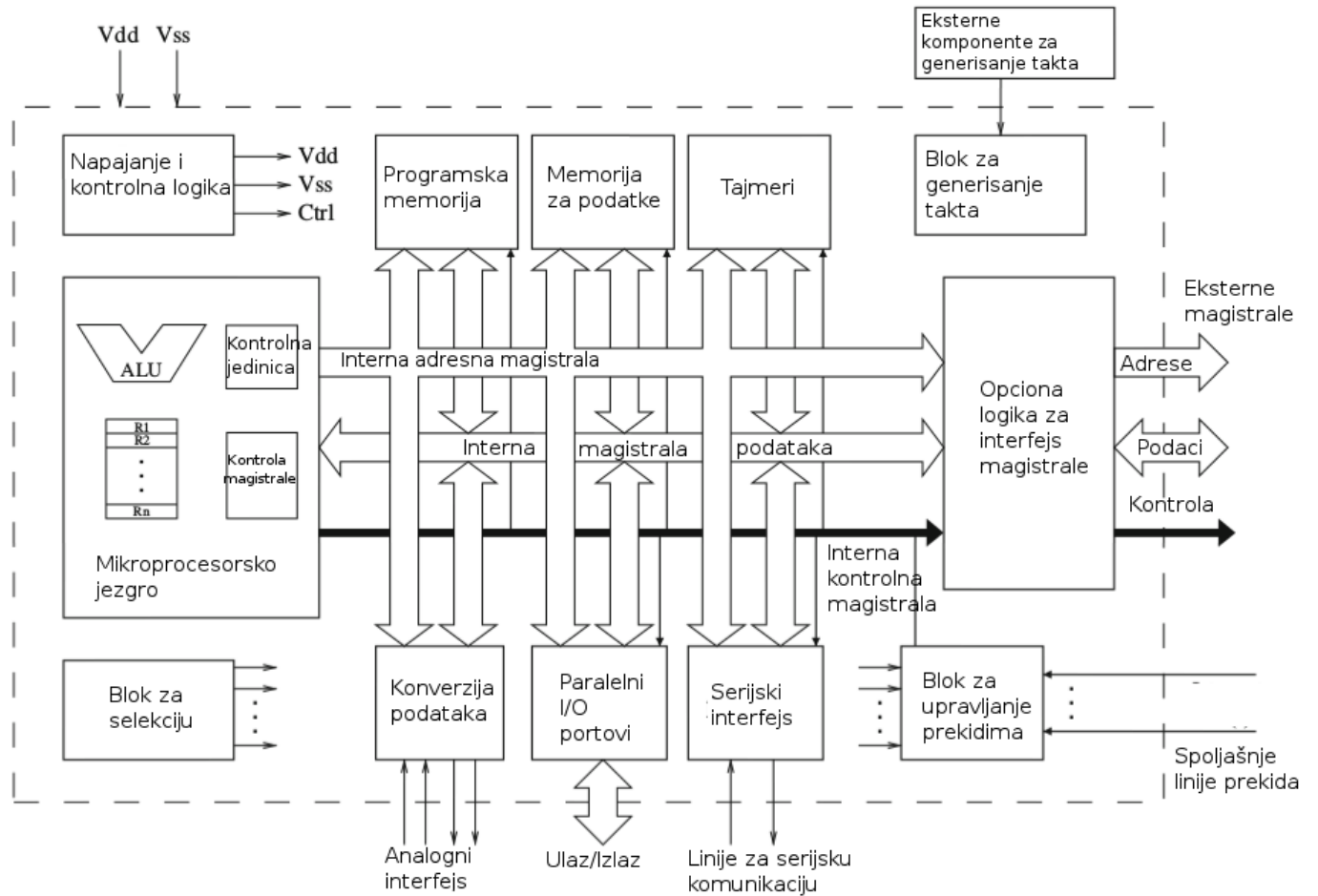
- Vodeći proizvođači mikroprocesora specijalizovane namene (DSP, GPU):



