



LOGIČKO PROJEKTOVANJE CENTRALNOG PROCESORA

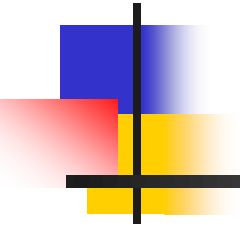
NRS Predavanje br. 4



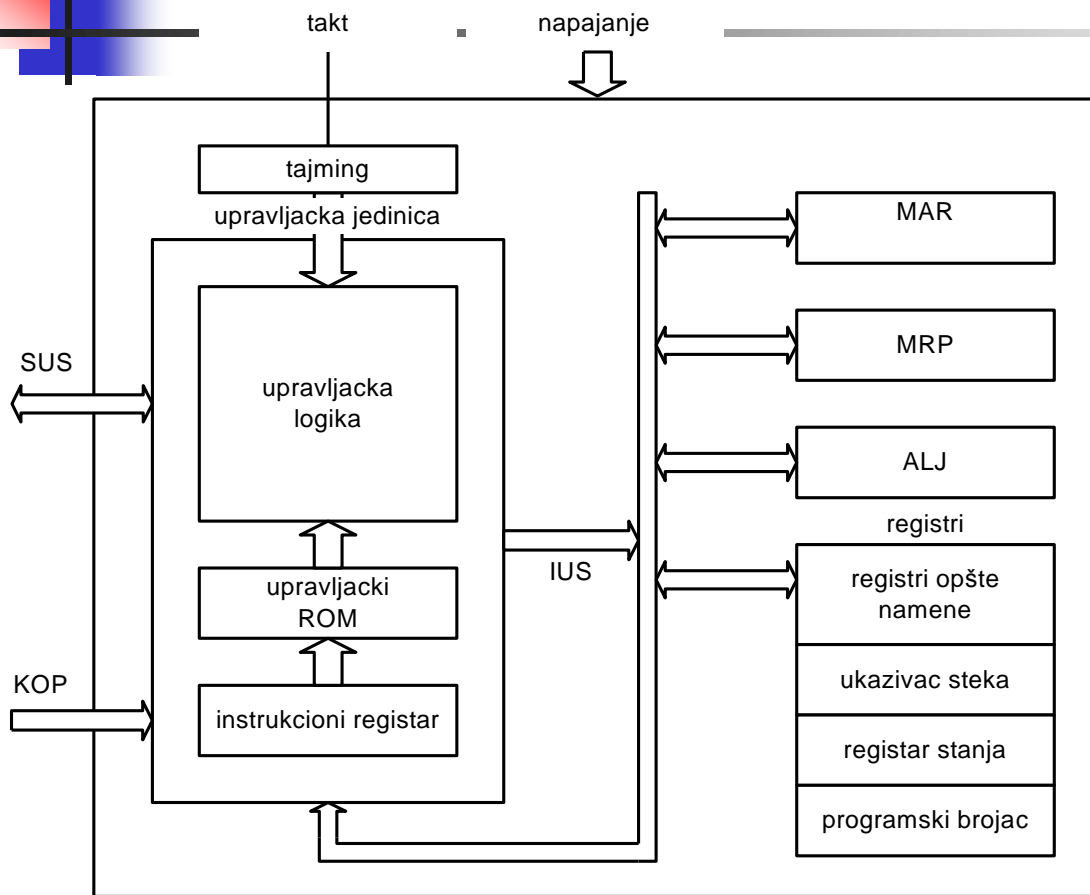
Centralni procesor

- Osnovni deo računarskog sistema
 - Obavlja aritmetičke i logičke operacije,
 - upravlja memorijom i ulazno-izlaznim podsistemom
- Mikroprocesor – CP kao VLSI komponenta
 - silicijumska podloga zatvorena plastičnim ili keramičkim omotom,
 - okružena nožicama (izvodima ili terminalnim tačkama)

ORGANIZACIJA REGISTARA



Osnovna organizacija



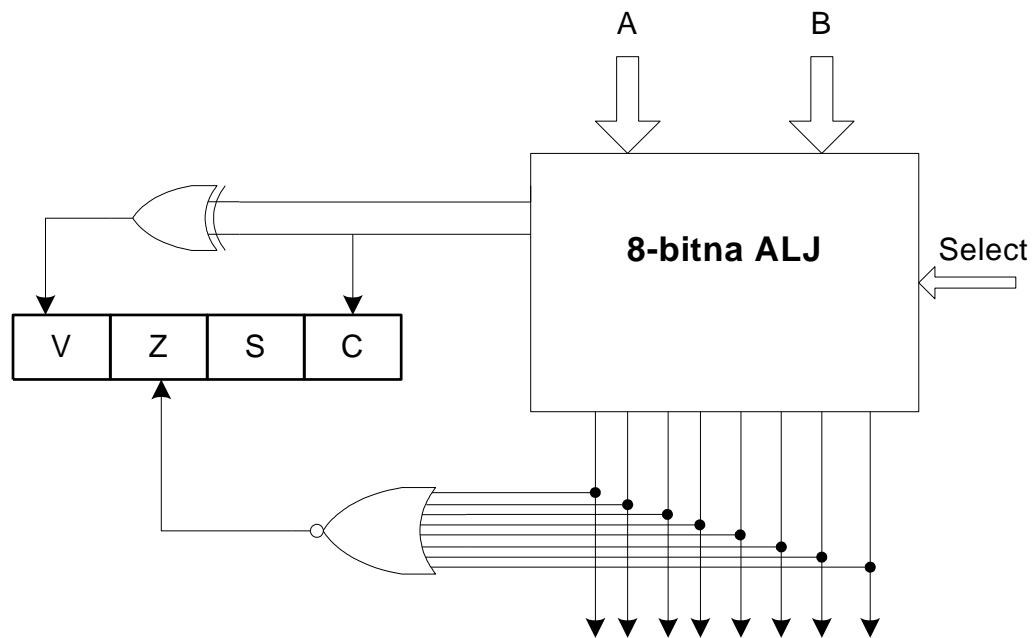
- ALJ (int, float)
- Registri (data, adr)
- Interne magistrale
- UJ - upravljački signali
 - interni
 - spoljni
- Specijalni registri
 - PC, IR
 - MAR, MRP
 - SR, SP

MAR - adresni prihvatni registar
MRP - Prihvatni registar podataka
ALJ - Aritmeticko-logicka jedinica

KOP - Spoljni ulazi
SUS - Spoljni upravljacki signali
IUS - Interni upravljacki signali

Registar stanja (SR)

- Kontrolni biti
 - Dozvola prekida (*interrupt enable*)
- Indikatorski biti
 - Praćenje AL operacija
 - V, Z, S, C



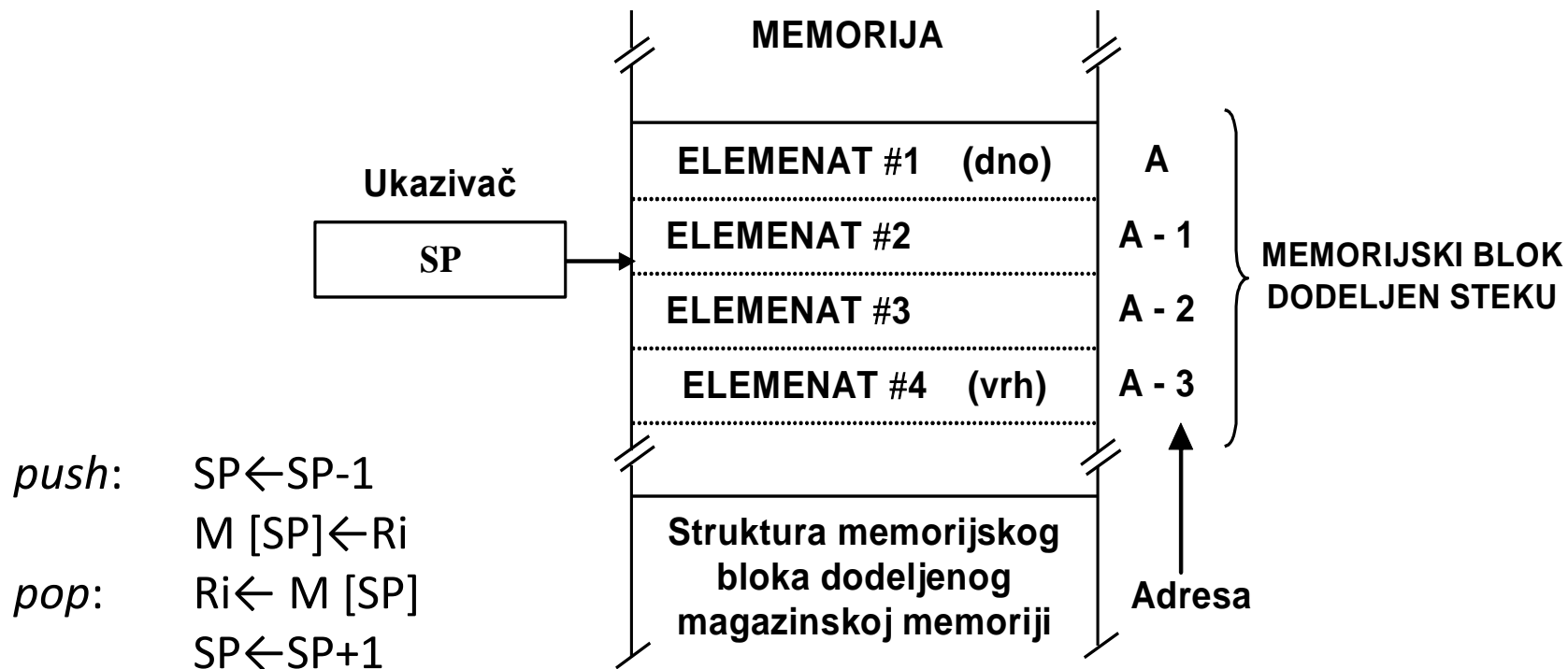
Provera vrednosti bita SR

- Uslovna grananja: JC, JNC, JNZ, JZ, JGE
- Ispitivanje vrednosti jednog bita registra \Rightarrow
- Upoređivanje dva broja \Downarrow (oduzimanje+test)

A	101x1100
B	00010000
A ^ B	000x0000

	Neoznačeni brojevi		Označeni brojevi	
A > B	$C=1 \wedge Z=0$	\overline{CZ}	$Z=0 \wedge (S=0, V=0 \vee S=1, V=1)$	$\overline{Z} \cdot \overline{S \oplus V}$
A ≥ B	$C=1$	C	$Z=1 \vee (S=0, V=0 \vee S=1, V=1)$	$Z + \overline{S \oplus V}$
A < B	$C=0$	\overline{C}	$Z=0 \wedge (S=1, V=0 \vee S=0, V=1)$	$\overline{Z} \cdot (S \oplus V)$
A ≤ B	$C=0 \vee Z=1$	$\overline{C} + Z$	$Z=1 \vee (S=1, V=0 \vee S=0, V=1)$	$Z + (S \oplus V)$
A = B	$Z=1$	Z	$Z=1$	Z
A ≠ B	$Z=0$	\overline{Z}	$Z=0$	\overline{Z}

Ukazivač steka (SP)





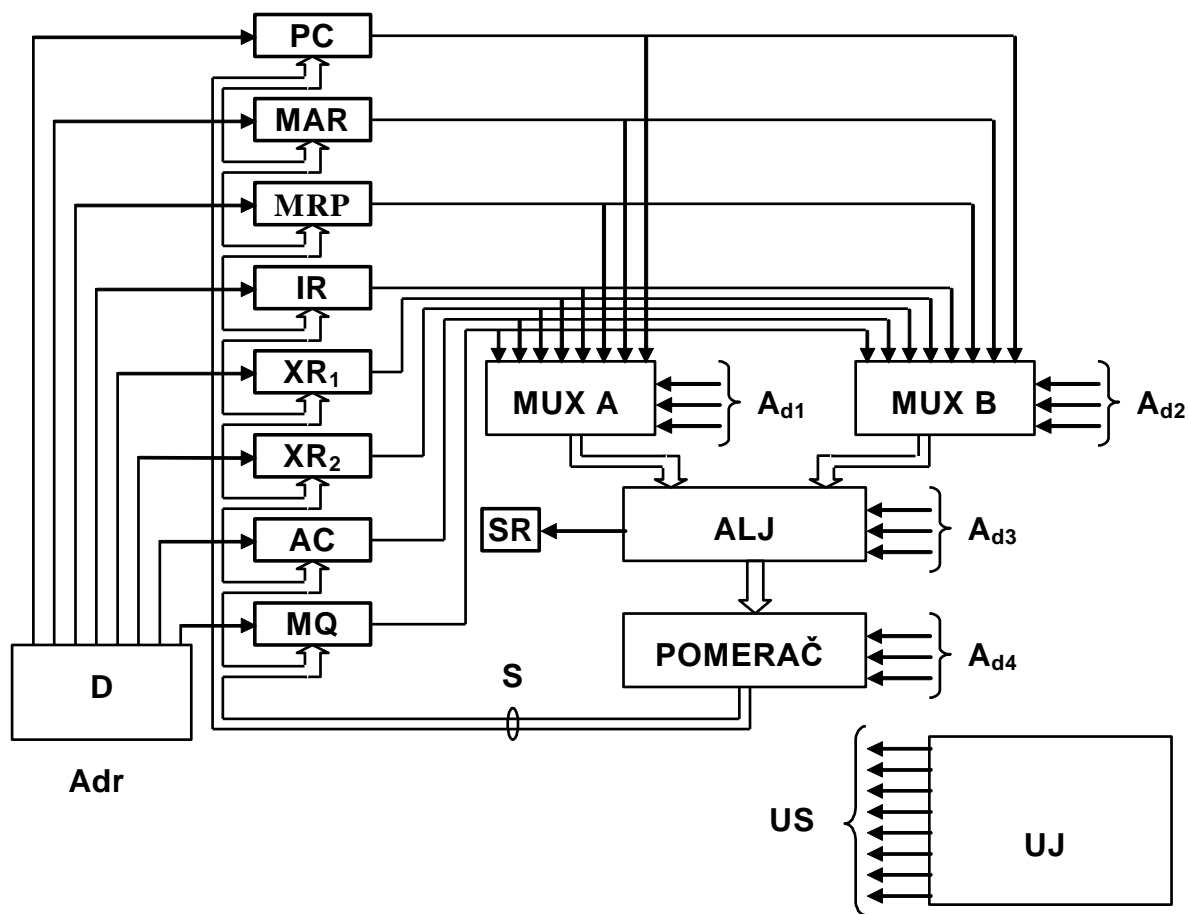
ORGANIZACIJA INTERNIH PRENOSNIH PUTEVA



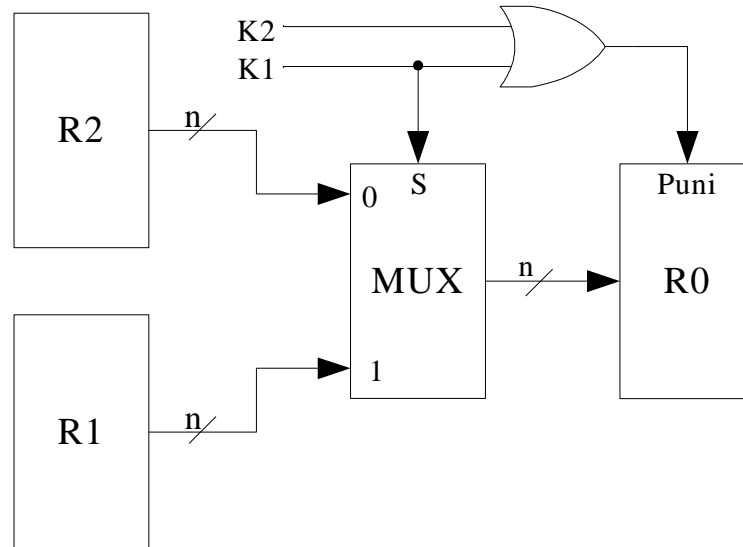
Zadatak i varijante rešenja

- Međuregistarski transfer
- Magistralna organizacija
 - Jedinствен prenosni put
 - Multiplekseri i registri sa tri stanja
- Memorijska organizacija registara

Magistralna organizacija

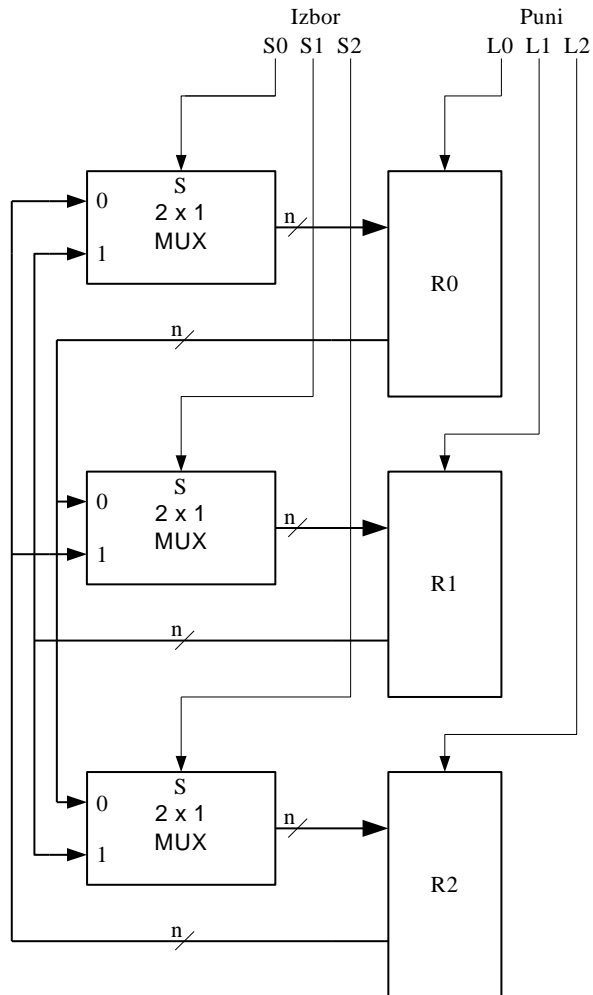


Magistrala sa dodeljenim multiplekserima

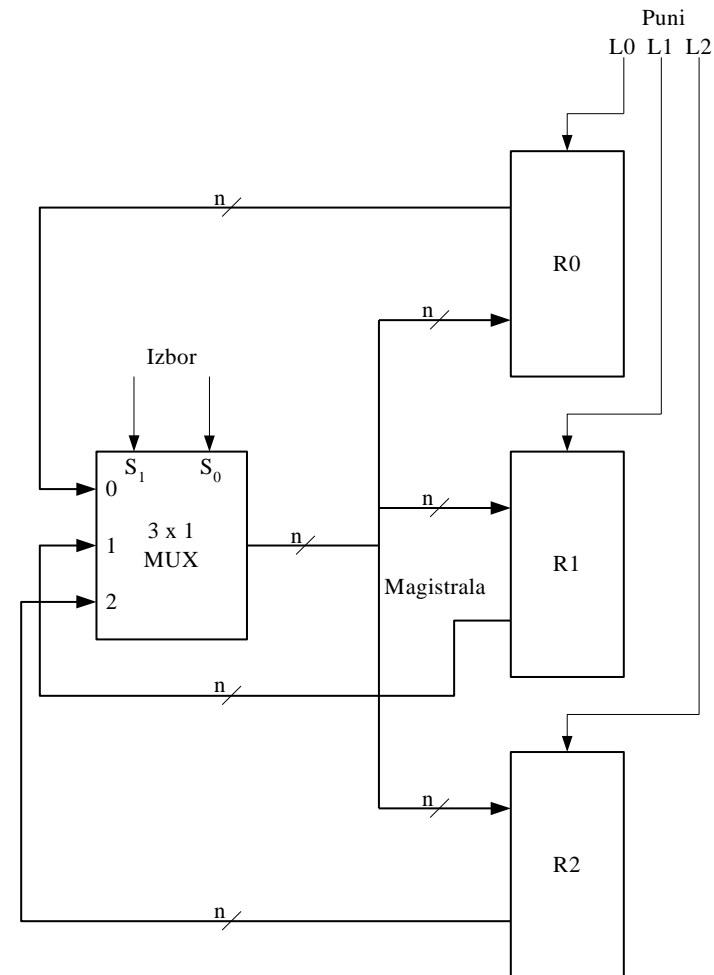


- Osnovna mana - veliki broj veza

Jednostruka magistrala (*single-bus*)



