



Digitalna vezja UL, FRI



P11, Pomnilnik

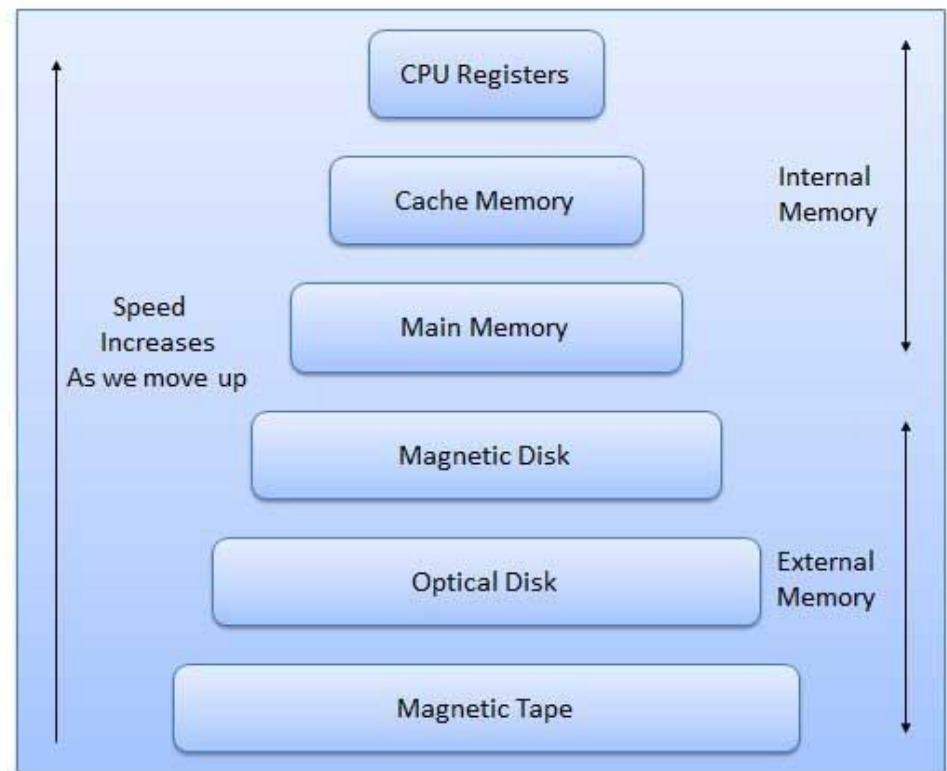
Vsebina

- Pomnilniki - uvod
- ROM (Read Only Memory)
- RAM (Random Access Memory)

- Literatura:
 1. Widmer, S.W, Moss, G.L., Tocci, R.J., Digital Systems, Principles and Applications, Pearson, 2018, poglavje 12 (887 – 959)
 2. Wakerly, J.F., Digital Design, Principles & Practices, Prentice Hall International, 2001, poglavje 9, (799 – 840)

Pomnilnik

- Digitalni sistemi - velike količine podatkov
- Shranjevanje podatkov in ukazov.
- Pomnilne naprave in sistemi
- Tip pomnilnika:
 - Notranji pomnilnik
 - Registri CPE
 - Predpomnilnik
 - Glavni pomnilnik
 - Zunanji pomnilnik
 - Magnetni disk
 - Optični disk
 - Magnetni trak
- Bipolarna in MOS tehnologija
 - Najhitrejši pomnilniki
 - Nižanje cene



http://www.tutorialspoint.com/computer_logical_organization/memory_devices.htm

Osnovni principi

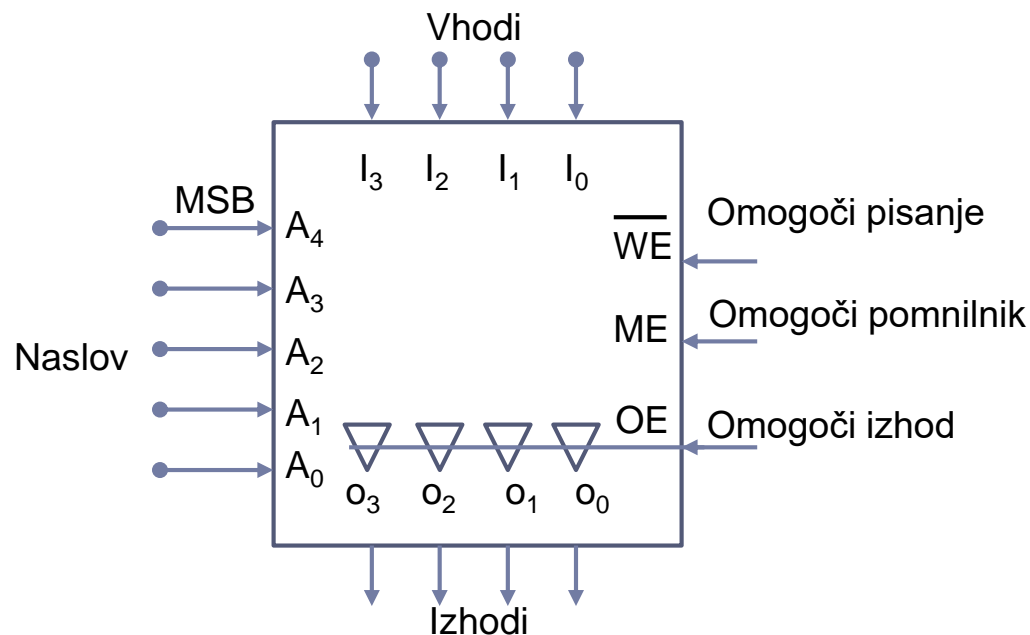
Vsak tip pomnilnika, ki jih bomo spoznali, ima v osnovi drugačne notranje operacije.

Za vse pomnilne sisteme pa veljajo osnovni principi delovanja:

- Uporaba dvojiškega naslova za dostop do pomnilne lokacije.
- Omogočiti pomnilniku, da odgovori na krmilne vhode.
- Posredovanje podatkov, ki so shranjeni na določeni lokaciji, na notranje podatkovne linije (vodilo).
- Branje – omogočiti tri-stanjski izhod, ki posreduje podatke na izhodne pine.
- Pisanje – uporaba podatkov za shranjevanje iz vhodnih pinov.
- Omogočiti operacijo pisanja – podatki se bodo shranili na določeno lokacijo.
- Sprostitev krmiljenja Branje/Pisanje po izvedbi branja ali pisanja in onemogočanje delovanja pomnilnega integriranega vezja (IC-integrated circuit)

Pomnilnik: 32 x 4

Osnovne funkcije poenostavljenega diagrama pomnilnika, ki shrani 32 4-bitnih besed.



- \sim WE – write enable
- ME – memory enable
- OE – output enable
- A_i – naslovni vhodi
- I_i – podatkovni vhodi
- O_i – podatkovni izhodi

Pomnilne celice

Naslov				
00000	0	0	1	1
00001	0	1	1	0
00010	1	1	1	1

...

11101	0	0	0	1
11110	1	0	1	1
11111	1	1	0	1

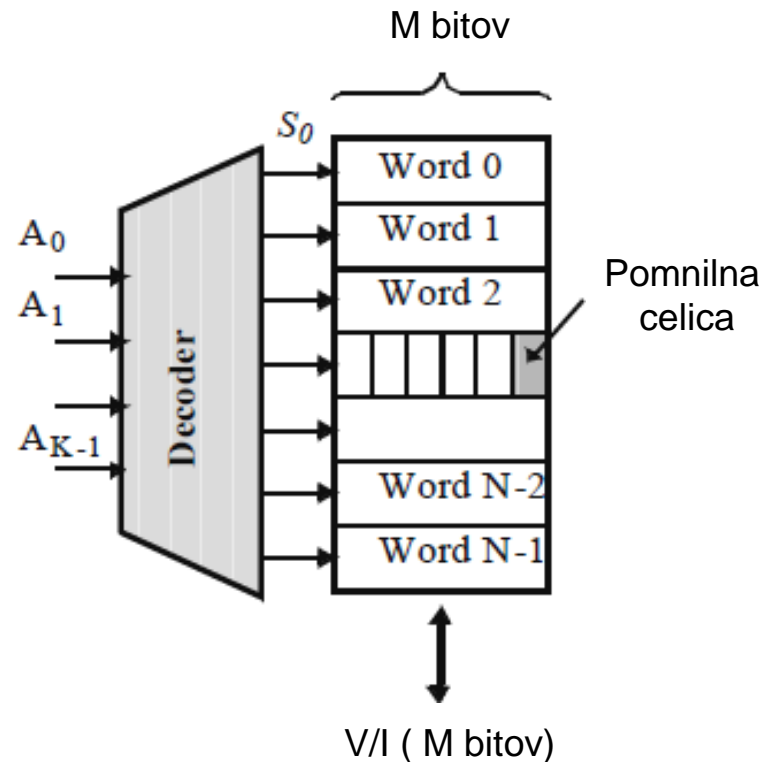
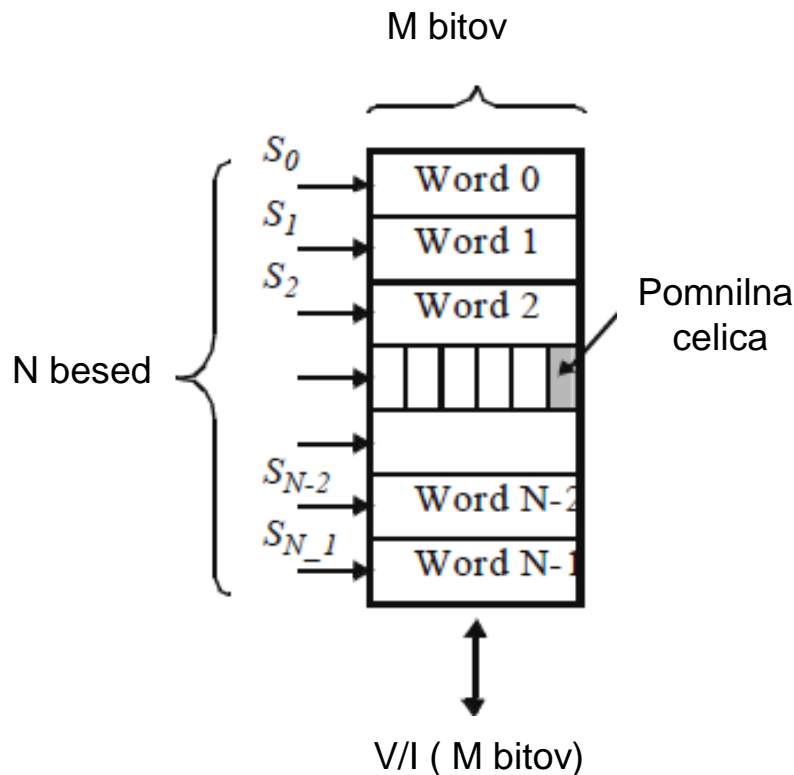
Oznaka za negacijo: \sim