Mikrokontroleri

- U okviru mikrokontrolera, osnovne komponente sistema su realizovane u okviru istog integrisanog kola. Ovo omogućava fleksibilnost i realizaciju kompletnog sistema korišćenjem minimalnog broja eksternih komponenti.
- U poređenju sa mikroprocesorima opšte namene, obično ih odlikuje manja složenost, niža cena i niža brzina rada, tj. frekvencija sistemskog takta.
- U okviru mikrokontrolera obično se nalaze sledeće periferijske jedinice:
 - Grupe digitalnih ulazno/izlaznih priključaka opšte namene (portovi)
 - Brojači i tajmeri
 - Podsystem za generisanje i obradu prekida
 - Interfejsi za serijsku komunikaciju
 - Komparatori, A/D i D/A konvertori
- Trenutno najzastupljenije familije mikrokontrolera na tržištu:
 - PIC (Proizvođač Microchip)
 - AVR (Proizvođač Microchip, ranije Atmel)
 - ARM (ARM holdings + razni proizvođači)
 - 8051 (Intel + razni proizvođači)





Interna struktura mikrokontrolera

RISC i CISC arhitektura

- Mikroračunarski sistemi se uglavnom optimizuju po pitanju softvera, ili hardvera. U skladu sa tim, postoje dve standardne arhitekture mikroračunara: CISC i RISC.
- **CISC (Complex Instruction Set Computing)** mašine karakteriše:
 - Promenljiva dužina instrukcijske reči (različit broj bita se koristi za kodovanje instrukcija)
 - Manja dužina programa u odnosu na RISC arhitekture
 - Instrukcije koje se izvršavaju tipično u nekoliko sukcesivnih perioda sistemskog takta.
- **RISC (Reduced Instruction Set Computing)** procesore odlikuje:
 - Fiksirana dužina instrukcijske reči, tipično 1 bajt
 - Veća dužina programa u odnosu na CISC
 - Jednostavniji hardver u odnosu na CISC procesore
 - Za izvršenje pojedinačne instrukcije obično je dovoljna jedna perioda sistemskog takta.



Centralna procesorska jedinica (CPU) - hardverske i softverske komponente

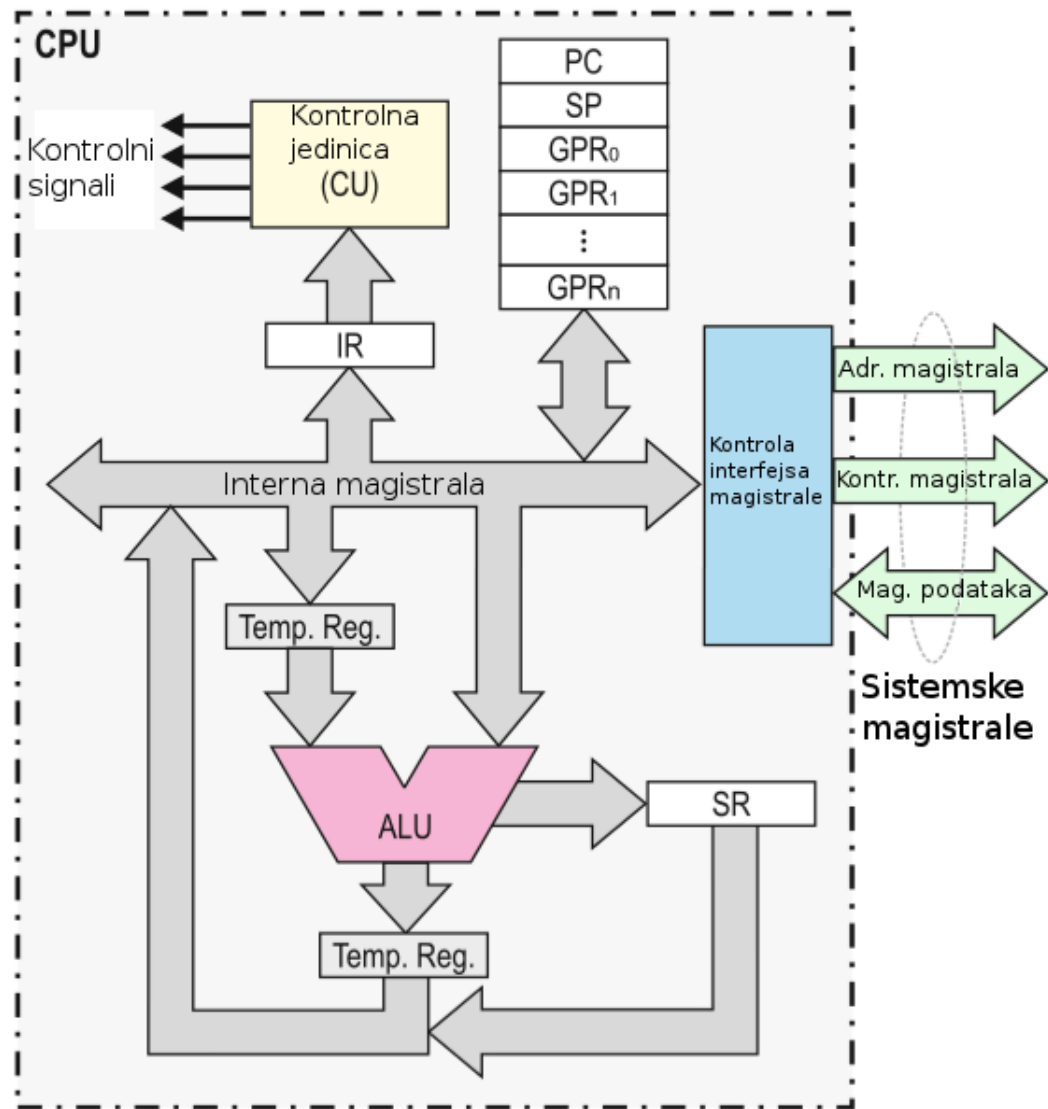
- Centralna procesorska jedinica predstavlja najvažniji deo mikroračunarskog sistema.
- CPU je zadužen za izvršavanje instrukcija, odnosno njegova uloga je generisanje odgovarajuće sekvence signala koja proizvodi akciju koja se izvršava na hardveru mikroračunarskog sistema.
- Minimalan skup komponenti koje sacinjavaju arhitekturu centralne procesorske jedinice su:

Hardverske komponente

- Aritmetičko-Logička jedinica (engl. *Arithmetic Logic Unit - ALU*)
- Kontrolna jedinica (engl. *Control Unit - CU*)
- Skup registara
- Kontrola interfejsa magistrala

Softverske komponente

- Skup mašinskih instrukcija
- Modovi (načini) adresiranja

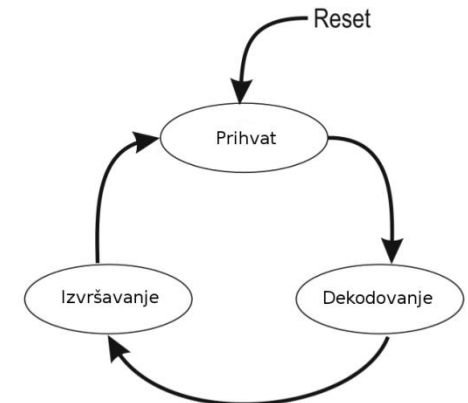


Centralna procesorska jedinica (CPU)

Kontrolna jedinica (CU)

Kontrolna jedinica (CU) upravlja funkcionalnošću centralne procesorske jedinice, implementirajući konačni automat (engl. *Finite State Machine - FSM*) koji ciklično ponavlja tri stanja:

1. **Prihvat instrukcije:** Tokom stanja prihvata nove instrukcije, instrukcija se iz programske memorije učitava u CPU. Programski brojač (PC) obezbeđuje adresu u memoriji sa koje se instrukcija čita. Adresirana instrukcija se preko magistrale podataka prebacuje u instrukcijski registar (IR).
2. **Dekodovanje instrukcije:** Nakon prihvata instrukcije, CU prelazi u stanje dekodovanja, u kome se značenje instrukcije "dešifruje". Dekodovana informacija se koristi kako bi se slali odgovarajući upravljački signali ka CPU komponentama u cilju izvršavanja aktivnosti predviđenih samom dekodovanom instrukcijom.
3. **Izvršavanje instrukcije:** U ovom stanju, CU šalje komande odgovarajućim funkcionalnim jedinicama u okviru CPU u cilju izvršavanja aktivnosti određenih instrukcijom. Na kraju ove faze, sadržaj programskog brojača se inkrementira, kako bi pokazivao na adresu sledeće instrukcije u programskoj memoriji, koja će se izvršiti tokom narednog instrukcijskog ciklusa.