(a) Dodeljeni multiplekseri

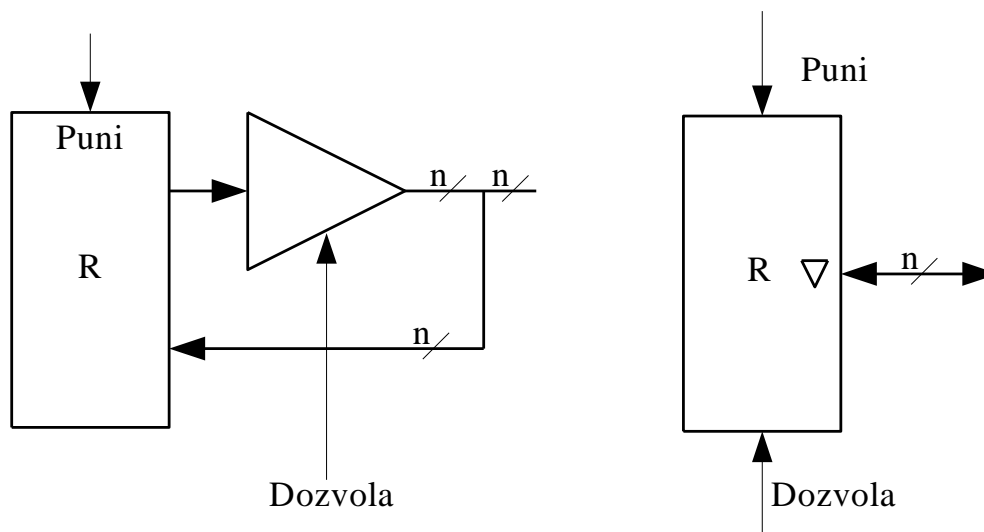


(b) Jednostruka magistrala

- a) **6n** ulaza, što zahteva ukupno $(2I + ILI) * 3n = 9n$ logičkih kola;
- b) **3n** ulaza, odnosno $(3I + ILI) * n = 4n$ logičkih kola.

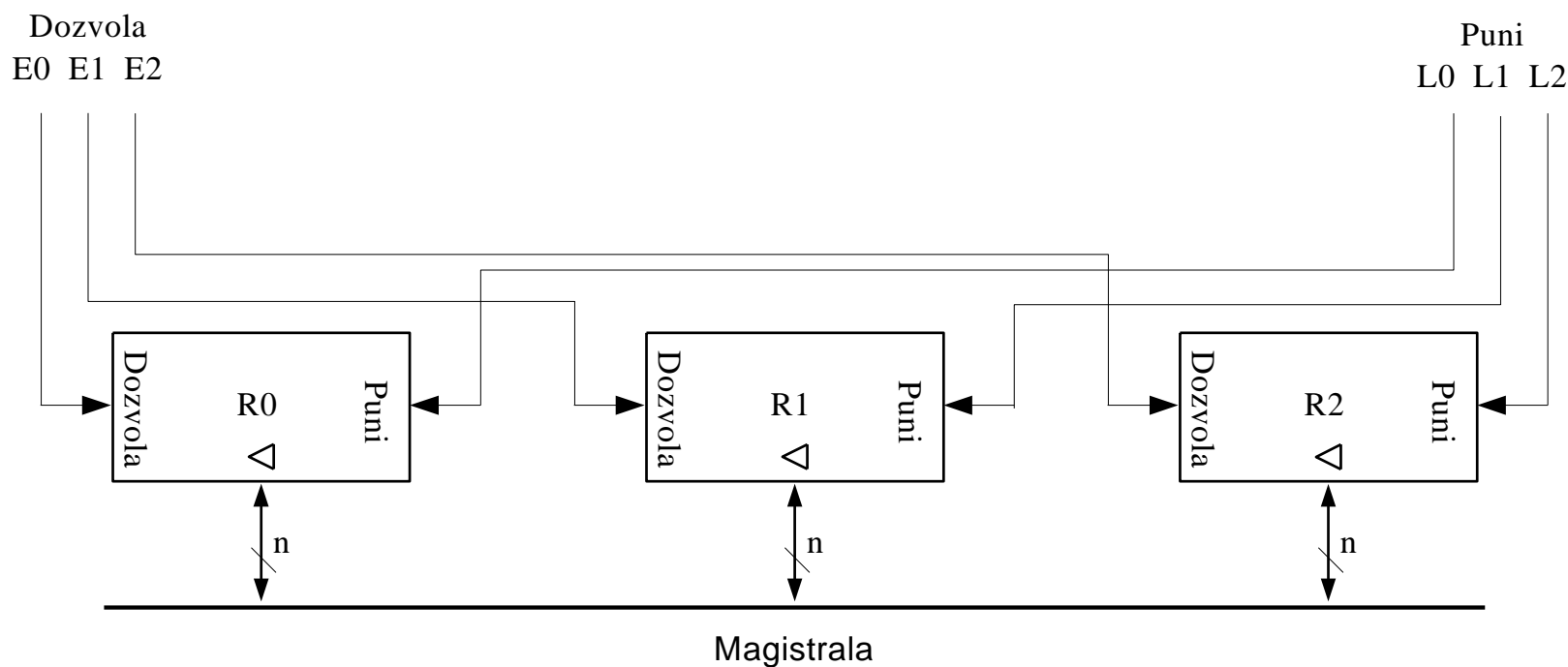
Magistrale realizovane pomoću registara sa tri stanja

- Redukcija broja linija
- Skraćenje kašnjenja - jedan nivo logičkih kapija
- Dvosmerni (bidirekcionni) registar sa tri stanja
 - 0, 1, stanje visoke impedanse (*hi-Z*, *tri-stated*) na izlazima



Realizacija magistrale sa 3 stanja

- Polazni registar - postavljanjem signala dozvole
- Odredišni registar - signali za punjenje registra

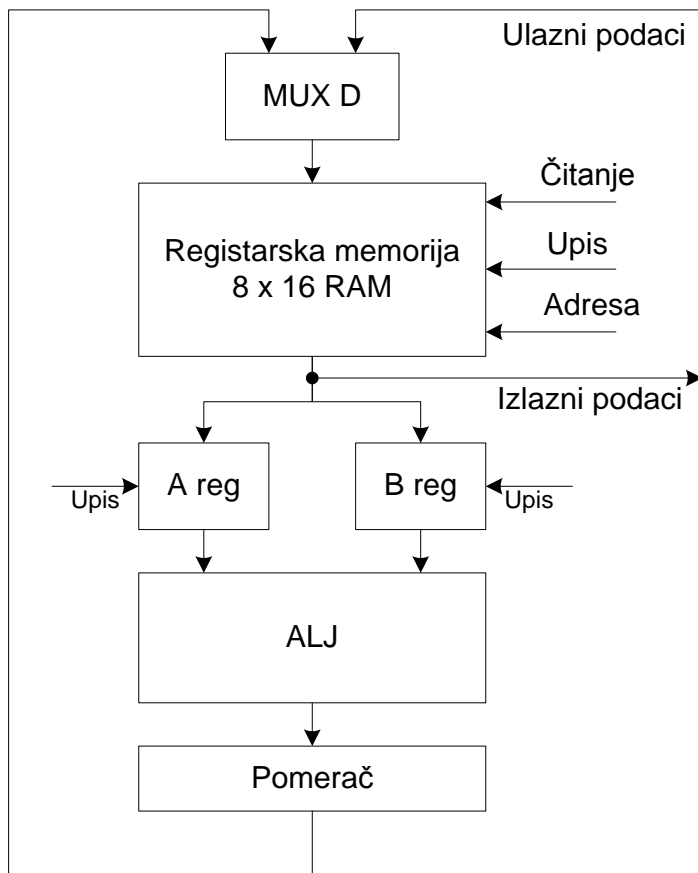




Registarska memorija (*scratch-pad*)

- Ekonomičnija varijanta organizacije
- Forma memorije – *register file*
- Pristup registru preko adrese
- Memorija malog obima, smeštena unutar procesora i ultra-brza
- Za razliku od OM, ne sadrži instrukcije

Organizacija registara u obliku memorije

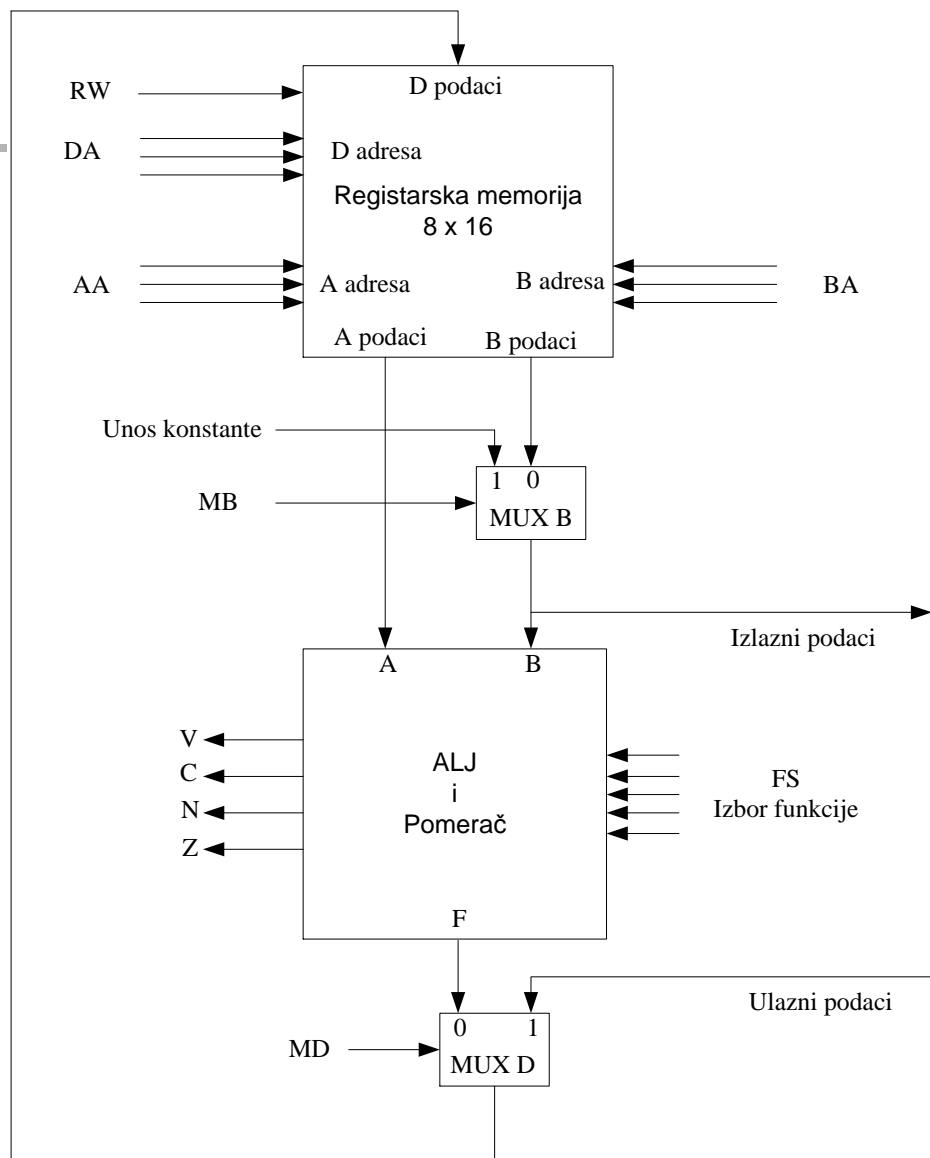
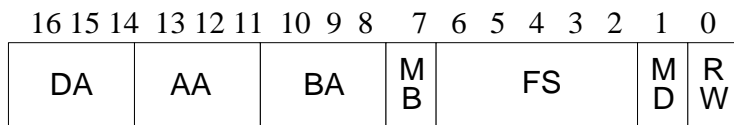


- Mana - višestruki pristup memoriji radi izvršenja standardnih instrukcija

Instrukcija		Izvršne mikroinstrukcije
$R2 \rightarrow R3 + R4$	\Rightarrow	t1: $A \rightarrow M[011]$
		t2: $B \rightarrow M[100]$
		t3: $M[010] \rightarrow A + B$

Unapređena organizacija

- Dvostruki pristup (*dual-port*)
 - A i B - port
 - $R_a \leftarrow R_a * R_b$ za 1 takt
- D - odredišni port
- Upravljačka reč

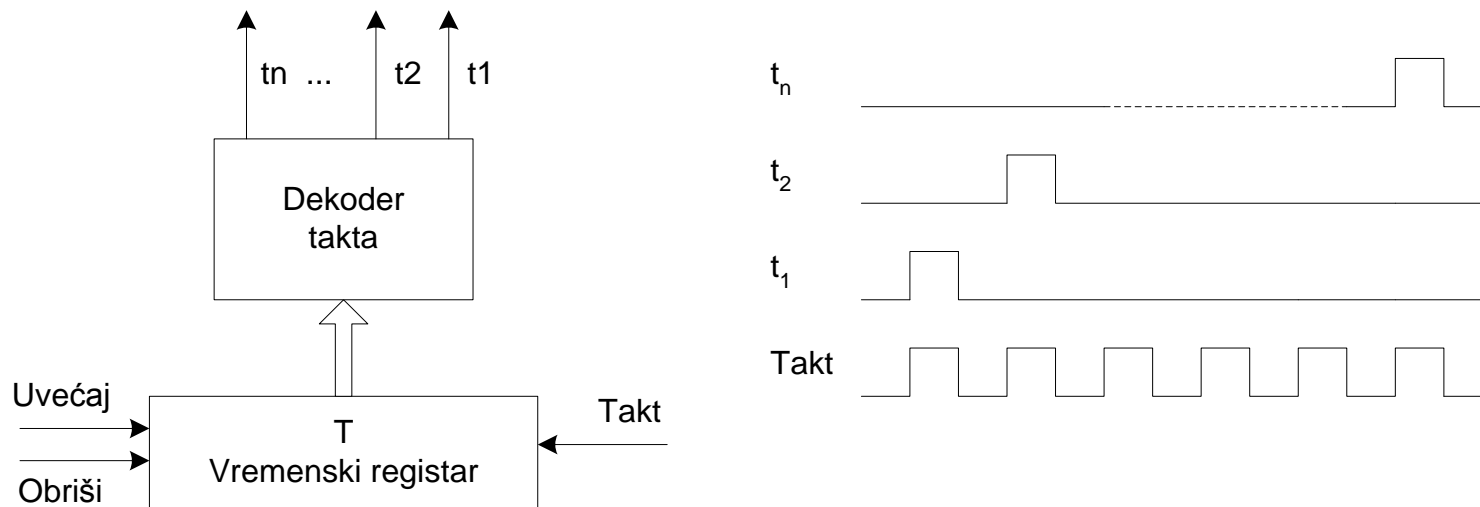




VREMENSKI REDOSLED SIGNALA CENTRALNOG PROCESORA

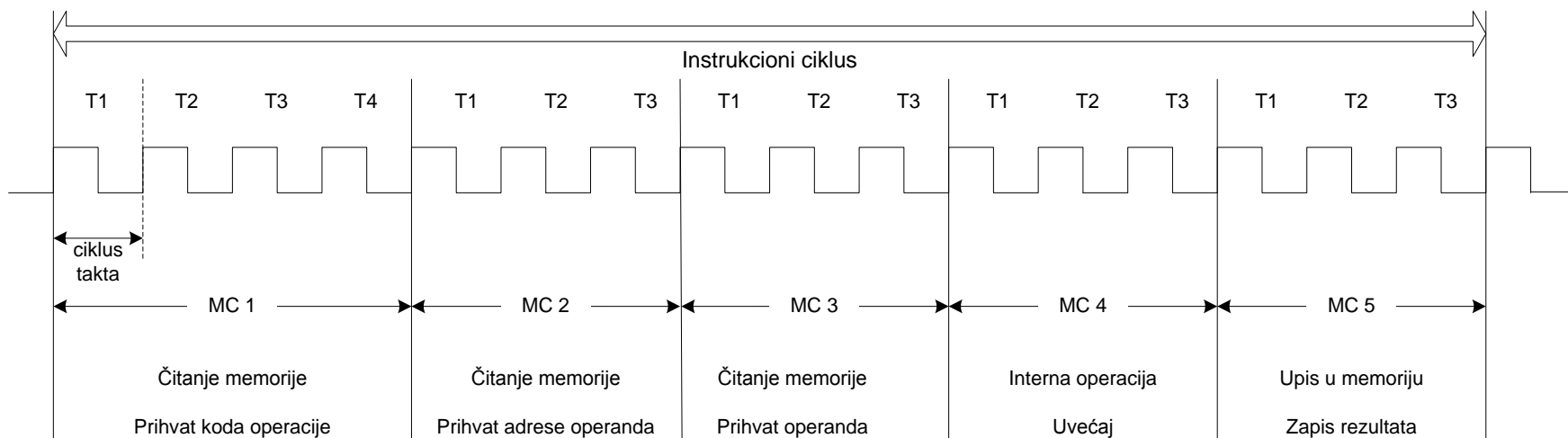
Takt centralnog procesora

- CP je sinhroni automat
- Ciklus takta predstavlja elementarni interval aktivnosti obrade – mikroinstrukcije
- Generator vremenskih funkcija



