

Digitalna vezja, BVS-RI

Mira TREBAR

P1 1 Pomnilnik in programabilni gradniki

Vsebina

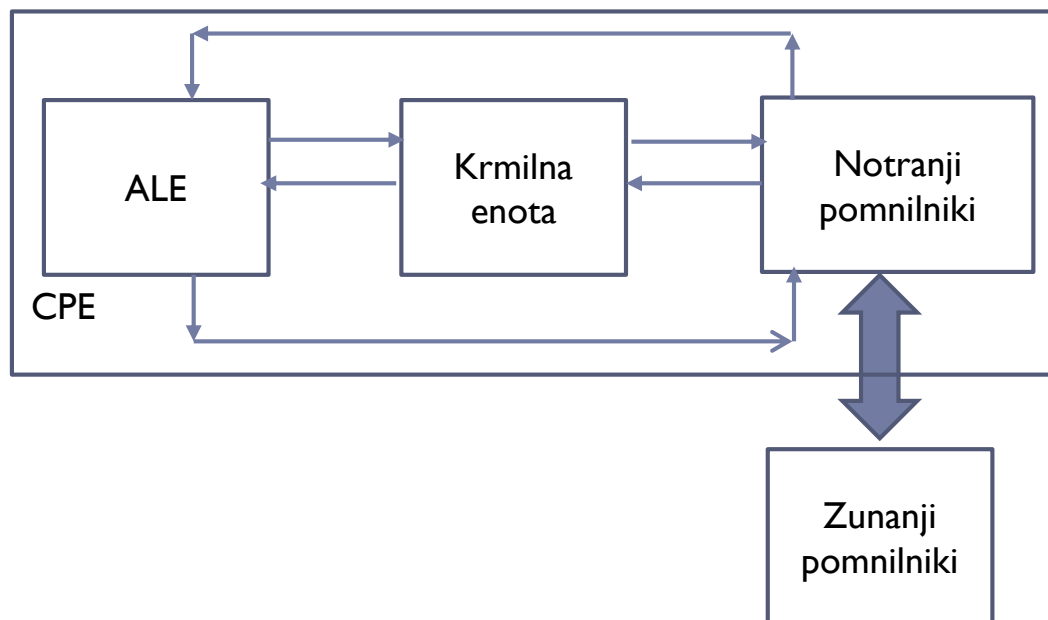
1. Pomnilnik
2. Povezava CPE – Pomnilnik
3. ROM (Read Only Memory)
4. RAM (Random Access Memory)
 - SRAM (Static RAM),
 - DRAM (Dynamic RAM)
5. Programabilni gradniki
6. Programabilno vezje (P)ROM
7. Naloge

□ Literatura:

- Widmer, S.W, Moss, G.L., Tocci, R.J., Digital Systems, Principles and Applications, Pearson, 2007 (PI2 Memory Devices, PI3 Programmable Logic device architectures)

1 Pomnilnik

- ❑ Digitalni sistemi - velike količine podatkov
- ❑ Shranjevanje podatkov in ukazov.
- ❑ Notranji pomnilniki
 - Glavni pomnilnik
 - Predpomnilnik
- ❑ Zunanji pomnilniki
 - Disk (HDD, SSD)
 - Bliskovni pomnilnik

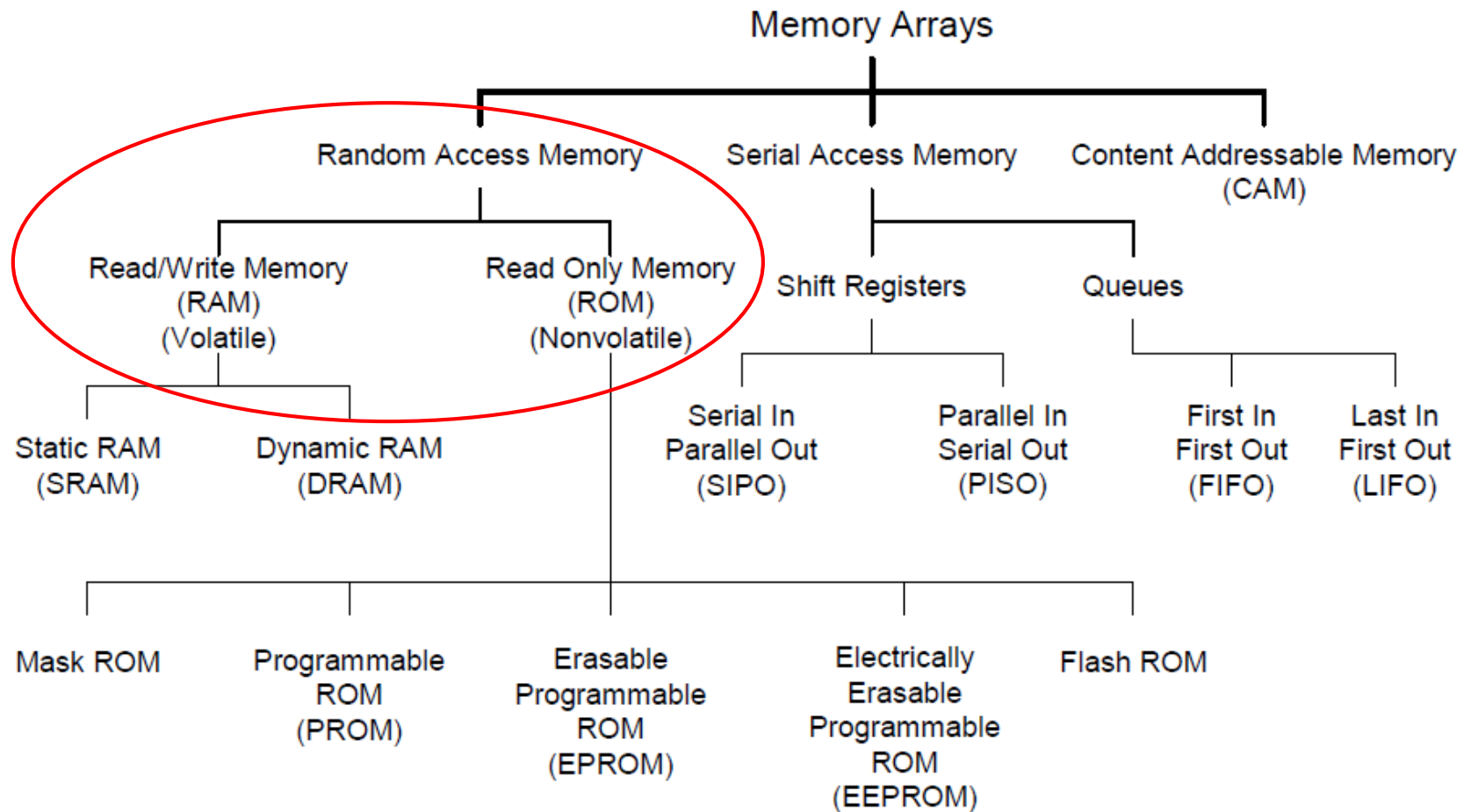


Terminologija

- ❑ Pomnilna **celica** (ang. Memory Cell) – vezje, ki shrani en bit informacije (0 ali 1)
- ❑ Pomnilna **beseda** (ang. Memory Word) – skupina bitov (celic), ki določa ukaz ali podatek
- ❑ **Bajt** (ang. Byte) – izraz, ki določa skupino osmih bitov
- ❑ **Kapaciteta** (ang. Capacity ali Density) – koliko bitov je lahko shranjenih v določeni pomnilni napravi ali v celotnem pomnilnem sistemu. Primer zapisa in kapaciteta:
 - ❑ 4096 20-bitnih besed – kapaciteta je 81920 bitov.
 - ❑ zapis je podan tudi kot 4096 x 20 (število besed x število bitov na besedo)
 - ❑ ali kot 4K x 20
- ❑ **Naslov** (ang. Address) – število, ki določa lokacijo besede v pomnilniku
- ❑ Operacija **branja** (ang. Read Operation) – beseda se iz pomnilnika prenese v drugo napravo
- ❑ Operacija **pisanja** (ang. Write operation) – nova beseda se shrani v pomnilnik
- ❑ **Čas dostopa** - t_{ACC} (ang. Access Time) – čas, ki je potreben za izvedbo operacije branja

-
- ❑ **Neobstojni** pomnilnik (ang. Volatile Memory) – potrebuje napajanje za shranjevanje podatkov
 - ❑ Pomnilnik **z naključnim dostopom** (ang. Random Access Memory) – vsaka beseda ima svoj dvojiški naslov
 - ❑ Pomnilnik **z zaporednim dostopom** (ang. Sequential Access Memory) – primer je magnetni trak za arhiviranje
 - ❑ Pomnilnik **z operacijo Branje/Pisanje** (ang. Read/Write Memory) – pomnilnik iz katerega beremo ali v njega pišemo
 - ❑ **Bralni** pomnilnik (ang. Read Only Memory) – pomnilnik iz katerega samo beremo
 - ❑ Naprave **s statičnim** pomnilnikom (ang. Static Memory Devices) – permanentno shranjevanje podatkov
 - ❑ Naprave **z dinamičnim** pomnilnikom (ang. Dynamic Memory Devices) – osveževanje pomnilnika
 - ❑ **Glavni** pomnilnik (ang. Main Memory) – shranjuje ukaze in podatke
 - ❑ **Predpomnilnik** (ang. Cache Memory) – hitri pomnilnik za shranjevanje podatkov
 - ❑ **Zunanji** pomnilnik (ang. Auxiliary Memory) – shranjuje velike količine podatkov