Primer:

- Pomnilnik - velikost 2K x 8:
 - Število pomnilnih besed: $2 * 1024 = 2048$
 - Dolžina besede: 8-bitov ali 1 bajt
 - Število bitov: $2048 * 8 = 16384$
- Kateri pomnilnik shrani več bitov: 5M x 8 ali 1M x 16?
 - $5M \times 8 = 5 * 2^{20} * 2^3 = 5 * 2^{23}$ (shrani več bitov)
 - $1M \times 16 = 1 * 2^{20} * 2 * 2^3 = 2 * 2^{23}$
- Za branje lokacije 00010 pomnilnika 32 x 4 so določeni vhodi, izhodi, signali:
 - Naslov: 00010
 - Podatkovni vhodi: xxxx
 - ~WE: HIGH
 - ME: HIGH
 - OE: HIGH
 - Podatkovni izhodi: 1111

Arhitektura pomnilnika

- N besed -> N izbiralnih signalov
- Dekoder – določa število izbiralnih signalov ($K = \log_2 N$)
- K - bitni naslov



Primer:

Shraniti želimo 16 različnih 8-bitnih podatkov – kakšna je struktura pomnilnika?

- Število bitov naslova:
- Velikost pomnilnika-število bitov:

Rešitev:

Določitev naslova:

$N = 16$ besed

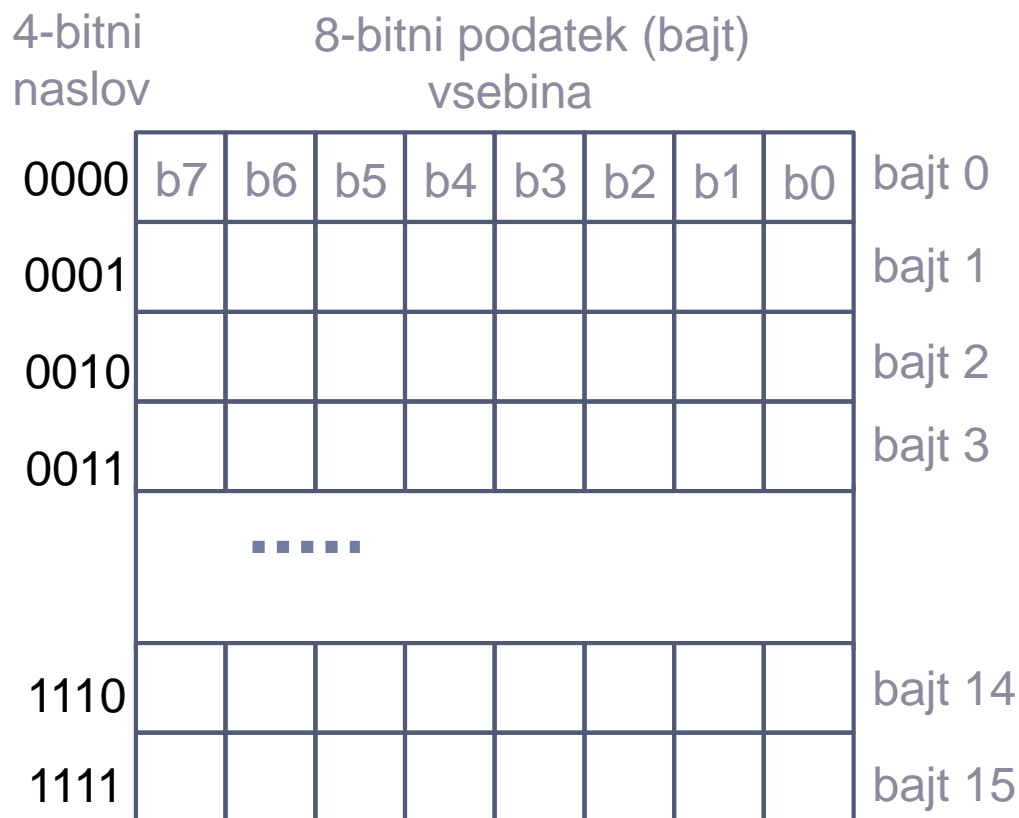
$K = \log_2 N \rightarrow 2^K = 16 = 2^4$

$K = 4$ (število bitov naslova)

Velikost pomnilnika:

16 bajtov (16B)

$16 \times 8 = 128$ bitov



Primer:

Zapis datuma v pomnilniku: 30.december 2016

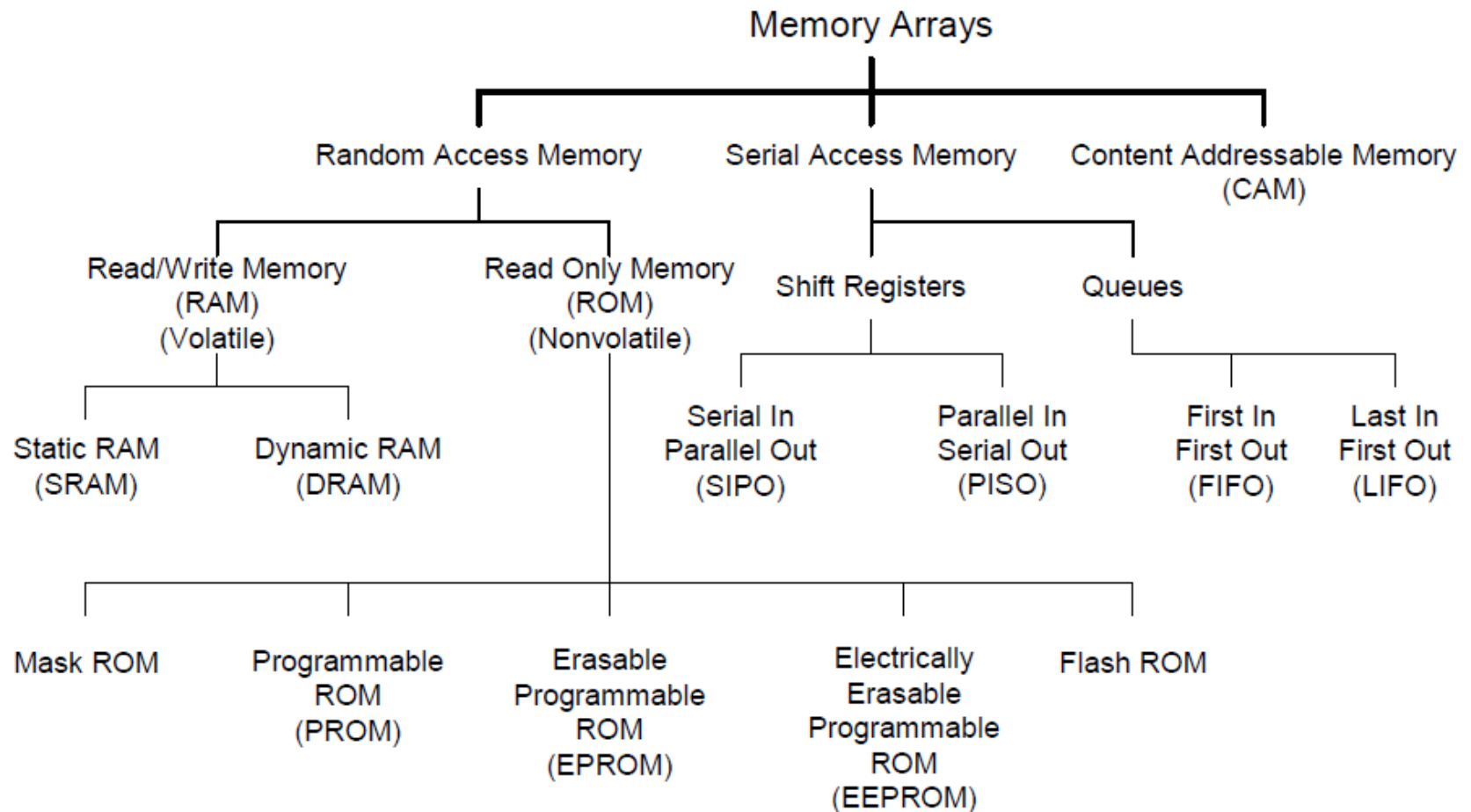
Vsebina pomnilnika: Vsaka beseda je UTF-8 (ASCII) zapis znakov

Vir: <http://www.utf8-chartable.de/>

Naslov										Podatek	
0	0	0	1	1	0	0	1	1	3	33	
1	0	0	1	1	0	0	0	0	0	30	
2	0	0	1	0	1	1	1	0	.	2e	
3	0	1	1	0	0	1	0	0	d	64	
4	0	1	1	0	0	1	0	1	e	65	
5	0	1	1	0	0	0	1	1	c	63	
6	0	1	1	0	0	1	0	1	e	65	
7	0	1	1	0	1	1	0	1	m	6d	

