



## 8. Programirljivi moduli (2)

---



# Sadržaj predavanja

---

- koncept programirljivih modula
- permanentna memorija
- **programirljivo logičko polje**
  - **funkcionalnost i struktura**
  - **tehnologija izvedbe**
  - **ostvarivanje Booleovih funkcija**
- poluprogramirljivo logičko polje
- složeni programirljivi moduli
- programirljivo polje logičkih blokova



# Programirljivo logičko polje

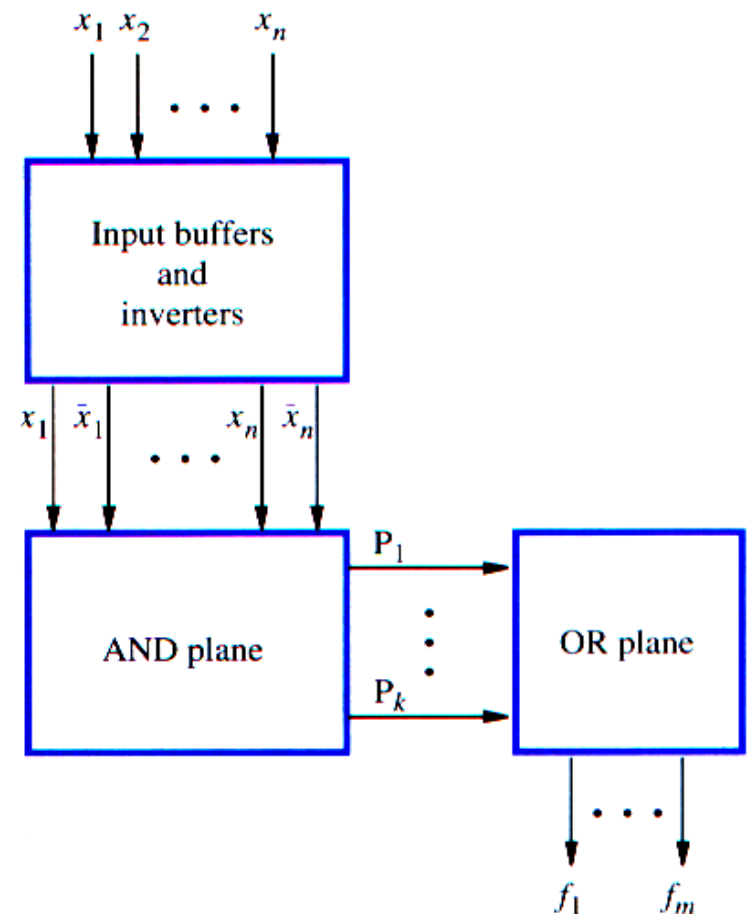
- zapažanje
  - ~ pri generiranju funkcija je spojno polje koda ROMa "slabo popunjeno"
  - obično se *ne* koriste *sve* kombinacije ulaza
    - ~ *nepotpuno specificirane funkcije*
  - ostvariti uštedu
    - ~ puno manji broj riječi od onog omogućenog dekodrom ( $W \ll 2^n$ )
  - relativno mali broj raspoloživih kombinacija ulaza
    - ~ obuhvatiti upravo one potrebne!
      - programiranje i dekodera
        - ~ *nepotpuno* dekodiranje

# Programirljivo logičko polje

- *programirljivo logičko polje*, PLA  
(engl. Programmable Logic Array)  
~ posebna logička struktura s *oba programirljiva* polja:
  - efikasno i fleksibilno rješenje
  - generirane funkcije  
~ u obliku *sume produkata*
  - ograničenje ostvarivosti funkcija  
~ veličina I polja (broj  $P_i$ ):
    - smanjenje broja riječi  
~ *minimizacija!*
  - višezlazna funkcija:
    - posebni postupak minimizacije
    - što više  $P_i$  *zajedničkih* za što veći broj funkcija

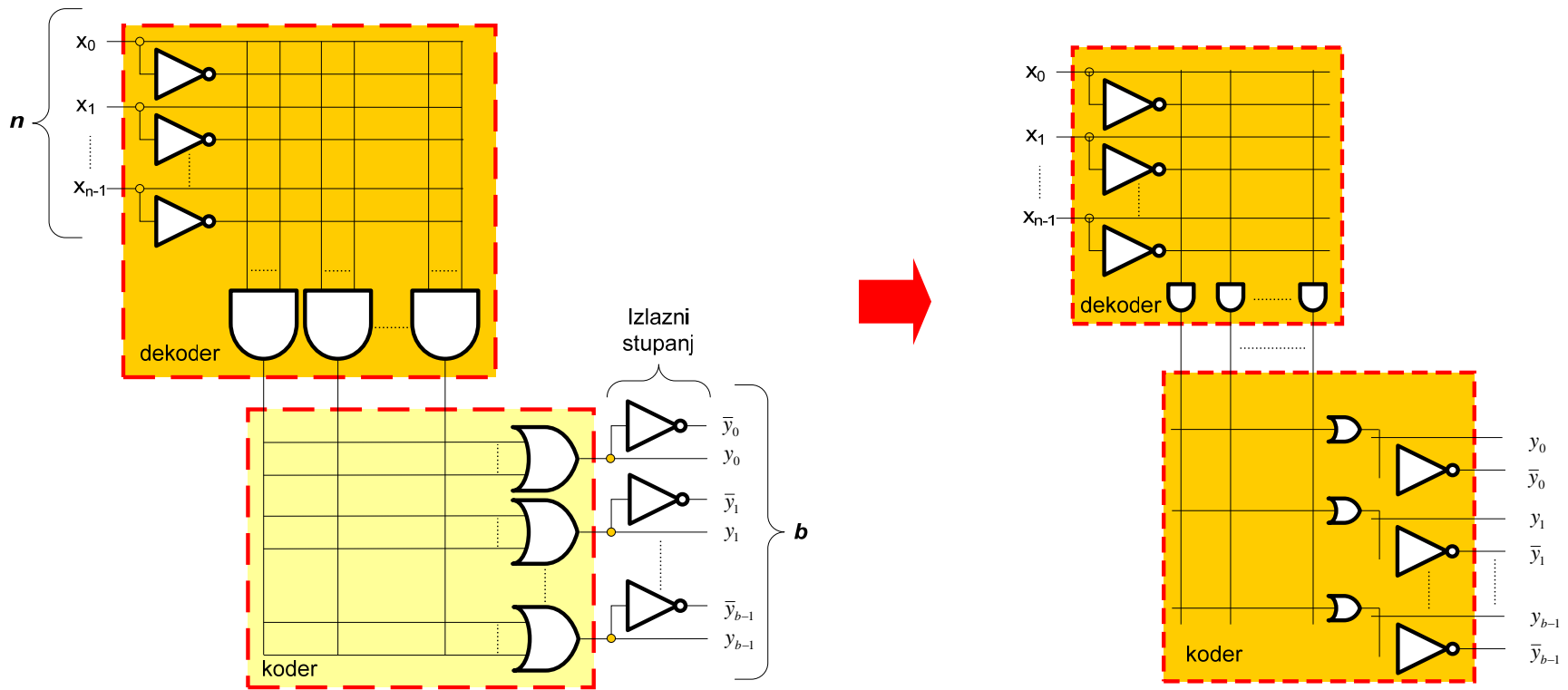
# Programirljivo logičko polje

- izvedba PLA:
  - odvojni sklopovi na ulazu  
~ generiranje  $x_i, \bar{x}_i, \forall i$
  - tipični parametri:  
 $n \times k \times m = 16 \times 32 \times 8$



# Programirljivo logičko polje

- struktura PLA:
  - izlaz  
~ funkcija i *komplement* (upravljani EX-ILI)