Instrukcioni ciklus

- Niz vremenskih intervala u kojima se vrši prihvata i izvršenje neke od instrukcija
- Instrukcioni ciklus = \sum mašinskih ciklusa
- Primer: INC adresa_operanda

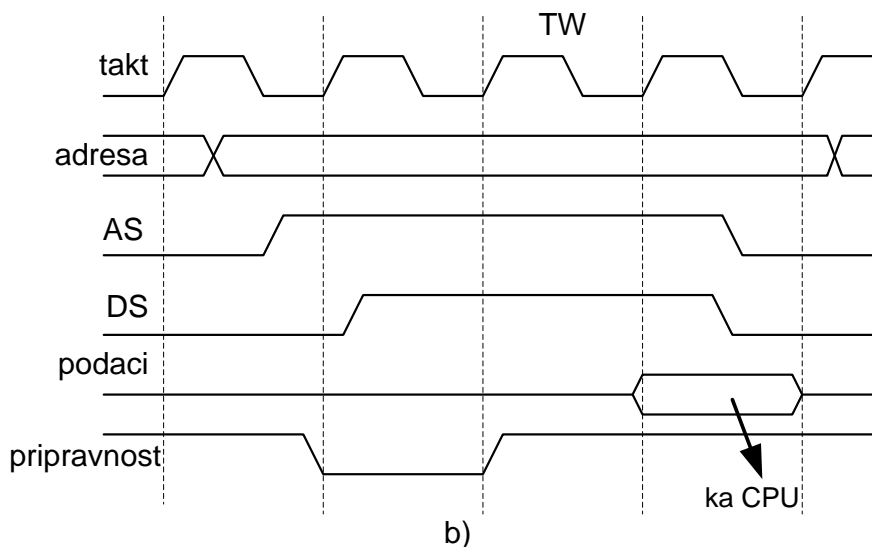
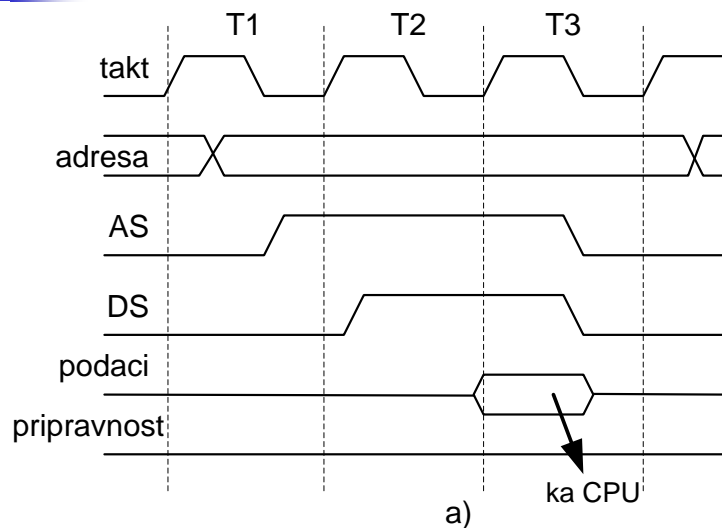




Mašinski ciklus

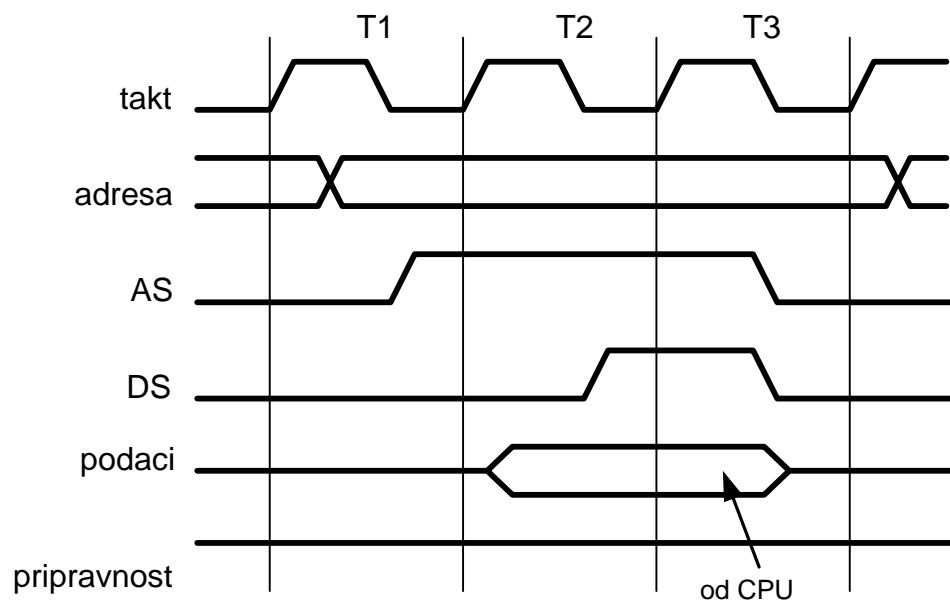
- Elementarne operacije procesora
- Najčešće vezane za operacije na magistralama
- Ograničen broj različitih tipova
 - Memorijska operacija čitanja
 - Memorijska operacija upisa
 - Interne operacije
 - Prepoznavanje prekida
 - Preuzimanje magistrale
 - Reset procesora

Mašinski ciklus čitanja



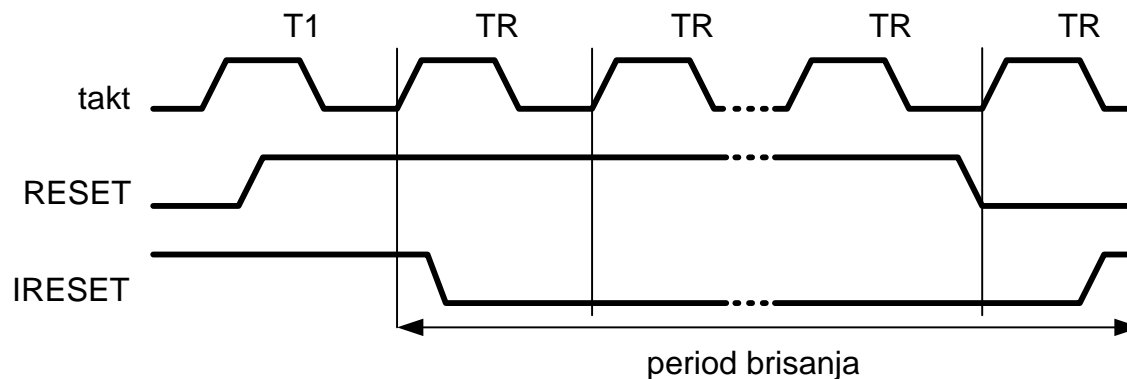
- Pristup memoriji ili UI uređaju, bez (a) i sa stanjem čekanja (b)
 - T1 – procesor postavlja adrese i aktivira AS
 - T2 – procesor aktivira DS, na opadajućoj ivici počne ispitivanje RDY
 - T3 – memorija postavi podatke, koji se učitavaju na opadajućoj ivici DS, praćeno deaktiviranjem AS
 - TW – čekaj jedan ciklus ako je RDY neaktivan

Mašinski ciklus upisa



- T1 - procesor postavlja adresu i AS
- T2 – procesor postavlja podatke i aktivira DS
- TR – upis u memoriju
- RDY – kao i pri čitanju

Mašinski ciklus brisanja (RESET)



- Prevodi procesor u poznato stanje
- Procesor ispituje ulaz RESET na opadajućoj ivici **svakog** ciklusa takta
- Prema tome, ne radi se o završetku tekuće instrukcije ili mašinskog ciklusa, tj. bilo kada se može desiti
- Po deaktiviranju RESET, postavlja se početno stanje procesora



MAŠINSKI JEZIK



Mašinski jezik

- Binarni format
 - Mašinske instrukcije (kod operacije)
 - Adresni režimi (adresa operanda)
 - Mašinski program
- Tekstualni format
 - Asemblerski jezik
 - Asemblerski program



Tipovi instrukcija

- Na osnovu funkcije
 - aritmetičko-logičke instrukcije (izmena podataka u registrima),
 - instrukcije grananja - uslovnog, bezuslovnog, veze sa potprogramom.
- Po načinu pristupa podacima
 - instrukcije sa obraćanjem memoriji (najmanje jedna adresa mora biti memorijska adresa),
 - instrukcije bez obraćanja memoriji (prenos podataka između registara).



Format instrukcija

Kod Operacije	Polazni operandi	Odredišni operandi
---------------	------------------	--------------------

- Format instrukcija definiše sintaksu mašinskog jezika
- Kod operacije definiše operaciju koja će se izvršiti nad polaznim operandima radi generisanja rezultata koji se upisuju u odredišni operand.
- Broj operanada (0, 1, 2, ...)
- Adresnost - broj pristupa memoriji u toku izvršenja instrukcije

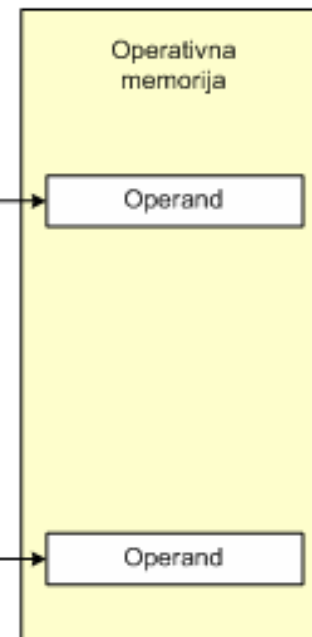
Adresni režimi:

metodi generisanja adresa podataka (operanada) u vreme izvršenja nekog programa

Direktno (neposredno) adresiranje operanda



Apsolutno adresiranje (neposredno adresiranje memorije)



Adresiranje registra



Registar

Operand

Indirektno registarsko adresiranje



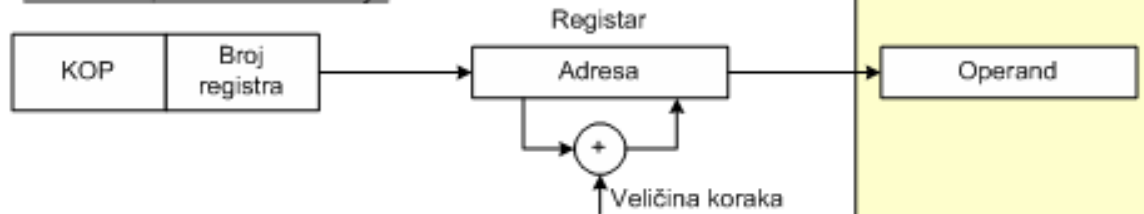
Registar

Adresa

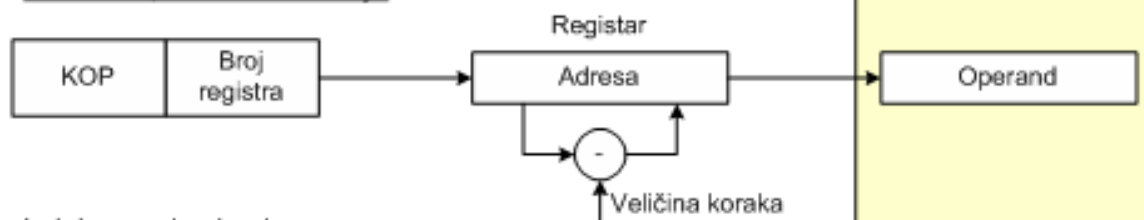
Operand

Adresni režimi:

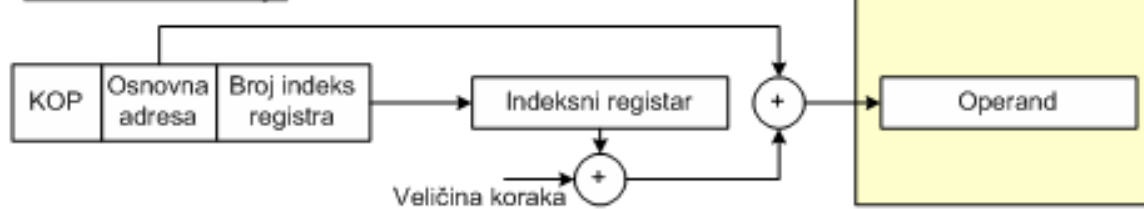
Adresni režim samouvećanja



Adresni režim samouvećanja

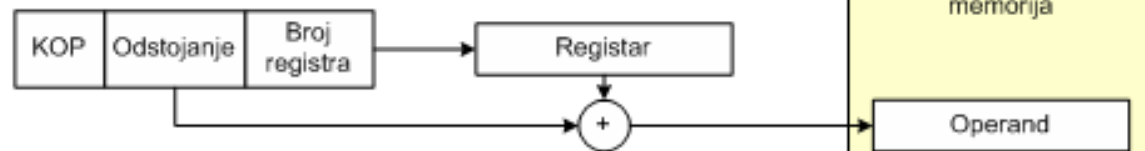


Indeksno adresiranje

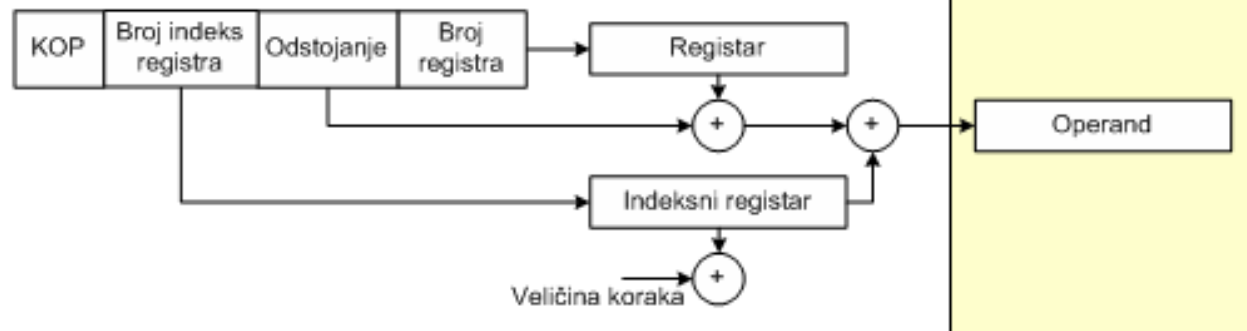


Adresni režimi:

Adresiranje sa odstojanjem

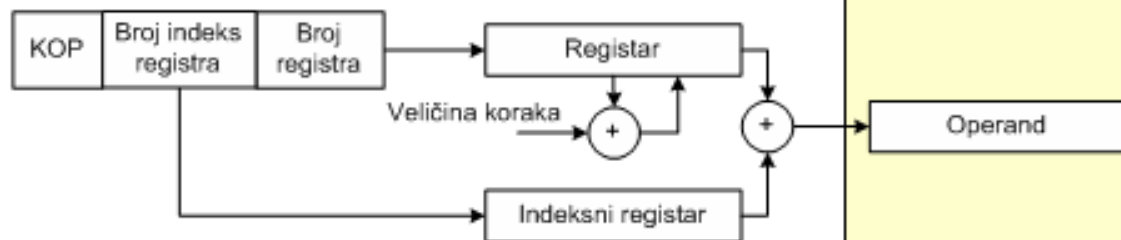


Indeksno adresiranje sa odstojanjem

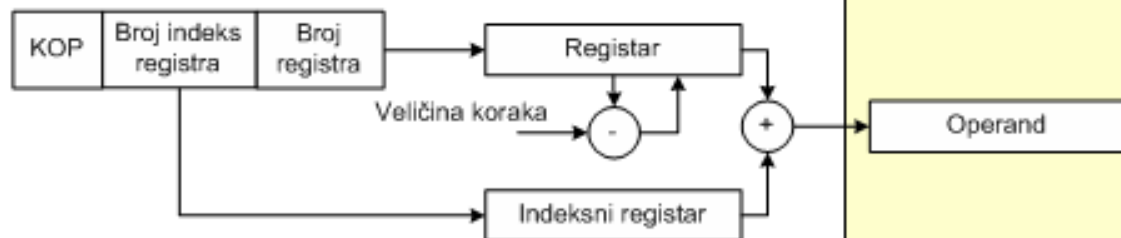


Adresni režimi:

Indeksno adresiranje sa samouvećanjem



Indeksno adresiranje sa samoumanjenjem





Upoređenje procesora sa kompleksnim i redukovanim setom instrukcija

- **CISC** (*Complex Instruction Set Computers*)
 - optimizovani za više programske jezike i kompaktne programe
- **RISC** (*Reduced Instruction Set Computers*)
 - postizanje visoke propusnosti i brzog izvršavanja instrukcija

	CISC	RISC
Pristup memoriji	Dostupan većini instrukcija	Load, Store (ostale su reg-reg)
Adresni režimi	brojni	ograničeni
Format instrukcija	varijabilan	fiksne dužine
Skup instrukcija	elementarne i kompleksne instrukcije	elementarne instrukcije