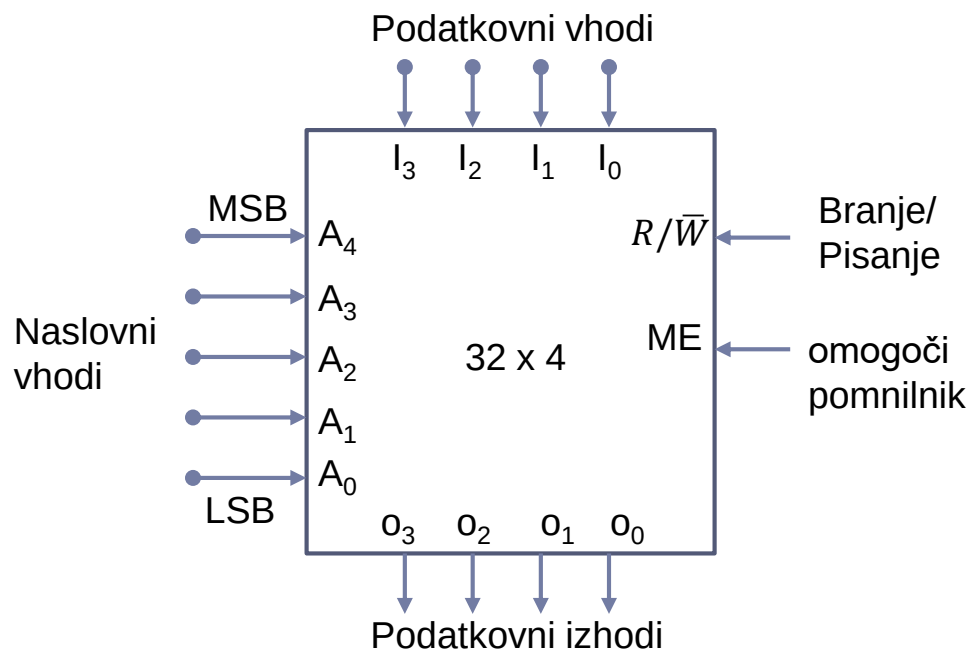
Pomnilniška polja



-
- ❑ Vsak pomnilniški sistem ima enaka osnovna načela delovanja, ki zahtevajo več vrst vhodov in izhodov za izvajanje naslednjih funkcij:
 - **Izbira naslova** v pomnilniku, ki je povezan z operacijo branja ali pisanja.
 - **Izbira operacije** branja ali pisanja, ki naj se izvede.
 - **Posredovanje vhodnih podatkov**, ki se med operacijo zapisovanja shranijo v pomnilnik.
 - Med operacijo branja **zadrži podatke na izhodu pomnilnika**.
 - **Omogoči (ali onemogoči)** pomnilnik, tako da se bo (ali ne bo) odzval na naslovne vhode in operacijo za branje ali pisanje.

 - ❑ Predstavitev osnovnih funkcij si bomo pogledali na poenostavljenem diagramu pomnilnika **32 x 4** (pomnilnik shrani 32 4-bitnih besed).

Pomnilnik: 32 x 4: Funkcije poenostavljenega diagrama pomnilnika

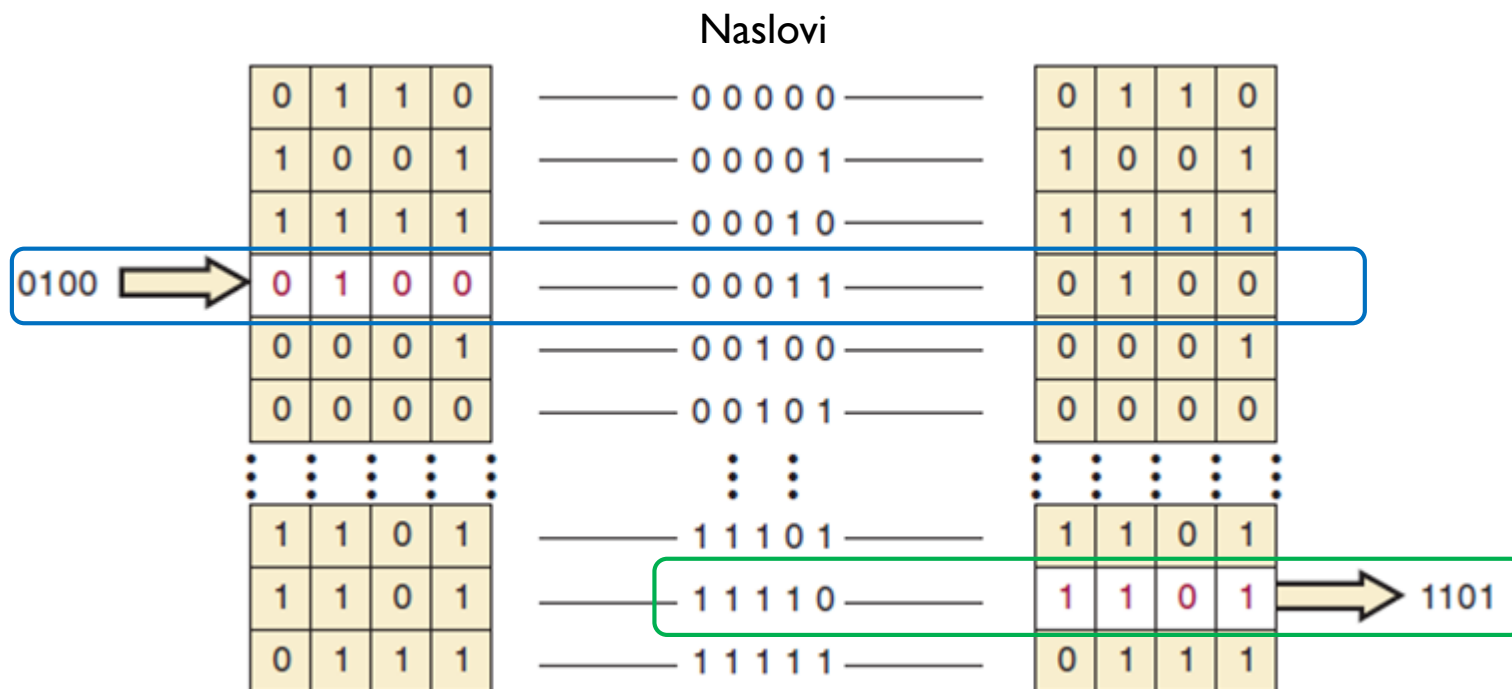


- $R/\overline{W}$ – Read/Write
- ME – Memory enable
- A_i – Address inputs
- I_i – Data inputs
- O_i – Data outputs

[illegible]

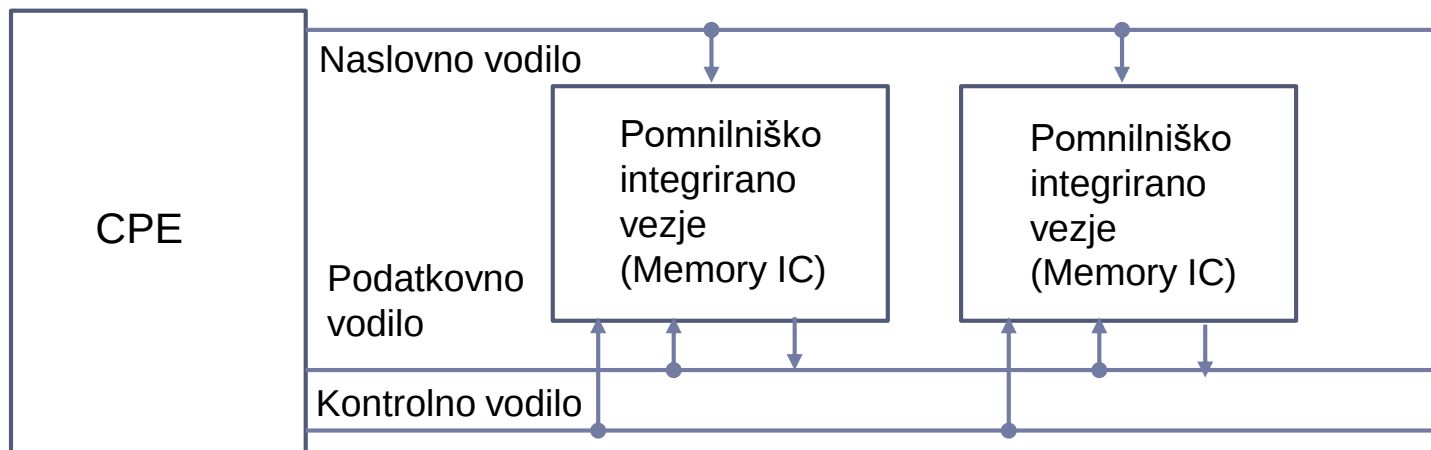
Operacija branja in pisanja:

- a) Pisanje besede 0100 v pomnilnik na naslov 00011
- b) Branje besede 1101 iz pomnilnika na naslovu 11110



2. 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 iz CPE v pomnilnik in iz pomnilnika v CPE.
 - Kontrolno vodilo – signal $R/\bar{W}$ (read/write data) iz CPE v pomnilnik – izvede se operacija branja ali pisanja.
- ❑ Postopek **pisanja v pomnilnik**: CPE (naslov, podatki, $\bar{W}$), Memory IC (dekodira naslov, podatki se shranijo na izbrani naslov).
- ❑ Postopek **branja iz pomnilnika**: CPE (naslov, R), Memory IC (dekodira naslov, podatki izbranega naslova se pošljejo na podatkovno vodilo).



3. ROM (Read Only Memory)

- ❑ ROM ima samo operacijo **branja**.
- ❑ Enkrat (večkrat) ga je mogoče **programirati** pred uporabo.
- ❑ Obstoje, **trajno shranjevanje** podatkov, kjer se ob izklopu napetosti ohrani informacija (ang. non-volatile memory).
- ❑ **Naključni dostop** do podatkov (ang. '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** (trigonometrične funkcije: 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 – 16 (beseda ima 8-bitov)
- Dekodirnik vrstic
- Dekodirnik kolon
- Izhodni 'tri-state buffer' (izhodni medpomnilnik)

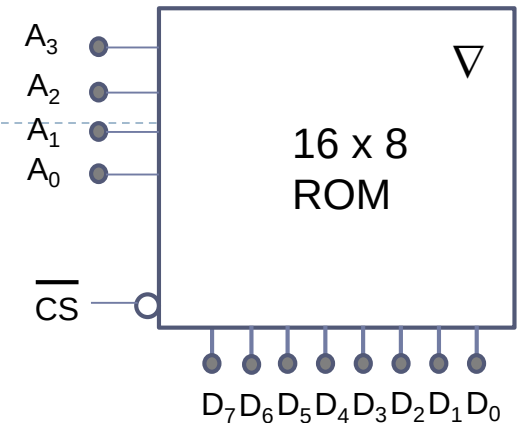
□ Polje registrov - shranjevanje podatkov, ki so programirani v ROM, vsebuje več pomnilnih celic, pozicija registra je določena z vrstico in kolono.