Aritmetičko-logička jedinica (ALU)

Aritmetičke operacije:

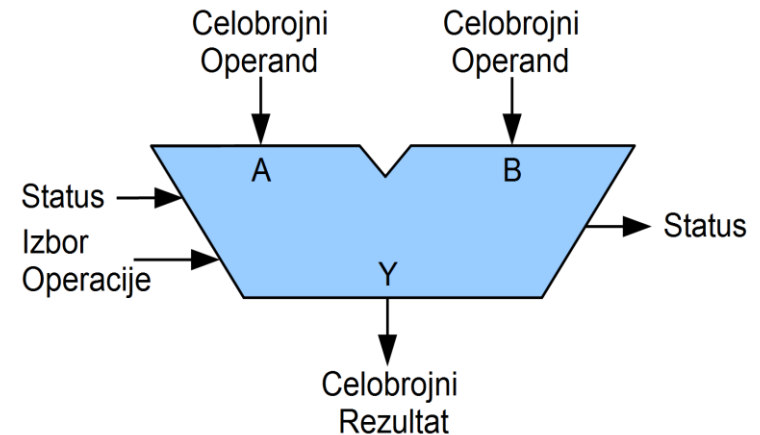
- Sabiranje (sa ili bez prenosa)
- Oduzimanje (sa ili bez pozajmice)
- Inkrement
- Dekrement
- Drugi komplement
- Množenje
- Deljenje

Logičke operacije:

- Bitsko I (AND)
- Bitsko ILI (OR)
- Bitsko EKS-ILI (XOR)
- Bitska negacija (NOT)

Operacije pomeranja:

- Pomeranje (ulevo ili udesno)
- Rotacija (ulevo ili udesno)

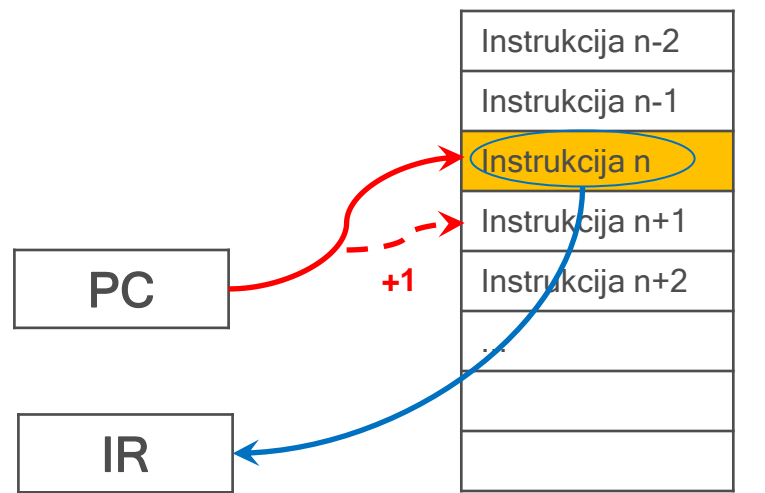


Najvažniji registri mikroprocesora

- Registri procesora omogućavaju privremeno smeštanje podataka, memorijskih adresa i kontrolnih informacija, tako da im se može brzo i jednostavno pristupiti.
 - Oni predstavljaju najbržu memoriju u mikroračunarskom sistemu, koja ujedno ima i najmanji kapacitet.
 - Sadržaj registara u okviru CPU gubi se prestankom napona napajanja.
 - U načelu, registri se mogu podeliti u dve grupe:
1. **Registri opšte namene** su registri koji nisu "vezani" za specifične funkcije procesora i mogu da se koriste za smeštanje podataka, promenljivih ili pokazivača na adrese, po potrebi. U skladu sa ovim, često se u literaturi klasifikuju kao adresni ili registri za podatke. U zavisnosti od arhitekture procesora, CPU može sadržati do nekoliko desetina registara opšte namene.
 2. **Registri specijalne namene** su registri koji su zaduženi za specifične funkcije u okviru CPU. Navažniji registri koji su sadržani u okviru CPU strukture su:
 - Instrukcijski registar (IR)
 - Programski brojač (engl. *Program Counter* - PC) koji se često naziva i instrukcijski pokazivač (engl. *Instruction Pointer* - IP)
 - Pokazivač steka (engl. *Stack Pointer* - SP)
 - Statusni registar (SREG), koji sadrži indikatore ("zastavice", engl. *Flags*)

Programski brojač (PC) i instrukcijski registar (IR)