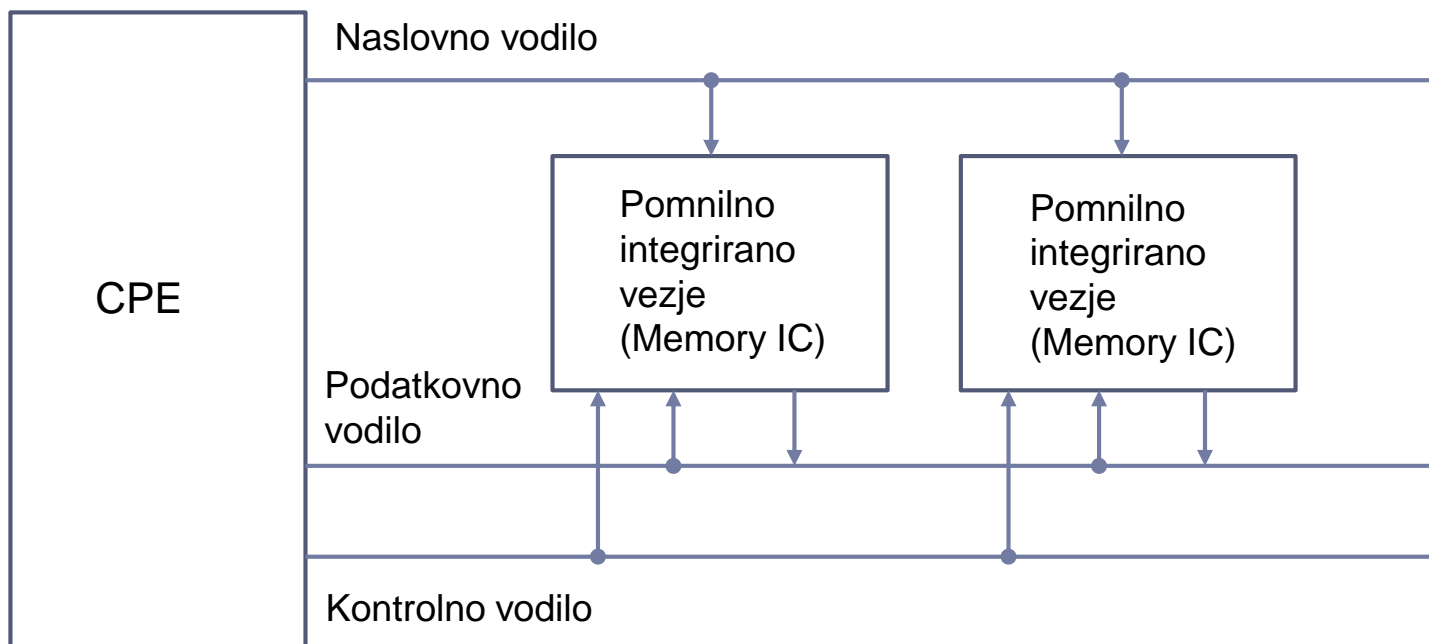
Naslov										Podatek	
8	0	1	1	0	0	0	1	0	b	62	
9	0	1	1	0	0	1	0	1	e	65	
10	0	1	1	1	0	0	1	0	r	72	
11	0	0	1	0	0	0	0	0		20	
12	0	0	1	1	0	0	1	0	2	32	
13	0	0	1	1	0	0	0	0	0	30	
14	0	0	1	1	0	0	0	1	l	31	
15	0	0	1	1	0	1	1	0	6	36	

Pomnilniki – pomnilniška polja



Povezava: CPE - Pomnilnik

- ▶ Glavni pomnilnik – polprevodniški pomnilnik (angl. semiconductor)
- ▶ Naslovno vodilo – enosmerno vodilo, kjer naslov iz CPE določa lokacijo.
- ▶ Podatkovno vodilo – dvosmerno vodilo za prenos podatkov v/iz CPE.
- ▶ Kontrolno vodilo – signala \sim RD (read data), \sim VWR (write data) iz CPE v pomnilnik. Izvajata se operaciji branja in pisanja.

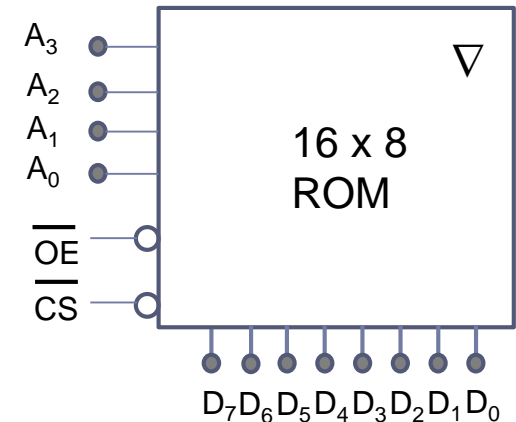


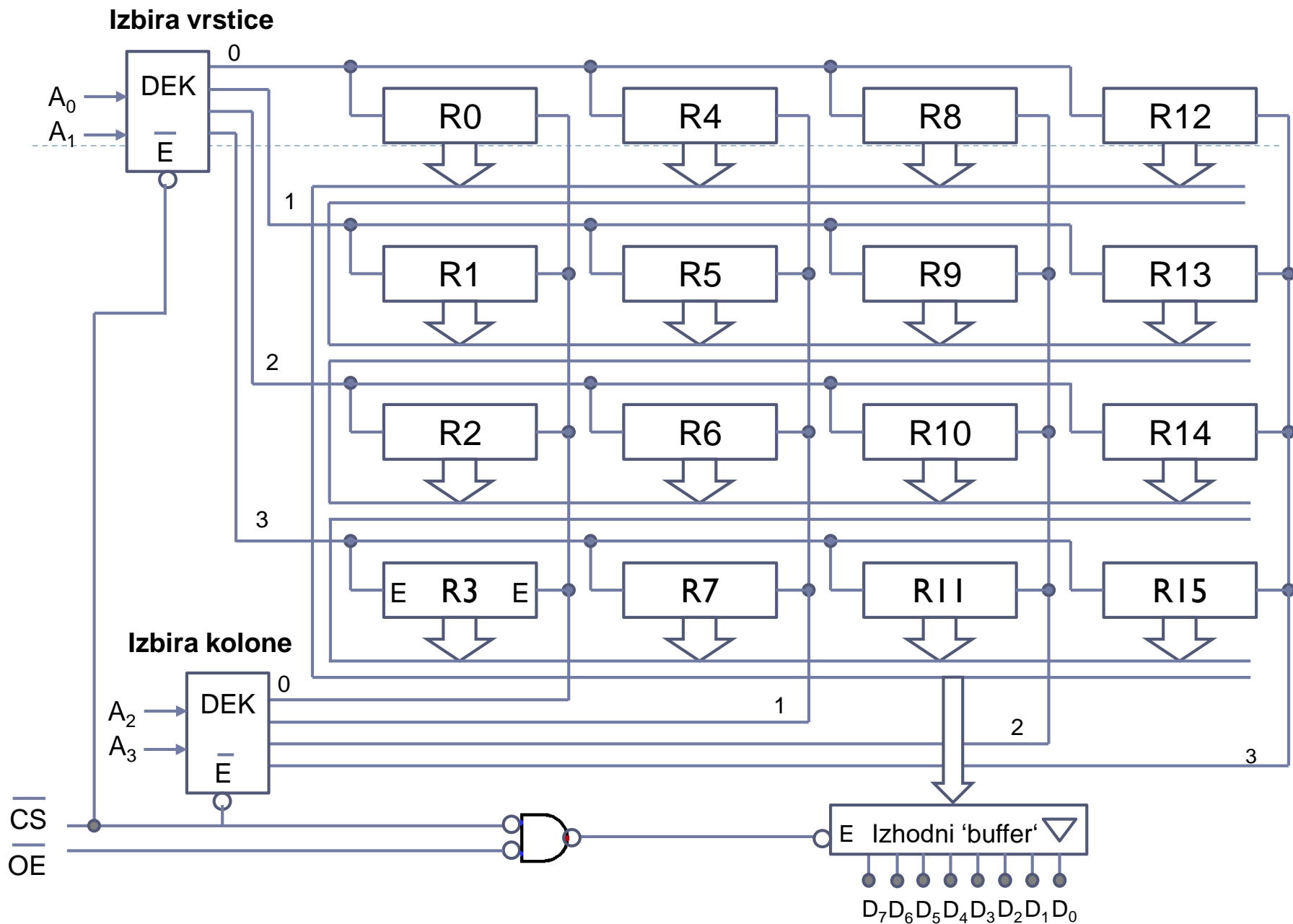
ROM (Read Only Memory)

- ROM ima samo operacijo branja.
- Enkrat (večkrat) ga je mogoče programirati pred uporabo.
- Obstojno, trajno shranjevanje podatkov, kjer se ob izklopu napetosti ohrani informacija (angl. Nonvolatile memory).
- Naključni dostop do podatkov ('random access') - Čas dostopa je enak za vse lokacije pomnilnika.
- Brisanje in ponovno programiranje:
 - PROM (Programmable ROM)
 - EPROM (Erasable Programmable ROM).
 - EEPROM (Electrically Erasable Programmable ROM)
- Uporaba:
 - shranjevanje preslikovalnih tabel (tabela trigonometričnih funkcij: sin, cos,, ...), pretvornik kod (BCD koda → 7-segmentni prikazovalnik)
 - Program za zagon naprave (Bootstrap memory)
 - Posebne, enostavne programske aplikacije

16 x 8 ROM

- Osnovne komponente:
 - Polje registrov (8-bitov)
 - Dekodirnik vrstic
 - Dekodirnik kolon
 - Izhodni 'tri-state buffer' (izhodni medpomnilnik)
- Polje registrov - shranjevanje podatkov, ki so programirani v ROM, vsebuje več pomnilnih celic, pozicija registra je določena z vrstico in kolono.
- Naslovni dekodirniki – z naslovi $A_3A_2A_1A_0$ določajo kateri register v polju bo izbran, da se bodo podatki pojavili na vodilu.
 - Naslova A_1A_0 določata vrstico v polju registrov
 - Naslova A_3A_2 določata kolono v polju registrovEn register bo izbran v polju in omogočen – podatki se pojavijo na izhodu.
- Izhodni 'buffer' – Register, ki je izbran z naslovnimi linijami posreduje podatke na podatkovno vodilo. Če imata vhoda $\sim CS$ in $\sim OE$ vrednost 0, se bodo podatki posredovali na zunanje podatkovne izhode.