□ 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 registrov

En 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 ima vhod $\overline{CS}$ (**Chip Select**) vrednost 0, se bodo podatki posredovali na zunanje podatkovne izhode.

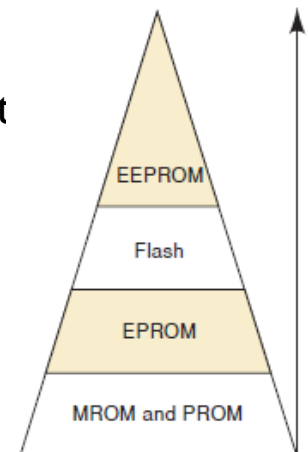




- 4 x 4 besed
- Vsaka beseda ima 8 bitov
- $E(\overline{CS})$ – omogočanje izhoda

ROM - tipi

- ❑ Mask-Programmed ROM (**MROM**) – informacija se shrani **v času proizvodnje** integriranega vezja (Struktura MOS MROM – MOSFET (metal-oxide-semiconductor field-effect tranzistor je uporabljen za vsako pomnilno celico).
- ❑ Programmable ROM (**PROM**): je programiran enkrat na enak način, kot bo podano pri programabilnih gradnikih (PLD), Primer: TMS27PC256 je popularen CMOS ROM s kapaciteto 32K x 8.
- ❑ Erasable PROM (**EPROM**): brisanje in ponovno programiranje zunaj vezja, po programiranju se vsebina ne spreminja (pomnilni element je MOS tranzistor).
- ❑ Electrically Erasable PROM (**EEPROM**): izboljšani EPROM, njegova prednost je električnem brisanju (brisanje in pisanje celotne vsebine, brisanje in pisanje posameznih bajtov).
- ❑ Compact Disc (**CD-ROM**): digitalni podatki (0,1) so shranjeni bit za bit
- ❑ Bliskovni pomnilnik (**Flash memory**) –
brisanje celotne vsebine (ang. bulk erase) ali sektorjev
- ❑ Relacija: Kompleksnost in cena – slika

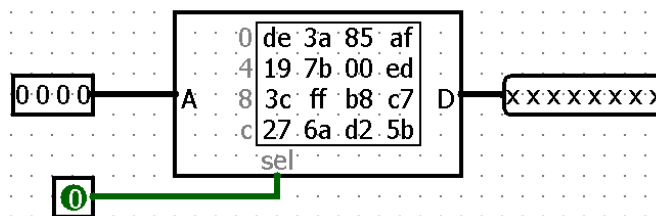


- Primer zapisa v ROMu (Tabela v binarnem zapisu in tabela v šestnajstiškem zapisu)

Naslov					Podatek							
Word	A ₃	A ₂	A ₁	A ₀	D ₇	D ₆	D ₅	D ₄	D ₃	D ₂	D ₁	D ₀
0	0	0	0	0	1	1	0	1	1	1	1	0
1	0	0	0	1	0	0	1	1	1	0	1	0
2	0	0	1	0	1	0	0	0	0	1	0	1
3	0	0	1	1	1	0	1	0	1	1	1	1
4	0	1	0	0	0	0	0	1	1	0	0	1
5	0	1	0	1	0	1	1	1	1	0	1	1
6	0	1	1	0	0	0	0	0	0	0	0	0
7	0	1	1	1	1	1	1	0	1	1	0	1
8	1	0	0	0	0	0	1	1	1	1	0	0
9	1	0	0	1	1	1	1	1	1	1	1	1
10	1	0	1	0	1	0	1	1	1	0	0	0
11	1	0	1	1	1	1	0	0	0	1	1	1
12	1	1	0	0	0	0	1	0	0	1	1	1
13	1	1	0	1	0	1	1	0	1	0	1	0
14	1	1	1	0	1	1	0	1	0	0	1	0
15	1	1	1	1	0	1	0	1	1	0	1	1

Word	A ₃	A ₂	A ₁	A ₀	D ₇ -D ₀
0	0				DE
1		1			3A
2		1			85
3			3		AF
4			4		19
5			5		7B
6			6		00
7			7		ED
8			8		3C
9			9		FF
10			A		B8
11			B		C7
12			C		27
13			D		6A
14			E		D2
15			F		5B

- Zapis ROMa v logisimu

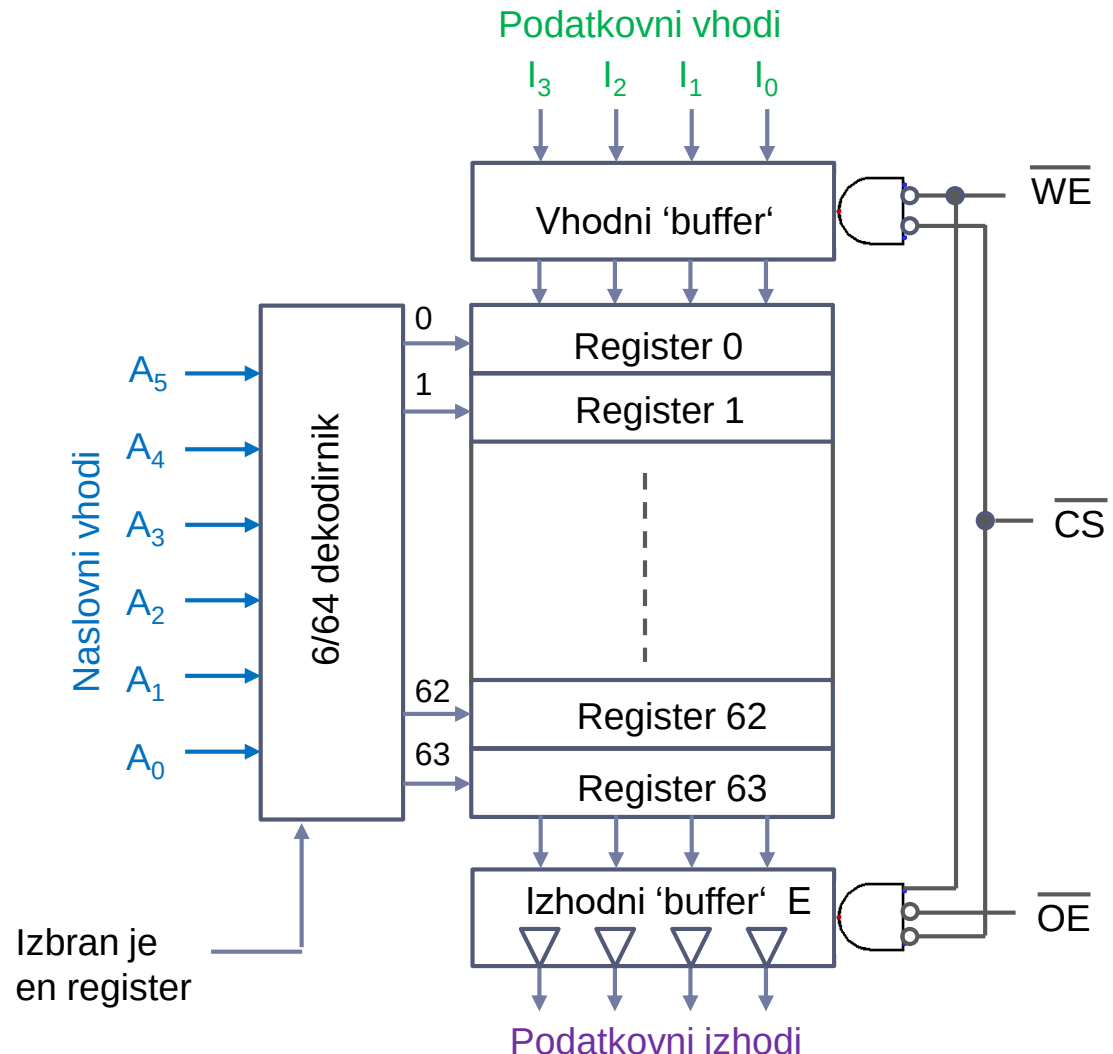


4. RAM (Random Access Memory)

- ❑ Pomnilnik z naključnim dostopom (ang. random access memory)
 - Čas dostopa enak za vse lokacije
 - Hiter dostop (visoka cena)
- ❑ Notranji pomnilnik, ki je povezan s CPE in začasno shranjuje:
 - podatke in
 - programe
- ❑ Bralno/pisalni pomnilnik (read/write memory)
- ❑ Neobstojni pomnilnik (ang. Volatile memory) - izguba vsebine ob izklopu:
 - Statični RAM
 - Dinamični RAM – osveževanje podatkov

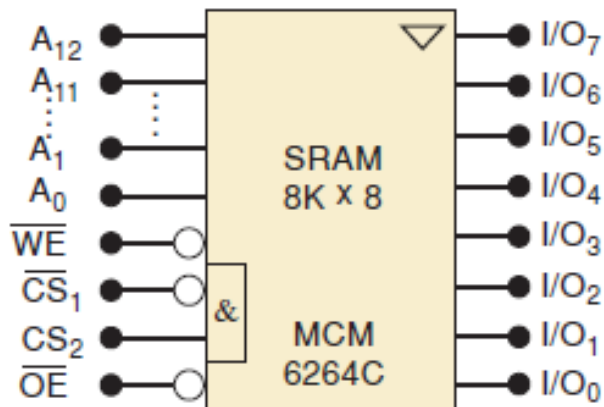
RAM 64 x 4 - Arhitektura

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



4.1 SRAM (Statični RAM)

- ❑ Na prejšnji prosojnici opisane operacije se nanašajo na SRAM.
- ❑ V pomnilniku je vsebina shranjena vse dokler je čip pod napajanjem.
- ❑ Izvedba, tehnologije: bipolarna, MOS,
večinoma poznamo čipe SRAM, ki so na voljo v NMOS ali CMOS izvedbi.
- ❑ SRAM se uporablja kot **predpomnilnik**.
- ❑ Primer: CMOS MCM6264C, 8K x 8 RAM



Mode	Inputs				I/O pins
	$\overline{WE}$	$\overline{CS}_1$	CS_2	$\overline{OE}$	
READ	1	0	1	0	DATA _{OUT}
WRITE	0	0	1	X	DATA _{IN}
Output disable	1	X	X	1	High Z
Not selected (power down)	X	1	X	X	High Z