SPOLJNI SPREŽNI SIGNALI PROCESSORA

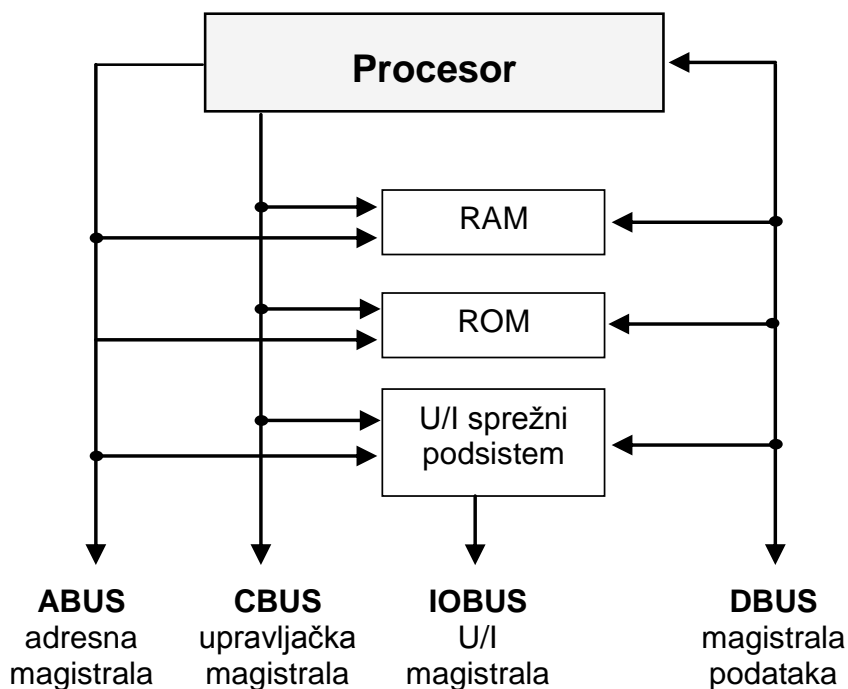




Spoljni sprežni signali procesora

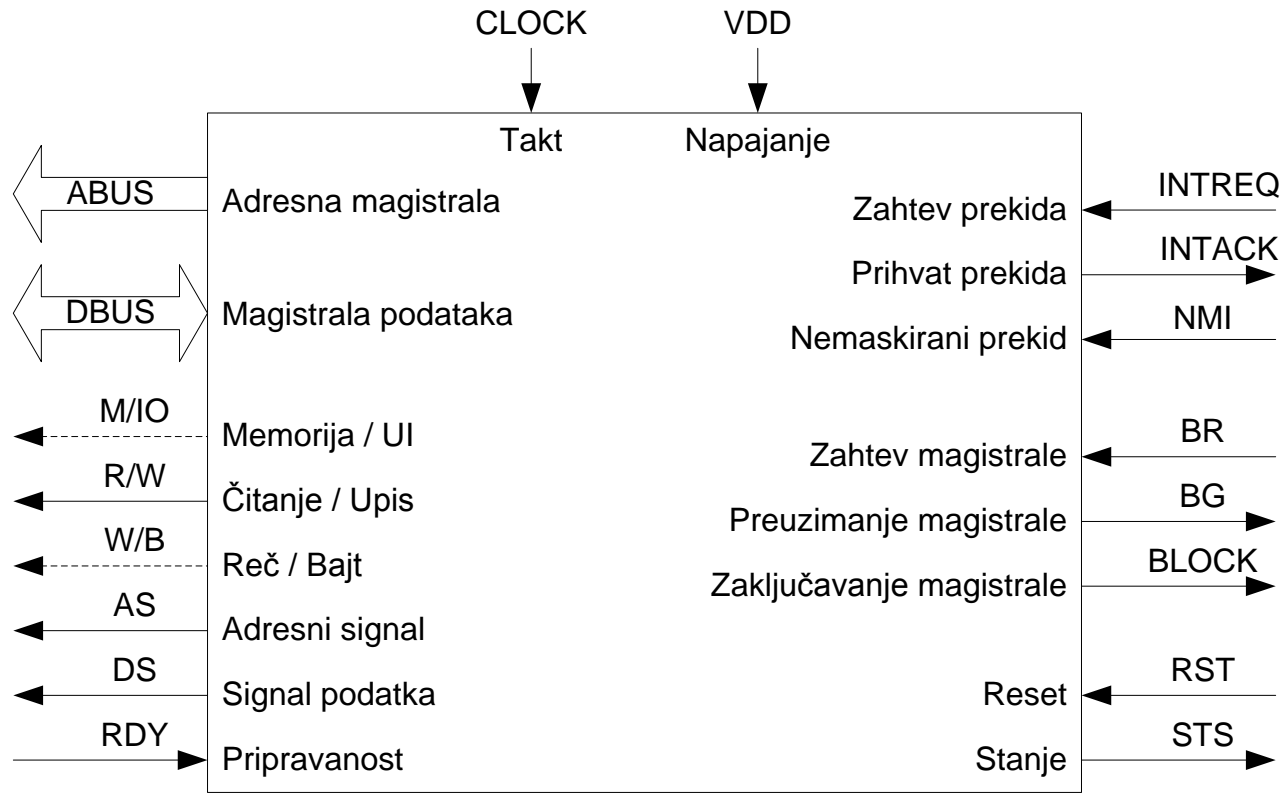
- Standardizacija sprežnih signala
 - napajanje,
 - signal takta,
 - signali pomoću kojih procesor komunicira
 - sa memorijom,
 - ulazno-izlaznim podsistemom, ili
 - sa drugim procesorima.

Sprežne magistrale mikroprocesora



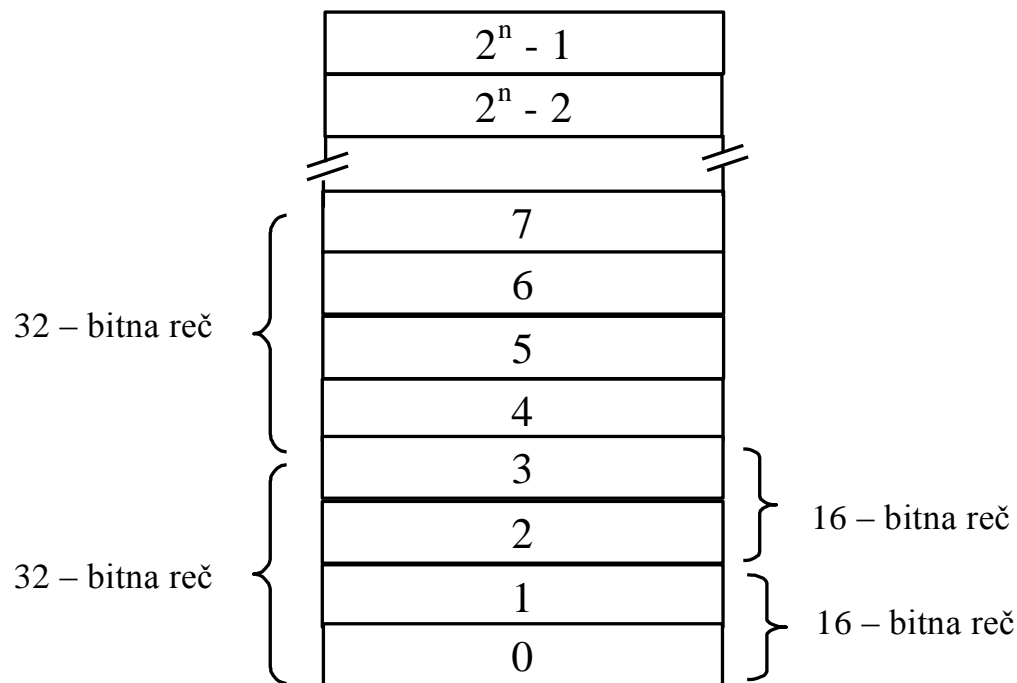
- **ABUS** – adresiranje memorije i U/I uređaja
 - jednosmerna, **m** linija
 - određuje kapacitet memorije (2^m)
- **DBUS** - prenos podataka
 - dvosmeran prenos, **n** linija (reč)
 - korišćenje kapija sa tri stanja
- **CBUS** - skup kontrolnih signala za
 - komunikaciju preko dve magistrale, i
 - sve ostale potrebno za uspešno funkcionisanje računarskog sistema
- **IOBUS** – sprega sa UI uređajima
 - IC sprega – SPI, I2C

Spoljni sprežni sistem hipotetičkog procesora



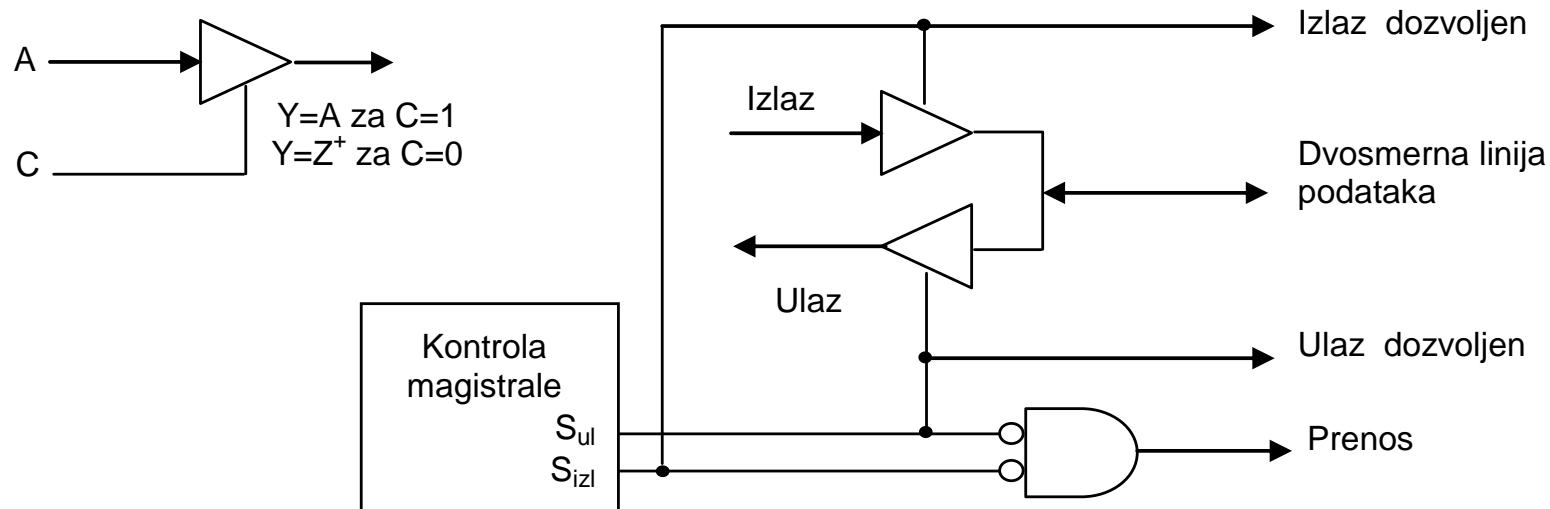
Adresna magistrala

- Organizacija adresnog prostora računarskog sistema
 - Byte (oktet) – reč (8, 16, 32, ...)



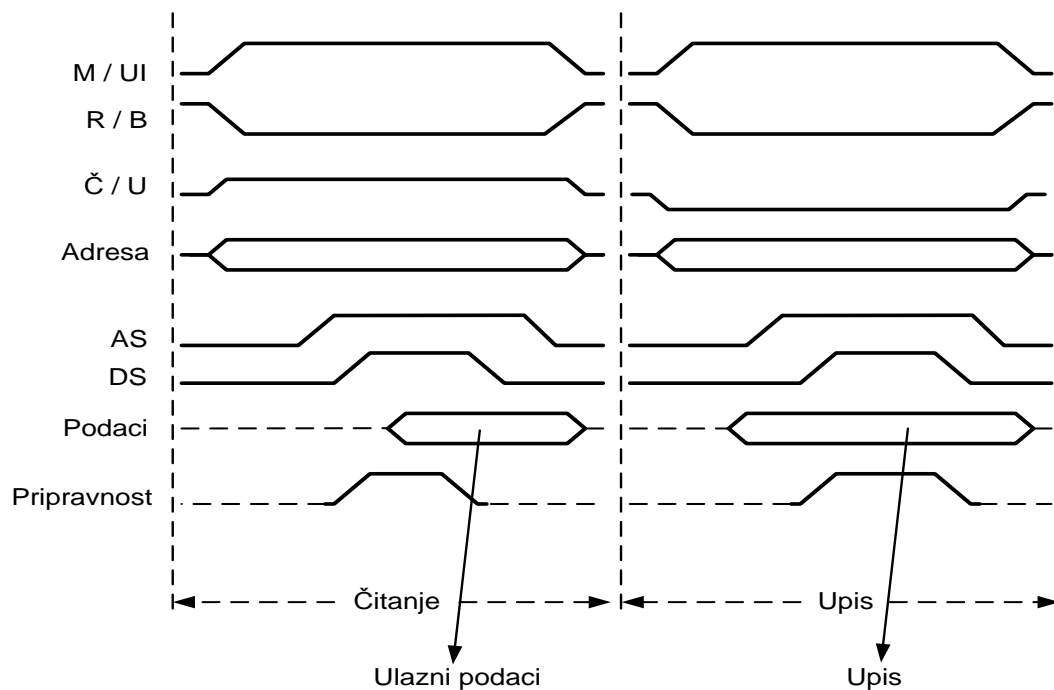
Magistrala podataka

- Broj linija podataka = dužina reči
- Realizacija dvosmerne magistrale podataka



Upravljačke linije za prenos podataka

- M/UI: *Memory/IO*
- R/W: *Read/Write*
- W/B: *Word/Byte*
- AS: *Address Strobe*
- DS: *Data Strobe*
- RDY: *Ready*

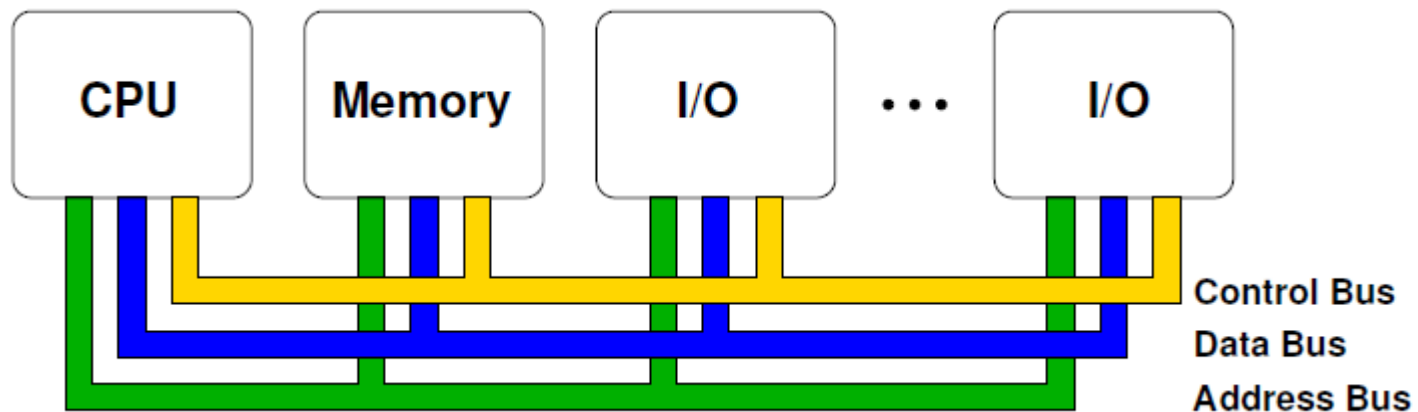




Ostali kontrolni signali

- Linije za rukovanje prekidima
 - *IntReq, IntAck*
- Linije za rukovanje magistralom
 - *BusReq, BusGranted, BusLock*
- Druge upravljačke linije
 - *Reset, InReset*, stanje procesora i sl.

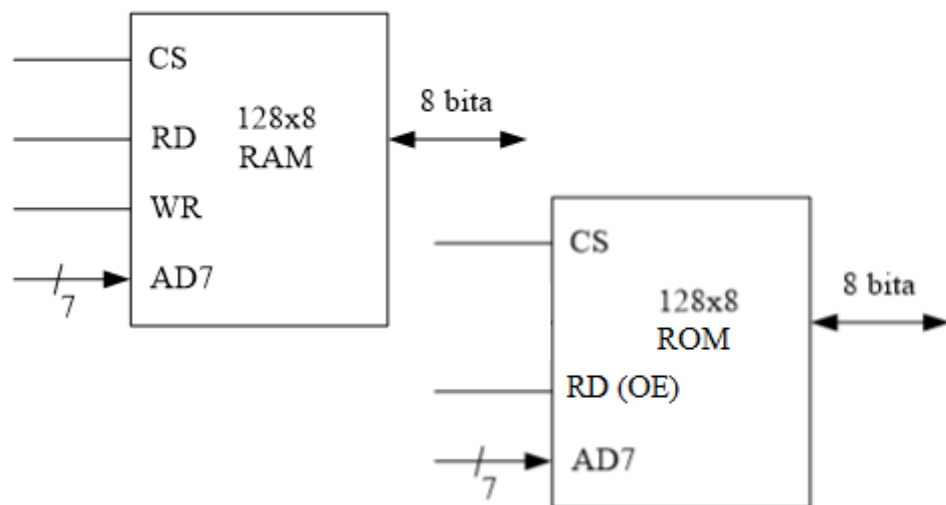
SPREGA CENTRALNOG PROCESORA SA MEMORIJOM I UI UREĐAJIMA



Memorija i UI uređaji: periferije

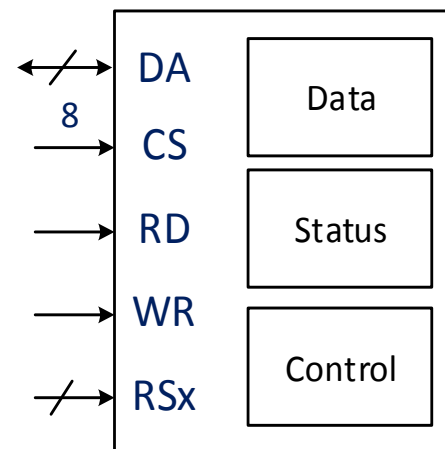
■ Memorija

- Skup registara za podatke
- Pristup po adresi – AD
- **RAM**, Flash: (rw)
- **ROM**, EPROM, Flash: (r)