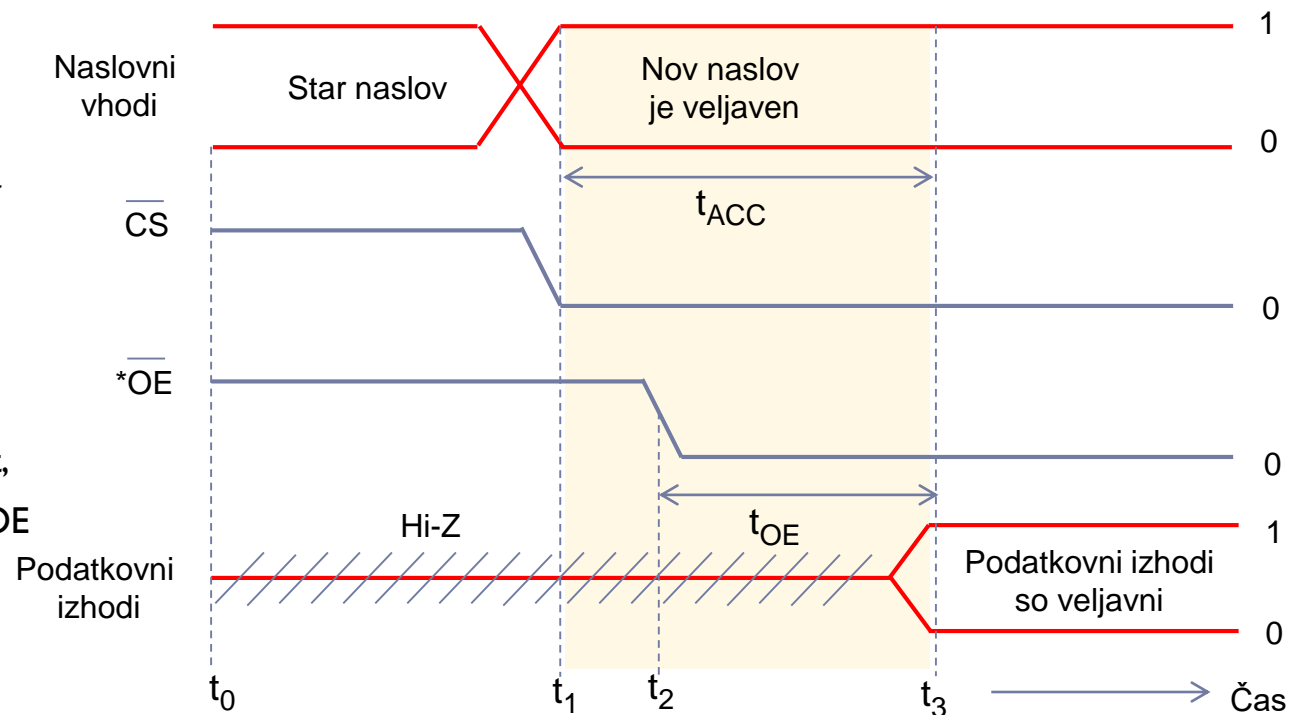




ROM - časovni diagram

- Spreminjanje naslova poteka pred časom t_1 - naslov postane veljaven.
- Signal $\sim\text{CS}$ bo aktiviran, notranje vezje v ROMu dekodira nov naslov za izbiro registra, ki pošlje podatke na izhodni 'buffer'.
- V času t_2 sta aktivirana tako $\sim\text{CS}$ in $\sim\text{OE}$, da omogočita izhodni 'buffer'.
- V času t_3 se izhodi spremenijo iz stanja HI-Z (tri-stanjski izhodi) v veljavne podatke.
- t_{ACC} - čas dostopa
- t_{OE} - čas za omogočanje izhoda

* t_{OE} - čas se meri od takrat, ko sta bila signala $\sim\text{CS}$ in $\sim\text{OE}$ potrjena

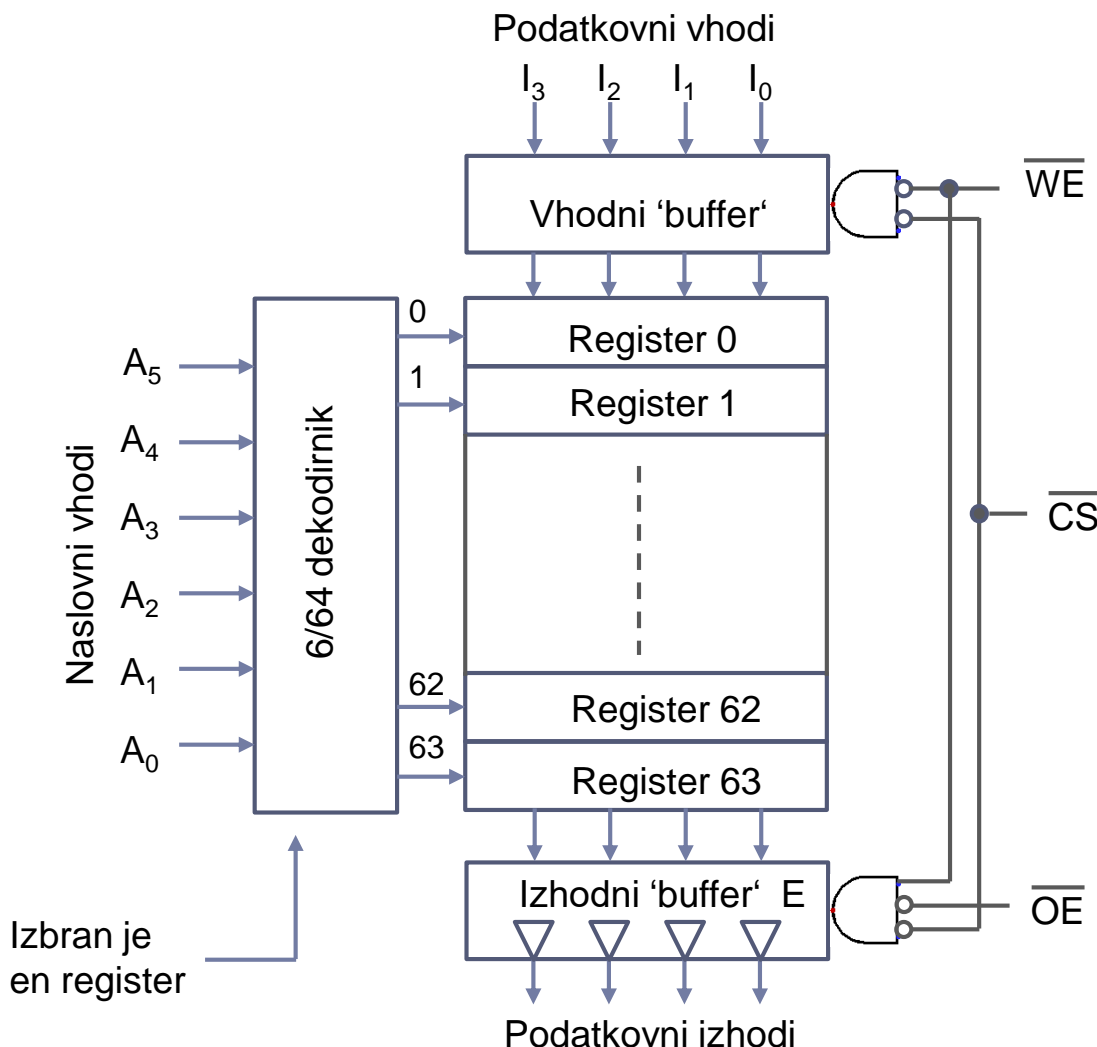


RAM (Random Access Memory)

- ▶ Pomnilnik z naključnim dostopom (random access memory)
 - ▶ Čas dostopa enak za vse lokacije
 - ▶ Hiter dostop (visoka cena)
- ▶ Notranji pomnilnik, ki je povezan s CPE in začasno shranjuje:
 - ▶ Podatke
 - ▶ Programe
- ▶ Bralno/pisalni pomnilnik (read/write memory), ki ne upočasnjuje izvajanje operacij
- ▶ Neobstojni pomnilnik (angl. Volatile memory) - izguba vsebine ob izklopu:
 - ▶ Statični RAM
 - ▶ Dinamični RAM

RAM 64 x 4 - Arhitektura

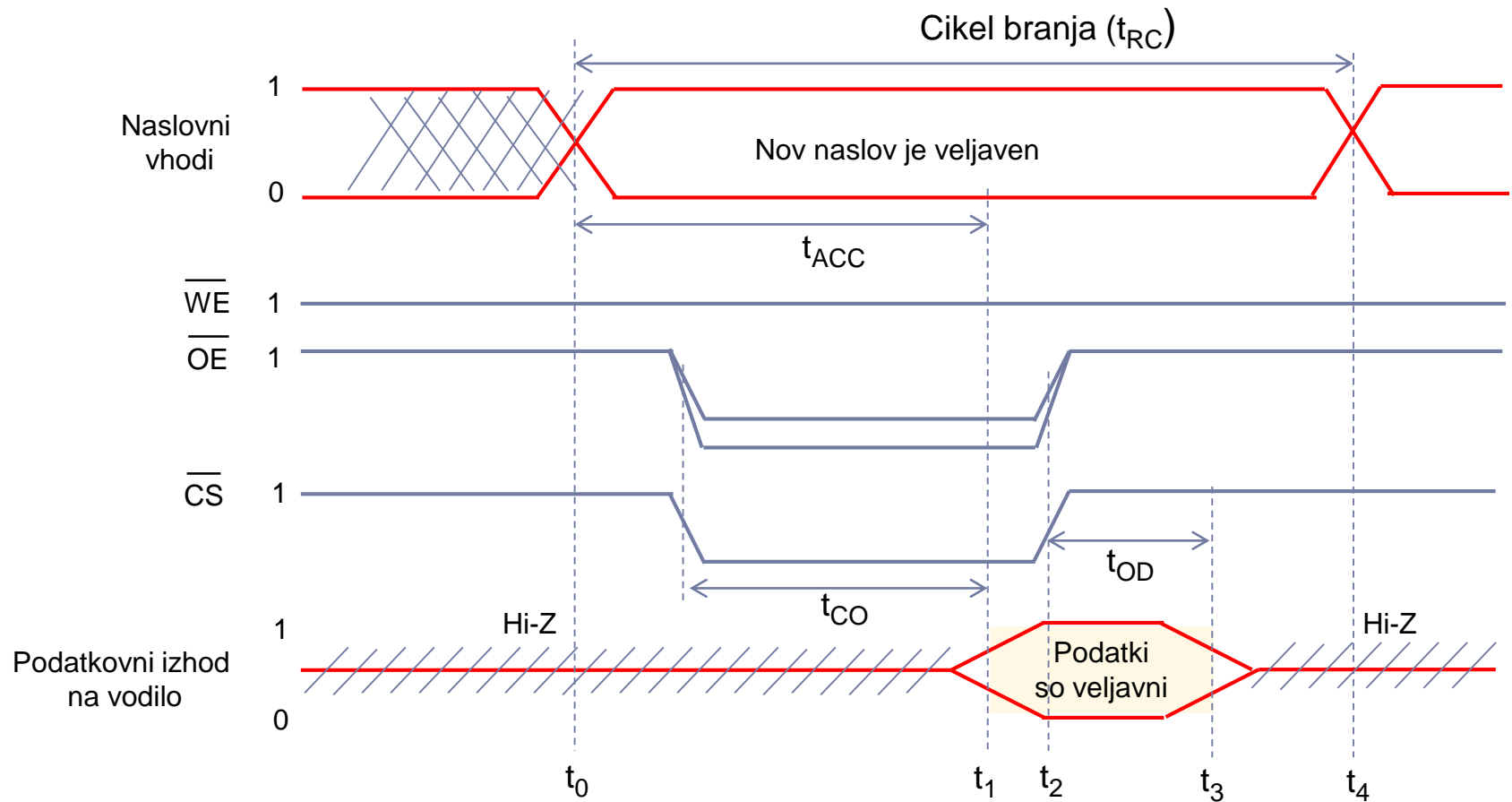
- ▶ RAM sestavljajo registri, ki shranjujejo besedo, za katero obstaja unikaten naslov
- ▶ $\sim\text{CS}$ (Chip Select): 0 – omogoči celoten čip za operacijo branja ali pisanja (READ ali WRITE)
- ▶ $\sim\text{OE}$ (Output Enable): 0 – omogoči izhode za branje (READ)
- ▶ $\sim\text{WE}$ (Write Enable) – omogočena je operacija:
 - ▶ 0 – write
 - ▶ 1 - read