4.2 Dinamični RAM (DRAM)

❑ Arhitektura-polje pomnilnih celic

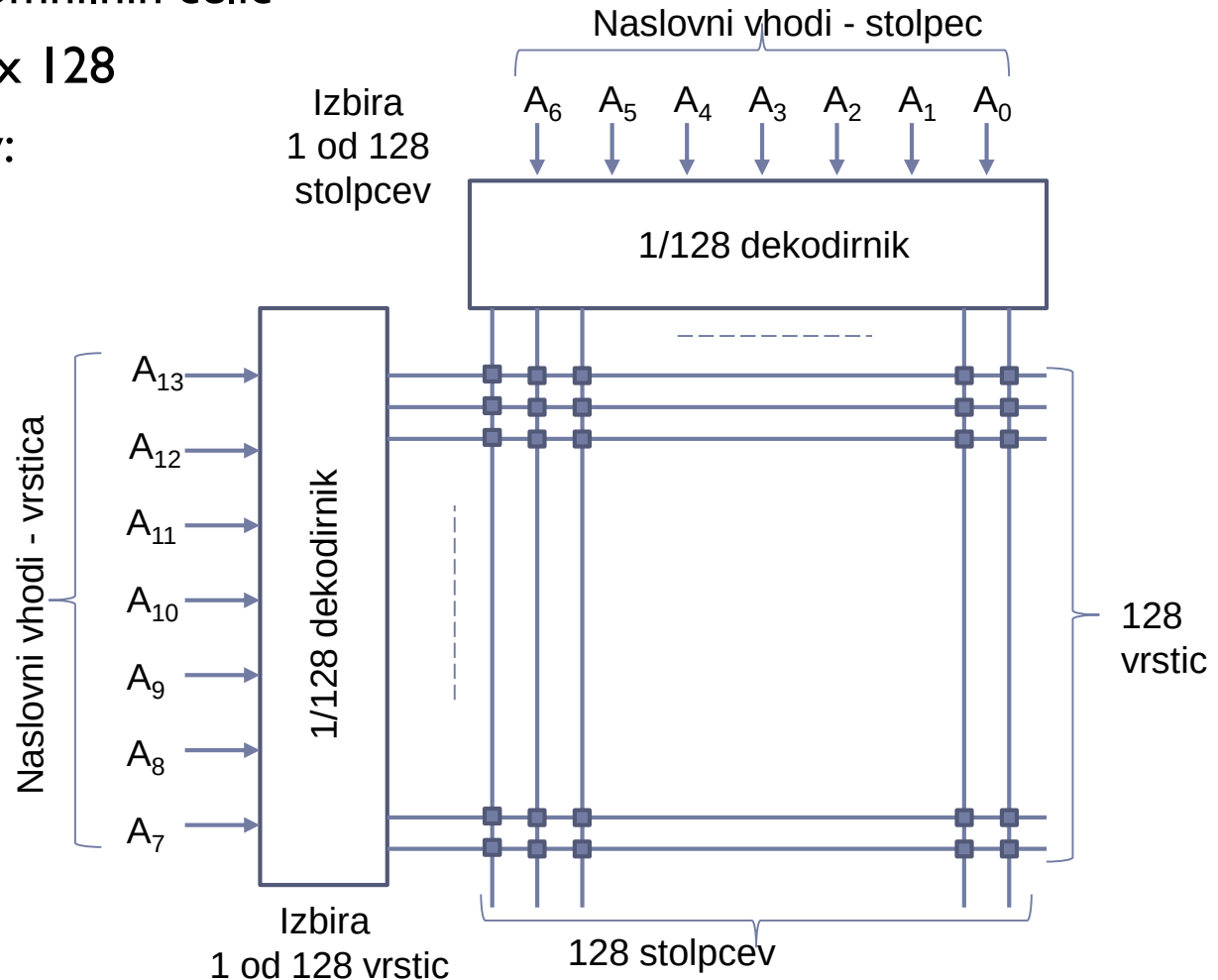
❑ 16,384 celic $\rightarrow 128 \times 128$

❑ 14 naslovnih vhodov:

- Stolpec: $A_0 - A_6$
- Vrstica: $A_7 - A_{13}$

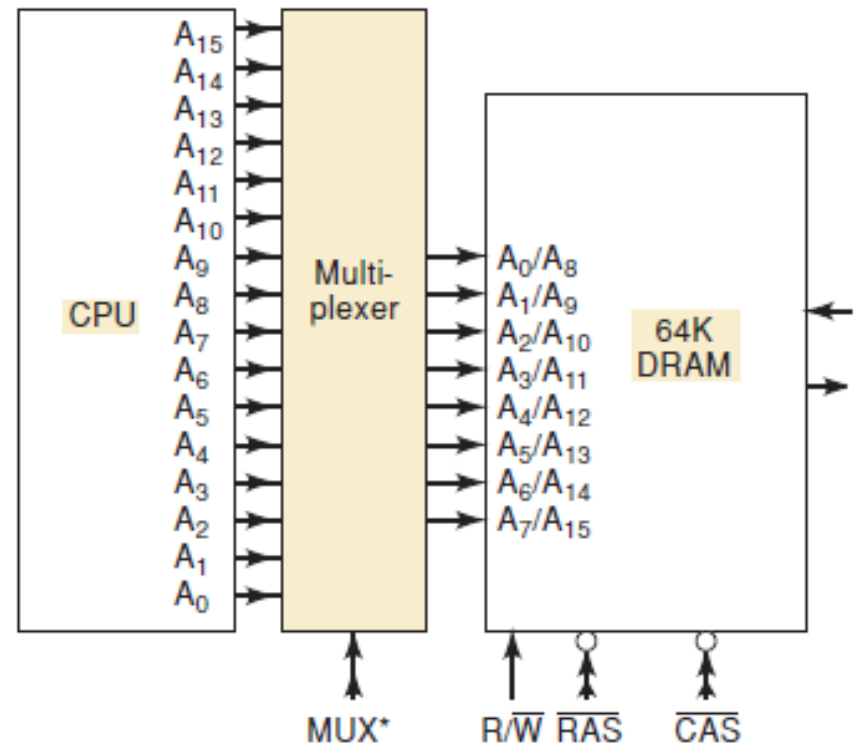
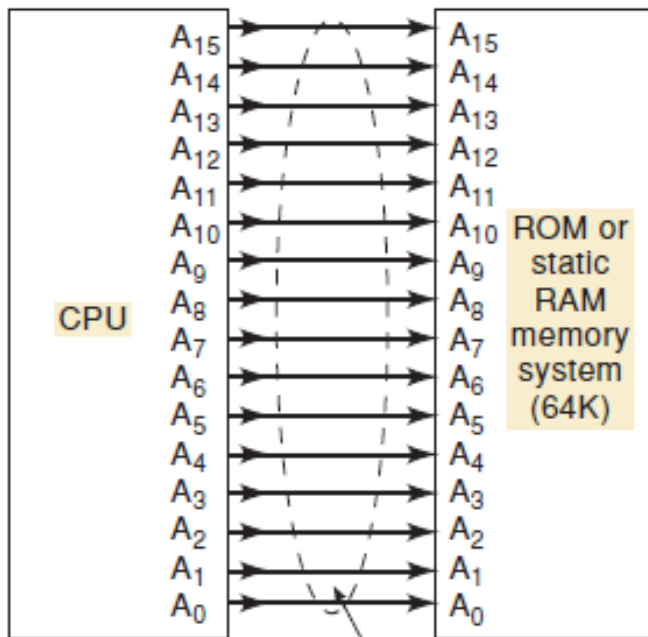
❑ Primer:

16K x 1 DRAM



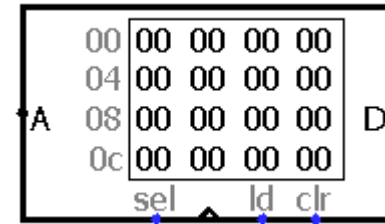
SRAM in DRAM naslavljanje

- DRAM - Shranjevanje naslova vrstice in stolpca krmilita signala:
 - $\overline{RAS}$ (row address strobe) – shrani naslovne vhode v register vrstice
 - $\overline{CAS}$ (column address strobe) - shrani naslovne vhode v register stolpca

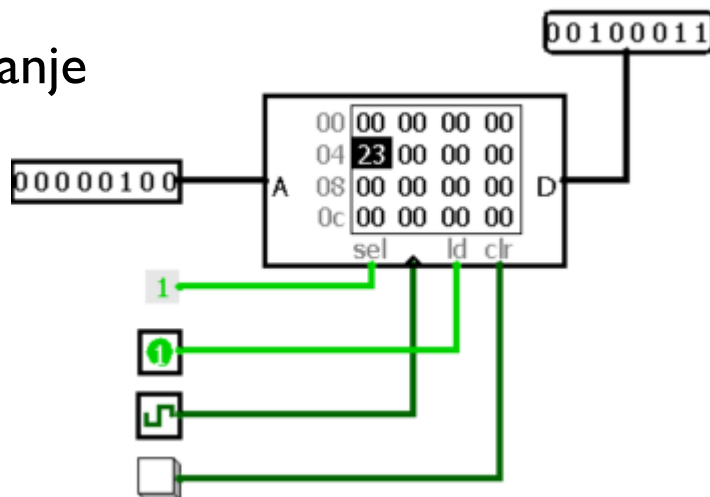


Primer: RAM v logisimu

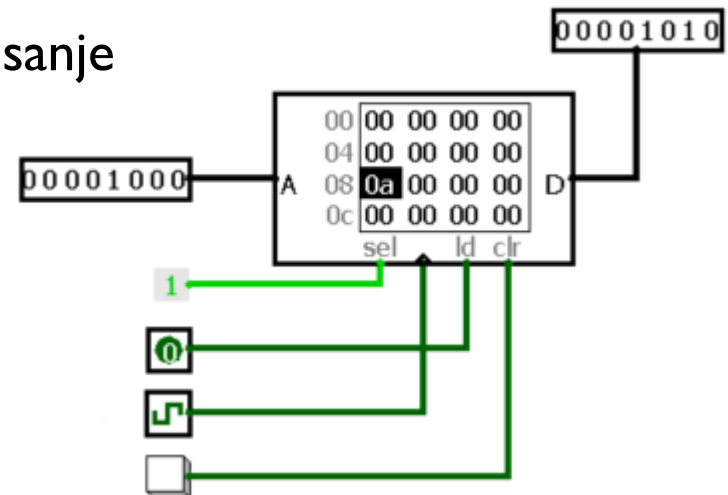
- ❑ A (ang. Address) – naslov
- ❑ D (ang. Data) - podatki
- ❑ Krmilni signali:
 - sel – onemogoči gradnik (sel = 0)
 - Ura
 - ld – branje (ld=1)/pisanje (ld=0)
 - clr – brisanje RAMa