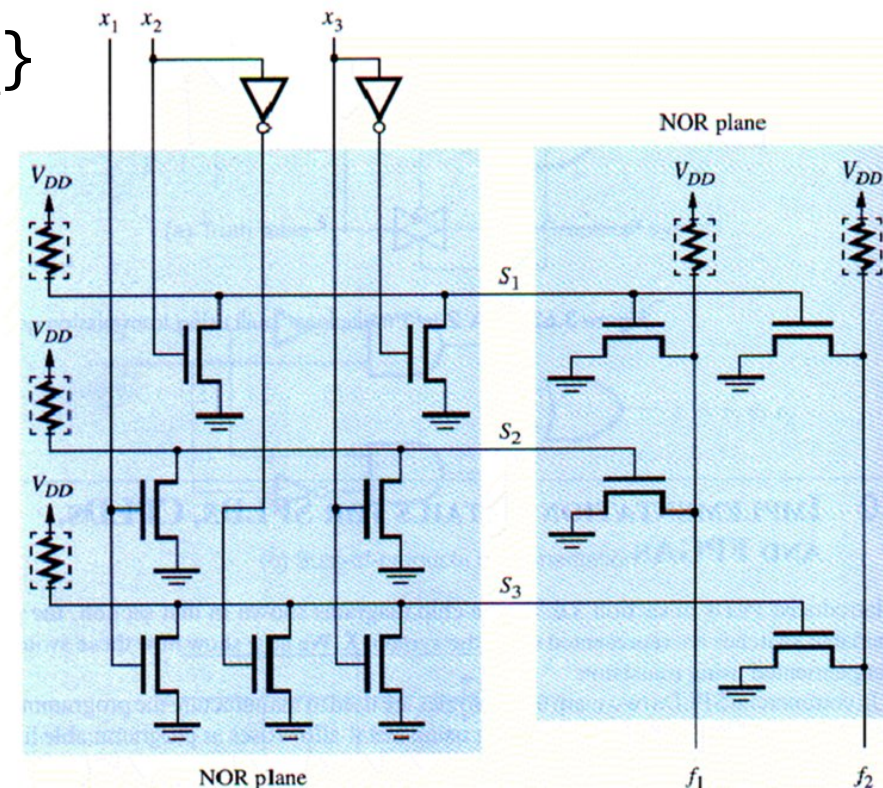


# Programirljivo logičko polje

*Primjer:* izvedba PLA na razini tranzistora

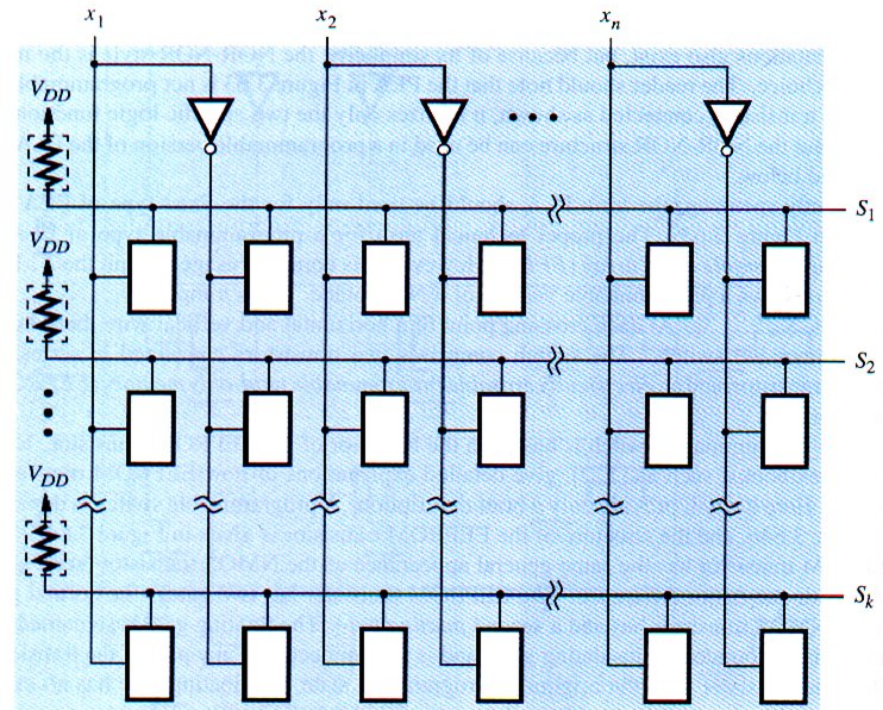
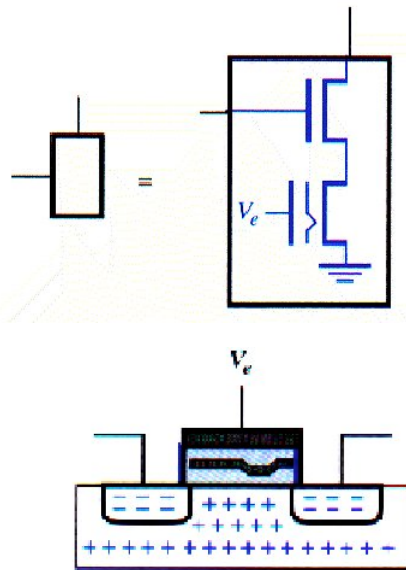
- programiranje u tvornici
- implementacija dvorazinske funkcije tipa NILI-NILI (NILI  $\sim$  paralelni NMOSovi)
- višezlazna funkcija:  $\{f_1, f_2\}$

$$\begin{aligned}f_1 &= \overline{S_1 + S_2} \\&= \overline{S_1} \cdot \overline{S_2} \\&= (x_2 + \overline{x_3}) \cdot (x_1 + \overline{x_3}) \\f_2 &= \overline{S_1 + S_3} \\&= \overline{S_1} \cdot \overline{S_3} \\&= (x_2 + \overline{x_3}) \cdot (x_1 + \overline{x_2} + x_3)\end{aligned}$$



# Programirljivo logičko polje

- izvedba FPLA (engl. Field-Programmable Logic Array):
  - (višekratno) programiranje "na licu mjesta"
  - NMOS Tr + programirljiva sklopka
  - programirljiva sklopka  $\sim$  EEPROM





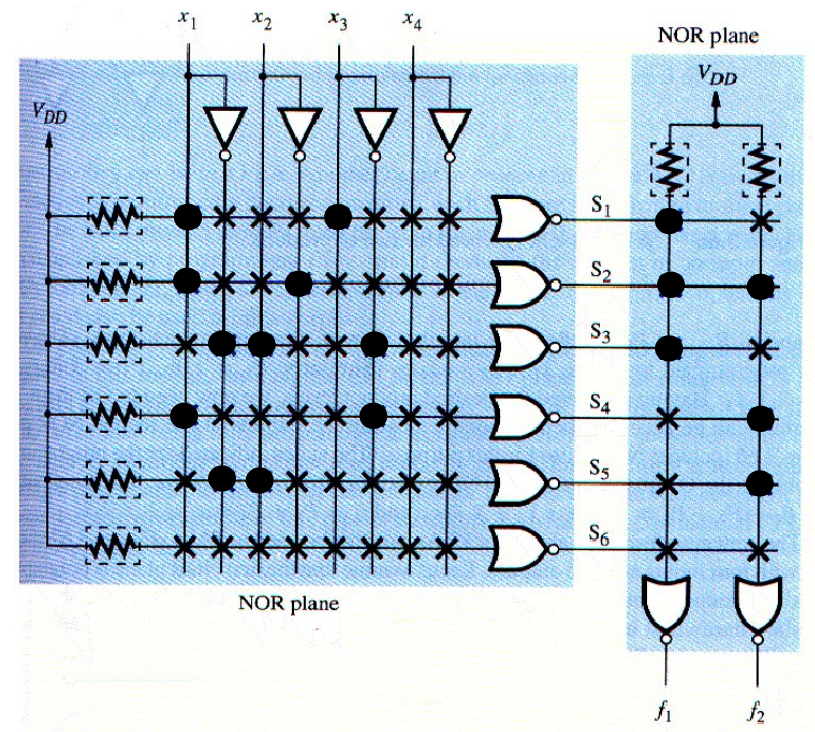
# Programirljivo logičko polje

*Primjer*: ostvarenje funkcija s FPLA

●: neprogramirana sklopka

x: programirana, *nema* spoja

$$\begin{aligned}f_1 &= \overline{S_1 + S_2 + S_3} \\&= \overline{S_1} \cdot \overline{S_2} \cdot \overline{S_3} \\&= (x_1 + x_3) \cdot (x_1 + \overline{x_2}) \cdot (\overline{x_1} + x_2 + \overline{x_3}) \\f_2 &= \overline{S_2 + S_4 + S_5} \\&= \overline{S_2} \cdot \overline{S_4} \cdot \overline{S_5} \\&= (x_1 + \overline{x_3}) \cdot (\overline{x_1} + x_2) \cdot (x_1 + \overline{x_2})\end{aligned}$$