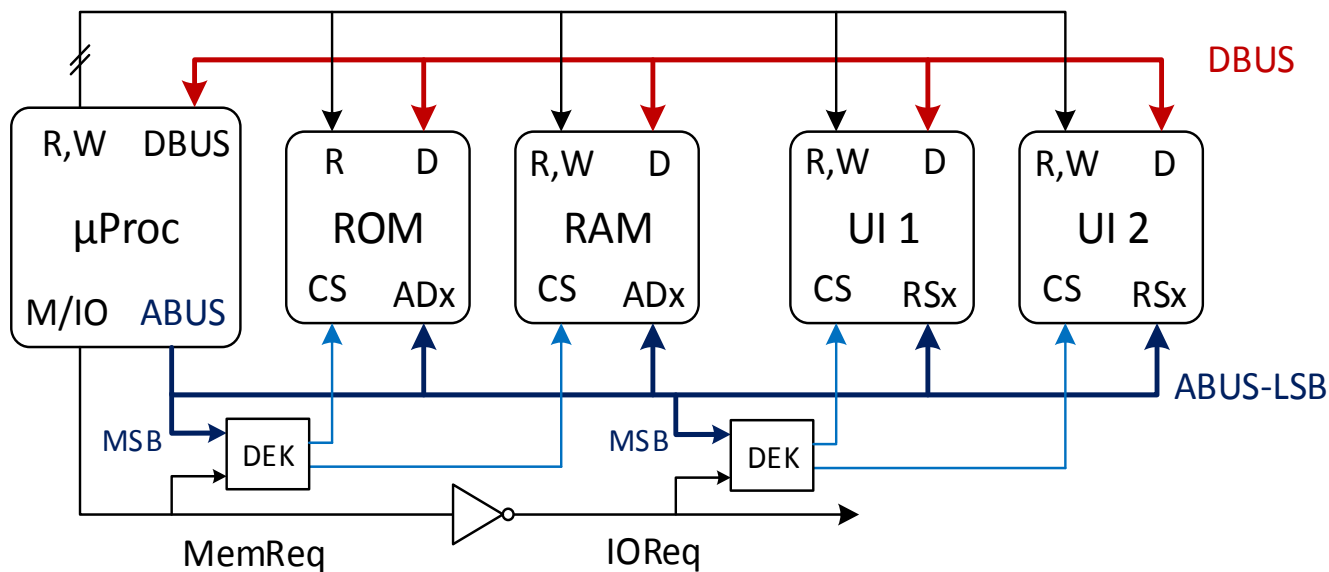


■ UI kontroler

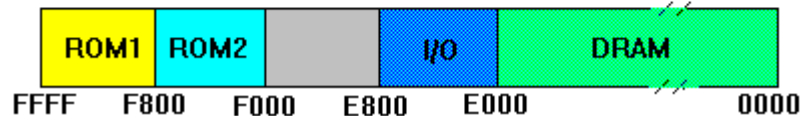
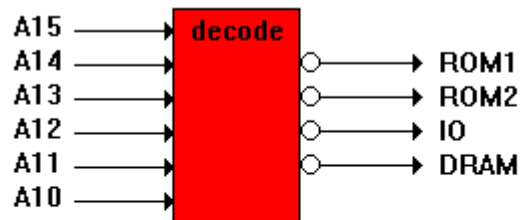
- Skup registara
- Status, kontrola, podaci
- Pristup po adresi - RS



Sprega sa periferijama

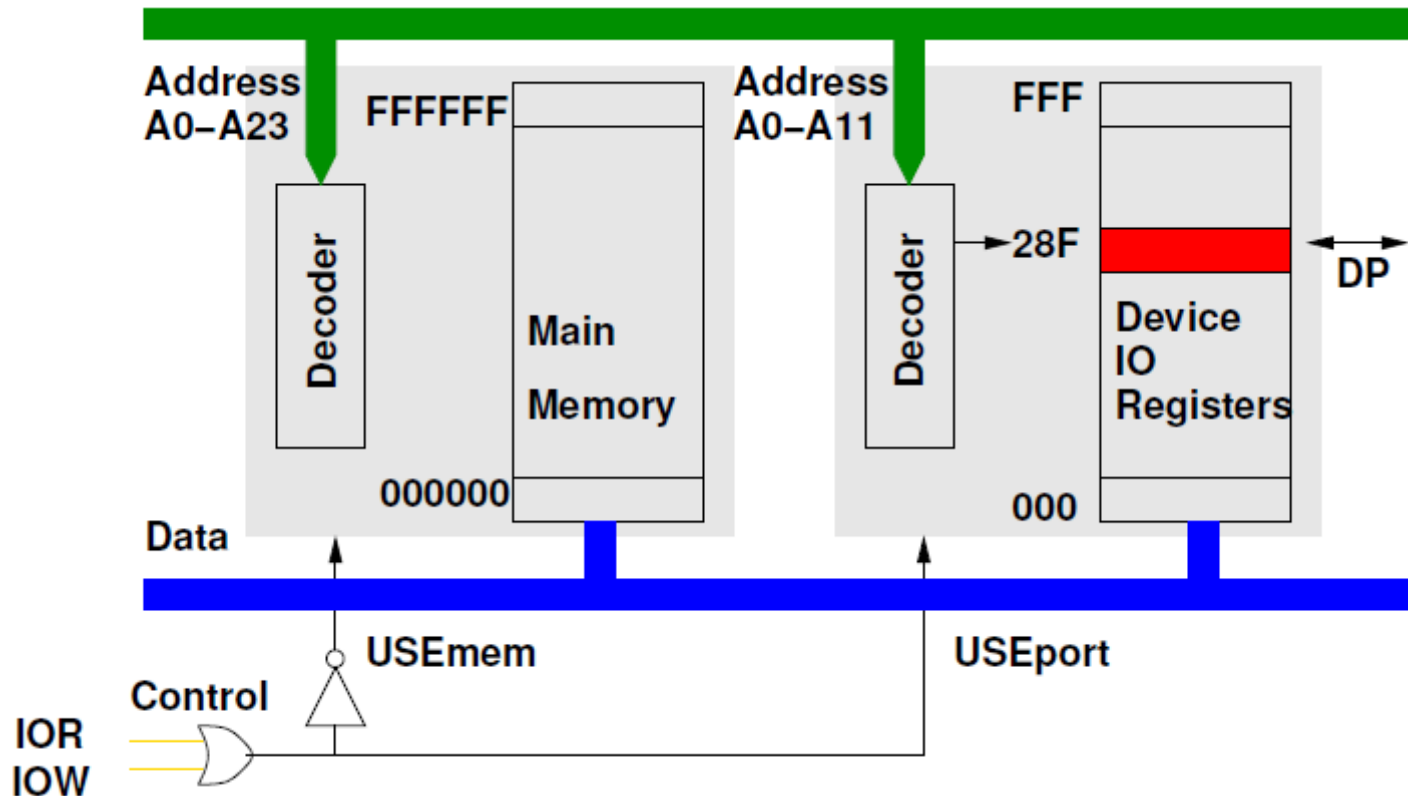


- R,W + DBUS
- ABUS, adresni dekođer
- Adresna mapa
 - Memorijski adresni prostor
 - UI adresni prostor



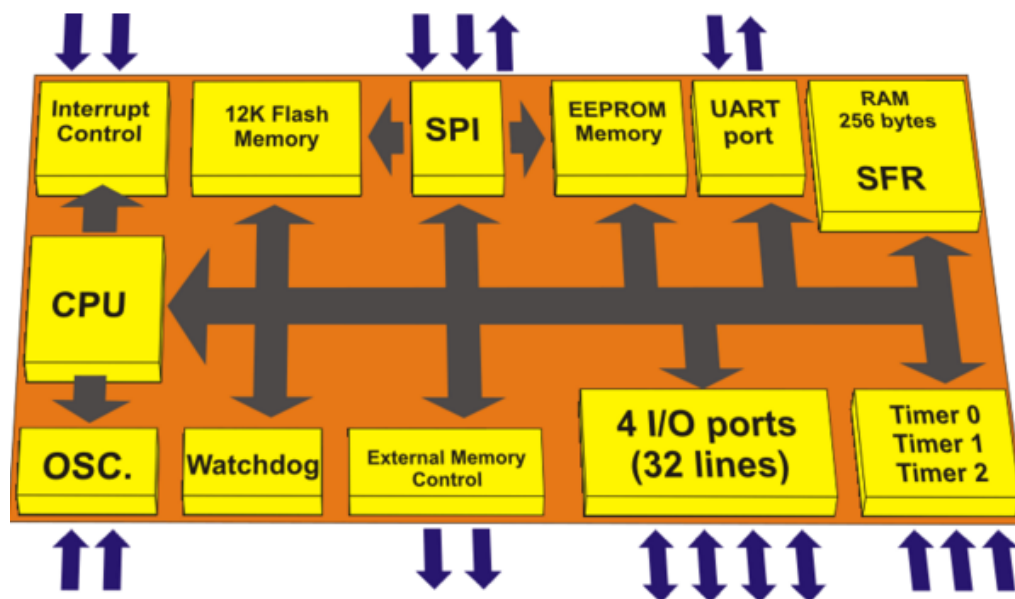
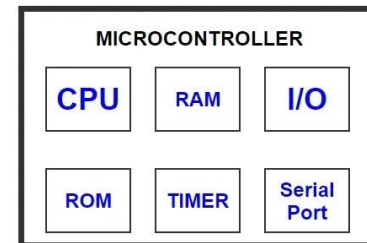
Primer:

MEM/UI prostor sa 24/12 adresnih bita



Mikrokontroler : Mikroprocesor ?

- Mikroprocesor – CPU u integrisanom kolu
- Mikrokontroler – CPU+Mem+IO u jednom kućištu
 - *single chip computer*
- Primer: Atmel **AT89S8253**, 8051 CPU



(T2) P1.0	1	40	VCC
(T2 EX) P1.1	2	39	P0.0 (AD0)
P1.2	3	38	P0.1 (AD1)
P1.3	4	37	P0.2 (AD2)
(\overline{SS}) P1.4	5	36	P0.3 (AD3)
(MOSI) P1.5	6	35	P0.4 (AD4)
(MISO) P1.6	7	34	P0.5 (AD5)
(SCK) P1.7	8	33	P0.6 (AD6)
RST	9	32	P0.7 (AD7)
(RXD) P3.0	10	31	\overline{EA}/VPP
(TXD) P3.1	11	30	ALE/ \overline{PROG}
($\overline{INT0}$) P3.2	12	29	\overline{PSEN}
($\overline{INT1}$) P3.3	13	28	P2.7 (A15)
(T0) P3.4	14	27	P2.6 (A14)
(T1) P3.5	15	26	P2.5 (A13)
(\overline{WR}) P3.6	16	25	P2.4 (A12)
(\overline{RD}) P3.7	17	24	P2.3 (A11)
XTAL2	18	23	P2.2 (A10)
XTAL1	19	22	P2.1 (A9)
GND	20	21	P2.0 (A8)

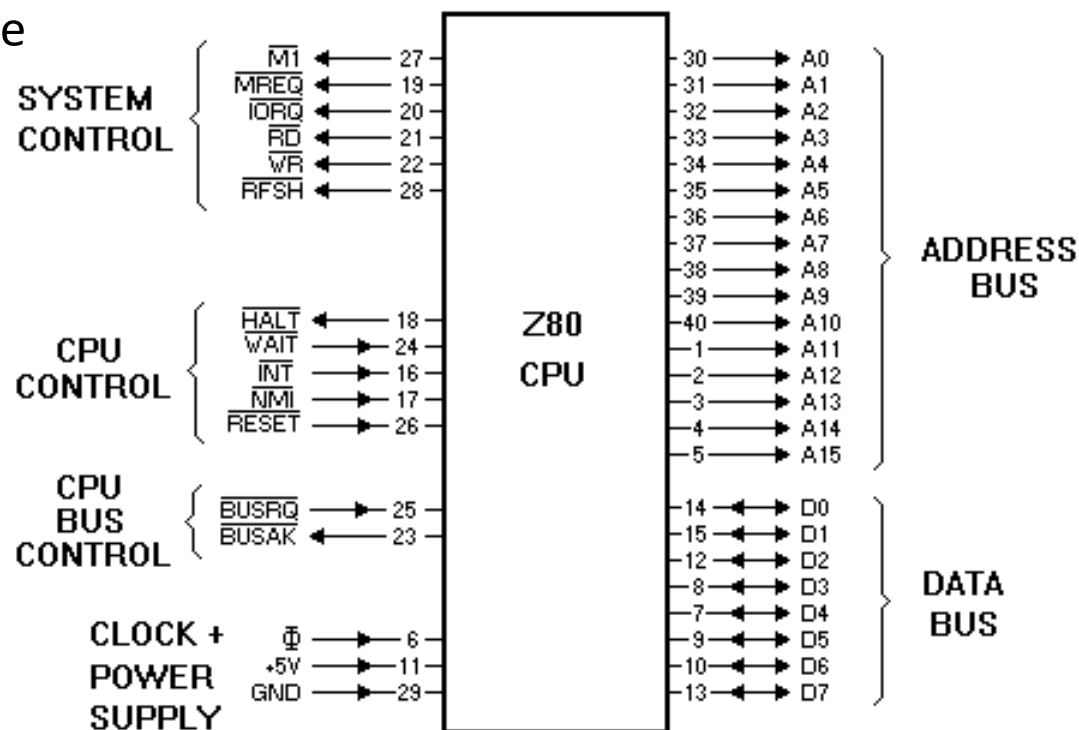


Sprega sa memorijom i UI uređajima

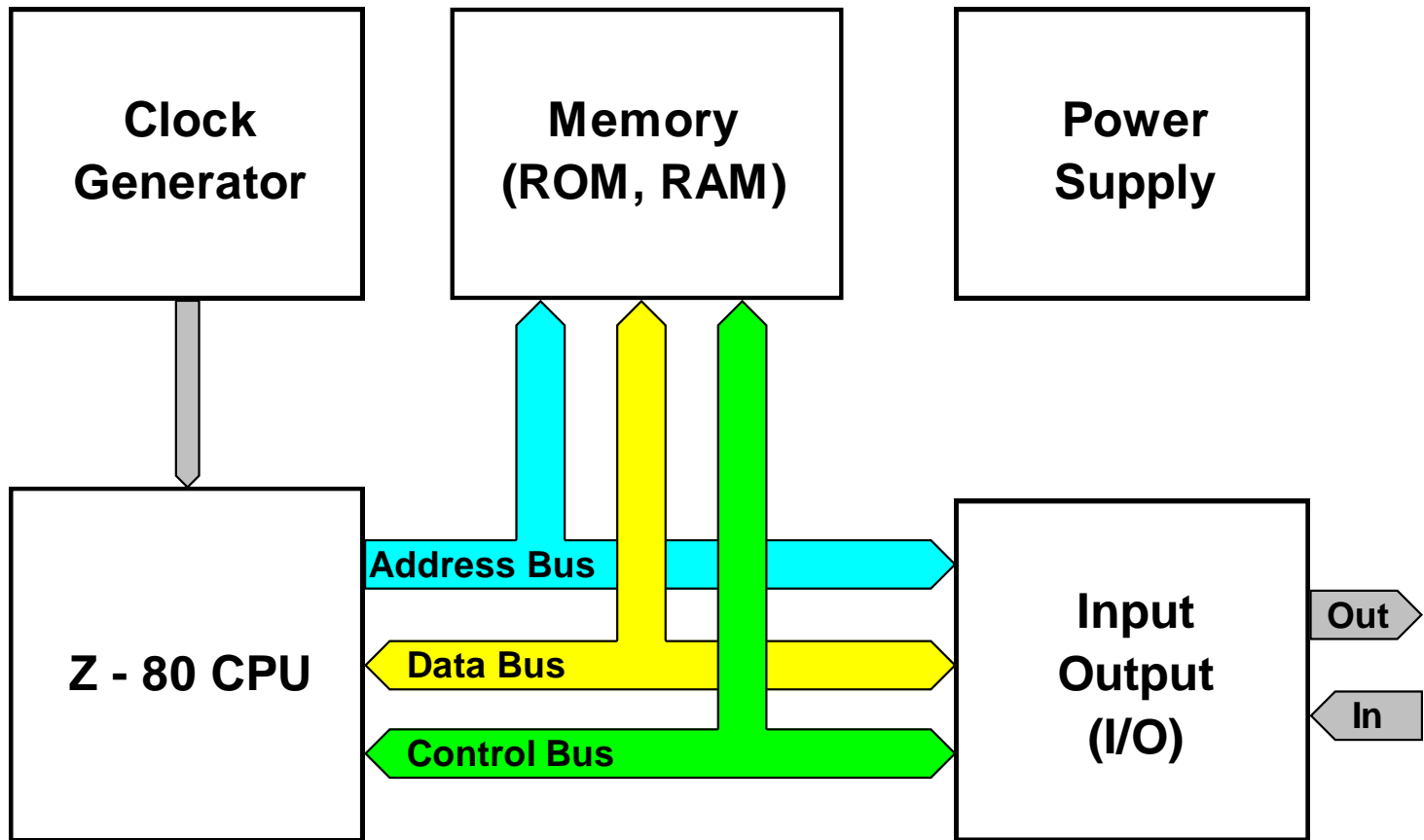
Povezivanje preko ABUS/DBUS

Pregled izlaganja

- Z80 procesor, i njegova
- Sprega sa memorijom i UI
- Memorijska mapa
- Puno i parcijalno dekodiranje