Branje

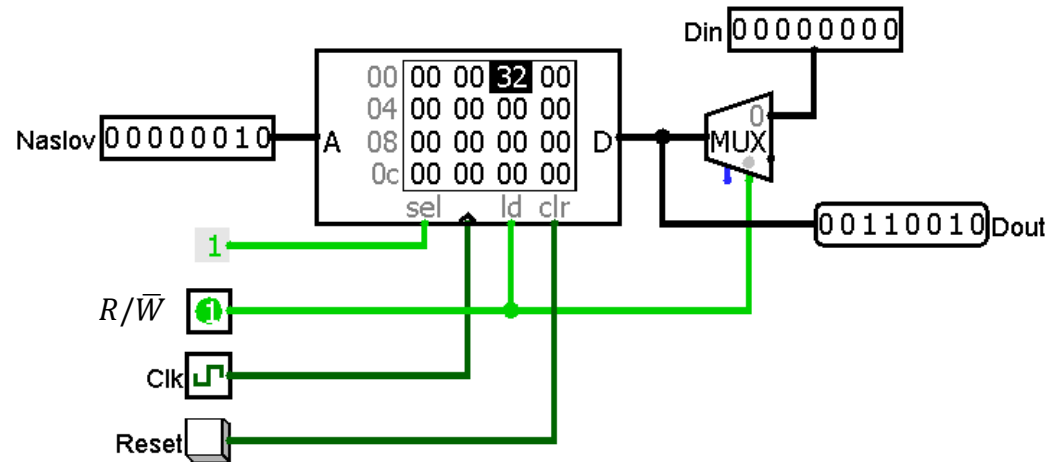


Pisanje

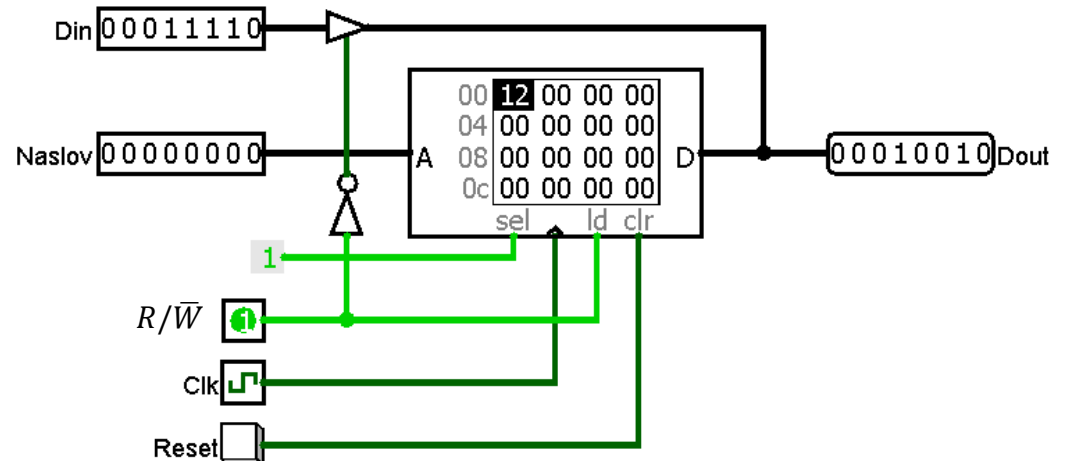


RAM - Vhod D je uporabljen za branje in pisanje

- RAM Din (2/I MUX)
- $R/\bar{W}$: 0 – write, 1 – read

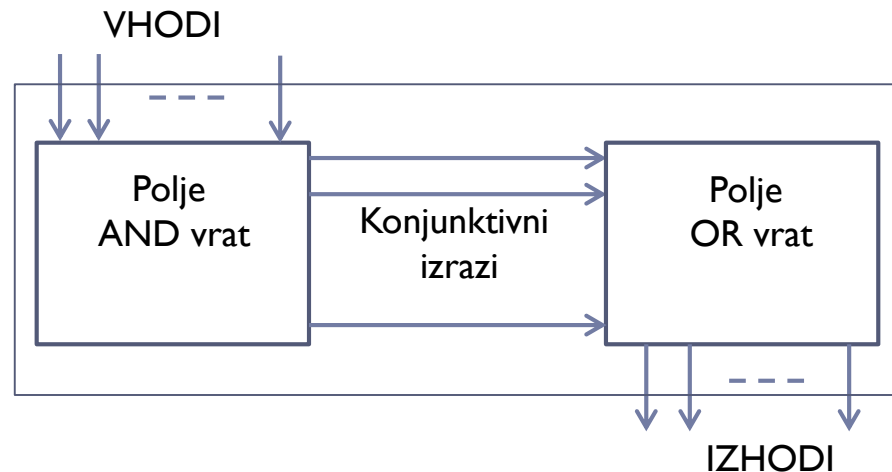


- RAM, Din (NEG, Controlled Buffer)



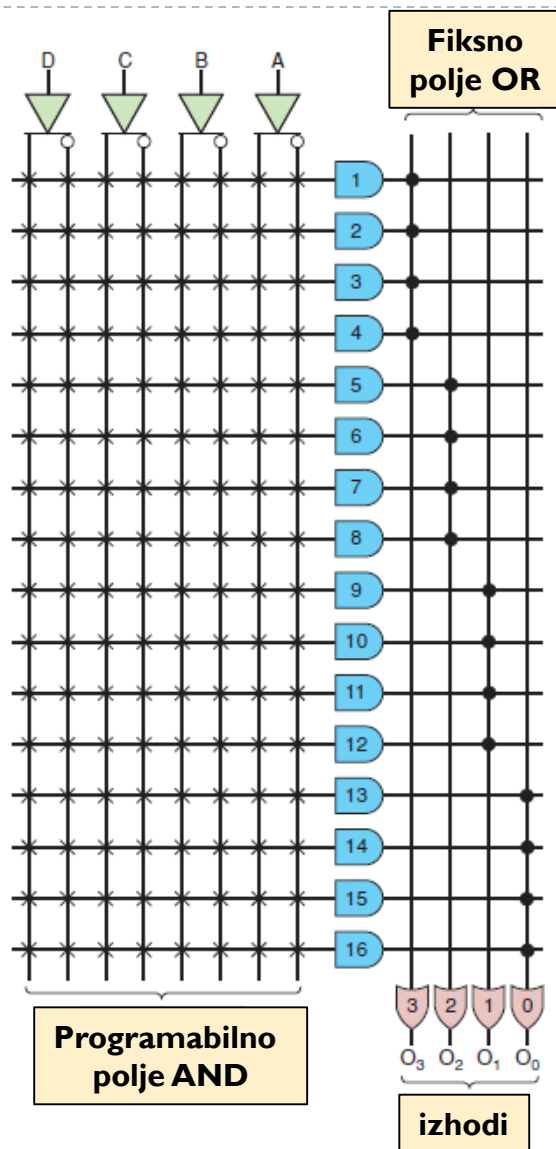
5. Programabilni gradniki

- ❑ Programabilni gradniki so namensko zasnovani gradniki za snovanje: enostavnih digitalnih vezij v obsežnih digitalnih sistemih.
- ❑ Arhitektura programabilnih gradnikov je podana z logičnimi vrati, ki so povezana v polje:
 - AND vrat z vhodi digitalnega vezja in
 - OR vrat z izhodi AND vrat