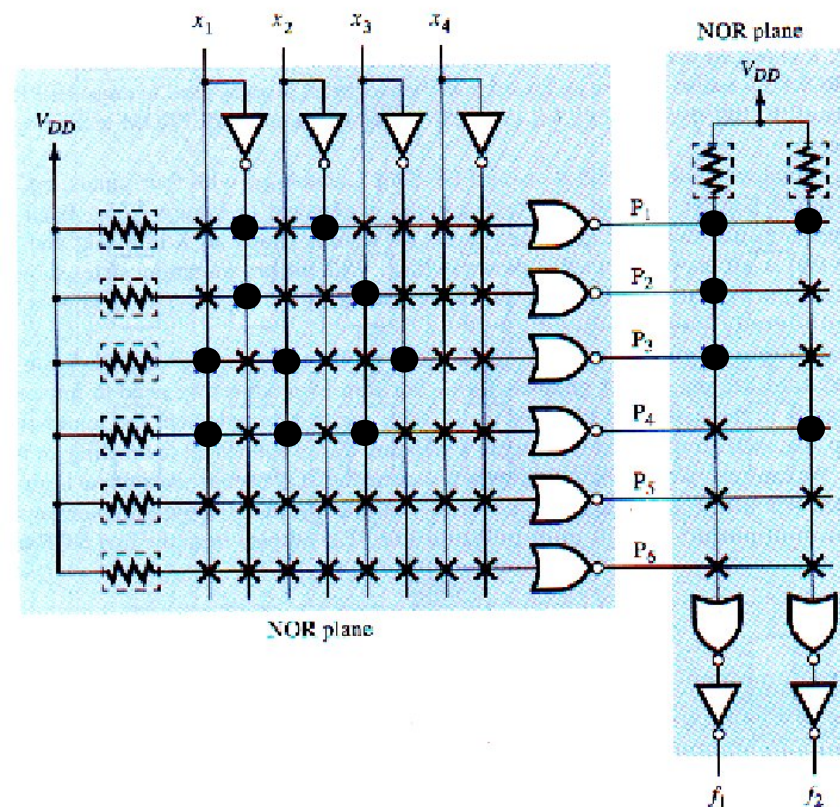


# Programirljivo logičko polje

- FPLA: ostvarenje dvorazinske funkcije tipa ILI-I
  - invertirati ulaze ( $x_1, \dots, x_4$ )
  - invertirati izlaz ( $f_1, f_2$ )

$$f_1 = P_1 + P_2 + P_3$$
$$= x_1 \cdot x_2 + x_1 \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot \overline{x_3}$$

$$f_2 = P_1 + P_4$$
$$= x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3}$$



# Programirljivo logičko polje

*Primjer:* ostvarivanje logičkih funkcija s PLA

$$Z_0 = \sum (0,1,3,6,8,9,10,13,14,21,24,27,30)$$

$$Z_1 = \sum (2,3,5,7,8,14,15,23,27,28,31)$$

$Z_0$   
DE

	000	010	110	100	001	011	111	101
00	1	1	1					
01	1	1				1		1
11	1		1					
10		1			1	1	1	

ABC

$Z_1$   
DE

	000	010	110	100	001	011	111	101
00		1					1	
01					1			
11	1		1		1	1	1	1
10	1					1		

ABC

$$Z_0 = \overline{A}\overline{C}\overline{D} + \overline{B}\overline{C}\overline{D}\overline{E} + \overline{A}\overline{B}\overline{C}\overline{E} + \overline{A}\overline{B}\overline{C}E + \overline{A}\overline{B}\overline{D}E + \overline{A}C\overline{D}\overline{E} + B\overline{C}\overline{D}\overline{E} \\ + A\overline{B}\overline{C}DE + A\overline{B}C\overline{D}E$$

$$Z_1 = CDE + \overline{A}\overline{B}CE + \overline{A}BCD + \overline{A}\overline{B}\overline{C}D + ABDE + \overline{A}\overline{B}\overline{C}\overline{D}\overline{E} + ABC\overline{D}\overline{E}$$

# Programirljivo logičko polje

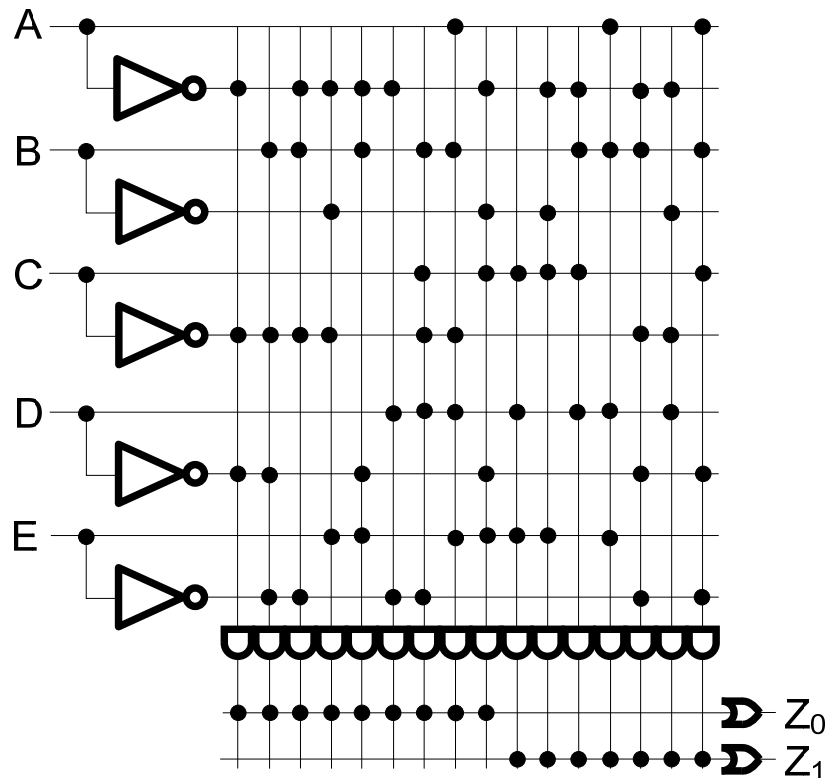
- matrični prikaz

$$Z_0 = \sum (0,1,3,6,8,9,10,13,14,21,24,27,30)$$

$$Z_1 = \sum (2,3,5,7,8,14,15,23,27,28,31)$$

$$\begin{aligned} Z_0 = & \overline{A}\overline{C}\overline{D} + \overline{B}\overline{C}\overline{D}\overline{E} + \overline{A}B\overline{C}\overline{E} \\ & + \overline{A}\overline{B}\overline{C}E + \overline{A}B\overline{D}E + \overline{A}C\overline{D}\overline{E} \\ & + \overline{B}C\overline{D}\overline{E} + \overline{A}B\overline{C}DE + \overline{A}\overline{B}C\overline{D}E \end{aligned}$$

$$\begin{aligned} Z_1 = & CDE + \overline{A}\overline{B}CE + \overline{A}BCD \\ & + \overline{A}\overline{B}\overline{C}D + ABDE \\ & + \overline{A}B\overline{C}\overline{D}\overline{E} + \overline{A}BC\overline{D}\overline{E} \end{aligned}$$