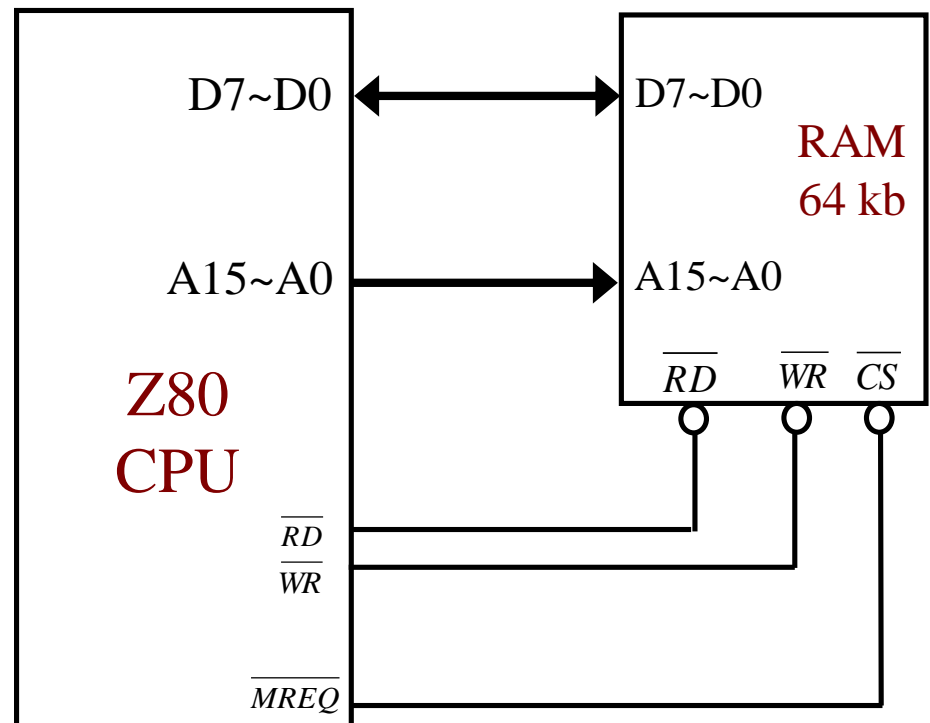


Minimalna konfiguracija Z80 μ -računara



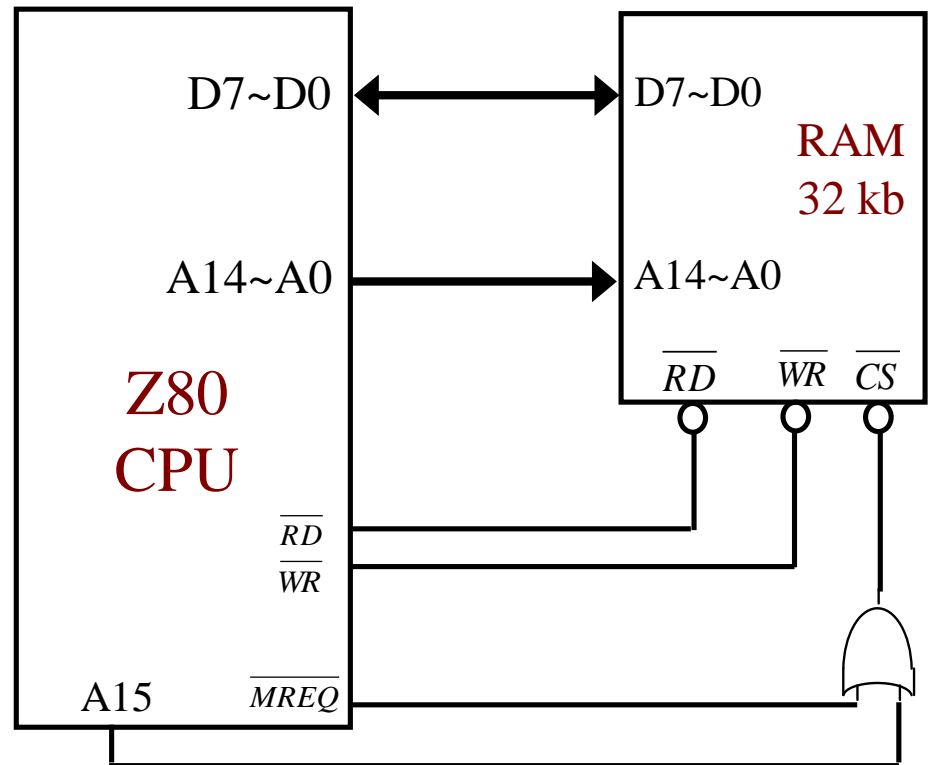
Z80 sprega sa memorijom

- 16 adresnih bita ABUS → 64k reči (max adresa FFFFh)
- 8 bita podataka DBUS → 8-bitna reč
- Po pravilu treba povezati
 - Data ↔ Data
 - Address ↔ Address
 - WR ↔ WR
 - RD ↔ RD
 - MREQ ↔ CS
- 1 RAM modul
 - 64 kb veličine
 - Pun memorijski opseg



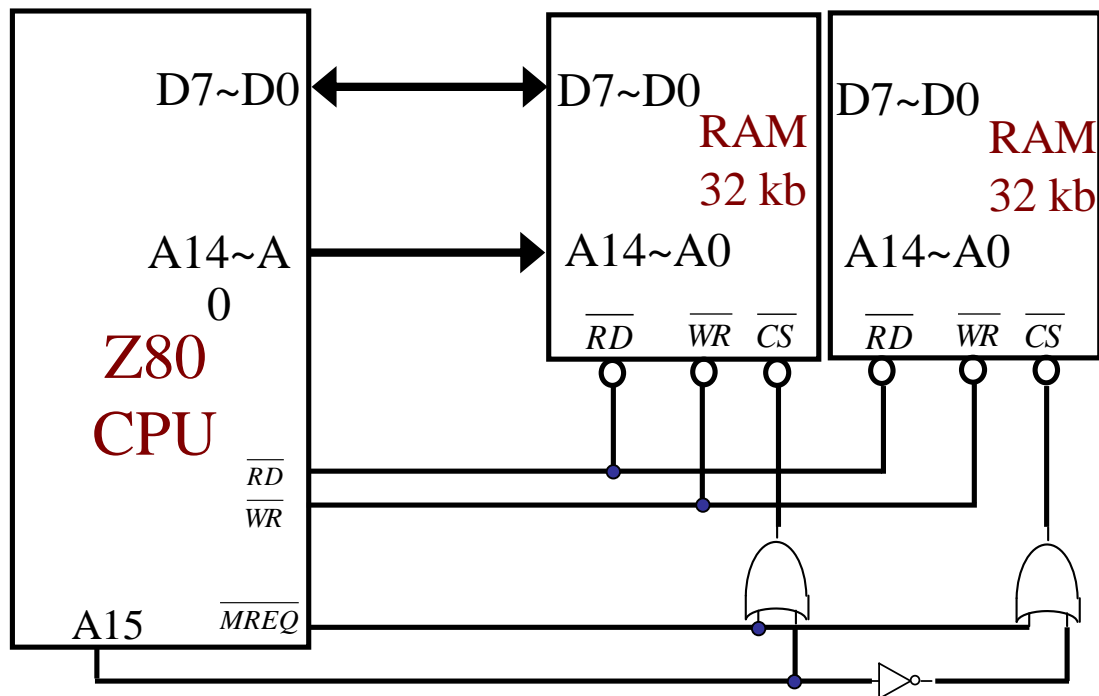
Ako je RAM čip samo 32 kB

- A15 se kombinuje sa MREQ
 - A15=1 je nelegalno
- RAM opseg 0000h - 7FFFh



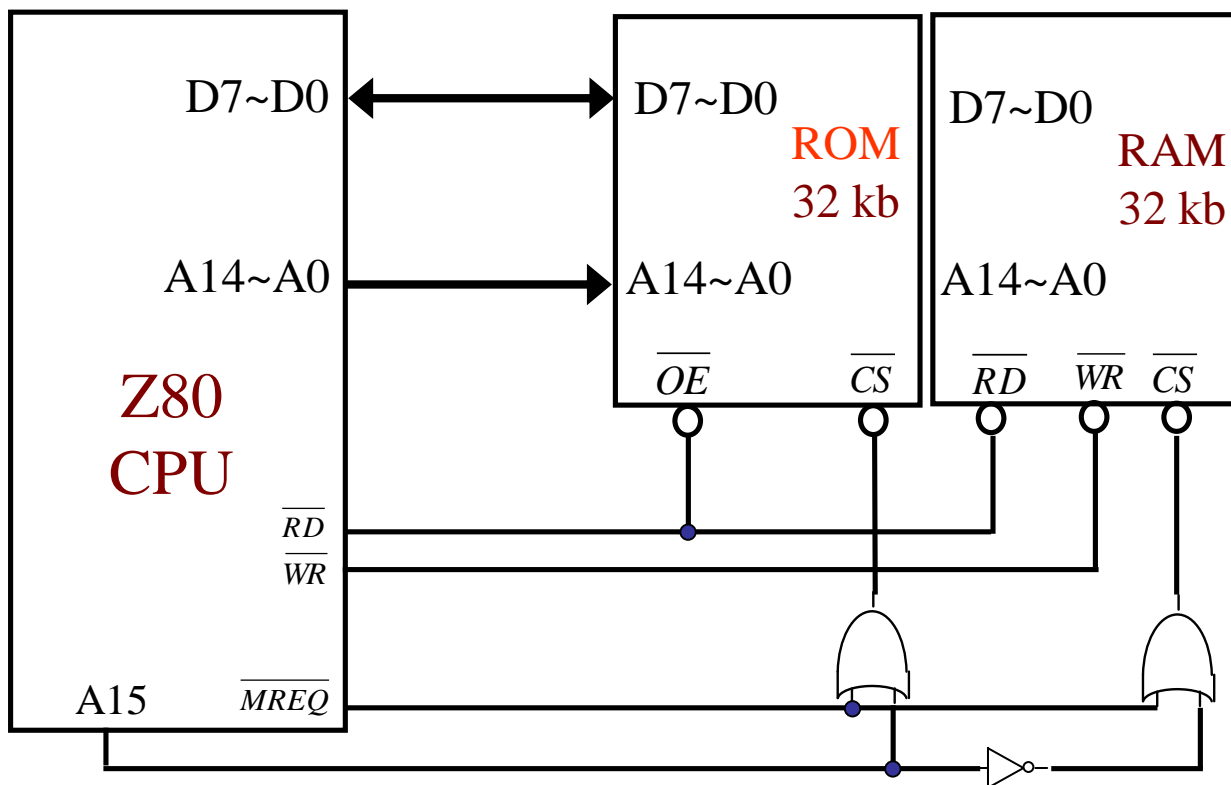
Ako imamo dve RAM memorije po 32 kB

- Potencijalni problem: konflikt na magistrali
 - Oba čipa isporučice podatke istovremeno, kada μ -procesor pokrene ciklus čitanja memorije
- Rešenje: korišćenje A15 adresne linije za arbitražu
 - dozvola pristupa višoj ili nižoj memoriji
- Adresna mapa
 - 0000h - 7FFFh RAM1
 - 8000h - FFFFh RAM2

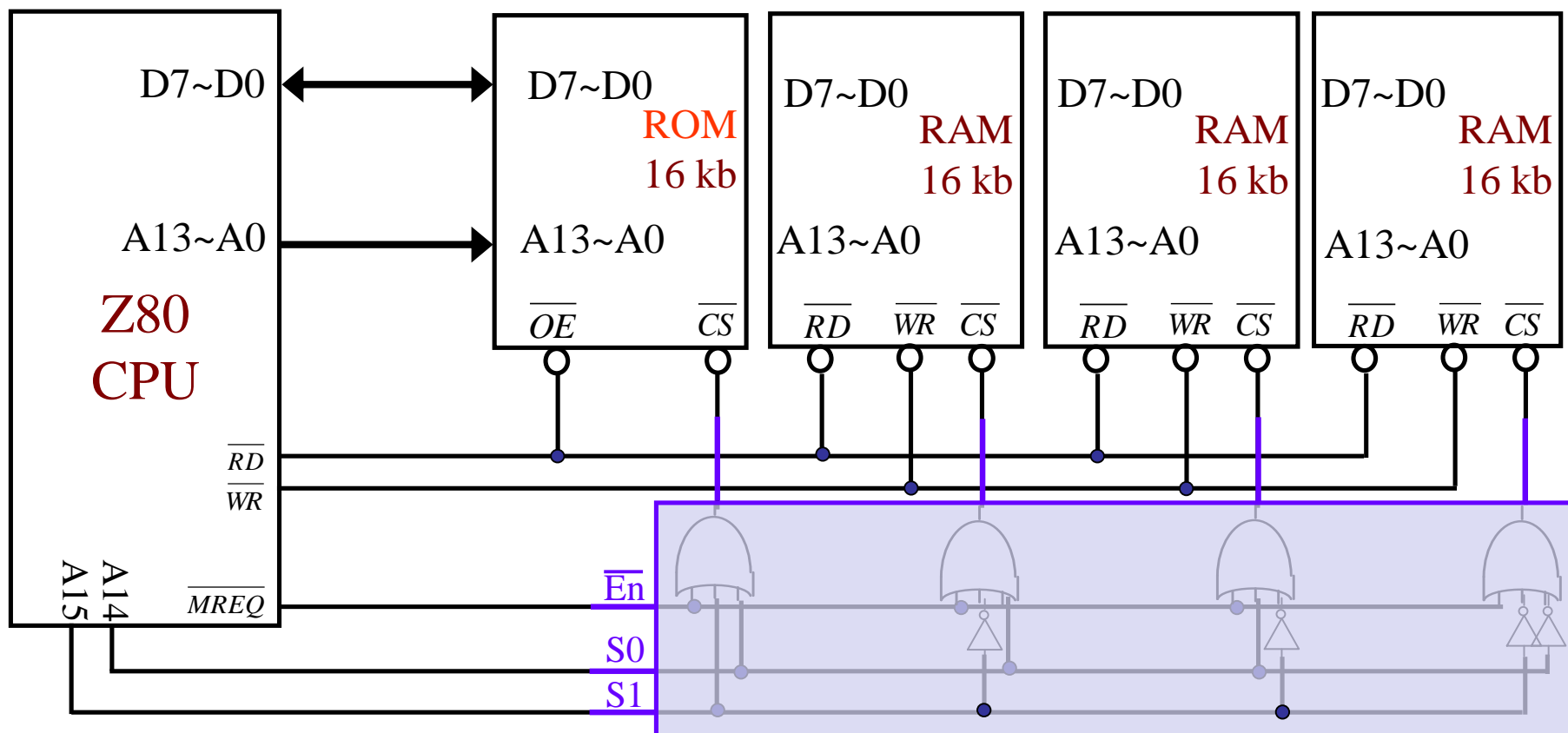


Kombinovanje ROM i RAM memorije

- 32 kB svaki
- ROM nema WR
- Adresna mapa
 - 0000h-7FFFh ROM
 - 8000h-FFFFh RAM



Korišćenje 4 memorijska modula



A14 i A15 za selekciju čipa

Address Bit Map

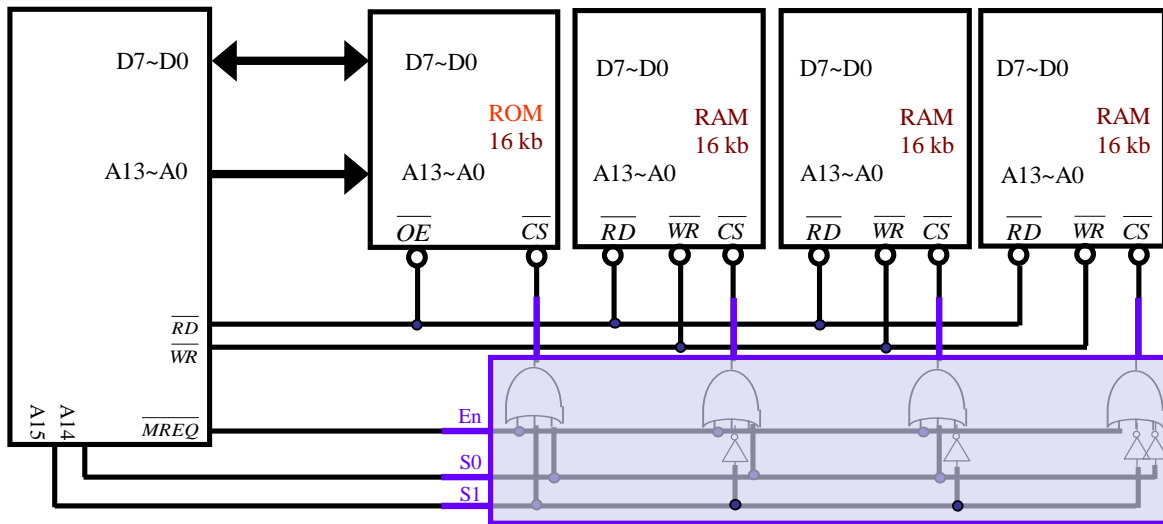
Izbor čipa

Adresa lokacije unutar čipa

A15 to A0 (HEX)	AA 11 54	AA 11 32	AAAA 1198 10	AAAA 7654	AAAA 3210	Memory Chip
0000h	00	00	0000	0000	0000	ROM
3FFFh	00	11	1111	1111	1111	
4000h	01	00	0000	0000	0000	RAM ₁
7FFFh	01	11	1111	1111	1111	
8000h	10	00	0000	0000	0000	RAM ₂
BFFFh	10	11	1111	1111	1111	
C000h	11	00	0000	0000	0000	RAM ₃
FFFFh	11	11	1111	1111	1111	

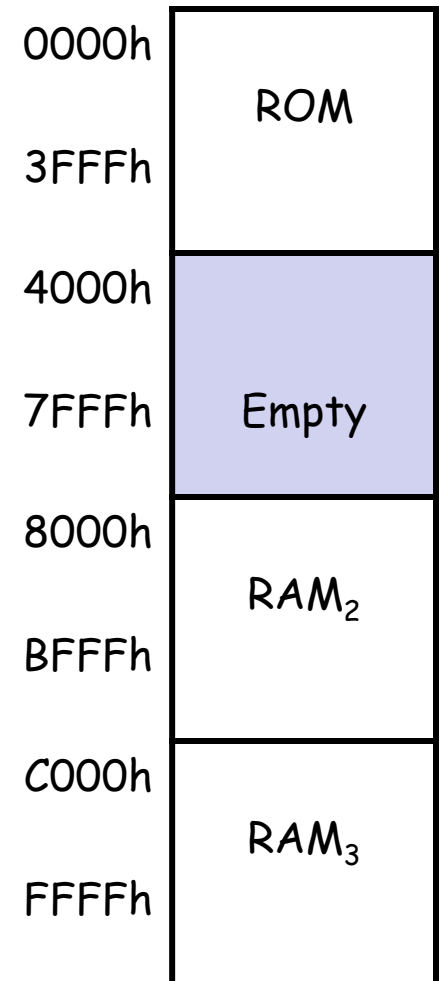
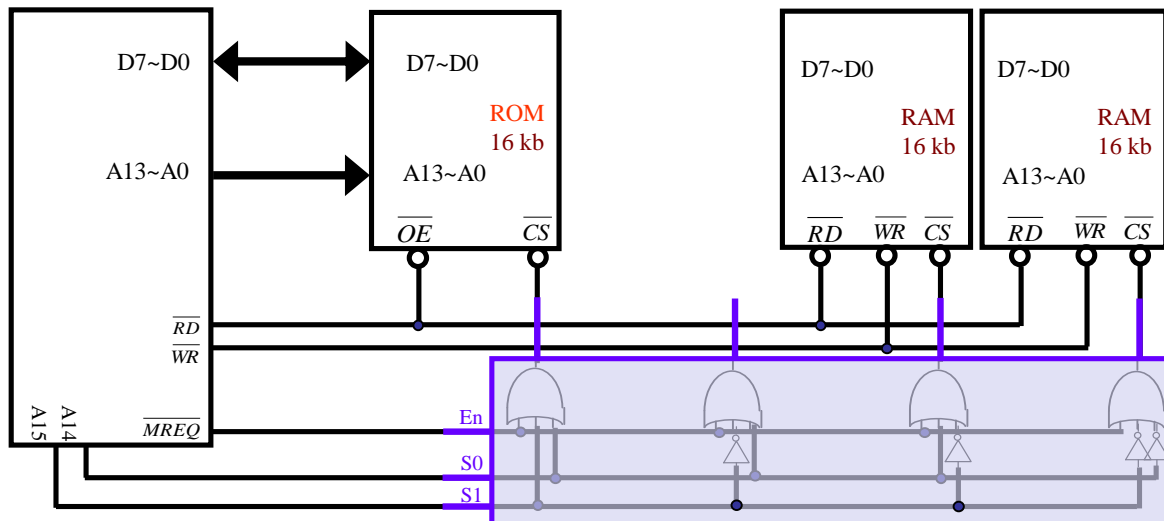
Memorijska Mapa