Arhitekture PLD: **PAL** (Programmable Array Logic) - programabilno polje AND

PRED
PROGRAMI-
RANJEM



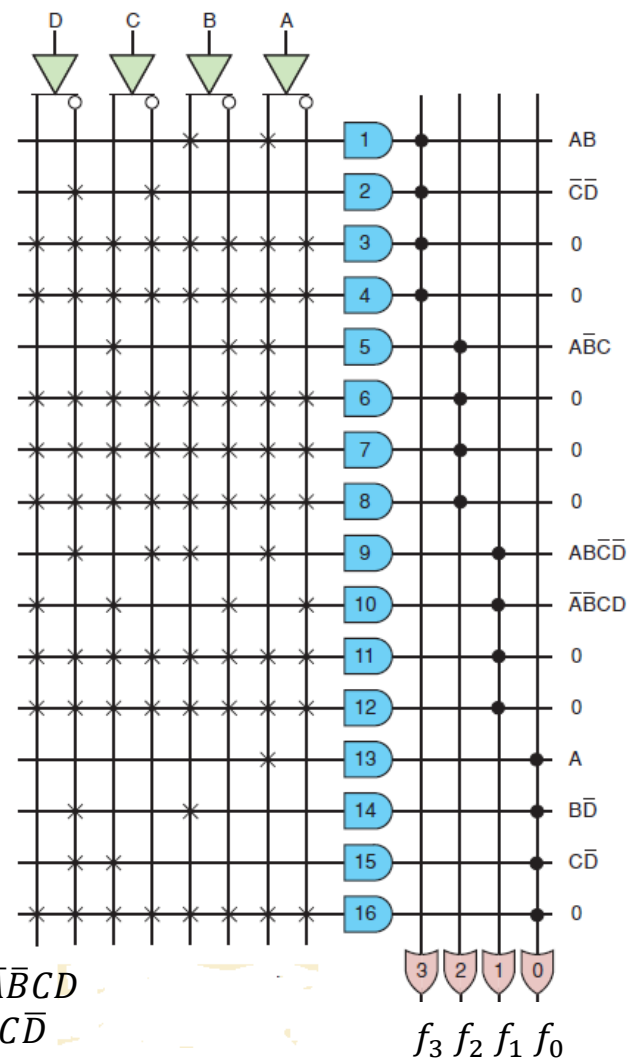
PO
PROGRAMI-
RANJU

$$f_3 = AB \vee \bar{C}\bar{D}$$

$$f_2 = A\bar{B}C$$

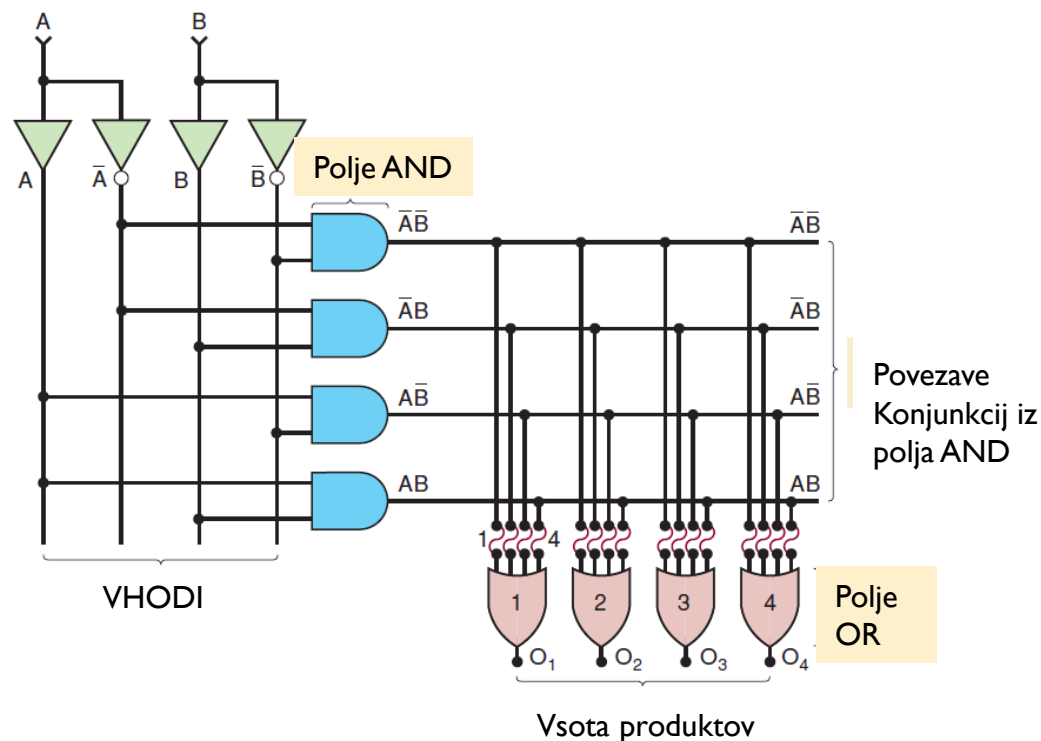
$$f_1 = AB\bar{C}\bar{D} \vee \bar{A}\bar{B}CD$$

$$f_0 = A \vee B\bar{D} \vee C\bar{D}$$



□ Primer enostavnega programabilnega vezja pred programiranjem

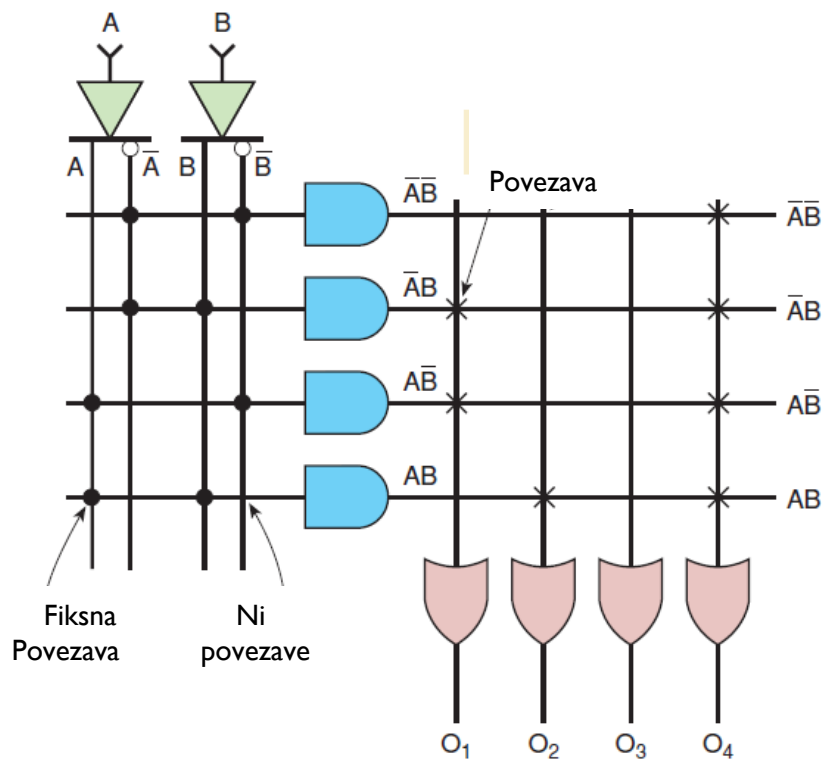
- Vsak izhod OR vrat je enak 1 ($O_1=O_2=O_3=O_4 = 1$)
- $O_1 = \bar{A} \cdot \bar{B} \vee \bar{A} \cdot B \vee A \cdot \bar{B} \vee A \cdot B = \bar{A} \cdot (\bar{B} \vee B) \vee A \cdot (\bar{B} \vee B) = \bar{A} \vee A = 1$
- Vsak izhod O_1, O_2, O_3, O_4 je programiran v odvisnosti od vhodov A, B.



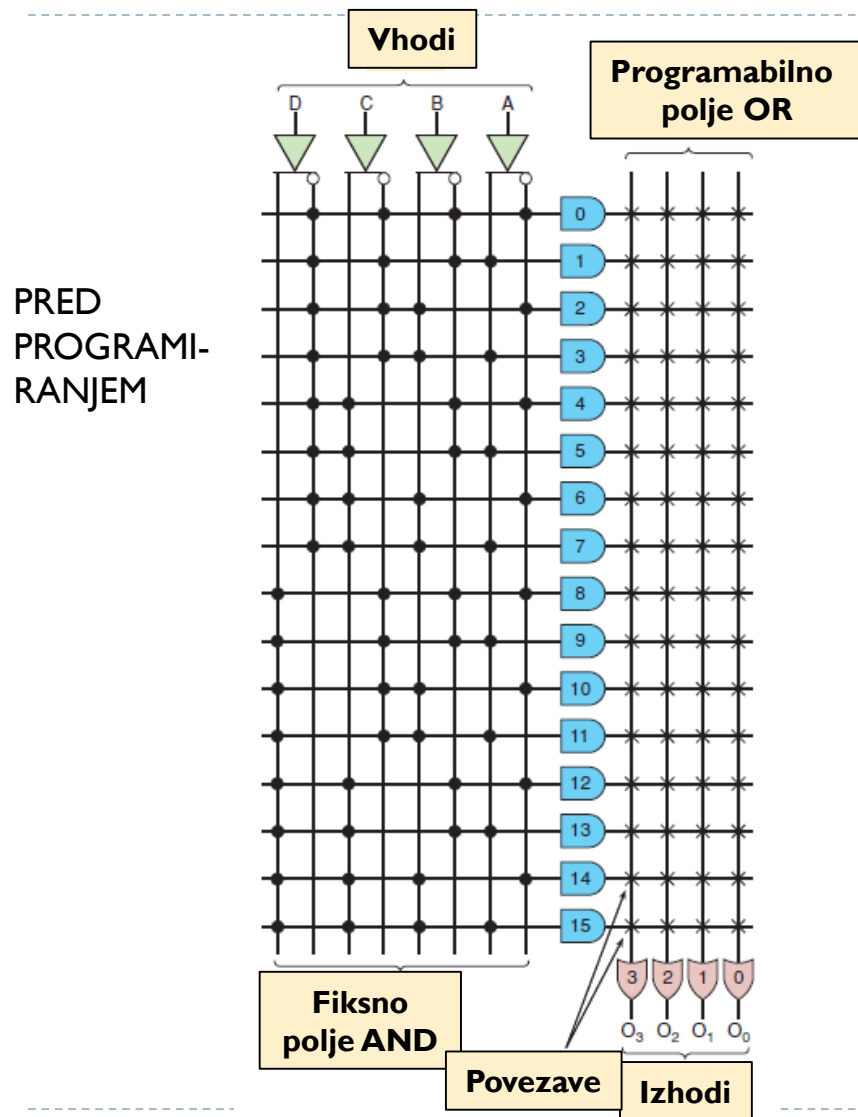
- Primer enostavnega programabilnega vezja po programiranju –
izhodi O_1, O_2, O_3, O_4 so programirani v odvisnosti od vhodov A, B.