# Programirljivo logičko polje

## *Zadatak:*

- primjerenim sklopom PLA ostvariti:
  - potpuno zbrajalo
  - potpuno zbrajalo/odbijalo
  - BCD zbrajalo
- primjenom K-tablica i metode QuineMcCluskey sklopom PLA ostvariti višezlaznu funkciju:
  - $f_1(A, B, C, D) = \sum m(0,1,2,3,6,7)$   
 $f_2(A, B, C, D) = \sum m(0,1,6,7,14,15)$   
 $f_3(A, B, C, D) = \sum m(0,1,2,3,8,9)$
  - $f_1(A, B, C, D, E) = \sum m(0,1,2,3,6,7,20,21,26,27,28)$   
 $f_2(A, B, C, D, E) = \sum m(0,1,6,7,14,15,16,17,19,20,24,27)$   
 $f_3(A, B, C, D, E) = \sum m(0,1,2,3,8,9,16,20,26,28,30)$



# Sadržaj predavanja

---

- koncept programirljivih modula
- permanentna memorija
- programirljivo logičko polje
- **poluprogramirljivo logičko polje**
  - **funkcionalnost i struktura**
  - **ostvarivanje Booleovih funkcija**
- složeni programirljivi moduli
- programirljivo polje logičkih blokova



# Poluprogramirljivo logičko polje

---

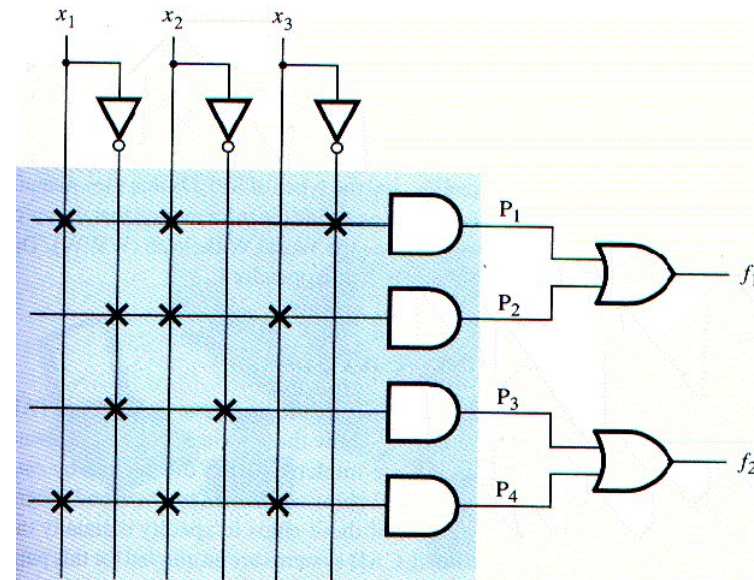
- problemi s PLA:
  - poteškoće s programirljivim sklopka:
    - teško ih je ispravno proizvesti
    - redukcija brzine rada sklopa ostvarenog s PLA
  - ograničiti mogućnost programiranja na *samo jedno* polje:
    - jednostavnije za proizvodnju
    - jeftinije
    - bolje performanse
    - smanjena fleksibilnost



# Poluprogramirljivo logičko polje

- *poluprogramirljivo polje*, PAL  
(engl. Programmable Array Logic)  
~ jako popularno rješenje:

- programira se samo I polje
- komercijalne izvedbe:  
~ 1000 programirljivih sklopki
- programiranje CAD sustavom
- kompenziranje reducirane funkcionalnosti  
~ proizvodnja u širokom rasponu veličina  
(broj ulaza i izlaza, broj ulaza u ILI sklopove)





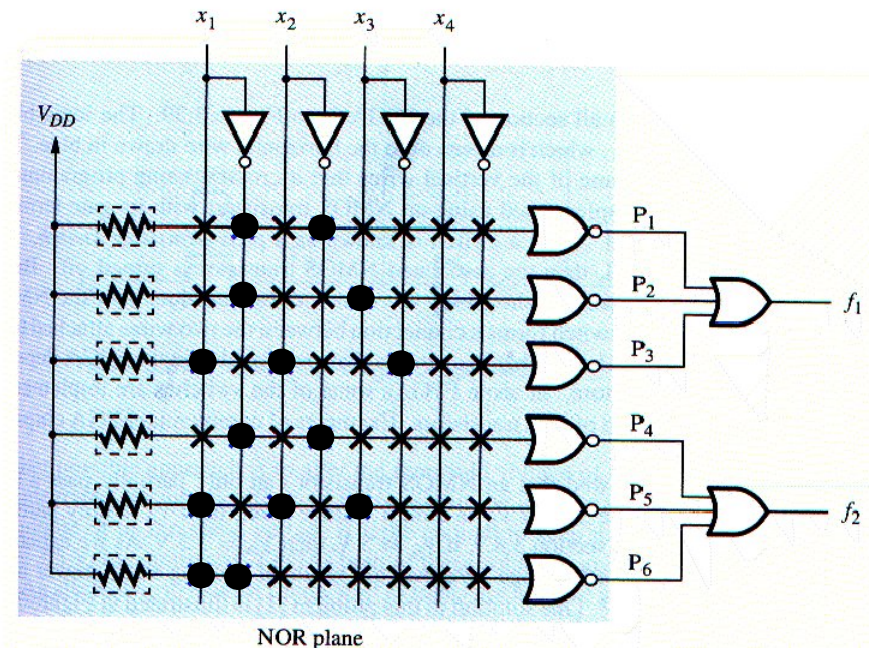
# Poluprogramirljivo logičko polje

*Primjer*: ostvarenje funkcije s PAL

- *nema* višestrukog korištenja  $P_i$   
(npr.  $P_1 = P_4 = x_1 \cdot x_2$ )  
~ *ne koristi* se minimiziranje višeizlaznih funkcija!!!
- nekorišteni  $\overline{P_i}$  programira se da daje 0:  
npr.  $P_6 = x_1 \cdot \overline{x_1} = 0$

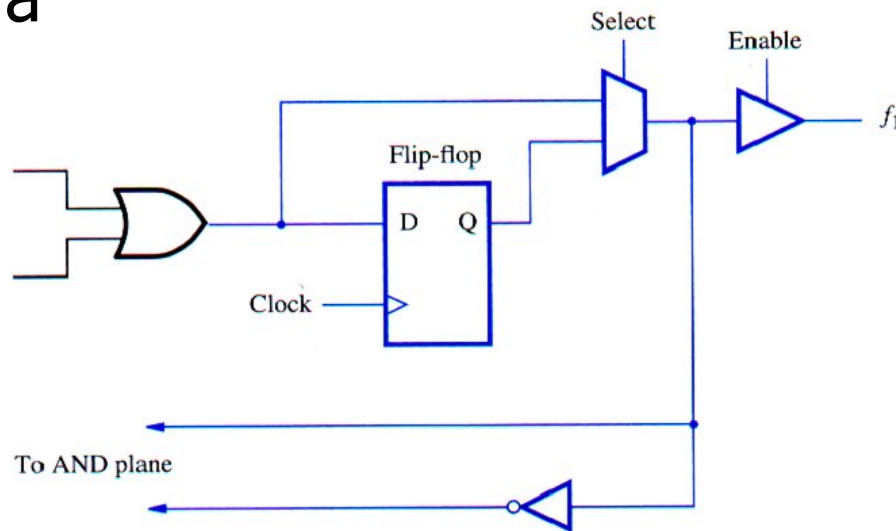
$$\begin{aligned} f_1 &= P_1 + P_2 + P_3 \\ &= x_1 \cdot x_2 + x_1 \cdot \overline{x_3} + \overline{x_1} \cdot \overline{x_2} \cdot x_3 \end{aligned}$$

$$\begin{aligned} f_2 &= P_4 + P_5 \\ &= x_1 \cdot x_2 + \overline{x_1} \cdot \overline{x_2} \cdot \overline{x_3} \end{aligned}$$