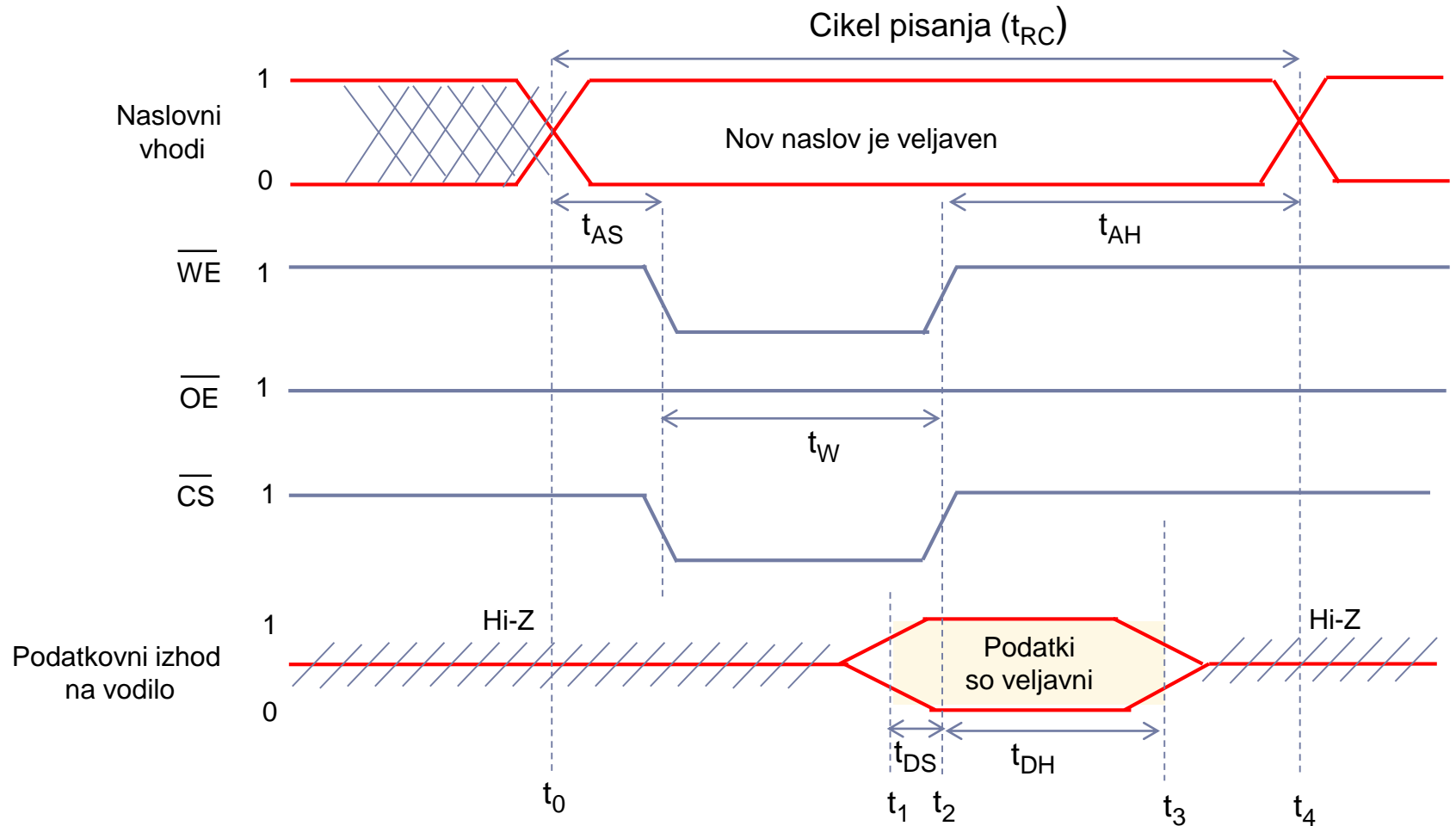
SRAM (Statični RAM)

- ▶ Na prejšnji prosojnici opisane operacije se nanašajo na SRAM.
- ▶ Shranjena vsebina ostane zapisana v pomnilniku vse dokler je čip pod napajanjem.
- ▶ Izvedba, tehnologije: bipolarna, MOS, BiCMOS; večinoma poznamo čipe SRAM, ki so na voljo v CMOS izvedbi.
- ▶ SRAM se uporablja kot predpomnilnik, ki je majhen in zelo hiter.
- ▶ Branje:
 - ▶ t_{ACC} – čas dostopa (access time)
 - ▶ t_{CO} – čas, ki je potreben za veljavnost podatkov, če sta $\sim OE$ in $\sim CS$ veljavna.
 - ▶ t_{OD} – čas, ko so podatki še veljavni, ko se $\sim OE$ in $\sim CS$ spremenita v HIGH
- ▶ Pisanje:
 - ▶ t_{AS} – čas, ki je potreben, da se $\sim WE$, $\sim CS$ spremenita v LOW (address setup time)
 - ▶ t_W – čas, ki je potreben za pisanje (write time interval)
 - ▶ t_{AH} – čas, ko mora biti naslov še veljaven (address hold time)

SRAM - branje



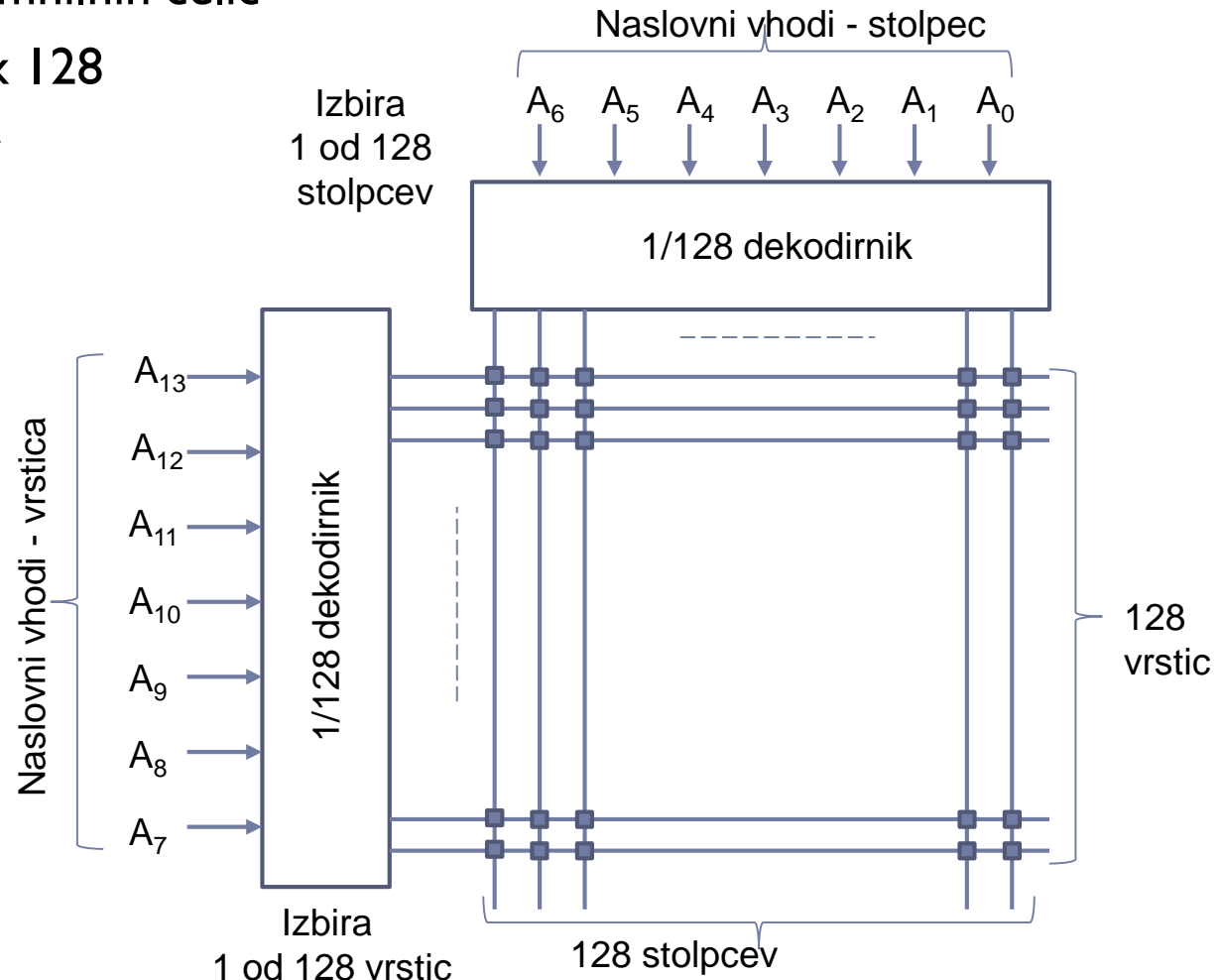
SRAM - pisanje



Veljavnost podatkov: t_{DS} (data setup time), t_{DH} (data hold time)