Izhodne funkcije:

- $O_1 = \bar{A}.B \vee A.\bar{B}$
- $O_2 = A.B$
- $O_3 = 0$
- $O_4 = 1$

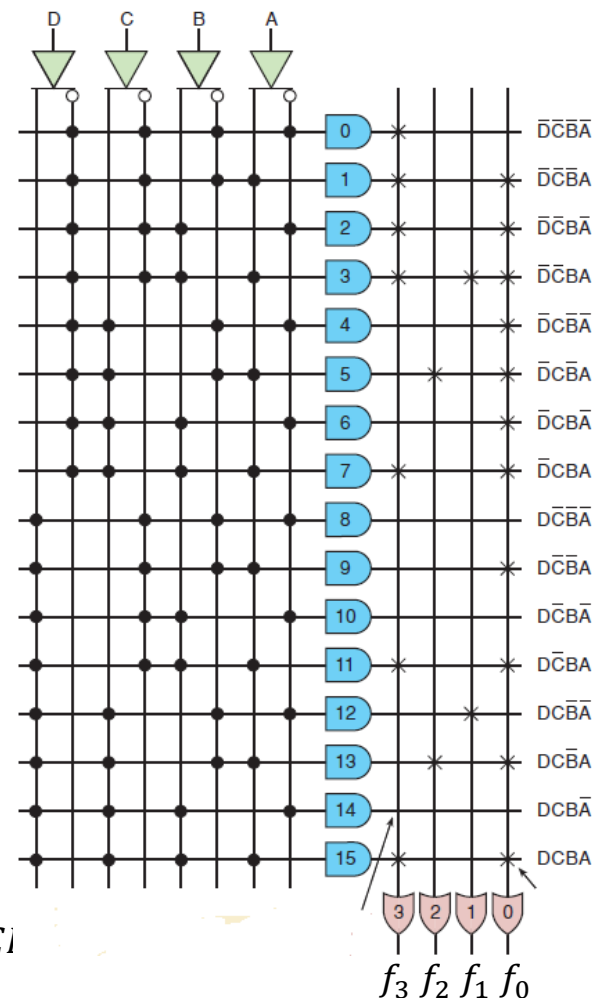


□ Arhitecture PLD: **PROM** (Programmable Read Only Memory)



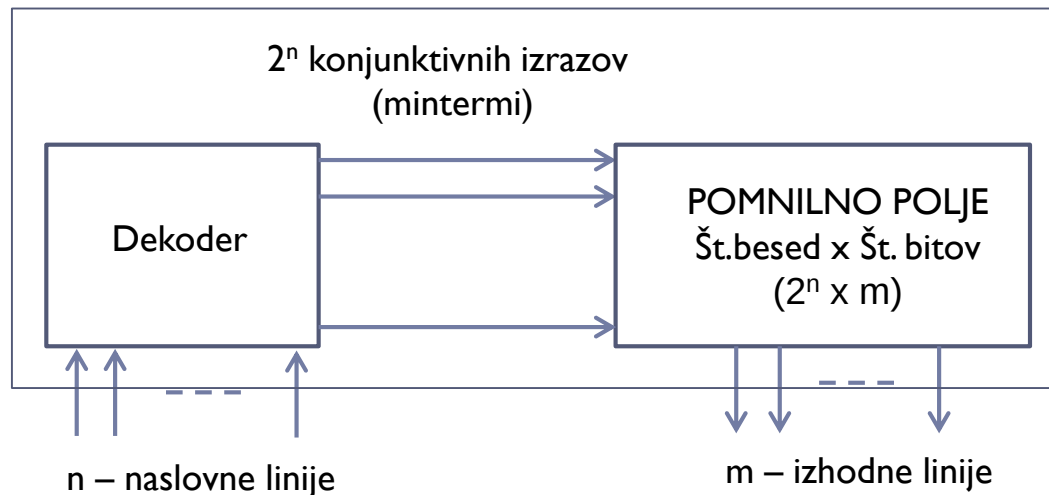
PO PROGRAMIRANJU

$$\begin{aligned}
 f_3 &= AB \vee \bar{C}\bar{D} \\
 f_2 &= A\bar{B}C \\
 f_1 &= AB\bar{C}\bar{D} \vee \bar{A}\bar{B}C\bar{D} \\
 f_0 &= A \vee B\bar{D} \vee C\bar{D}
 \end{aligned}$$



6 Programabilno vezje – (P)ROM

- ❑ Fiksno polje AND vrat je dekodeer, ki ima n naslovnih linij (vhodne spremenljivke in 2^n izhodov (vhodne kombinacije)
- ❑ Programabilno polje OR vrat, ki določajo izbiro pomnilne besede, v katero se vpišejo funkcijske vrednosti.



Realizacija funkcij z vhodi x_1, x_2, x_3 s programabilnim gradnikom PROM

Logične funkcije:

$$g_1 = x_1 \vee \overline{x_2} \cdot \overline{x_3}$$

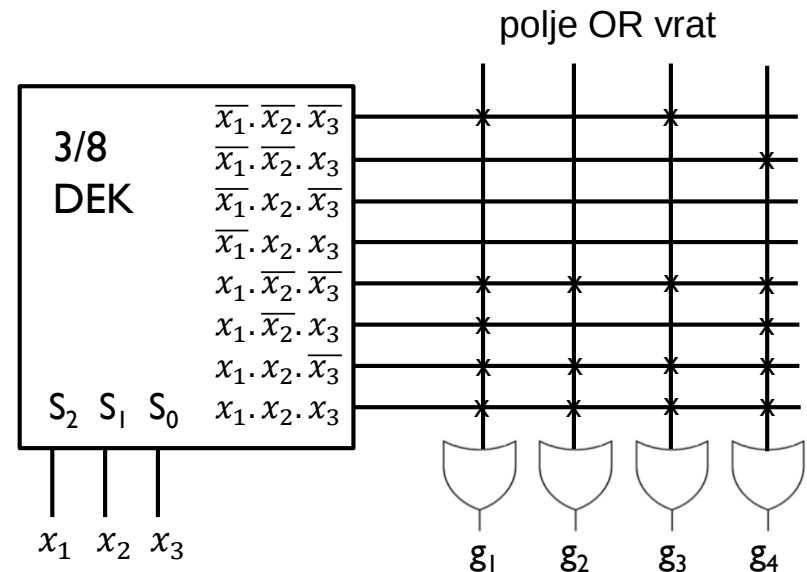
$$g_2 = x_1 \cdot \overline{x_3} \vee x_1 \cdot x_2$$

$$g_3 = \overline{x_2} \cdot \overline{x_3} \vee x_1 \cdot x_2$$

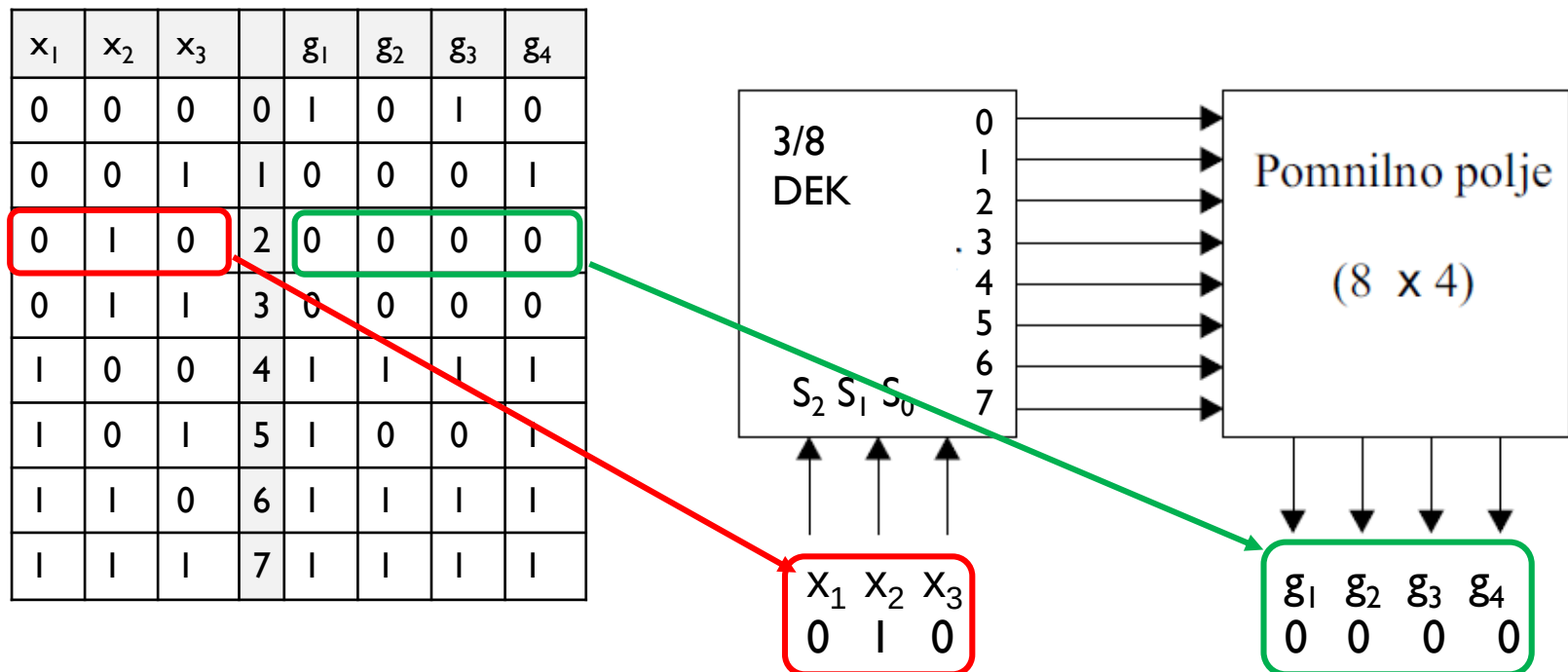
$$g_4 = x_1 \vee \overline{x_2} \cdot x_3$$

Pravilnostna tabela $\rightarrow$ vsota produktov $\rightarrow$ PROM

x_1	x_2	x_3		g_1	g_2	g_3	g_4
0	0	0	0	1	0	1	0
0	0	1	1	0	0	0	1
0	1	0	2	0	0	0	0
0	1	1	3	0	0	0	0
1	0	0	4	1	1	1	1
1	0	1	5	1	0	0	1
1	1	0	6	1	1	1	1
1	1	1	7	1	1	1	1



- Za realizacijo funkcije z uporabo PROMa je potreben zapis funkcije v pravilnostni tabeli ali popolna disjunktivna normalna oblika:
- Vhodne spremenljivke x_1, x_2, x_3 so krmilni vhodi dekodirnika (naslovni vhodi)
 - Izhod dekodirnika določa naslov za preslikavo funkcijskih vrednosti, ki so zapisane v pomnilnem polju na izhode g_1, g_2, g_3, g_4



Logisim - ROM

x1	x0	y1	y0	F
0	0	0	0	1
0	0	0	1	1
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	0
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