- Respektuje tip i adresni opseg svakog memorijskog modula



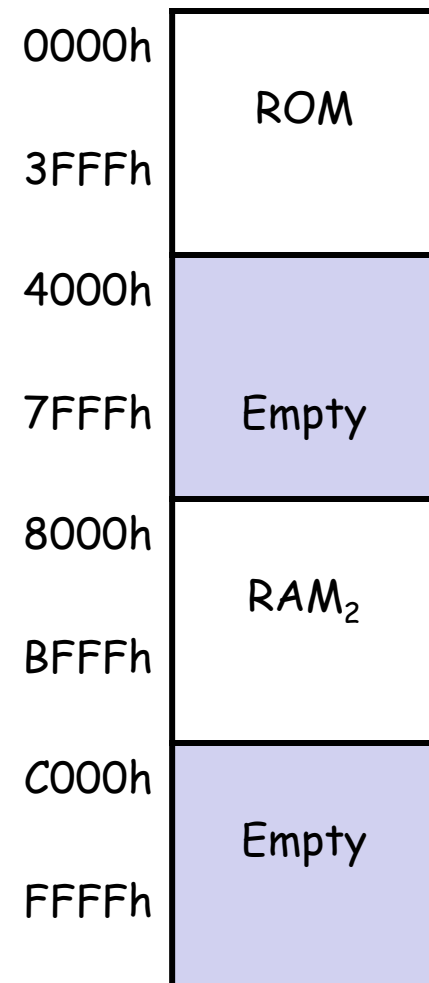
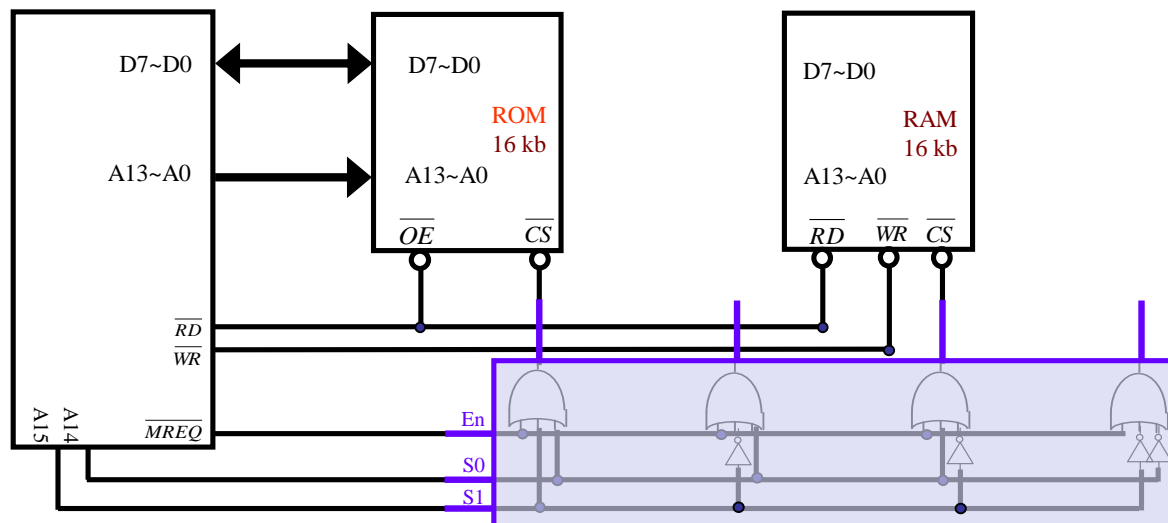
Memorija sa praznom zonom

- Prazna zona ne može biti korišćena
 - Read operacija vraća FFh (obično)
 - Write operacija prosto nema efekta



Memorijska mapa sa dve prazne zone

- Ako fale dva čipa





Puno i parcijalno dekodiranje

- Puno dekodiranje

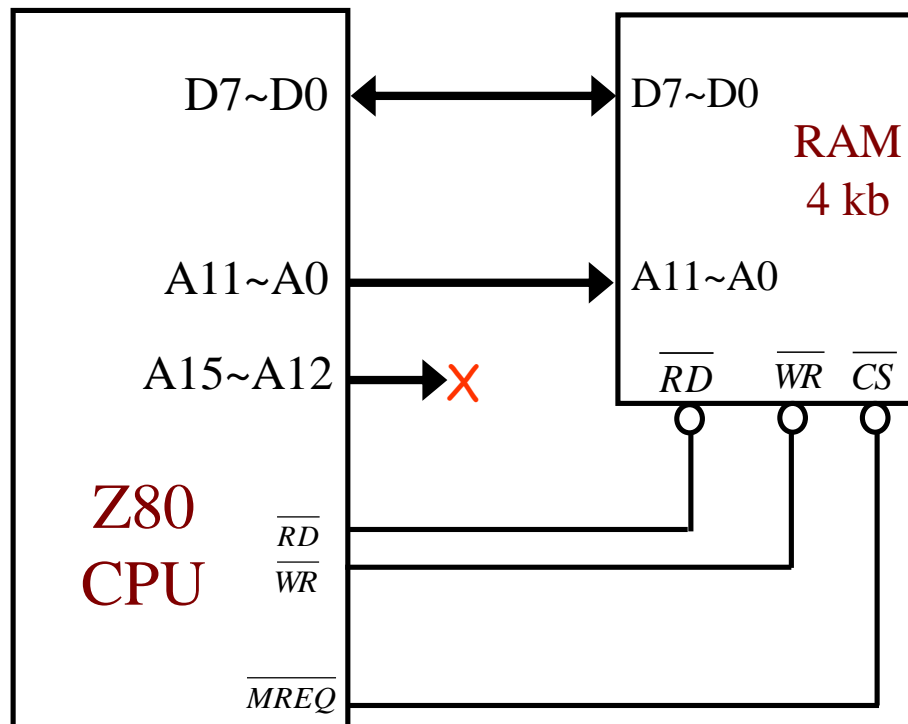
- Sve adresne linije su povezane i koriste se za izbor svakog od memorijskih ili UI modula (čipa)
- Apsolutna adresa – svaka lokacija ima samo jednu adresu

- Nepotpuno adresiranje

- Neke od adresnih linija se koriste za selekciju Mem/UI registara
- Ovo dovodi do preklapanja adresa i adresnih opsega (*fold back, shading*)
- Svaki registar (memorija ili UI) ima više od jedne adrese

Parcijalno dekodiranje

- Linije A15 - A12 nisu vezane, ne utiču na adresiranje uopšte.
- Kakva je adresna mapa?



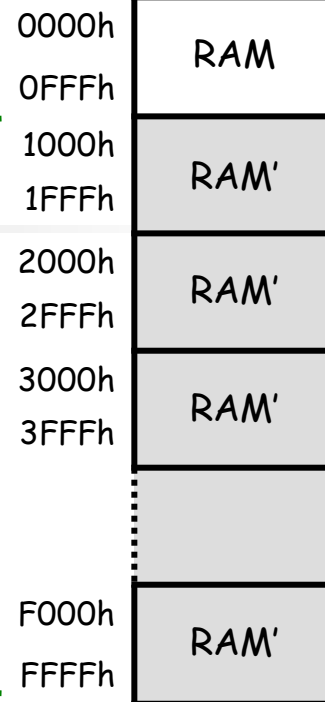
MMapa pri nepotpunom dekodiranju

Svaka lokacija ima **više** adresa

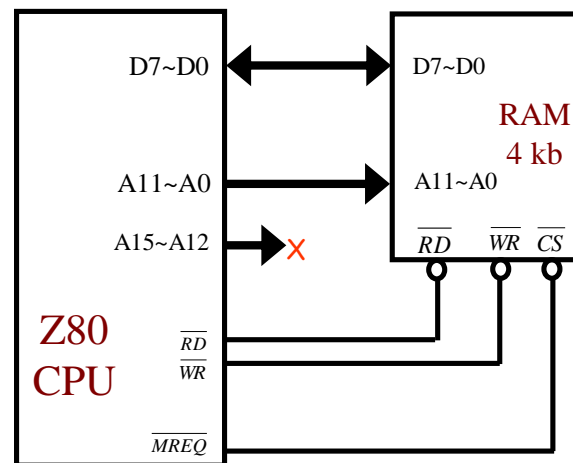
U primeru, prva RAM lokacija ima 1+15 adresa:

0000h
1000h
2000h
3000h
.....
.....
F000h

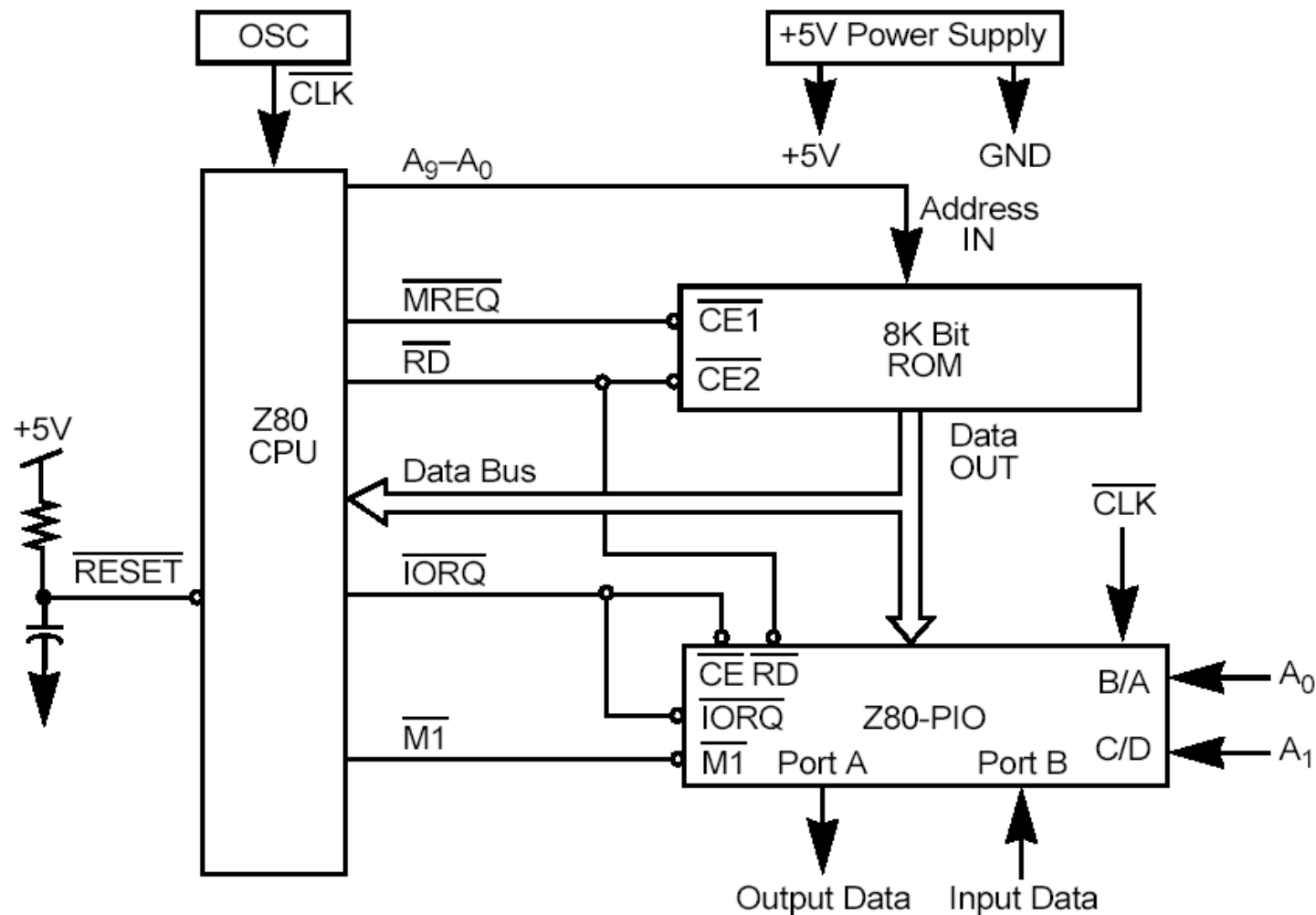
Dodatne Adrese



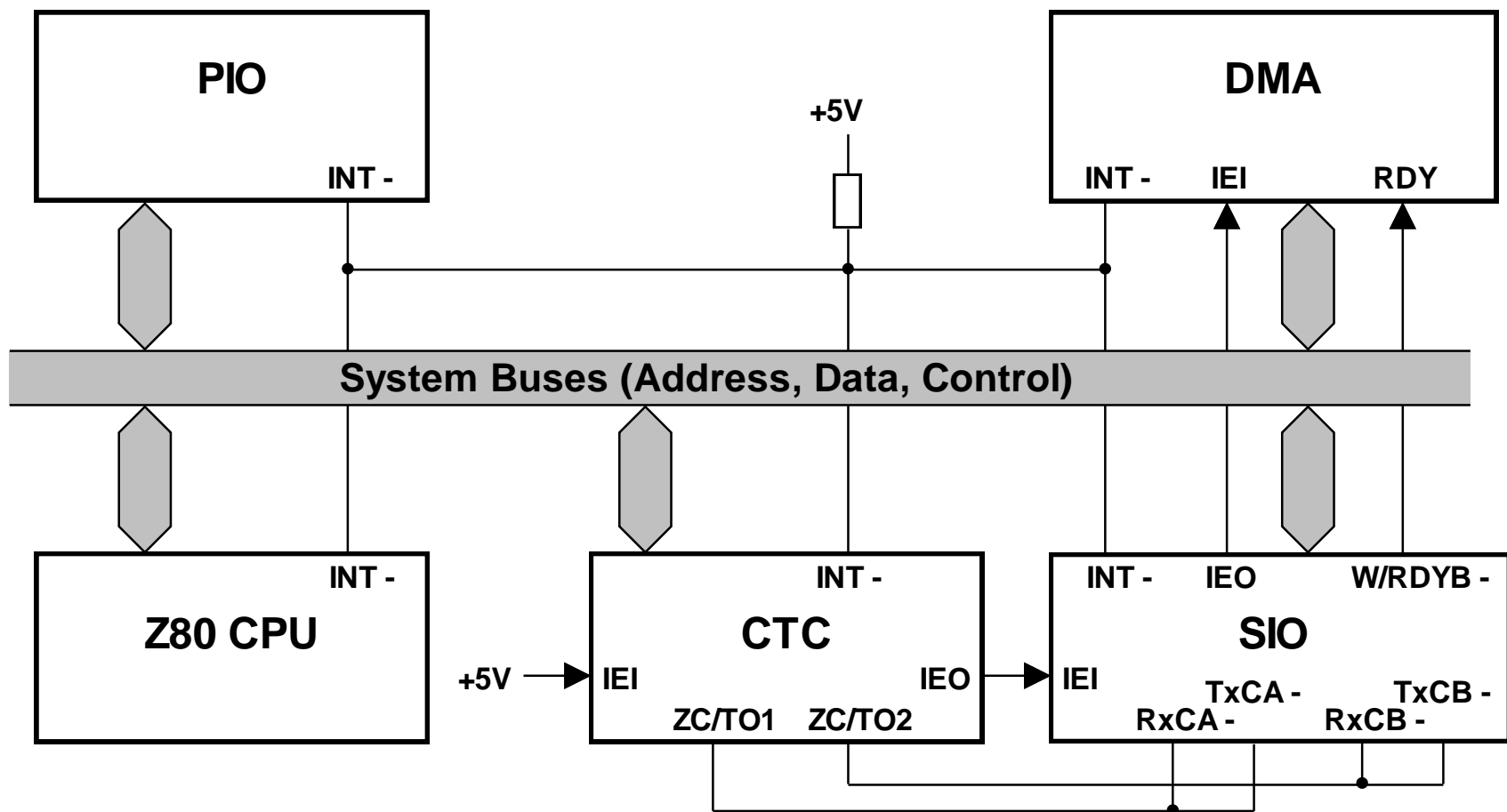
A15 to A0 (HEX)	AAAA 1111 5432	AAAA 1198 10	AAAA 7654	AAAA 3210	Memory Chip
X000h	xxxx	0000	0000	0000	RAM
XFFFh	xxxx	1111	1111	1111	



Minimalni RS sa Z80 procesorom



Z80- μ P-Familija (tipično okruženje)





Z80 Ulaz / Izlaz

- Ne više od 256 input i 256 output portova (registara), u UI adresnom prostoru (izolovan UI)
- Instrukcije IN i OUT
- 8-bitna UI adresa na linijama A7–A0

OUT (n), A

- n is 8 bit port address
- Content of A is data

OUT (C), r

- Content of C is a port address
- r is a data register

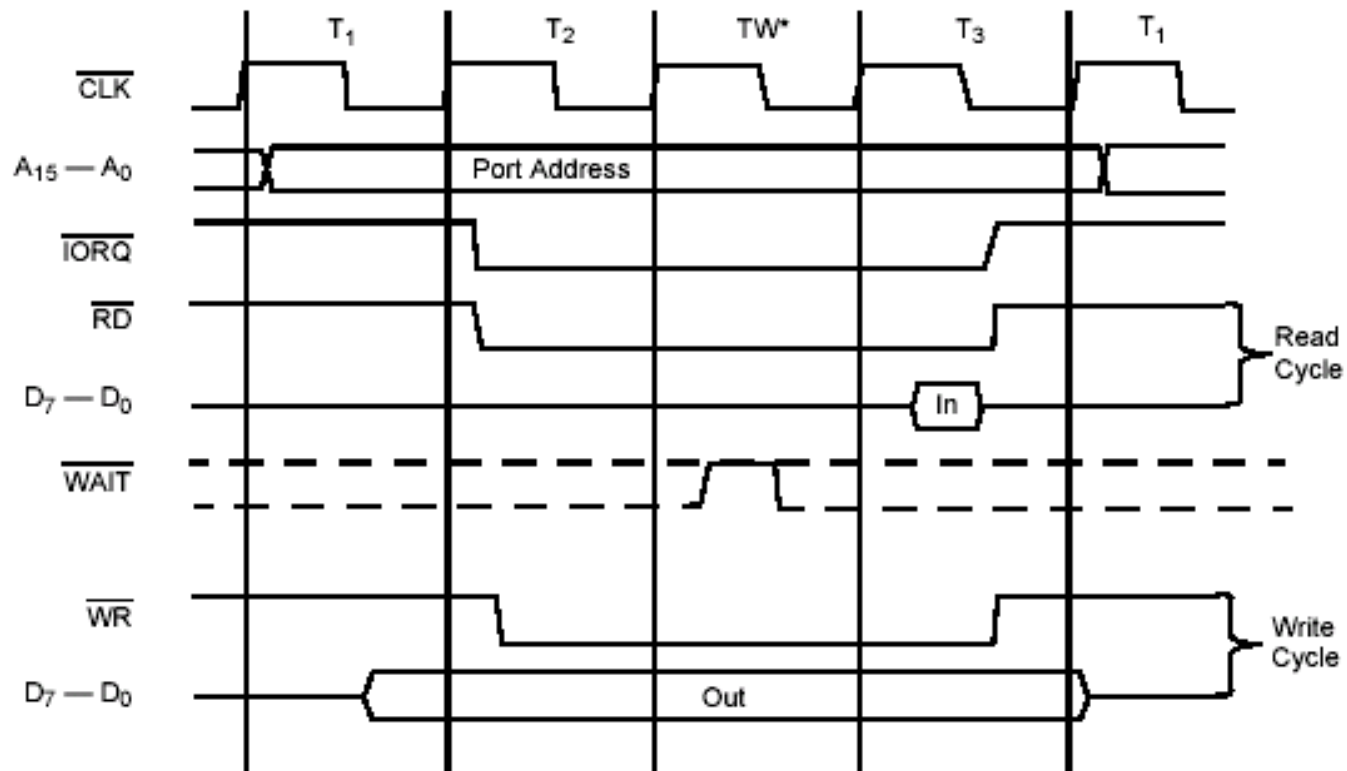
IN A, (n)