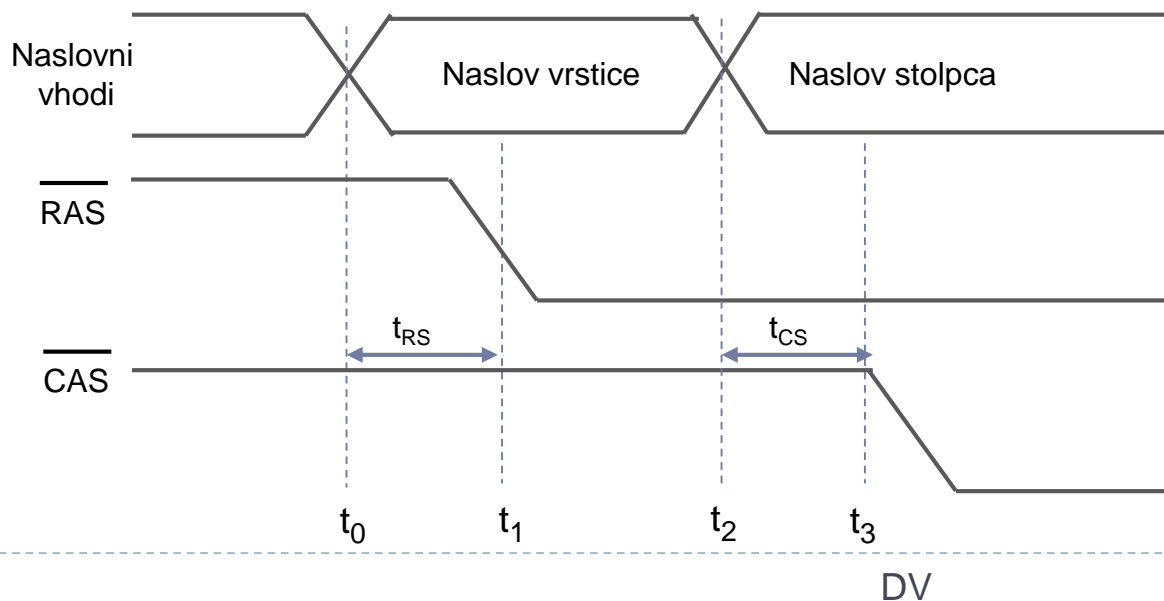
Dinamični RAM (DRAM)

- ▶ Arhitektura-polje pomnilnih celic
- ▶ 16,384 celic $\rightarrow 128 \times 128$
- ▶ 14 naslovnih vhodov:
 - ▶ Stolpec: $A_0 - A_6$
 - ▶ Vrstica: $A_7 - A_{13}$
- ▶ Primer:
16K x 1 DRAM



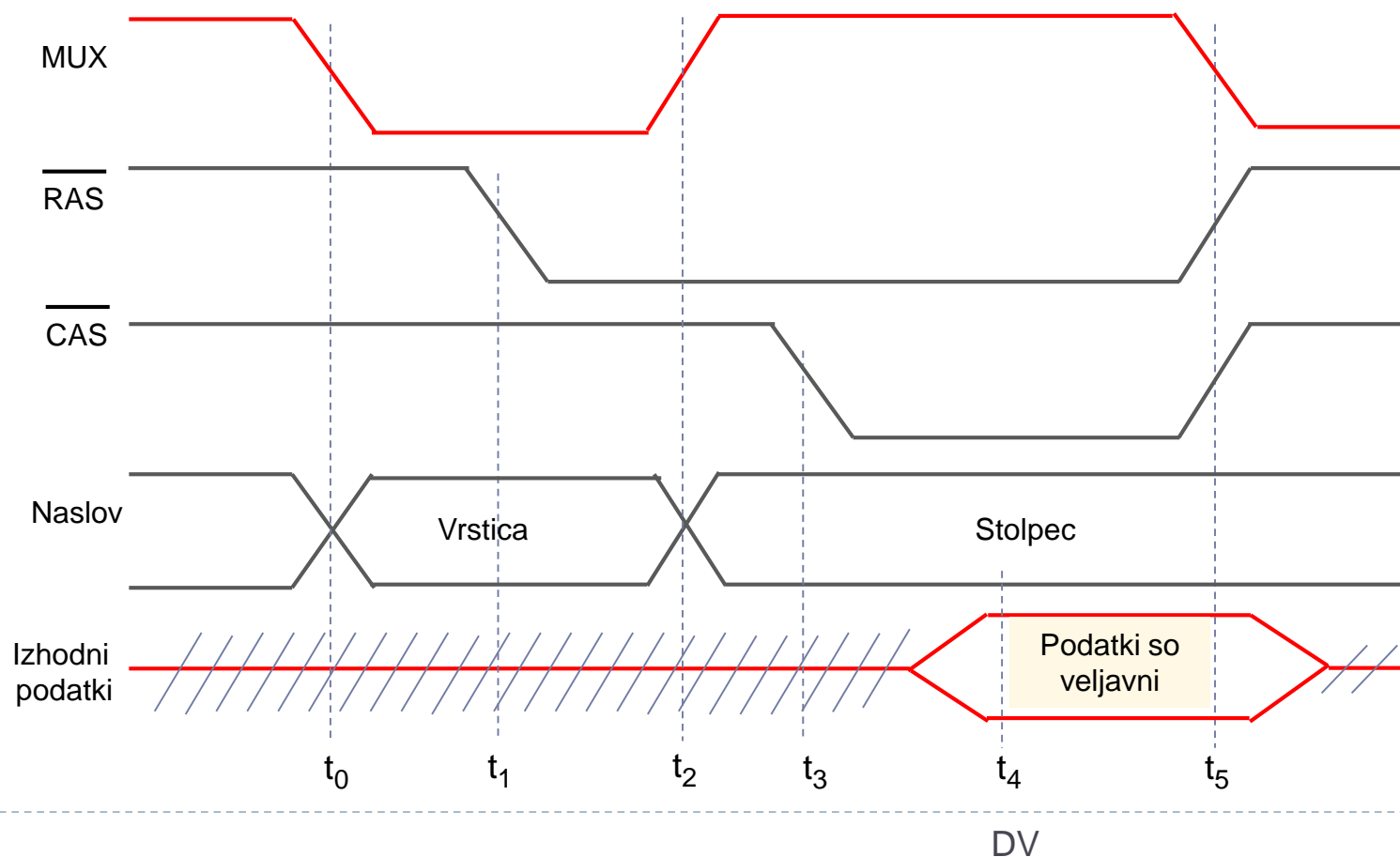
DRAM – izbiranje naslova

- ▶ Naslavljanje je izvedeno v dveh delih - naslovni vhodi so povezani na naslovni register vrstice in naslovni register stolpca.
- ▶ Shranjevanje naslova vrstice in vrstic stolpca krmilita:
- ▶ \sim RAS (row address strobe) – shrani naslovne vhode v register vrstice
- ▶ \sim CAS (column address strobe) - shrani naslovne vhode v register stolpca
- ▶ t_{RS} – čas vzpostavitve naslova vrstice
- ▶ t_{CS} – čas vzpostavitve naslova stolpca



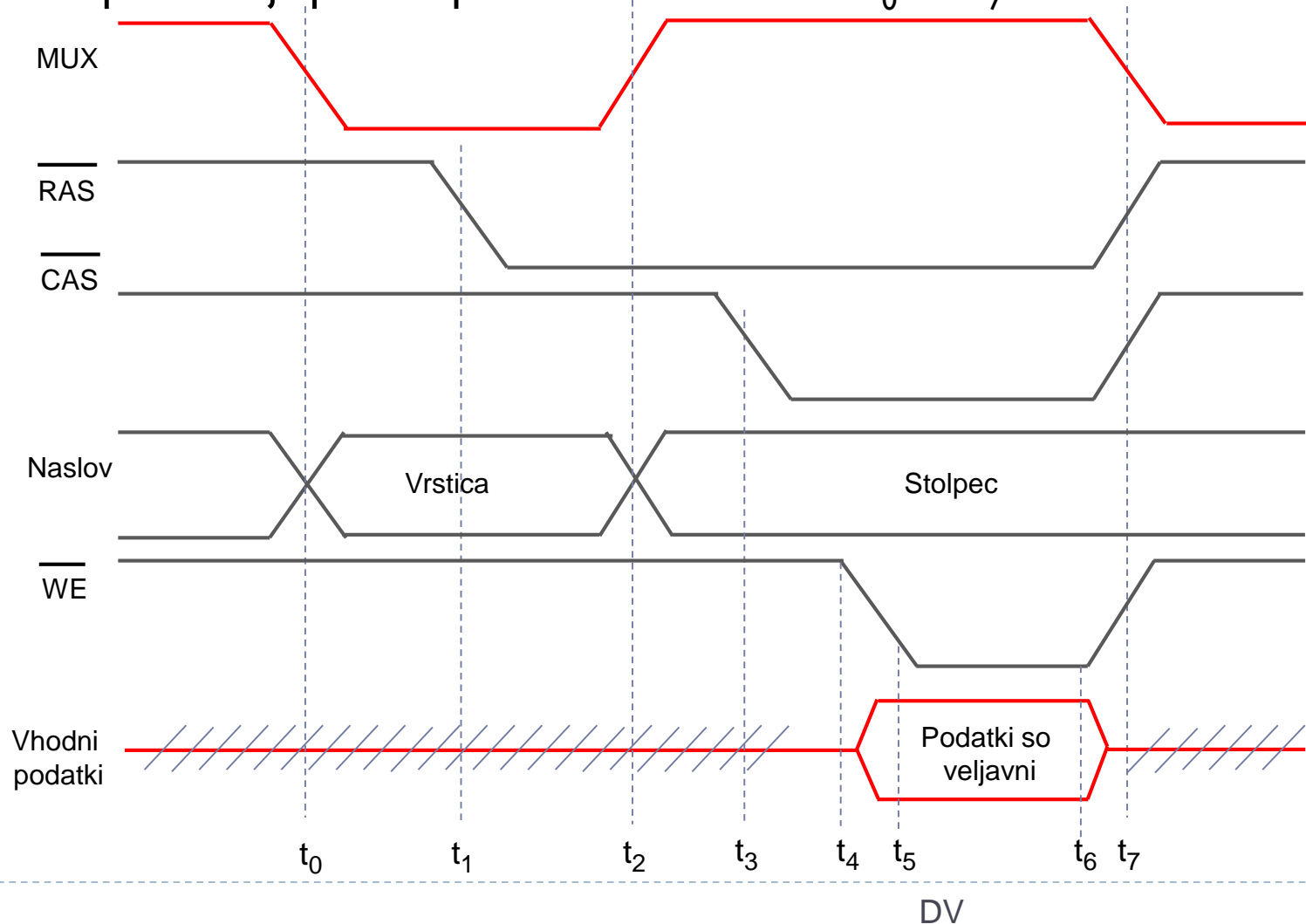
DRAM - branje

- ▶ Vhodni signal \sim WE je pri branju HIGH.
- ▶ Postopek branja poteka po korakih v časih od t_0 do t_5 .



