

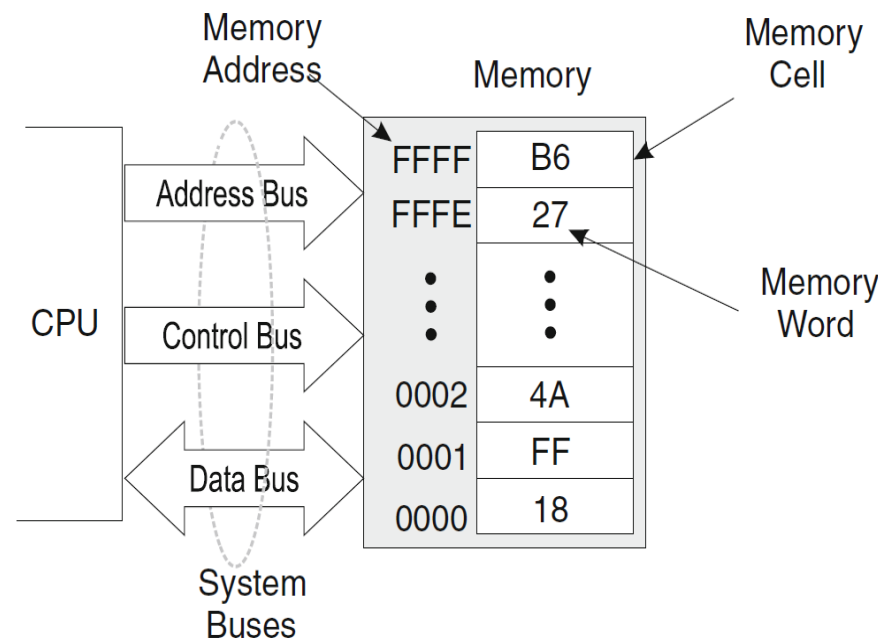


Organizacija memorije

Katedra za elektroniku

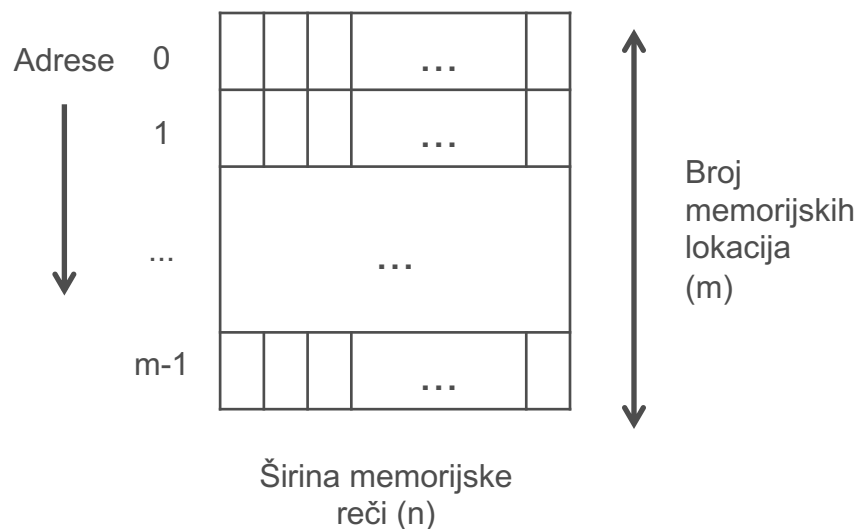
Organizacija memorije I

- Memorijski podsistem služi za **smeštanje instrukcija programa** kao i **podataka** koje taj program obrađuje
- Memorija se sastoji od velikog broja hardverskih komponenti koje su u stanju da skladište **1 bit informacije**
- Ovi bitovi su tipično organizovani u ***n*-bitne reči**, koje se tipično nazivaju registrima ili lokacijama
- **Sadržaj** svake memorijske lokacije zove se **memorijska reč**
- Pored toga, svaka memorijska lokacija identifikuje se pomoću jedinstvenog identifikatora, **memorijske adrese**
- Procesor koristi memorijsku adresu da bi pristupio željenoj memorijskoj lokaciji kako bi u nju upisao ili pročitao memorijsku reč



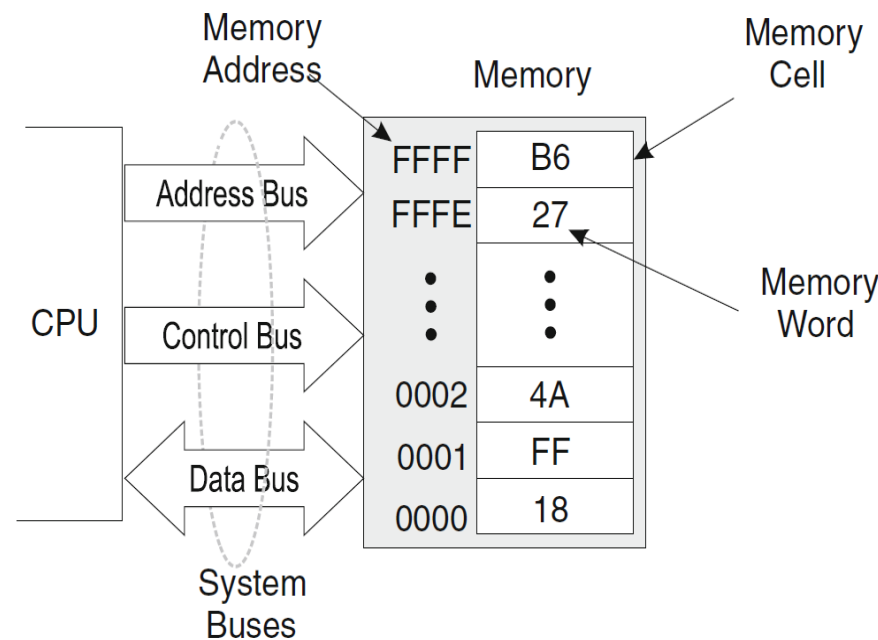
Organizacija memorije II

- U opštem slučaju, **memorijska jedinica** sastoji se iz **m memorijskih lokacija** veličine n bita
- Obično se memorija označava kao **$m \times n$ memorija**
- Kapacitet memorije navodi se brojem iza kojega sledi slovo b ili B
- *Na primer* kada kažemo da je memorija veličine 1Mb (jedan megabit) ili 1MB (jedan megabajt) organizacija može biti $1M \times 1$ i $1M \times 8$, respektivno
- **Adrese memorijskih lokacija** su obično numerisane u sekvencijalnom redosledu



Organizacija memorije III

- Memorija prikazana na slici sastavljena je od **64K memorijskih lokacija**
- Svaka **memorijska lokacija** u stanju je da skladišti **8-bitnu reč** (bajt)
- Imajući ovo u vidu, **kapacitet** memorije iznosi **64KB**
- Procesor koristi **adresnu magistralu** da selektuje memorijsku lokaciju sa kojom želi da komunicira
- **Prenos podataka** između procesora i memorijske lokacije obavlja se preko **magistrale podataka**
- Procesor koristi **kontrolnu magistralu** da bi **specificirao tip operacije** koju je potrebno realizovati: upis ili čitanje iz memorije



Vrste memorija I

- Memorije se klasifikuju prema dva kriterijuma:
 - Mogućnost **očuvavanja upisanog sadržaja**
 - Mogućnost **upisa u memoriju**
- Sve memorije omogućavaju čitanje sadržaja iz memorije
- Sa stanovišta očuvanja upisanog sadržaja memorije se dele na:
 - **Memorije bez gubitka sadržaja (*nonvolatile*)** – sadržaj upisan u memoriju ne gubi se sa prestankom napajanja
 - **Memorije sa gubitkom sadržaja (*volatile*)** – sadržaj upisan u memoriju gubi se sa prestankom napajanja