- n is 8 bit port address
- Data is transferred to A

IN r (C)

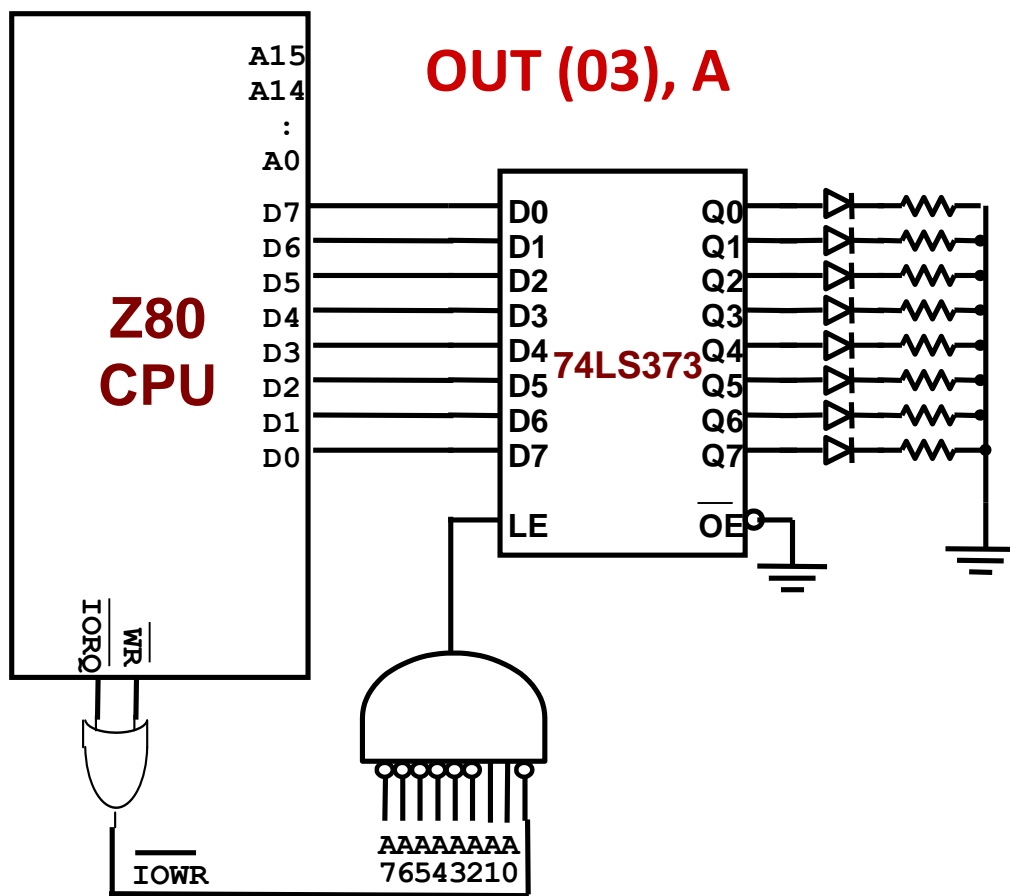
- Content of C is a port address
- Input data transferred to r (data reg)

IO Read/Write ciklus

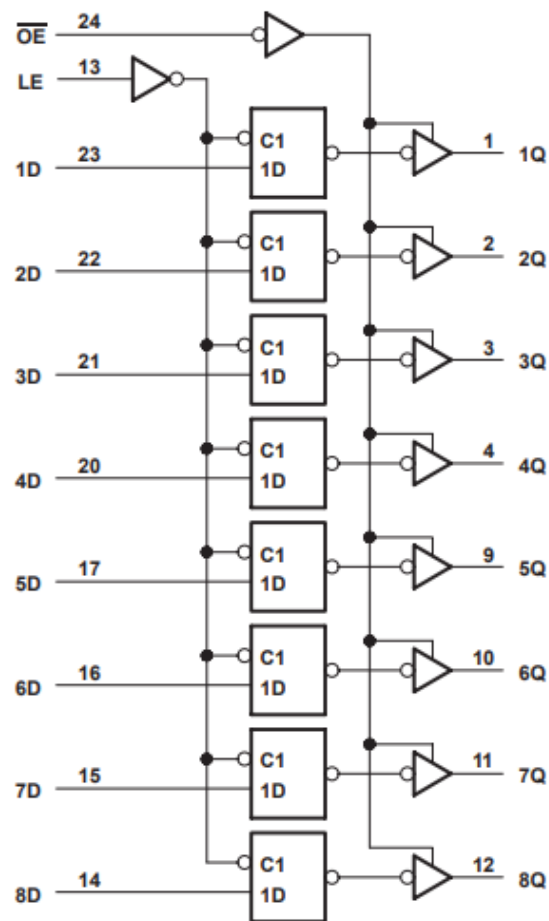


*Automatically inserted WAIT state

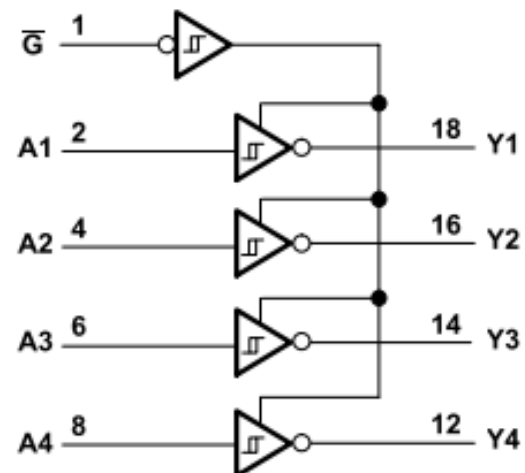
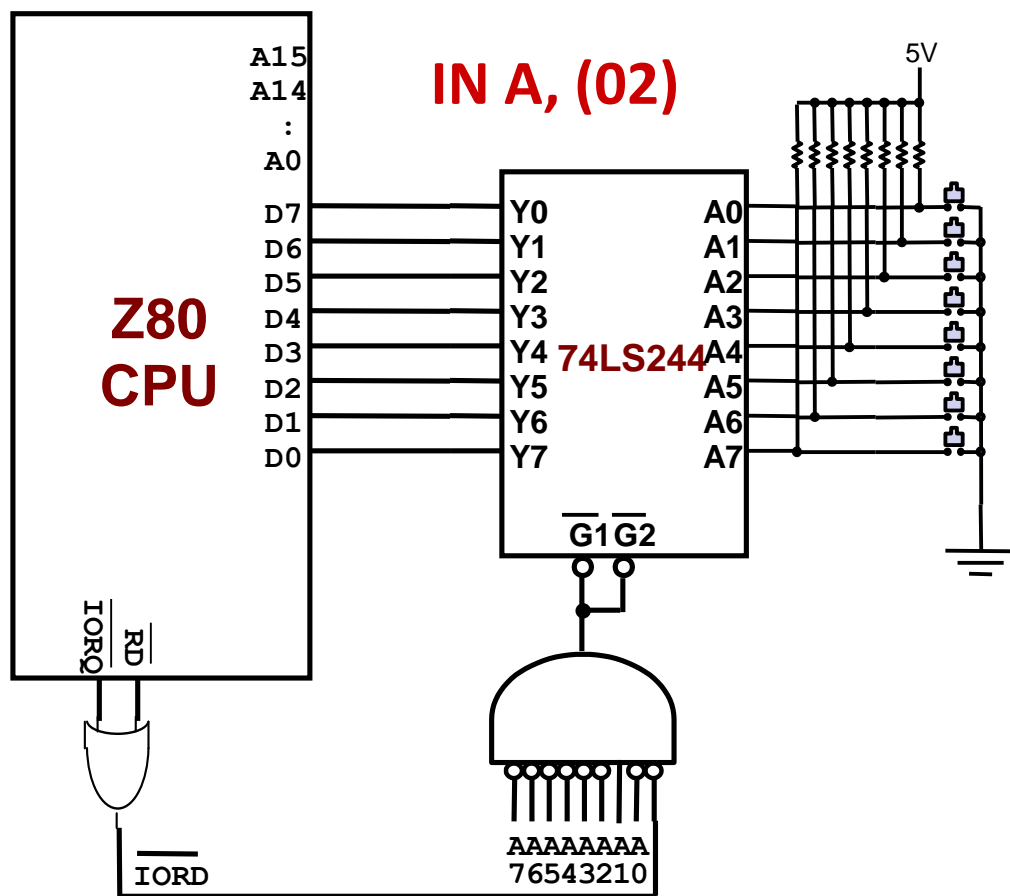
Z80 i jednostavan output port



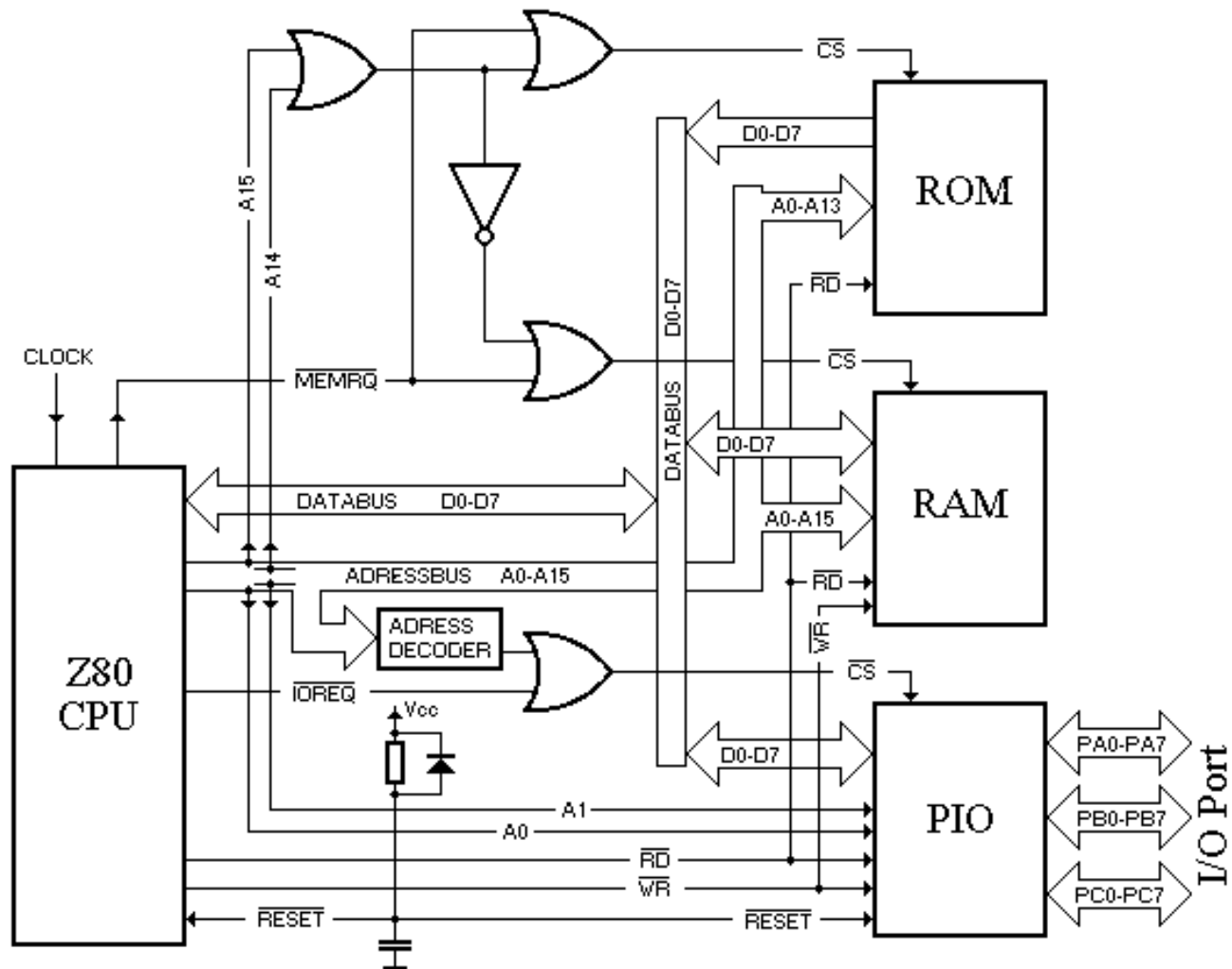
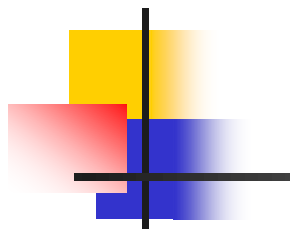
logic diagram (positive logic)



Z80 i jednostavan *input* port



Z80 računar sa memorijom i PIO kolom

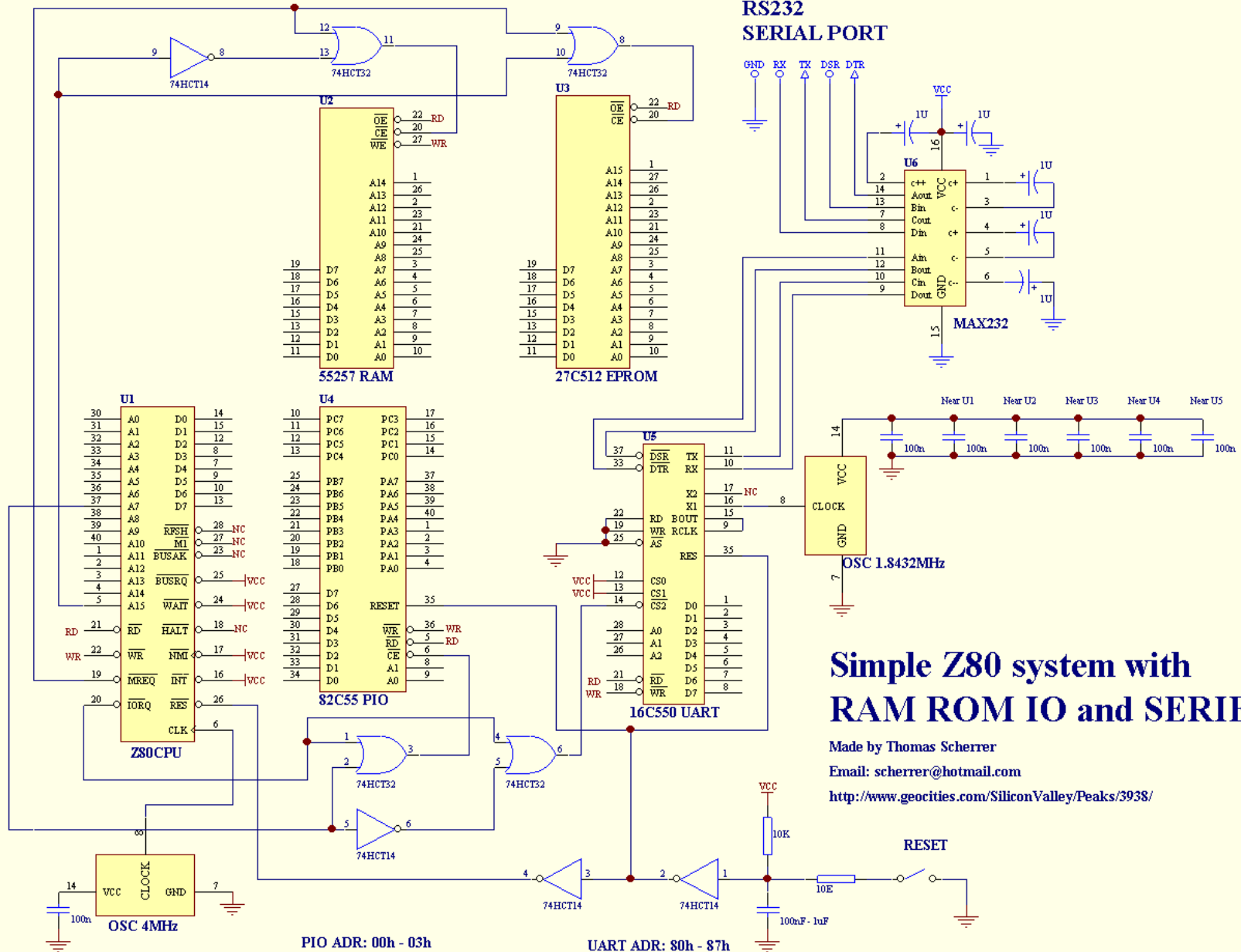


CPU..... Central Processing Unit
 ROM..... Read Only Memory
 RAM..... Random Access Memory
 PIO..... Peripheral Input Output

CS..... Chip Select
 RD..... Read
 WR..... Write
 MEMRQ..... Memory Request
 IOREQ..... Input Output Request

32Kb RAM: 8000h - FFFFh

32Kb EPROM: 0000h - 7FFFh



Simple Z80 system with RAM ROM IO and SERIAL

Made by Thomas Scherrer

Email: scherrer@hotmail.com

<http://www.geocities.com/SiliconValley/Peaks/3938/>