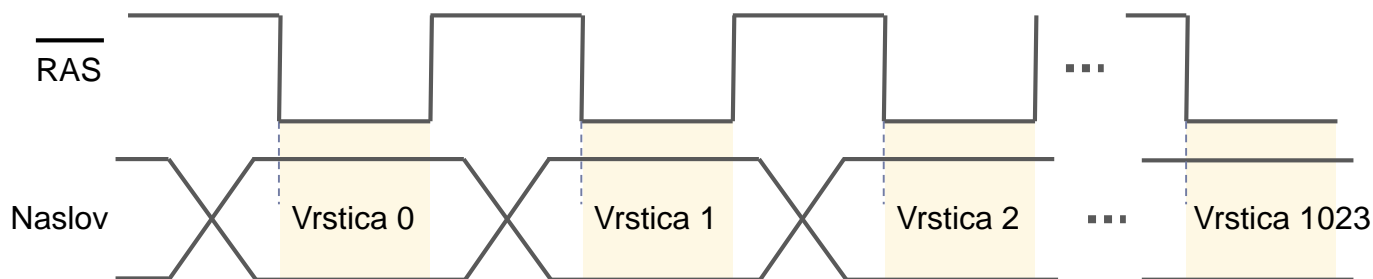
DRAM - pisanje

- Postopek branja poteka po korakih v časih od t_0 do t_7 .



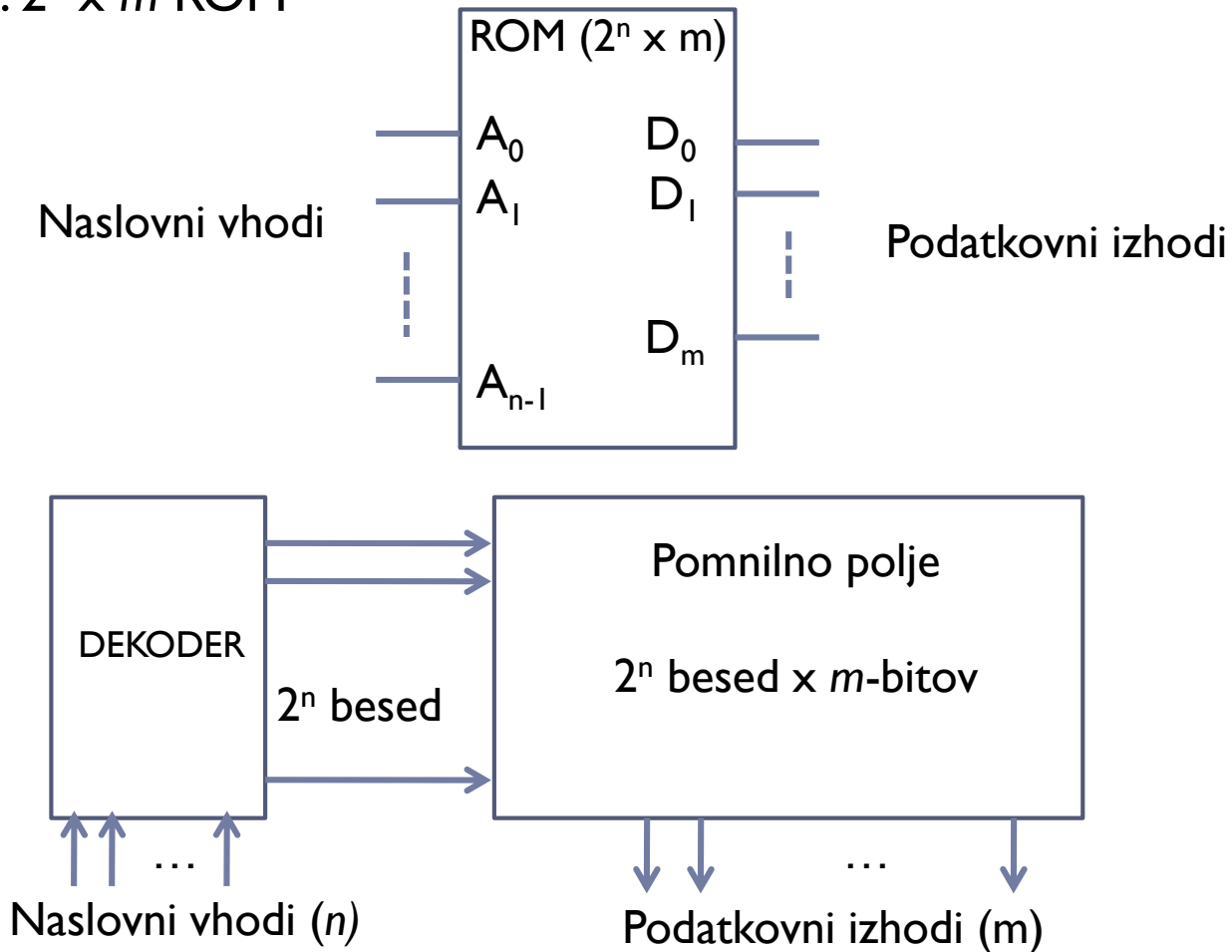
DRAM - osveževanje

- ▶ Celica DRAM se osveži takoj potem, ko je bila izvedena operacija branja.
- ▶ Vsaka pomnilna celica mora biti osvežena periodično (vsakih 2 do 8 ms).
- ▶ Proizvajalci so razvili integrirana vezja DRAM tako, da se v izvedbe operacije branja na celici izvede osvežitev na celotni vrstici.
- ▶ Razširitveno osveževanje (angl. burst refresh mode) – pomnilna operacija je prekinjena in vsaka vrstica DRAMa je osvežena dokler niso osvežene vse vrstice.
- ▶ Univerzalna metoda osveževanja: '∼RAS-only refresh' – izvede se z naslovom vrstice z ∼RAS takrat, ko sta ∼CAS in ∼WE HIGH.



Naloga 1: Opis

- ROM: naslovni vhodi (n), podatkovni izhodi (m)
- Oznaka: $2^n \times m$ ROM



Naloga 2: Uporaba pomnilnika ROM

- Zapis logičnih funkcij v pomnilnik ROM - logične funkcije z n spremenljivkami se iz pravilnostne tabele shrani v bralni pomnilnik velikosti $2^n \times m$.

Primer:

$n=3$ - število spremenljivk

(število naslovnih linij)

$2^n = 8$ – število vhodnih kombinacij

(število pomnilnih besed)

$m=4$ – število funkcij (f_a, f_b, f_c, f_d)

(število izhodov)

x	y	z	f_a	f_b	f_c	f_d
0	0	0	0	0	1	1
0	0	1	0	1	1	0
0	1	0	1	1	1	1
0	1	1	0	1	0	1
1	0	0	1	1	1	0
1	0	1	0	0	0	1
1	1	0	1	0	1	1
1	1	1	1	1	0	1



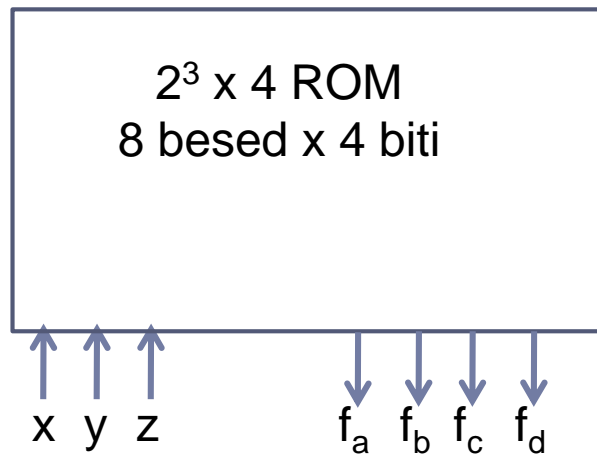
Naslov



Podatki

Realizacija funkcij v ROMu - izvede se
preslikava funkcij v pomnilno polje

- 8 vhodnih kombinacij funkcije
pomeni 8 besed v ROM-u,
- 4 funkcije so 4 izhodi ROM-a



3-bitni
naslov

4-bitni podatek

000	0	0	1	1
001	0	1	1	0
010	1	1	1	1
011	0	1	0	1
110	1	1	1	0
111	0	0	0	1
110	1	0	1	1
111	1	1	0	1

Naloga 3

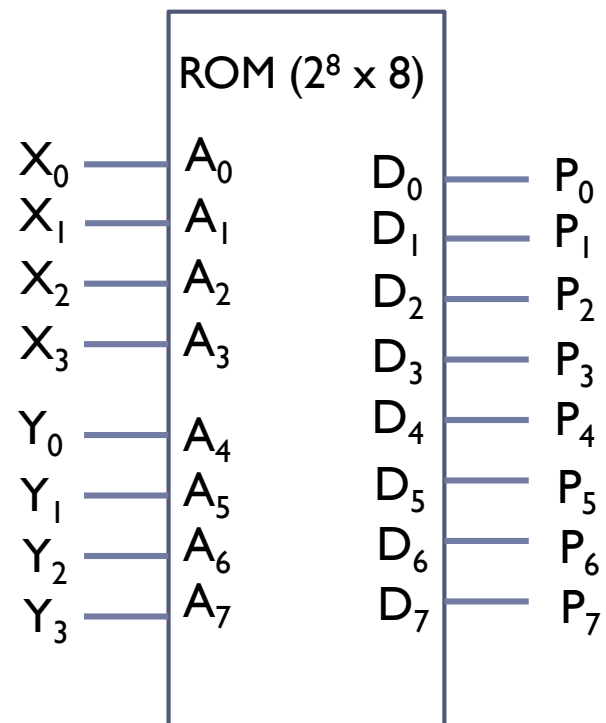
➤ Realizacija 4-bitnega množilnika $P=X*Y$.