Vrste memorija II

- Sa stanovišta mogućnosti upisa memorije se dele na:
 - **write/read memorije**
 - **In-system programmable (ISP) memorije**
 - **read only memorije**
- U prvu grupu spadaja memorije u koje **procesor** može **neometano** da **upisuje** podatke tokom rada embeded sistema
- Većina memorija sa gubitkom sadržaja spada u ovu grupu, te su vrlo zgodne za smeštanje **privremenih podataka** ili podataka koji će biti generisani ili modifikovani od strane programa
- Neke od memorija bez gubitka sadržaja (FRAM, *Ferro electric RAM*) takođe spadaja u ovu grupu

Vrste memorija III

- U grupu **in-system programabilnih memorija** spadaju memorije bez gubitka sadržaja u koje je upis po pravilu dozvoljen samo **tokom učitavanja programa**, ali **ne i tokom izvršavanja programa**
- Jedan od glavnih razloga leži u činjenici da je **brzina upisa** u ovu grupu memorija po pravilu vrlo **mala**
- Drugi aspekt koji upis u memoriju čini otežanim je potreba korišćenja **većih napona napajanja** prilikom upisa od napona koji su neophodni prilikom čitanja podataka
- Ova potreba dovodi do **povećane potrošnje** u sistemu i zahteva složenije sisteme za napajanje

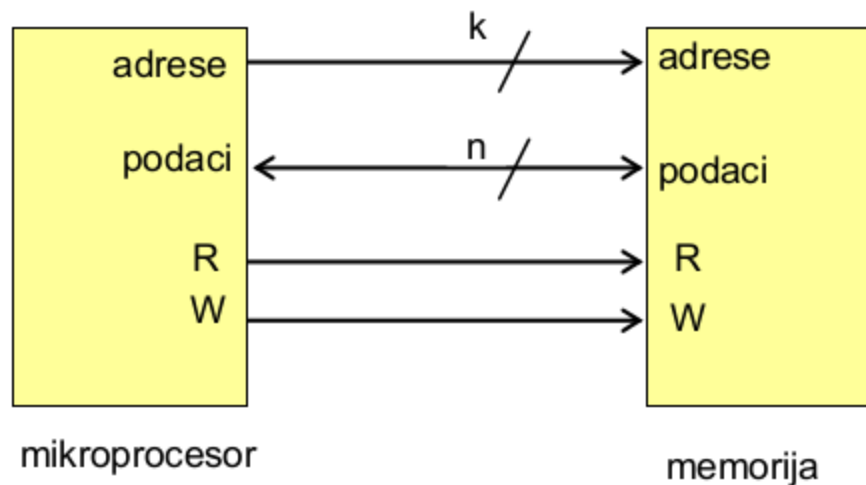
Vrste memorija IV

Mogućnost čuvanja podataka	Tip memorije	In-system programabilna	Komentar
Memorija bez gubitka sadržaja	Masked ROM	Ne	Nije programabilna
	OTPROM	Ne	Može se programirati samo jednom, pomoću programatora
	EPROM	Ne	Izbrisiva i programabilna uz korišćenje programatora
	EEPROM	Da	Spora za brisanje/upis. Nije preporučljiv upis tokom izvršavanja programa. Zahteva viši napon prilikom upisa.
	Flash	Da	Slična EEPROM memoriji
	FRAM	Da	Brz upis u memoriju uz korišćenje niskih napona napajanja
Memorija sa gubitkom sadržaja	SRAM (Static RAM)	Da	Najbrža od svih memorije što se tiče upisa/čitanja
	DRAM (Dynamic RAM)	Da	Brzi upis/čitanje, ali ne kao kod SRAM-a

Povezivanje memorije sa procesorom

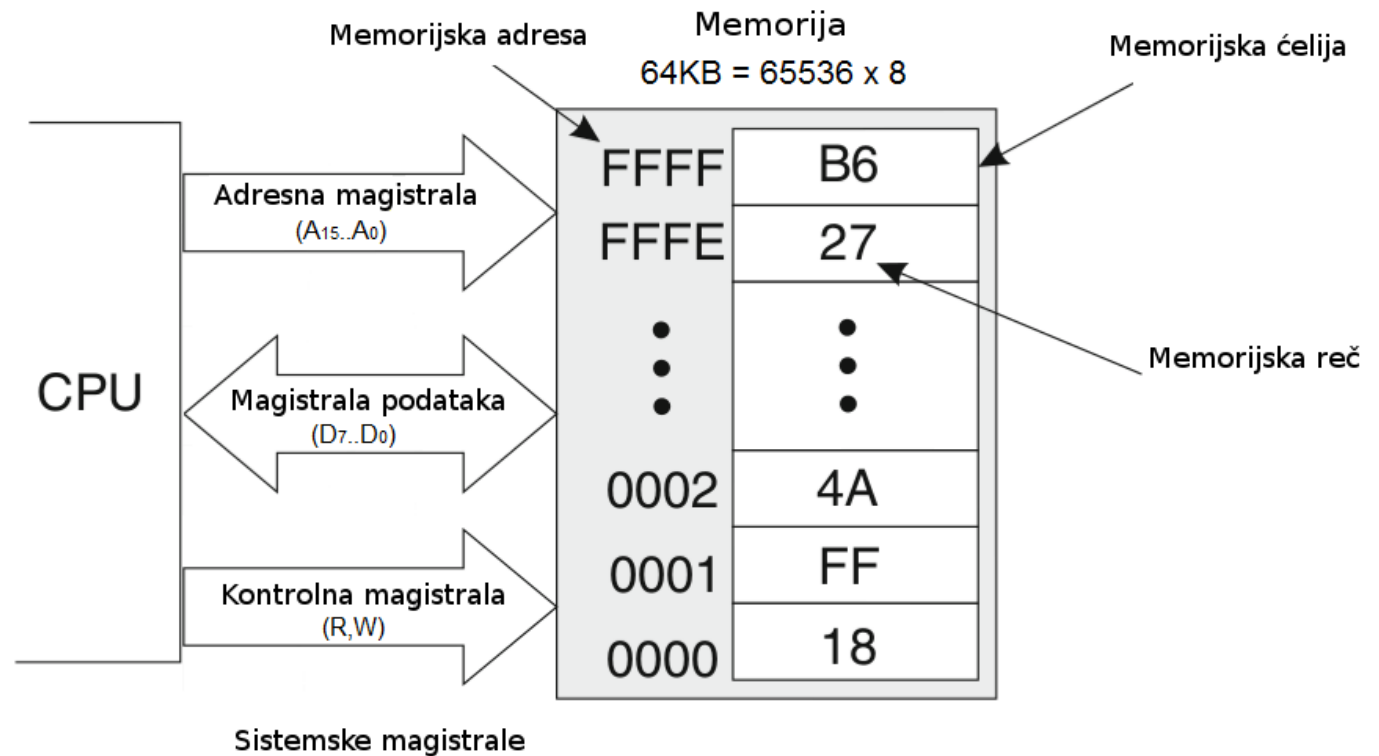
• Za povezivanje memorije sa procesorom koriste se grupe linija koje se nazivaju magistralama:

- **Adresna magistrala** (engl. *Address Bus*): linije preko kojih procesor zadaje adresu lokacije kojoj se pristupa. Memorija sa $m = 2^k$ lokacija ima adresnu magistralu širine k bita.
- **Magistrala podataka** (engl. *Data Bus*): linije preko kojih se vrši prenos podataka od procesora ka memoriji kod operacije upisa, odnosno od memorije ka procesoru prilikom čitanja. Širina magistrale podataka određena je širinom memorijske reči (n).
- U sastav **kontrolne magistrale** ulaze signali za dozvolu čitanja (**R**, engl. *Read*) i dozvolu upisa (**W**, engl. *Write*). Takođe, neki memorijski moduli posetuju dodatni kontrolni ulaz (**CS**, engl. *Chip Select*), preko kojeg se dozvoljava, odnosno zabranjuje pristup modulu.