# Poluprogramirljivo logičko polje

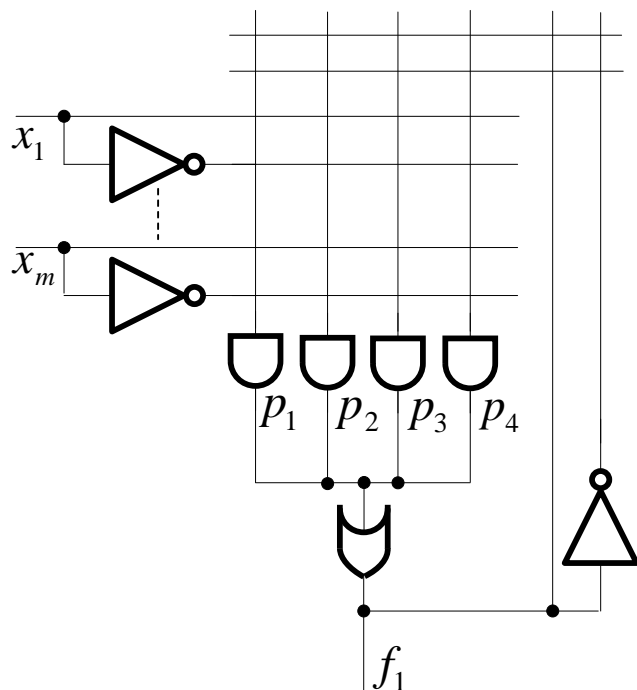
- povečanje fleksibilnosti  
~ *makročelija* (engl. macrocell):
  - dodatno sklopovlje na izlazu ILI sklopa
  - povratna veza na I polje  
~ ostvarivanja složenijih (~ *višerazinskih*) sklopova



# Poluprogramirljivo logičko polje

*Primjer:* potpuno zbrajalo/odbijalo ostvareno makroćelijom

- $K=0$ : zbrajalo,  $K=1$ : odbijalo
- uočiti:  $S_i = D_i$ , simetrija  $C_i$  i  $Z_i$
- primitivna makroćelija

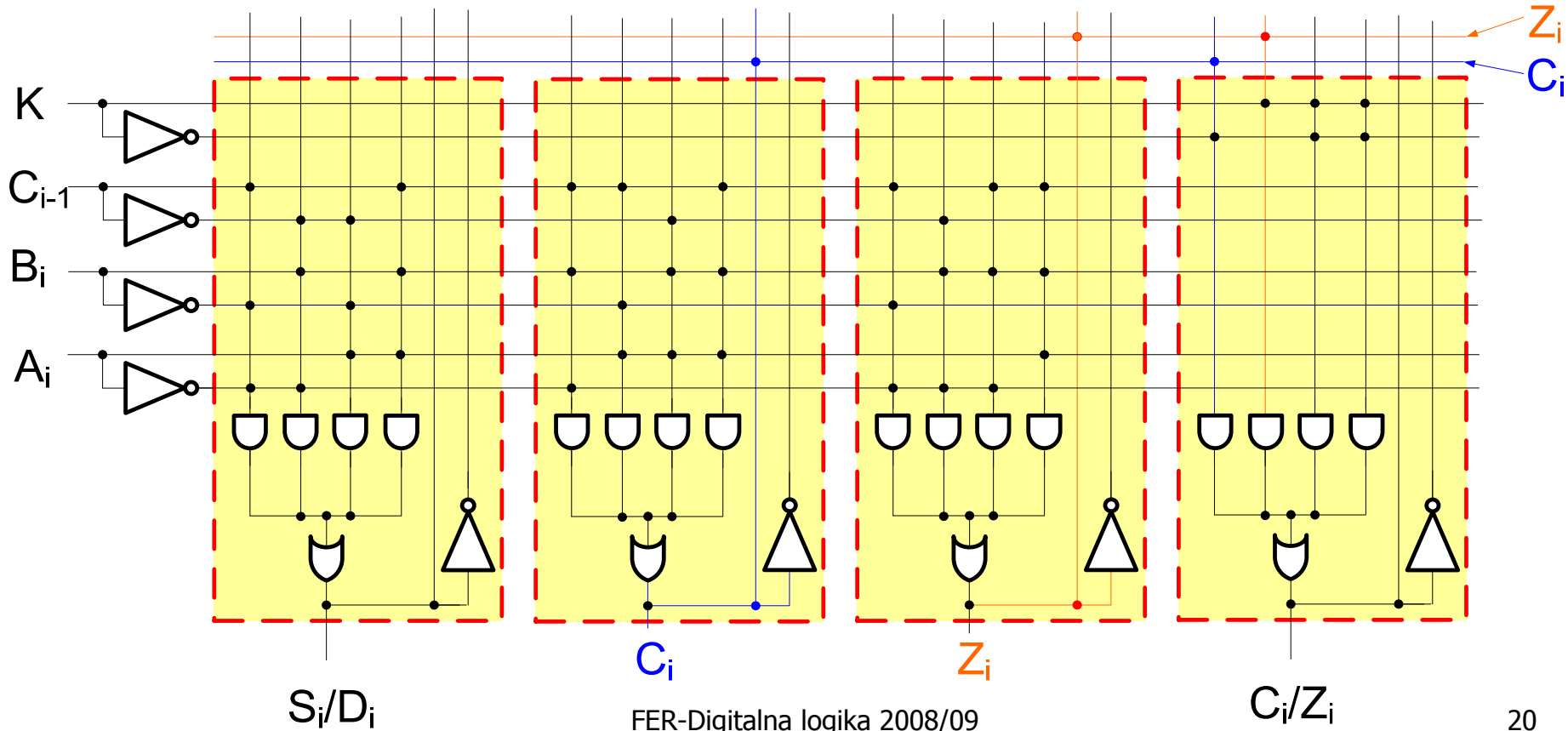


$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$	$D_i$	$Z_i$
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	1	0	0	1	0	0
1	1	1	1	1	1	1

# Poluprogramirljivo logičko polje

- potrebne su *četiri* makroćelije (zašto?)

$A_i$	$B_i$	$C_{i-1}$	$S_i$	$C_i$	$D_i$	$Z_i$
0	0	0	0	0	0	0
0	0	1	1	0	1	1
0	1	0	1	0	1	1
0	1	1	0	1	0	1
1	0	0	1	0	1	0
1	0	1	0	1	0	0
1	1	0	0	1	0	0
1	1	1	1	1	1	1





# Sadržaj predavanja

---

- koncept programirljivih modula
- permanentna memorija
- programirljivo logičko polje
- poluprogramirljivo logičko polje
- **složeni programirljivi moduli**
- programirljivo polje logičkih blokova