➤ Pomnilnik ROM:

$$2^8 \times 8 = 256B \text{ (256 bajtov)}$$

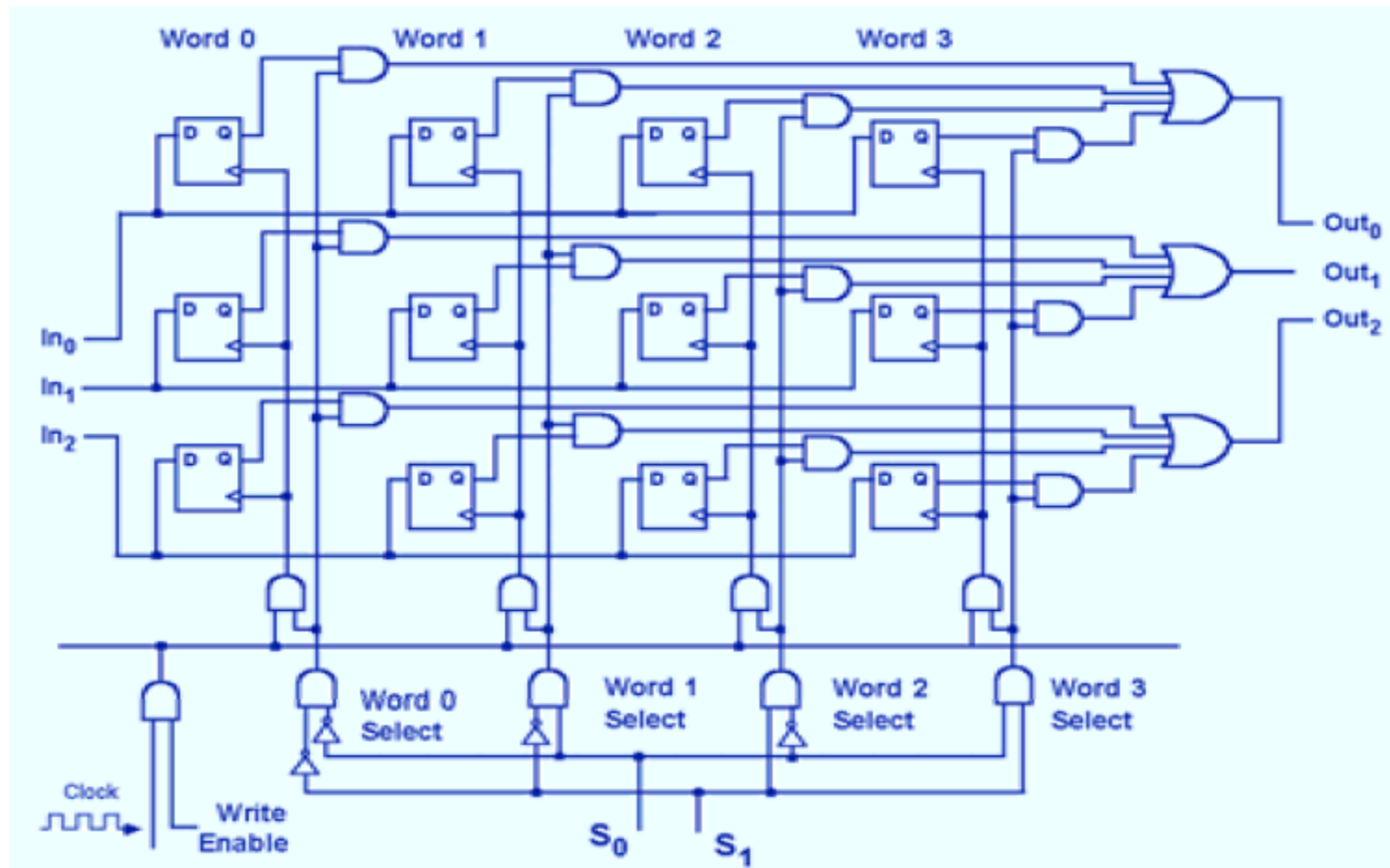
➤ Vsebina ROMa (hex zapis)

00:	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00
10:	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
20:	00	02	04	06	08	0A	0C	0E	10	12	14	16	18	1A	1C	1E
30:	00	03	06	09	0C	0F	12	15	18	1B	1E	21	24	27	2A	2D
40:	00	04	08	0C	10	14	18	1C	20	24	28	2C	30	34	38	3C
50:	00	05	0A	0F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
60:	00	06	0C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
70:	00	07	0E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
80:	00	08	10	18	20	28	30	38	40	48	50	58	60	68	70	78
90:	00	09	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A0:	00	0A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B0:	00	0B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C0:	00	0C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D0:	00	0D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E0:	00	0E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F0:	00	0F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

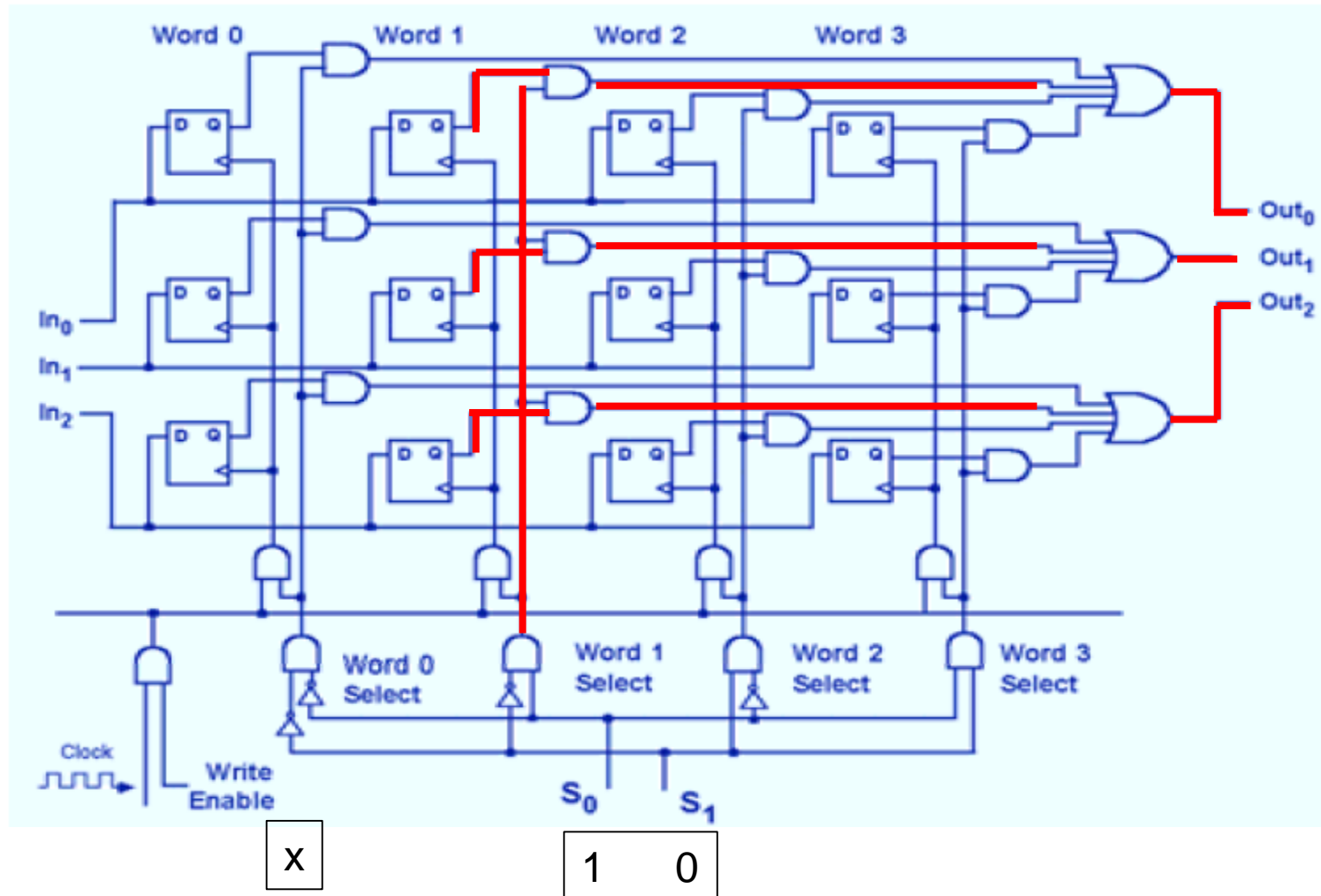


Naloga 4: Pomnilnik (4x3)

- ▶ Pomnilnik vsebuje 4 besede, vsaka beseda ima 3 bite
- ▶ Za predstavitev 4 besed potrebujemo 2-bita za naslov (S_1, S_0) – dekodeer za naslavljanje izvede izbiro posamezne besede oz. vrstice v pomnilniku.
- ▶ Za shranjevanje podatkov uporabimo eno D pomnilno celico za vsak bit, kar predstavlja 12 celic v pomnilniku.
- ▶ Potrebujemo tudi spremenljivko - Write Enable (WE) za izbiro branja ali pisanja, ki je povezana z urinim signalom (Clock):
 - ▶ $WE = 1$ – pisanje v pomnilnik
 - ▶ $WE = 0$ – branje iz pomnilnika
- ▶ Potrebujemo še drugo kombinacijsko vezje, logična vrata
- ▶ Podatkovni vhodi: In_2, In_1, In_0 ,
- ▶ Podatkovni izhodi: Out_2, Out_1, Out_0 ,



Branje besede z naslovom 1 (Word 1)



DV