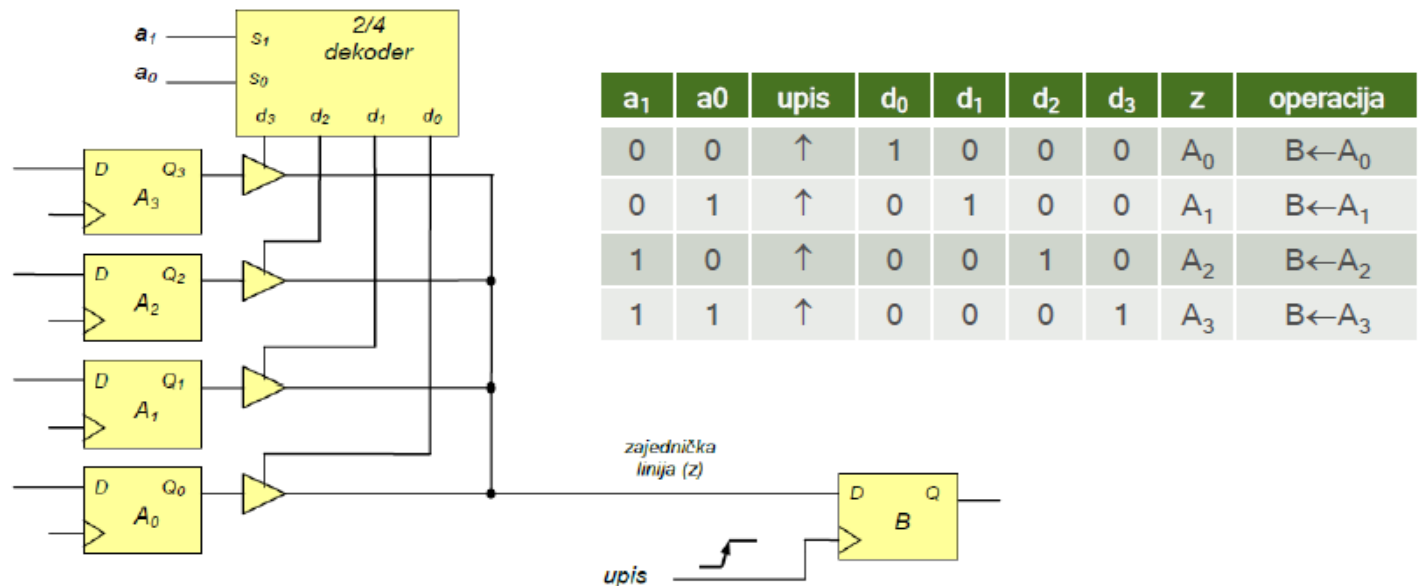
PRIMER: RAM memorija kapaciteta 64KB

- $m = 64^K = 64 \cdot 2^{10} = 2^6 \cdot 2^{10} = 2^{16} = 65536 \Rightarrow$ širina adresne magistrale je 16 bita (linije $A_{15} \dots A_0$)
- $n = 8 \Rightarrow$ širina magistrale podataka je 8 bita (linije $D_7 \dots D_0$)



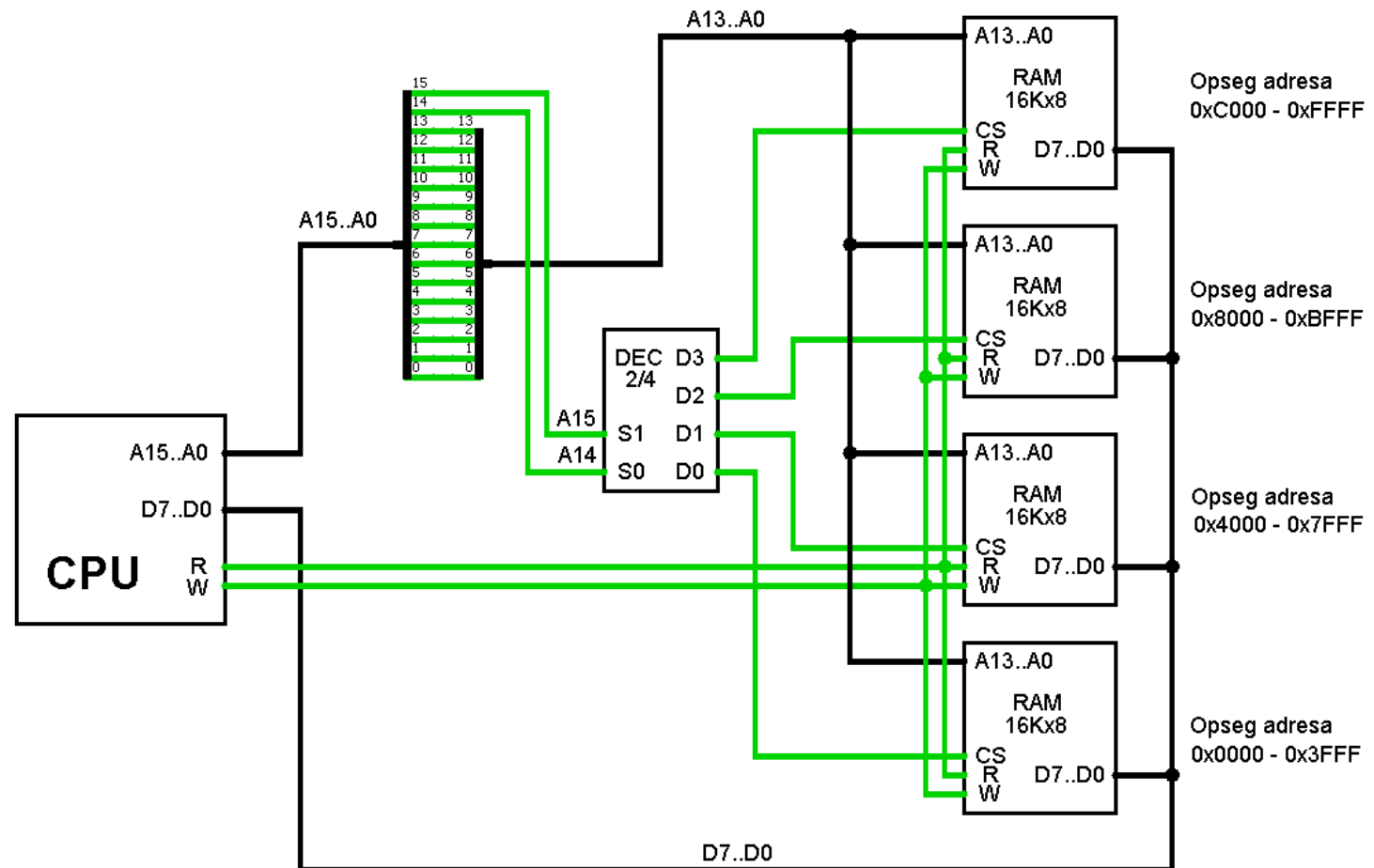
Povezivanje memorijskih elemenata na zajedničku magistralu

- Kada je potrebno istovremeno spojiti izlaze više memorijskih elemenata na zajedničku magistralu, neophodno je obezbediti mehanizam kojim se sprečava pojava **konflikta na magistrali**. Ova pojava se dešava kada dva elementa istovremenu pokušavaju da diktiraju različita logička stanja na magistrali, što može dovesti do fizičkog oštećenja, ili uništenja oba elementa.
- Tehnike za prevazilaženje konflikta na magistrali:
 - Korišćenje kola sa izlaznim stepenom sa "otvorenim kolektorom"
 - Korišćenje trostatičkih bafera
- Ukoliko se koriste trostatički baferi, njima upravlja dekoderska logika koja na magistralu propušta signal sa izlaza tačno jednog bafera (onog koji je trenutno adresiran), dok sve ostale postavlja u stanje visoke impedanse.