- **Instrukcijski registar (IR)** prima instrukcije iz programske memorije, kako bi kontrolna jedinica mogla da ih dekoduje i izvrši. Akcija prenosa instrukcije iz memorije u IR naziva se prihvatanje instrukcije.
- **Programski brojač (PC)** sadrži adresu instrukcije koja se učitava iz programske memorije u instrukcijski registar. Često se naziva i instrukcijski pokazivač.
- Svaki put kada se instrukcija prihvata i dekoduje, automatski se inkrementira vrednost PC registra, kako bi pokazivao na narednu instrukciju koja će se izvršavati tokom sledećeg instrukcijskog ciklusa. Izuzetak od ovakvog ponašanja predstavljaju instrukcije skoka i poziva potprograma.
- Obzirom da PC sadrži adresu, njegova širina mora biti usklađena sa kapacitetom programske memorije.



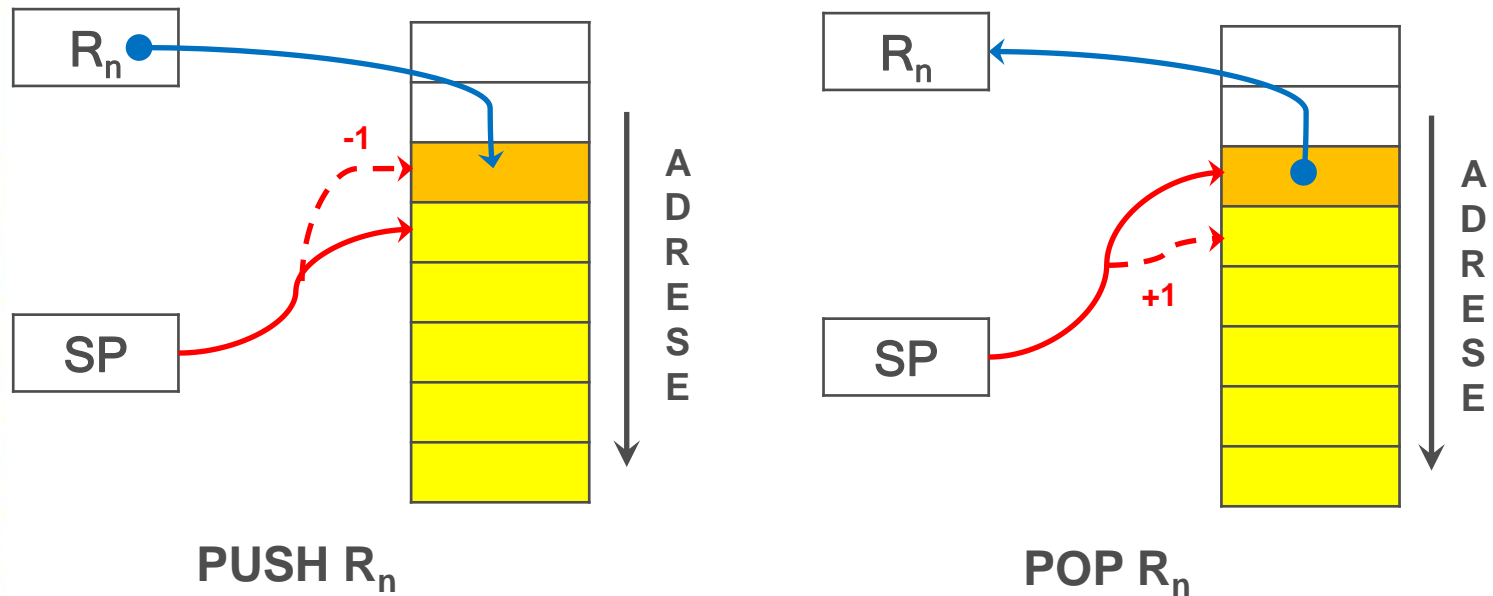
Programska memorija

Pokazivač steka (SP)