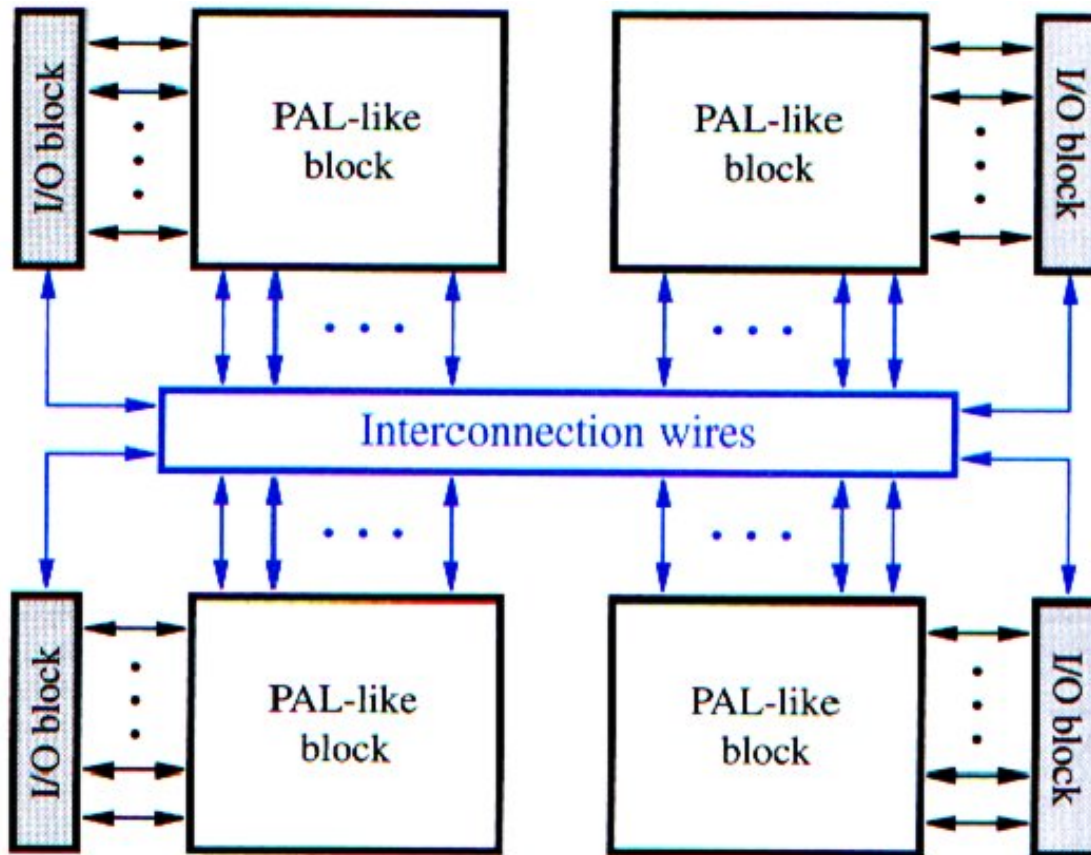


# Složeni programirljivi moduli

- programirljivi moduli:
  - SPLD (= PLA, PAL)  
~ skromne dimenzije sklopa
  - složeni PLD, CPLD:
    - *više* blokova sa sklopovljem (~ SPLD)
    - mogućnost internog povezivanja blokova
    - tipične dimenzije:
      - 2÷100 "PALu sličnih blokova "
      - 16 makroćelija u " PALu sličnom bloku "
      - 5÷20 ulaza u ILI sklopove
      - EX-ILI za programirljivo komplementiranje izlaza
      - izlaz iz makroćelije  
~ sklop s tri stanja

# Složeni programirljivi moduli

- struktura CPLD:





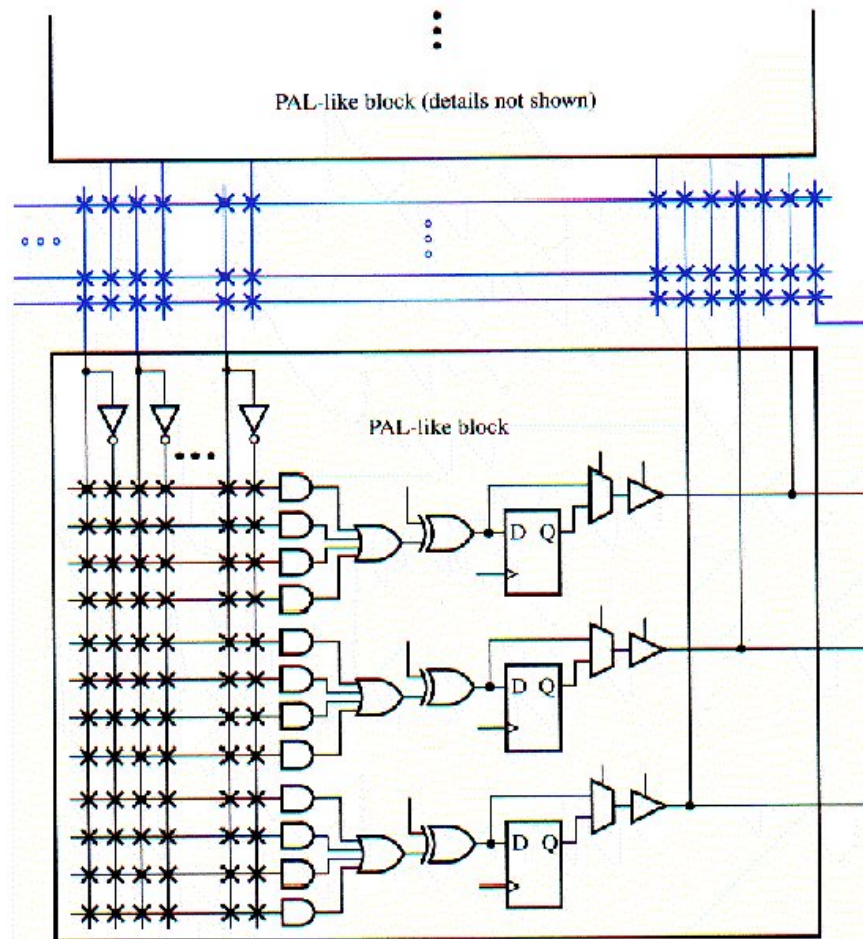
# Složeni programirljivi moduli

- struktura CPLD:
  - *PALu slični blokovi* (engl. PAL-like blocks)  
~ ostvarenje primitivnije funkcije
  - *UI blokovi*  
~ sučelje za svaki PALu slični blok
  - *povezno ožičenje* (engl. interconnection wires):
    - programirljive sklopke za povezivanje PALu sličnih blokova
    - broj programirljivih sklopki:  
~ pažljiva procjena! (fleksibilnost-efikasnost)
    - vertikalne linije  
~ ulazi u makroćeliju i izlazi iz nje
    - problem:  
~ ako je izvod IC korišten kao izlaz, pripadna je makroćelija neupotrebljiva



# Složeni programirljivi moduli

- struktura PALu sličnog bloka:



# Složeni programirljivi moduli

- programiranje CPLD
  - ~ tipično *u sustavu u kojem se koriste* (engl. In-System Programming, ISP)
- mehanički razlozi:
  - ~ IC s velikim brojem ( $\sim 100 \div 200$ ) krhkih i savitljivih izvoda
- *podnožja* (engl. sockets) su sumjerljivo skupa
- posebni *konektor* povezan na sve CPLD u sustavu
  - ~ *JTAG pristup* (engl. JTAG port, Joint Test Action Group), standard IEEE
- *stalno* (engl. nonvolatile) programiranje
- primjena CPLD u  $\sim 50\%$  slučajeva korištenja PLD



# Sadržaj predavanja

---

- koncept programirljivih modula
- permanentna memorija
- programirljivo logičko polje
- poluprogramirljivo logičko polje
- složeni programirljivi moduli
- **programirljivo polje logičkih blokova**
  - **struktura i logički blokovi**
  - **ostvarivanje Booleovih funkcija**