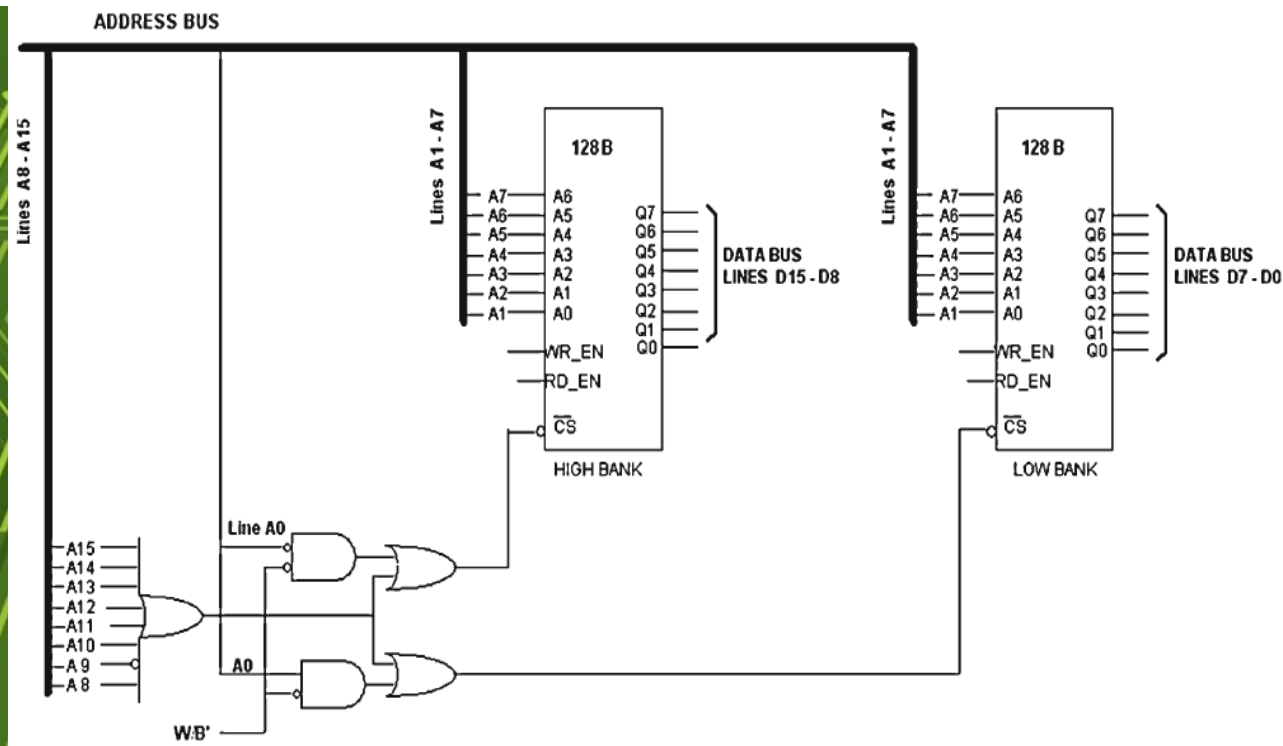
PRIMER: Povezavanje procesora sa 4 memorijska modula kapaciteta po 16KB, čime se dobija jedinstven memorijski prostor kapaciteta 64KB

- Napomena: signal CS (engl. *Chip Select*) na ulazima memorijskih modula u neaktivnom stanju (CS=0) onemogućava pristup memorijskom modulu i prevodi izlaze u stanje visoke impedanse, a u aktivnom stanju (CS=1) aktivira modul za čitanje ili upis, u zavisnosti od signala R i W i povezuje ga sa magistralom podataka.



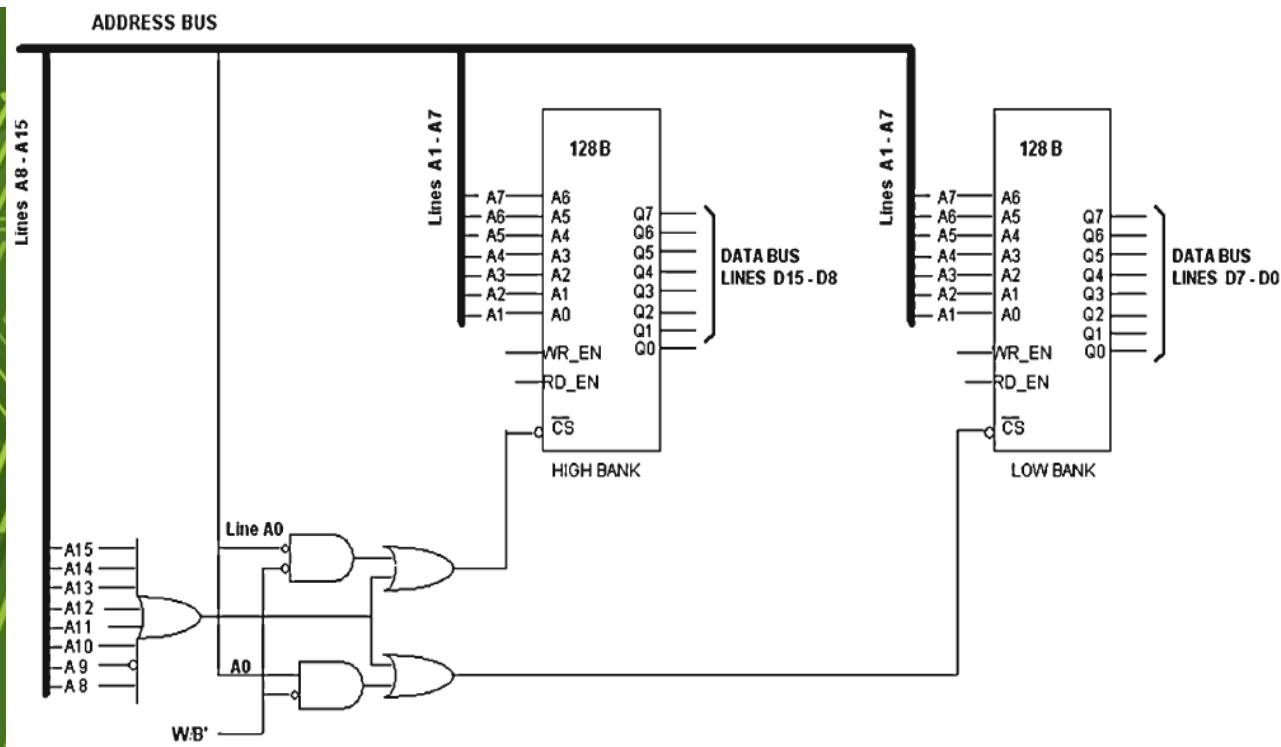
Primer povezivanja procesora i memorije I

- Na slici su prikazana **dva 128 B memorijska modula** povezana na **16-bitnu magistralu podataka**, **16-bitnu adresnu magistralu** i kontrolni signal **W/B**
- Obzirom da svaki od memorijskih modula ima magistralu podataka širine 8 bita, potrebna su dva modula da pokriju svih **16 bita magistrale podataka procesora**
- Kontrolni signal W/B** određuje da li procesor želi da pristupa **dvobajtnoj reči (word)**, ako je **W/B=1**, ili **jednobajtnoj reči (bajt)**, ako je **W/B=0**
- Odrediti opseg adresa memorijskog sistema kao i adresne opsege oba 128B modula