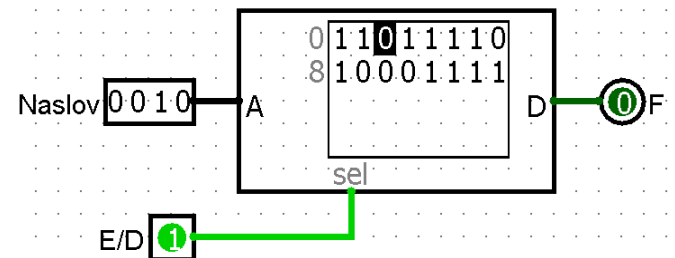
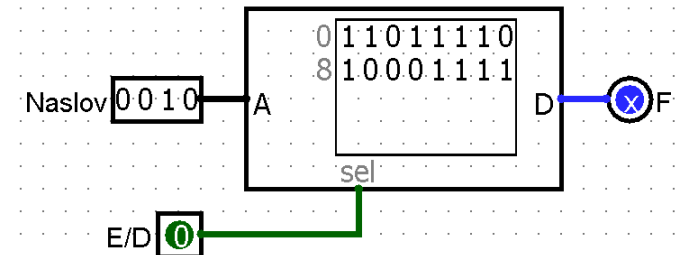
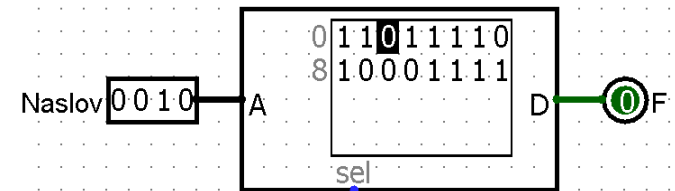
- A – Naslov (vhodne spremenljivke: x1, x0, y1, y0)

- D - funkcija $F(x1, x0, y1, y0)$

- sel – E/D (signal, ki onemogoči delovanje elementa)

- E/D=0 (izhod ni določen)

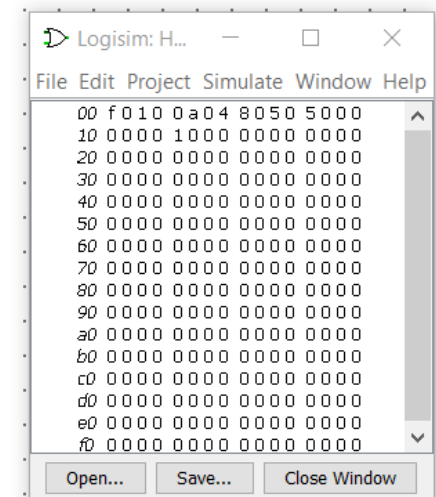
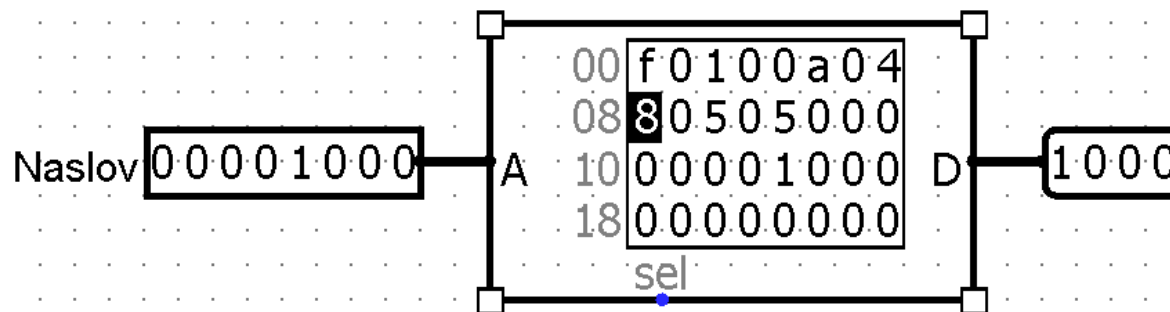
- E/D = 1 (izhod je določen z naslovom)



- ❑ Uporaba ROMa – preslikovalna tabela (ang. LUT – look-up table) za zapis logičnih funkcij:

- A - 8-bitni naslov
- D – 4-bitni podatek (krmilne funkcije)

- ❑ Logisim tabela in vpis podatkov



7. Naloge:

NI.)

Shraniti želimo $N=16$ 8-bitnih besed – kakšna je struktura pomnilnika?

1. Število bitov v naslovu je podano kot $K=\log_2 N$: ?
2. Velikost pomnilnika (kapaciteta-število bitov): ?

Rešitev:

1. Število bitov naslova:

$N=16$ besed

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

$$K = 4$$

2. Velikost pomnilnika:

$$16 \times 8 = 128 \text{ bitov}$$

ali

$$1 \text{ bajt (B)} = 8 \text{ bitov}$$

$$16 \times 1B = 16B$$

4-bitni naslov	8-bitni podatek (bajt) vsebina								
0000	b7	b6	b5	b4	b3	b2	b1	b0	bajt 0
0001									bajt 1
0010									bajt 2
0011									bajt 3
									
1110									bajt 14
1111									bajt 15

N2.)

- ❑ Zapis datuma v pomnilniku: 24.december 2021

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

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

- ❑ Dvojiški in šestnajstiški zapis

Naslov										Podatek	
0	0	0	1	1	0	0	1	0	2	31	
1	0	0	1	1	0	1	0	0	4	36	
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	1	0	2	32	
15	0	0	1	1	0	0	0	1	1	30	

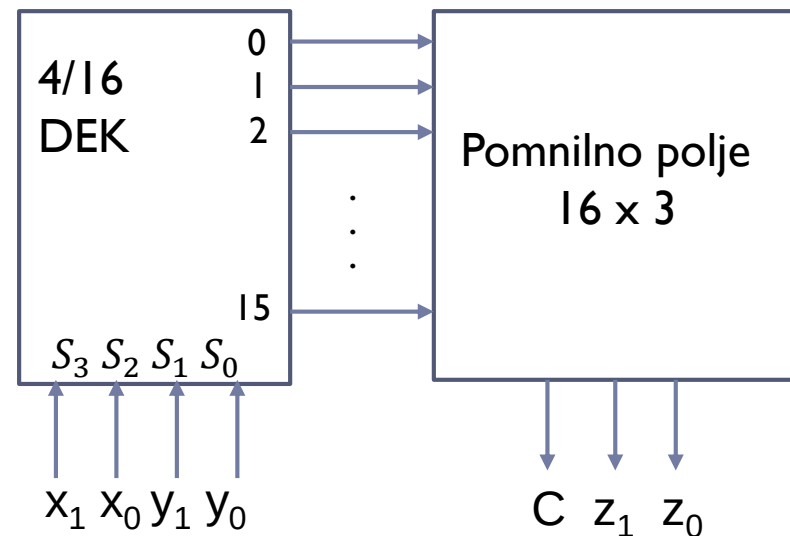
N3.) 2- bitni seštevalnik (PROM): Vhodi: $X=(x_1, x_0)$, $Y=(y_1, y_0)$, Izhodi: $Z=(z_1, z_0)$, C

x_1	x_0	y_1	y_0	C	z_1	z_0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	0	0	1
0	1	0	1	0	1	0
0	1	1	0	0	1	1
0	1	1	1	1	0	0
1	0	0	0	0	1	0
1	0	0	1	0	1	1
1	0	1	0	1	0	0
1	0	1	1	1	0	1
1	1	0	0	0	1	1
1	1	0	1	1	0	0
1	1	1	0	1	0	1
1	1	1	1	1	1	0

Zapis izhodov v programabilnem gradniku PROM, 16×3

Naslovni vhodi: x_1, x_0, y_1, y_0

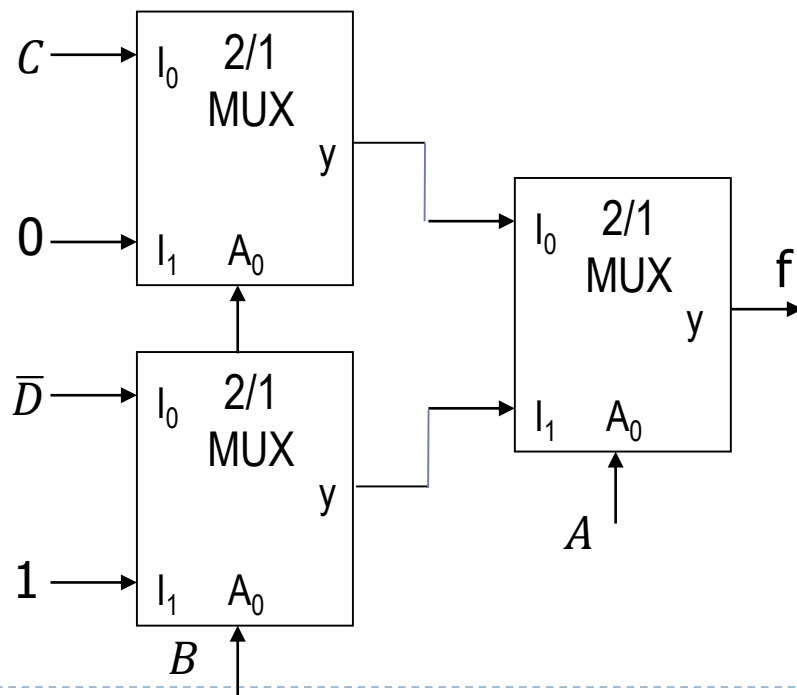
Izhodi: C, z_1, z_0



N4.)

Podana je logična funkcija $f(A,B,C,D)$, ki je realizirana z 2/1 MUXi. Naloge:

1. Zapišite funkcijo v disjunktivni normalni obliki iz podane sheme z MUXi.
2. Zapišite funkcijo v pravilnostni tabeli.
3. Zapišite rešitev za vpis logične funkcije v PROM.



I.) Funkcija iz sheme z MUXi

$$\begin{aligned} f(A, B, C, D) &= \bar{A} \cdot \bar{B} \cdot C \vee \\ &\quad \bar{A} \cdot B \cdot 0 \vee \\ &\quad A \cdot \bar{B} \cdot \bar{D} \vee \\ &\quad A \cdot B \cdot 1 = \\ &= \bar{A} \cdot \bar{B} \cdot C \vee A \cdot \bar{B} \cdot \bar{D} \vee A \cdot B \end{aligned}$$

2.)Zapis funkcije v pravilnostno tabelo 3.) zapis za realizacijo v PROMu

A	B	C	D	f
0	0	0	0	0
0	0	0	1	0
0	0	1	0	1
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Naslov (A,B,C,D)	Pomnilno polje
0	0
1	0
2	1
3	1
4	0
5	0
6	0
7	0
8	1
9	0
10	1
11	0
12	1
13	1
14	1
15	1

Realizacija funkcije f v logisimu (z MUXi, z ROMom)