# Programirljivo polje logičkih blokova

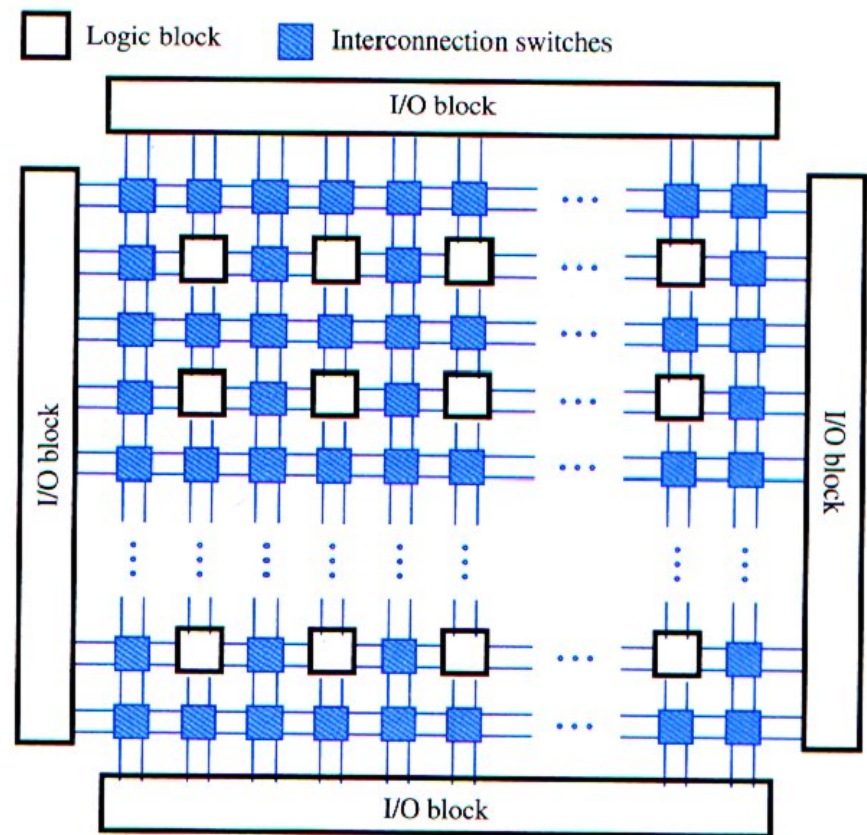
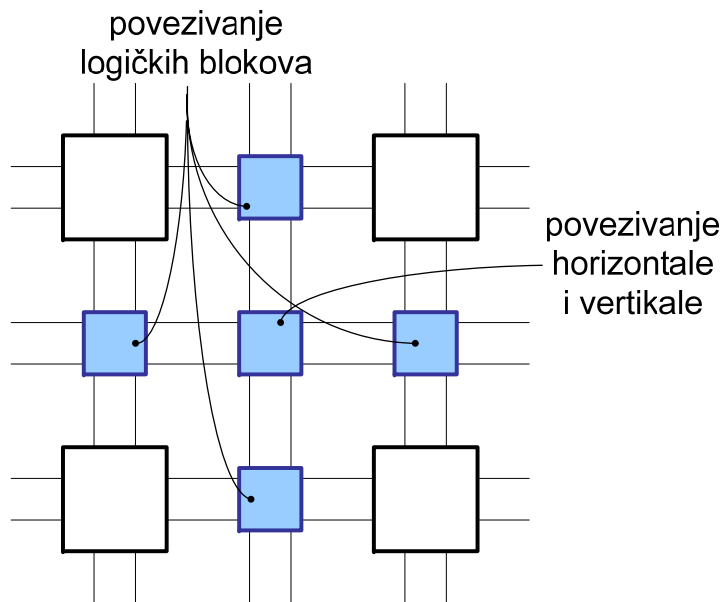
- mjera složenosti digitalnog sklopa  
~ broj *ekvivalentnih sklopova* (engl. equivalent gates):  
ukupni broj NI sklopova s 2 ulaza koji bi bili  
potrebni za njegovo ostvarenje
- SPLD/CPLD podržavaju ostvarenja  
relativno *jednostavnijih* sklopova
  - makroćelije SPLD/CPLD: ~20  
npr. PAL s 8 makroćelija: ~160  
CPLD s 1000 makroćelija: ~20.000
  - smanjenje troškova i povećanja performansi  
⇒ sklop sa *što manjim* brojem IC

# Programirljivo polje logičkih blokova

- *programirljivo polje logičkih blokova*  
(engl. Field Programmable Gate Array, FPGA)
  - PLD za ostvarivanje *relativno velikih* digitalnih sklopova ( $\geq 100.000$  ekvivalentnih sklopova)
  - *logički blokovi* (engl. logic blocks, LB) umjesto I/ILI polja
  - tipična struktura FPGA:
    - LB organizirani u *dvodimenzijско* polje
    - *UI blokovi* za sučelje (s izvodima IC)
    - vodovi i programirljive sklopke  
~ za međusobno povezivanje LB:  
horizontalni i vertikalni  
*kanali za usmjeravanje* (engl. routing channels)  
između redaka i stupaca LBova

# Programirljivo polje logičkih blokova

- općenita struktura FPGA:
  - bijela polja ~ logički blokovi (LBovi)
  - plava polja ~ programirljive sklopke



# Programirljivo polje logičkih blokova

- općenita struktura *logičkog bloka*:
  - više ulaza, jedan izlaz  
~ ostvaruje *jednostavnu* Booleovu funkciju
  - više tipova, najuobičajeniji koristi preglednu tablicu (LUT) s memorijskim ćelijama:
    - ostvarivanje funkcije primjerenim MUX  
~ izvedba multipleksorskim stablom
    - MUX u funkciji permanentne memorije!
    - tipično:  
LUT s 4/5 ulaza → 16/32 ćelija  
+ neko dodatno sklopovlje

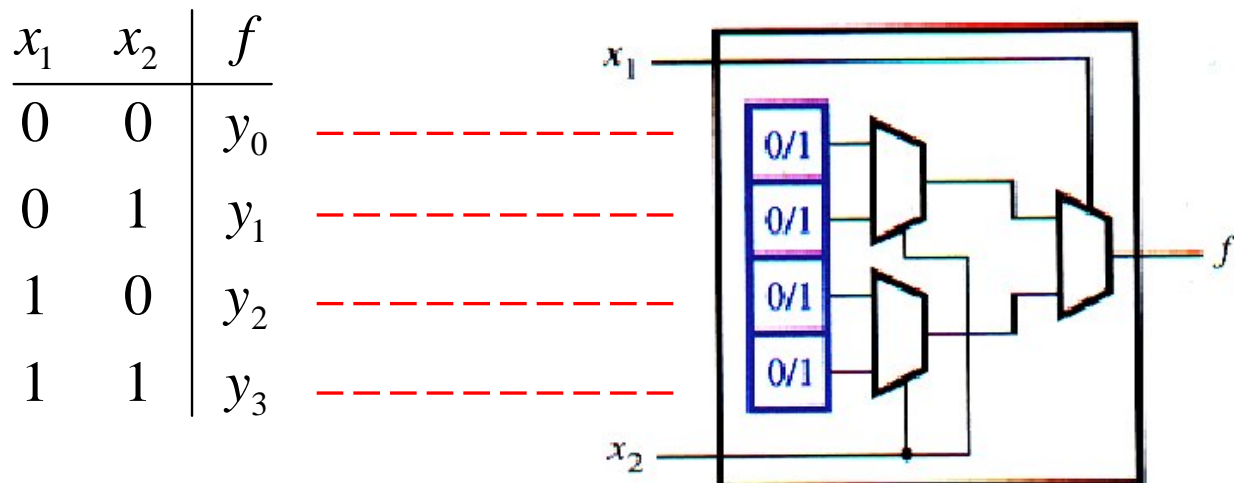
# Programirljivo polje logičkih blokova

- struktura logičkog bloka s 2 ulaza

2 ulaza  $\rightarrow$  2-ulazni MUX:  $f = f(x_1, x_2)$

$y_i$ : sadržaj memorijskih ćelija

$$\begin{aligned} f &= \overline{x_1} \cdot (\overline{x_2} \cdot y_0 + x_2 \cdot y_1) + x_1 \cdot (\overline{x_2} \cdot y_2 + x_2 \cdot y_3) \\ &= \overline{x_1} \cdot \overline{x_2} \cdot y_0 + \overline{x_1} \cdot x_2 \cdot y_1 + x_1 \cdot \overline{x_2} \cdot y_2 + x_1 \cdot x_2 \cdot y_3 \end{aligned}$$