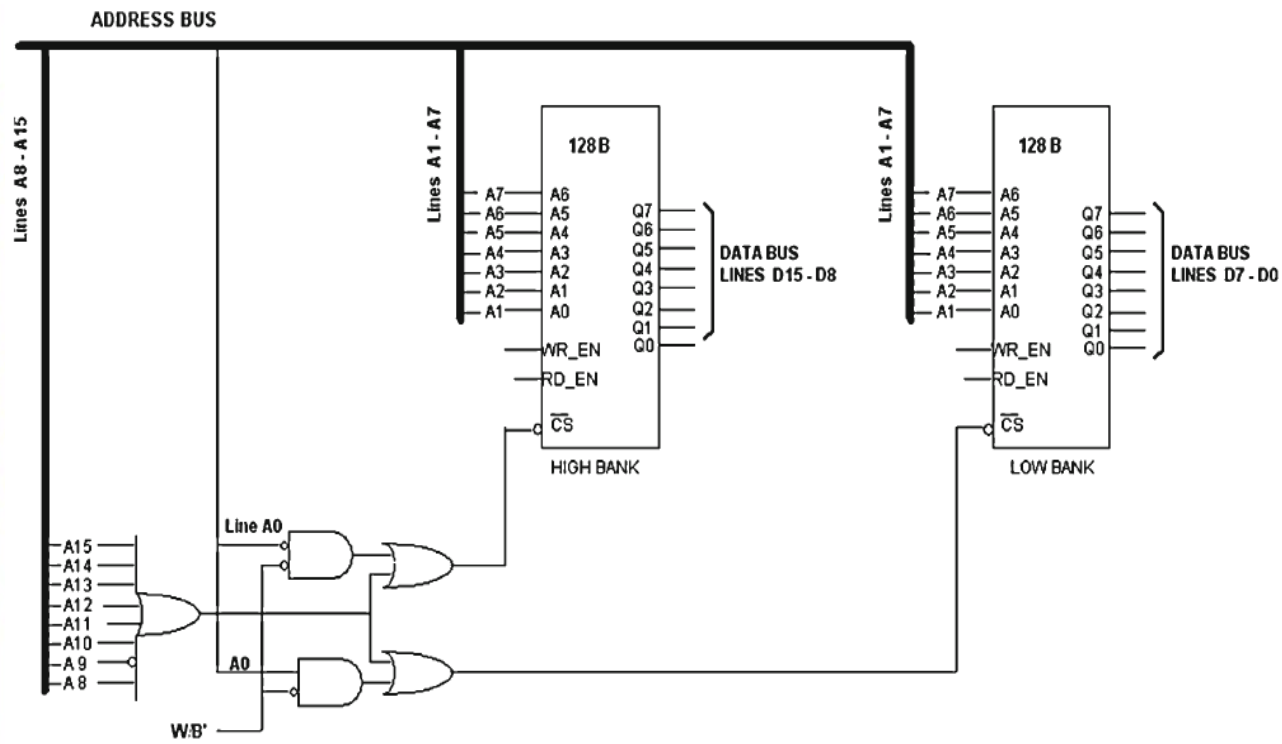
Primer povezivanja procesora i memorije II

- Da bi se memorijski moduli **aktivirali** potrebno je da njihovi **CS signali budu na niskom nivou**
- Analizom blok dijagrama možemo videti da je neophodan uslov da bilo koji od **CS** signala bude **na niskom nivou** $A_{15}=A_{14}=A_{13}=A_{12}=A_{11}=A_{10}=0$, $A_9=1$, $A_8=0$
- Adrese A_7-A_1 aktiviraju interne linije odgovarajućeg memorijskog modula
- Ukoliko je signal **W/B=1** oba modula su **aktivirana**, bez obzira na vrednost adresnog bita A_0
- U slučaju da je **W/B=0** tada $A_0=0$ aktivira **donju banku**, a $A_0=1$ aktivira **gornju banku**



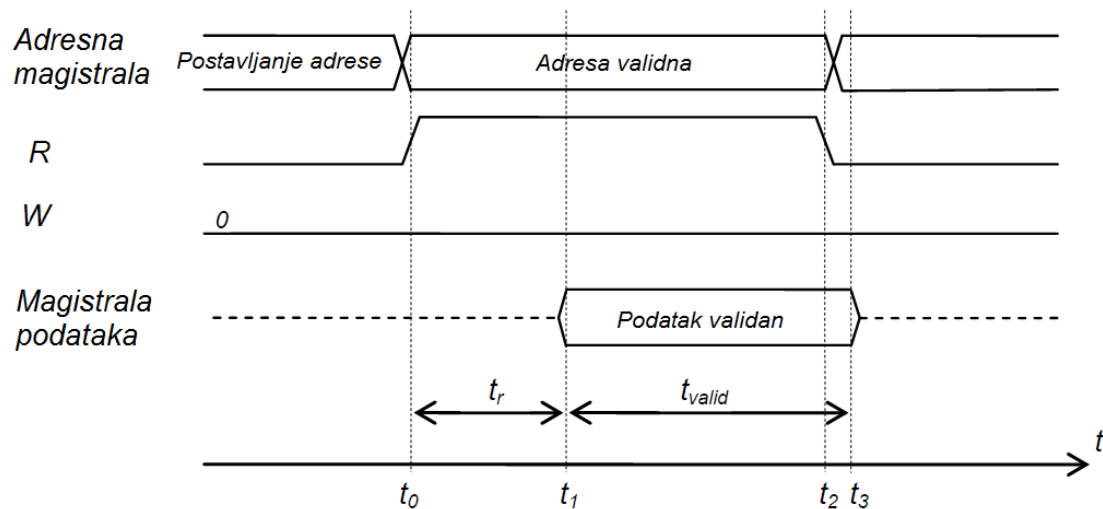
Primer povezivanja procesora i memorije II

- **Opseg adresa** koje pokriva cela memorija:
 - Početna adresa: 0000 0010 0000 0000B; 0200h
 - Krajnja adresa: 0000 0010 1111 1111B; 02FFh
- **Donja banka:**
 - Aktivirana kada je **A0=0**, što znači da pokriva sve parne adrese u opsegu 0200h-02FFh (0200h, 0202h, 0204h, ..., 02FEh)
- **Gornja banka:**
 - Aktivirana kada je **A0=1**, što znači da pokriva sve neparne adrese u opsegu 0200h-02FFh (0201h, 0203h, 0205h, ..., 02FFh)



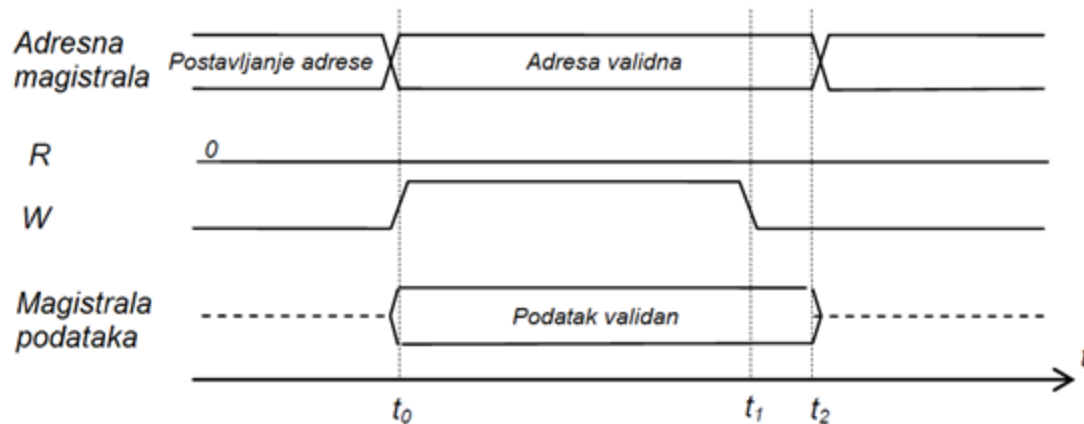
Vremenski dijagrami ciklusa čitanja podatka iz memorije