- Stek je memorijska struktura organizovana po principu **LIFO** (engl. *Last-In-First-Out*). Termin "stek" (engl. *stack*) je preuzet iz analogije sa slaganjem tanjira koje se vrši u prirodnom LIFO maniru: tanjir koji je poslednji stavljen na gomilu je onaj koji mora prvi biti uzet sa gomile, u suprotnom postoji realna opasnost da će tanjiri pasti i razbiti se.
- U datom trenutku je moguć pristup samo jednom podatku i to poslednjem koji je postavljen na stek (za njega se kaže da se nalazi na vrhu steka).
- Prilikom rada sa stekom postoje dve osnovne operacije:
 - Postavljanje podatka na vrh steka (PUSH)
 - Preuzimanje podatka sa steka (POP)
- Struktura steka realizuje se u okviru memorije za podatke. Pokazivač steka (**SP registar**) sadrži adresu na kojoj se nalazi podatak koji se trenutno nalazi na vrhu steka.
- Stek se može popunjavati od nižih ka višim adresama, ili češće od viših ka nižim adresama. Unapred se određuje početna adresa koja će inicijalno biti upisana u SP. Prilikom svakog dodavanja novog podatka na stek, ili preuzimanja podatka sa steka, SP se automatski inkrementira, ili dekrementira, u zavisnosti od operacije.

• U primerima prikazanim na sledećoj slici, prikazane su operacije PUSH i POP u kojima učestvuje registar R_n , pri čemu se stek popunjava od viših ka nižim adresama:

- Prilikom **PUSH** operacije, prvo se dekrementira sadržaj SP, a zatim se sadržaj registra R_n upisuje u memorijsku lokaciju čija adresa se nalazi u SP.
- Prilikom **POP** operacije, prvo se sadržaj memorijske lokacije koja je na vrhu steka (tj. čija adresa je u SP) upisuje u registar R_n , a potom se inkrementira sadržaj SP.



Statusni registar (SREG)

- Statusni registar sadrži skup indikatorskih bita ("zastavica", engl. *Flags*), koji se automatski setuju/resetuju u zavisnosti od rezultata aritmetičke ili logičke instrukcije koju obavlja ALU.
- Broj statusnih bita, kao i stanje koje je signalizirano takvim statusnim bitima su najčešće zavisni od same arhitekture mikroprocesora. Najčešći indikatorski biti koji su implementirani unutar većine procesora su:
 - **Indikator nule (Z - engl. *Zero Flag*)** se postavlja na logičku 1 kada je rezultat operacije jednak nuli, u suprotnom se postavlja na 0.
 - **Indikator prenosa (C - engl. *Carry Flag*)** se postavlja na 1 u slučajevima da je operacija dala rezultat sa prenosom (npr. sabiranjem dva 8-bitna broja je dobijen 9-bitni broj). Pojedine instrukcije koje vrše pomeranje podataka takođe utiču na sadržaj ovog indikatora.
 - **Indikator znaka (N - engl. *Negative Flag, Sign Flag*)** se postavlja na 1 kada je rezultat operacije obavljene nad označenim vrednostima negativan, dok je 0 u suprotnom.
 - **Indikator prekoračenja (V - engl. *Overflow Flag*)** signalizira prekoračenje prilikom operacije sabiranja ili oduzimanja označenih brojeva, koji mogu biti kako pozitivni tako i negativni.

Uticaj instrukcija sabiranja na stanje indikatora u okviru statusnog registra

PRIMER: 4Ah + 79h = C3h

```
  01001010
+ 01111001
-----
  01100011
```



C	Z	N	V
0	0	1	1

PRIMER: B4h + 4Ch = 100h

```
  10110100
+ 01001100
-----
  10000000
```



C	Z	N	V
1	1	0	0

PRIMER: 9Ah + B9h = 153h

```
  10011010
+ 10111001
-----
  101010011
```



C	Z	N	V
1	0	0	1

PRIMER: CAh + 1Bh = E5h

```
  11001010
+ 00011011
-----
  011100101
```



C	Z	N	V
0	0	1	0

AVR arhitektura

- AVR familija mikrokontrolera koristi Harvard arhitekturu sa odvojenim memorijskim prostorima za programske instrukcije i podatke (promenljive).
- Instrukcije iz programske memorije izvršavaju se protočnom obradom prvog nivoa: dok se jedna instrukcija izvršava, naredna instrukcija se zahvata iz programske memorije. Ovaj koncept omogućava da se u svakom taktu izvrši jedna instrukcija.
- Programska FLASH je memorija može da se reprogramira direktno u sistemu (engl. In-System-Programmable).
- 32 osmobitna registra opšte namene sa brzim pristupom (R0-R31) omogućavaju pristup podacima u jednom ciklusu sistemskog takta