- Nakon što procesor postavi ispravnu adresu na linije adresne magistrale, u trenutku t_0 aktivira se upravljački signal za čitanje (R).
- Kada upravljačka jedinica memorije prepozna da je signal R postao aktivan, pokreće operaciju čitanja. U trenutku t_1 se sadržaj adresirane memorijske lokacije pojavljuje na magistrali podataka, koja je pre toga bila u stanju visoke impedanse.
- Vreme $t_r = t_1 - t_0$ naziva se vreme odziva memorije kod operacije čitanja.
- Podatak se zadržava na magistrali podataka do trenutka t_2 . U intervalu (t_1, t_2) procesor očitava stanje magistrale podataka i upisuje ga u prihvatni registar, nakon čega se upravljački signal za čitanje R deaktivira. Podatak je validan još neko vreme po deaktiviranju signala R, do trenutka t_3 .
- Tokom čitavog trajanja operacije čitanja, upravljački signal za upis W je neaktivan.



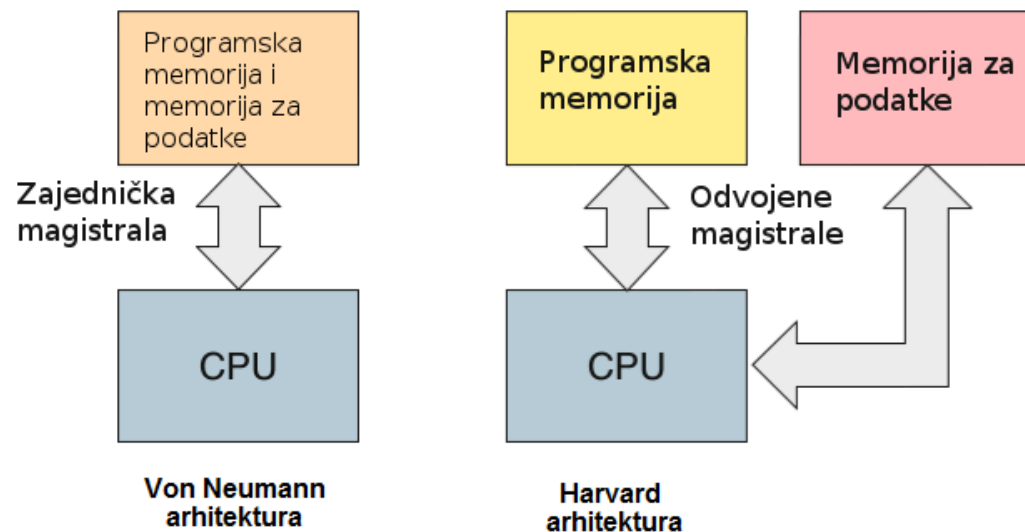
Vremenski dijagrami ciklusa upisa podatka u memoriju

- Ciklus upisa počinje tako što procesor simultano postavlja ispravnu adresu na linije adresne magistrale i podatak na magistralu podataka. Nakon toga, u trenutku t_0 aktivira se upravljački signal za upis (W).
- Kada upravljačka jedinica memorije prepozna da je signal W postao aktivan, pokreće operaciju upisa.
- U trenutku t_1 se upravljački signal za upis W deaktivira. Stanja adresne magistrale i magistrale podataka se održavaju još neko vreme, do trenutka t_2 .
- Interval (t_0, t_1) mora trajati dovoljno dugo da memorija stigne da obavi operaciju upisa. Vremenski interval $t_w = t_2 - t_0$ naziva se vreme odziva memorije kod operacije upisa.
- Tokom čitavog trajanja operacije upisa, upravljački signal za čitanje R je neaktivan.



Von Neumann i Harvard arhitektura

- Da bi se postigla željena funkcionalnost mikroračunarskog sistema, potrebno je u odgovarajući memorijski prostor smestiti programske instrukcije i podatke koji se obrađuju.
- Kod računara opšte namene, uobičajeno je da su program i podaci smešteni u isti memorijski prostor. Ovakva arhitektura nazvana je po svom tvorcu *Von Neumann* arhitektura, ili *Princeton* arhitektura.
- Organizacija mikroračunarskog sistema kod koje su programska memorija i memorija za podatke fizički razdvojene poznata je kao *Harvard* arhitektura. Struktura mikrokontrolera je obično zasnovana na ovoj arhitekturi, gde je programska memorija tipa FLASH, a memorija za podatke SRAM. Na ovaj način je program permanentno prisutan unutar mikrokontrolera, čak i po prestanku napajanja.

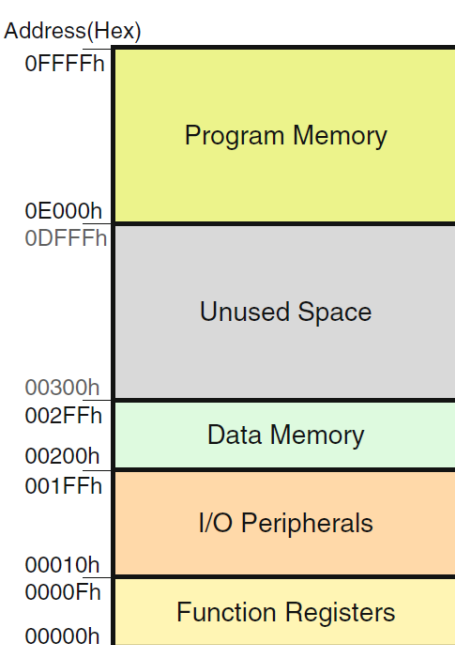


Memorijska mapa I

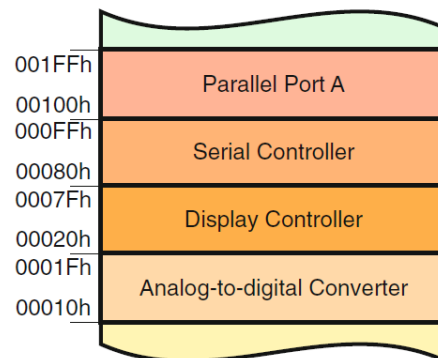
- **Memorijska mapa** predstavlja model načina korišćenja raspoloživog adresnog prostora unutar mikroprocesorskog sistema
- Predstavlja vrlo **važan podatak** prilikom **planiranja programa** i odabira odgovarajućeg procesora
- U memorijskoj mapi navedeni su **blokovi adresa** koji su adresirani različitim perifernim jedinicama prisutnim **unutar embeded sistema**
- Na slici je prikazan grafički prikaz moguće memorijske mape embeded sistema sa 16-bitnom **adresnom magistralom** i **adresnim prostorom** od 64 KB



(a)



(b)

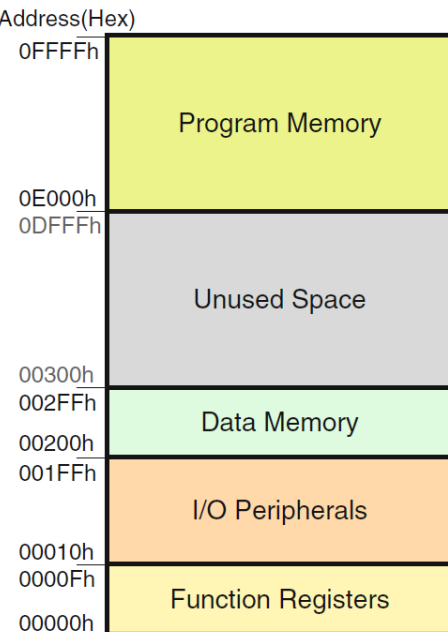


Memorijska mapa II

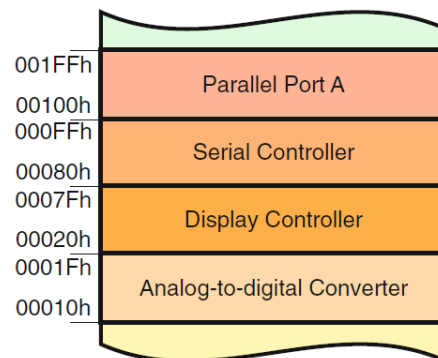
- Ovo je samo jedan od mogućih načina organizovanja memorijskog prostora
- U ovom primeru, prvih **16** memorijskih reči (adrese 0h do 0Fh) alocirano je za smeštanje **funkcijskih registara**
- Sledećih **496** (10h - 1FFh) lokacija alocirano je **ulazno/izlaznim periferijskim jedinicama**
- Narednih **256** lokacija (200h - 2FFh) rezervisano je **za memoriju podataka**
- **Programska memorija**, veličine **8K** lokacija, locirana je na kraj adresnog prostora (0E000h – 0FFFFh)
- Adrese iz opsega od 00300h do 0DFFFh predstavljaju **neiskorišćeni memorijski prostor** i mogu se koristiti u budućim verzijama sistema



(a)



(b)

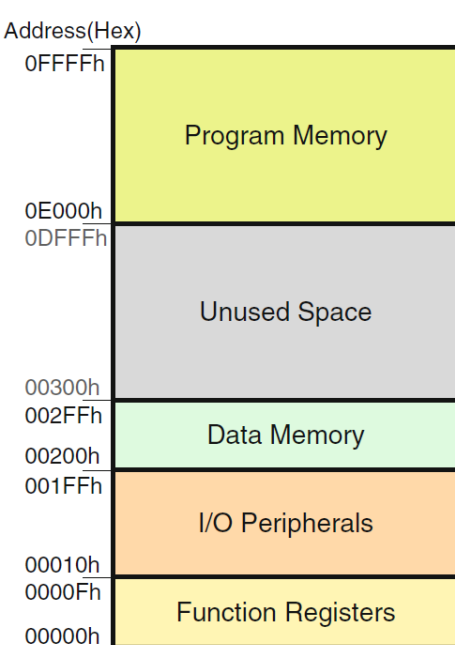


Memorijska mapa III

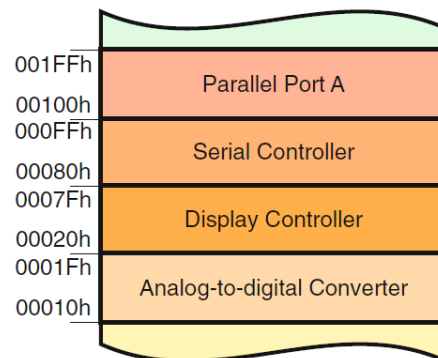
- Memorijske mape mogu biti **globalne** i **parcijalne**
- Globalna memorijska mapa prikazuje strukturu **čitavog adresnog prostora**, kao što je ilustrovano na slici a)
- Parcijalna memorijska mapa pruža **detalje o samo jednom delu adresnog prostora**, omogućavajući mnogo bolji uvid u taj deo globalne mape
- Na primer, na slici b) prikazana je parcijalna mapa moguće alokacije adresa između perifernih jedinica prisutnih u sistemu



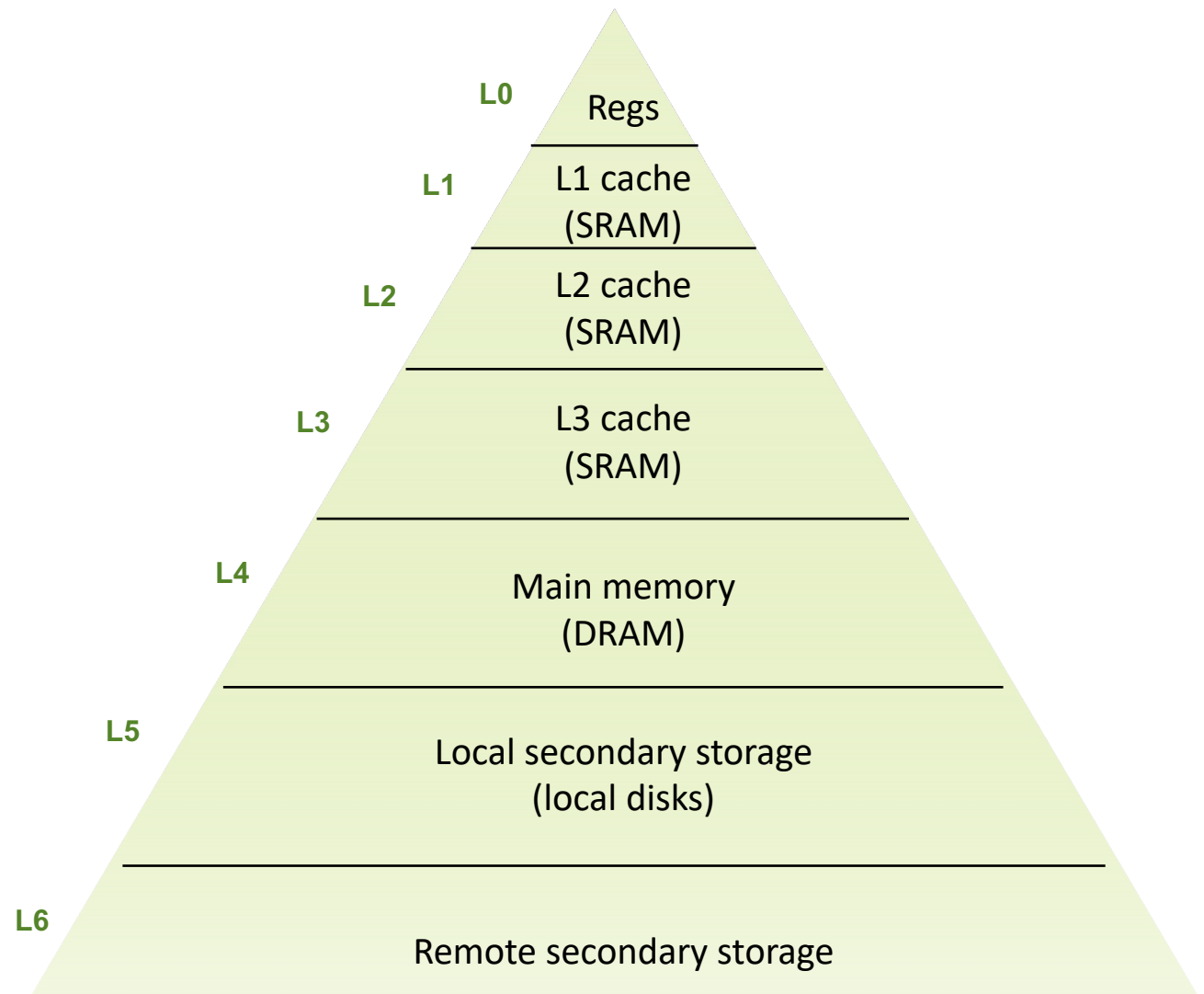
(a)



(b)

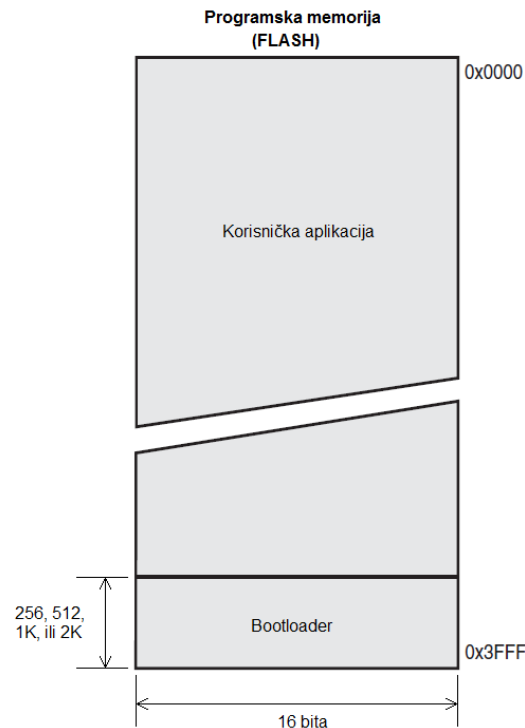


Memorijska hijerarhija



Mapa programske memorije mikokontrolera ATmega328p

- Programska FLASH memorija mikrokontrolera ATmega328p je kapaciteta 32KB. Pošto su programske instrukcije veličine 16 bita, ova memorija je organizovana kao 16Kx16, tj. 16K lokacija širine 16 bita.
- Korisnički program (aplikacija) smešta se od početne adrese (0x00).
- Vrh memorijskog prostora programske memorije rezervisan je za pomoćni program zvan *bootloader*. Njegova uloga je učitavanje programskih instrukcija preko serijskog porta i njihovo smeštanje (programiranje) u deo programske memorije rezervisan za korisničku aplikaciju. Za bootloader može biti rezervisan prostor veličine 256, 512, 1K, ili 2K reči širine 16 bita.



Mapa memorije za podatke mikokontrolera ATmega328p

- Memorija za podatke mikrokontrolera ATmega328p je tipa SRAM.
- Registri opšte namene u okviru CPU jezgra i ulazno-izlazni registri su memorijski mapirani u okviru prvih 256 bajta memorije za podatke.
- Počev od adrese 256 (odnosno 0x0100) nalazi se prostor za korisničke varijable, kapaciteta 2KB. Stoga je ukupan kapacitet memorije za podatke $256B + 2KB = 2304B$
- Opseg adresa je od 0 - 2303 decimalno, odnosno 0x0000 - 0x08FF heksadecimalno.

Memorija za podatke (SRAM)

32 registra opše namene (R0-R31)	0x0000 - 0x001F
64 U/I registra *	0x0020 - 0x005F
160 dodatnih U/I registara	0x0060 - 0x00FF
Interni SRAM (2048x8)	0x0100
	0x08FF

* Registri na adresama 0x0020-0x003F su bit-adresabilni

The background is a dark green field filled with a complex network of lighter green lines that resemble a printed circuit board (PCB) layout. These lines are interconnected by small yellow circles, which represent solder points or vias. The lines and dots are arranged in a way that creates a sense of depth and perspective, with some elements appearing to recede into the distance. The overall aesthetic is technical and digital.

Sistemske magistrale

Sistemske magistrale

- Procesor pristupa **memoriji** i **ulazno/izlaznim jedinicama** pomoću sistemskih magistrala
- Magistrala predstavlja **skup linija** koje ostvaruju zajedničku funkciju
- Svaka od **linija** nosi **jedan bit informacije**, a grupa ovakvih bitova može da se interpretira kao celina
- Sistemske magistrale grupisane su u tri klase:
 - **Magistralu podataka**
 - **Adresnu magistralu**
 - **Upravljačku magistralu**

Magistrala podataka

- Skup linija koje prenose podatke ili instrukcije od i ka procesoru naziva se **magistrala podataka**
- Dve operacije mogu se definisati za magistralu podataka:
 - Operacija **čitanja** - izvršava se svaki put kada se informacija prenosi **ka procesoru**
 - Operacija **upisa** - izvršava se svaki put kada podaci prenose **od procesora** ka memoriji ili periferijskoj jedinici
- Jedan transfer informacija naziva se **transakcijom** na magistrali podataka
- **Broj individulanih linija** koje čine magistralu podataka određuje **maksimalnu širinu podatka** koju procesor može da pošalje ili primi tokom jedne transakcije

Adresna magistrala I

- Procesor u svakom trenutku interaguje samo sa **jednom memorijskim blokom** ili **periferijskom jedinicom**
- Svaki memorijski ili registarski element, ima jedinstvenu identifikacionu oznaku koja se naziva **adresa**
- Skup linija koje prenose informaciju o adresi registra sa kojim se želi ostvariti komunikacija čine **adresnu magistralu**
- Linije koje čine adresnu magistralu su obično **unidirekcione** i polaze **od procesora**
- **Broj linija** koje čine adresnu magistralu određuje **maksimalnu veličinu adresnog prostora** koji dati procesor može da adresira
- Adresna magistrala od **m** bita može da adresira najviše **2^m** različitih **lokacija**

Adresna magistrala II

- **Primer 1:**

Koliko različitih memorijskih lokacija je moguće adresirati i koji je njihov adresni opseg (početna i krajnja vrednost adrese) u slučaju da se koristi adresna magistrala od:

- a) 12 bita
- b) 22 bita

Adresna magistrala II

- **Primer 1:**

Koliko različitih memorijskih lokacija je moguće adresirati i koji je njihov adresni opseg (početna i krajnja vrednost adrese) u slučaju da se koristi adresna magistrala od:

- a) 12 bita
- b) 22 bita

- **Rešenje:**

- a) U slučaju korišćenja 12-bitne adresne magistrale, postoji $2^{12}=4096$ različitih memorijskih lokacija koje je moguće adresirati. Početna i krajnja vrednost adresa su 0000 0000 0000B i 1111 1111 1111B ako koristimo binarnu reprezentaciju, odnosno 0x000 i 0xFFF ako koristimo heksadecimalnu reprezentaciju.
- b) U slučaju 22-bitne adresne magistrale, postoji $2^{22}=4194304$ različitih memorijskih lokacija koje je moguće adresirati. Početna i krajnja vrednost adresa su 0x000000 i 0x3FFFFFF.

Adresna magistrala III

- Primer 2:

Odgovarajući sistem raspolaže memorijom čija je veličina 32768 memorijskih lokacija. Koji je najmanji broj linija adresne magistrale koja se može koristiti za adresiranje ove memorije?

Adresna magistrala III

- Rešenje:

Minimalni broj potrebnih linija adresne magistrale, m , koji je neophodan da bi se adresirala memorija od k memorijskih lokacija, može se izračunati korišćenjem sledećeg izraza:

U našem primeru

$$m = \lceil \log_2(k) \rceil$$

$$m = \lceil \log_2(32768) \rceil = 15$$

Upravljačka magistrala

- **Upravljačka magistrala** grupiše sve linije koje prenose signale koji **upravljaju** nekom od aktivnosti unutar sistema
- Za razliku od adresne i magistrale podataka, kod kojih se vrednosit individualnih linija posmatraju kao celina (adresa, odnosno podatak), signali upravljačke magistrale obično rade i **interpretiraju se odvojeno**
- Upravljački signali uključuju:
 - **Signale koji pružaju informaciju** o tome da li procesor izvršava operaciju čitanja ili upisa
 - **Signale koji sinhronizuju prenos podataka**, označavajući kada transakcija počinje i završava
 - **Signale koji pružaju informaciju** da li neki od uređaja pristunih u sistemu zahteva servis od strane procesora
- Većina linija upravljačke magistrale su **unidirekcione** i one ili ulaze ili napuštaju procesor, u zavisnosti od njihove namene
- Broj i funkcija linija upravljačke magistrale zavisi od arhitekture procesora