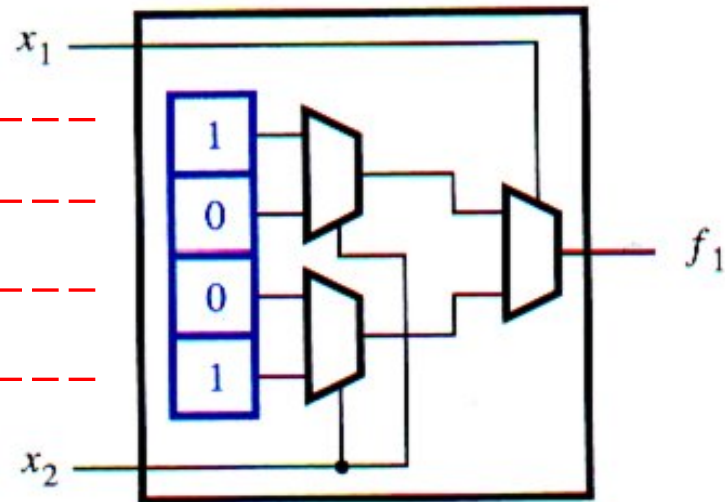
# Programirljivo polje logičkih blokova

*Primjer:* logički blok s 2 ulaza

sadržaj ćelija  $y = \langle 1, 0, 0, 1 \rangle$

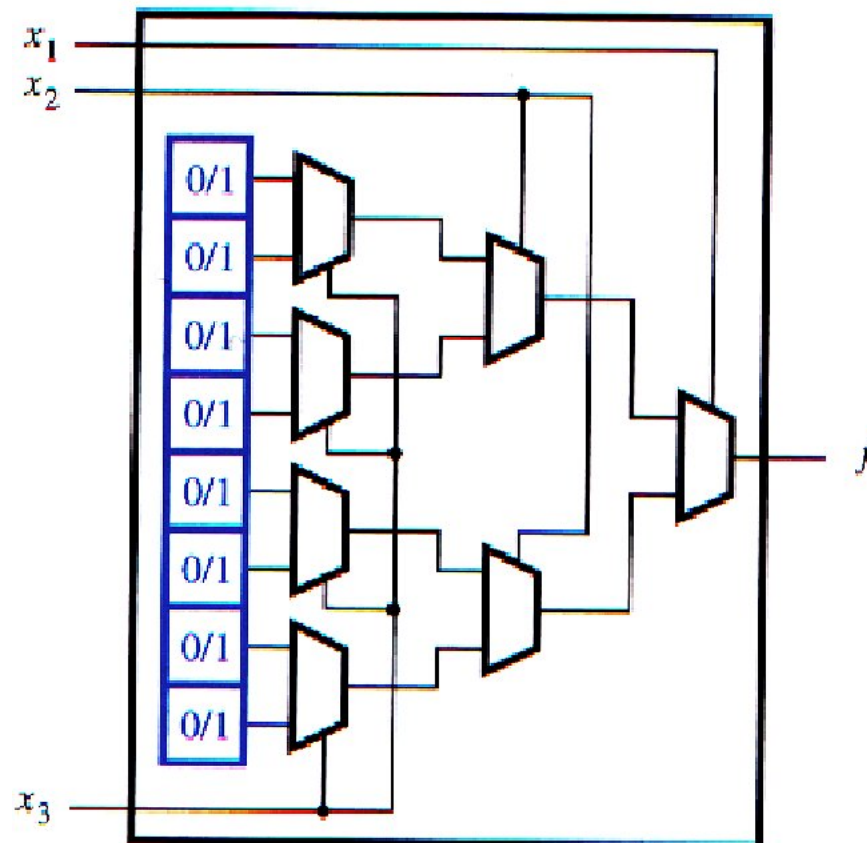
$x_1$	$x_2$	$f$
0	0	1
0	1	0
1	0	0
1	1	1

$$f_1 = \overline{x_1} \cdot \overline{x_2} + x_1 \cdot x_2$$



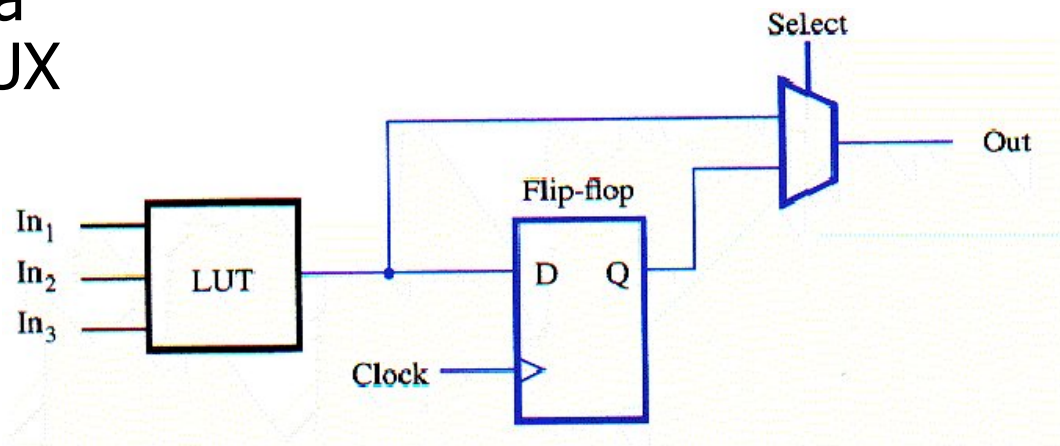
# Programirljivo polje logičkih blokova

- struktura logičkog bloka s 3 ulaza  
~ 8 memorijskih ćelija



# Programirljivo polje logičkih blokova

- "dodatna logika" u logičkom bloku  
~ makroćelije:
  - element za pamćenje  
(D bistabil: memorira 1 bit)
  - odabir izlaza  
~ izlazni MUX



$$Out = \overline{Select} \cdot f + Select \cdot f^{n-1}$$
$$f = f(In_1, In_2, In_3)$$



# Programirljivo polje logičkih blokova

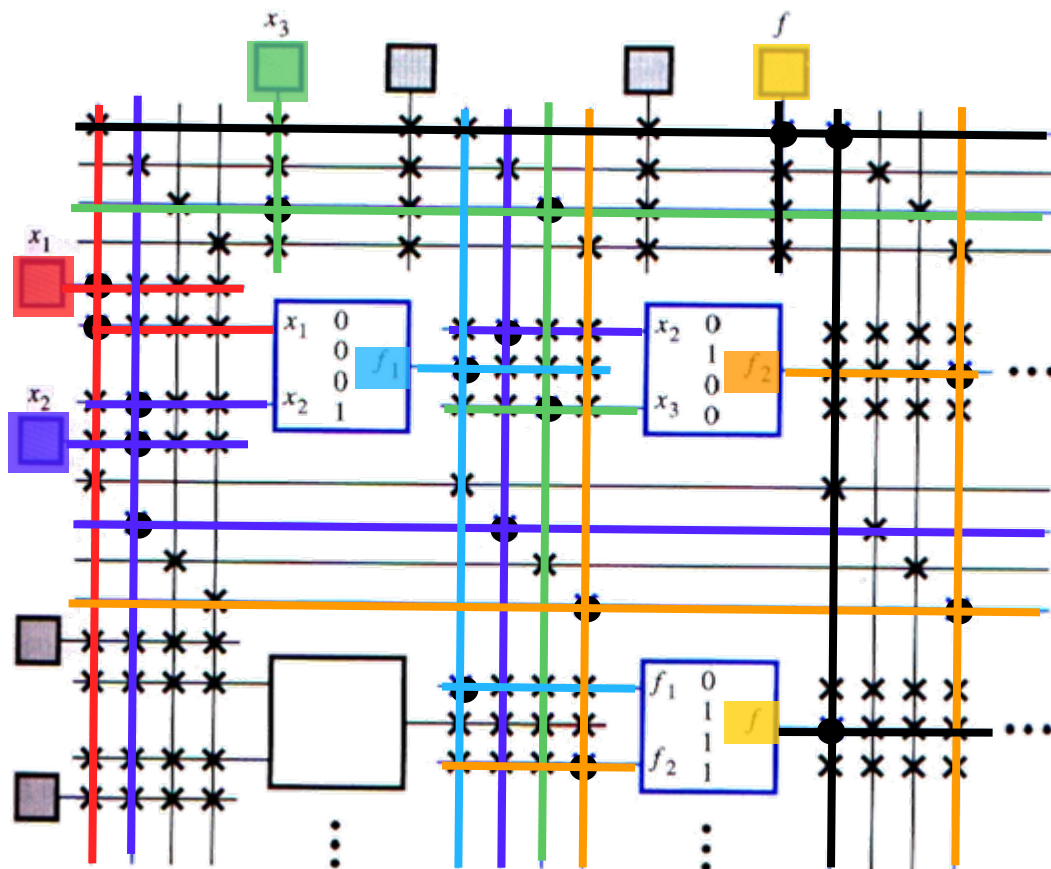
---

- *programiranje* FPGA  
~ također ISP:
  - memorijske ćelije LUT:  
~ *nestalne* (engl. volatile):  
(EA)ROM za pohranjivanje sadržaja LUT
  - automatsko *punjenje* (engl. loading)  
prilikom uključivanja uređaja

# Programirljivo polje logičkih blokova

*Primjer:* (dio) programiranog FPGA

$$f(x_1, x_2, x_3) = ?$$



$$f_1 = x_1 \cdot x_2$$

$$f_2 = \overline{x_2} \cdot x_3$$

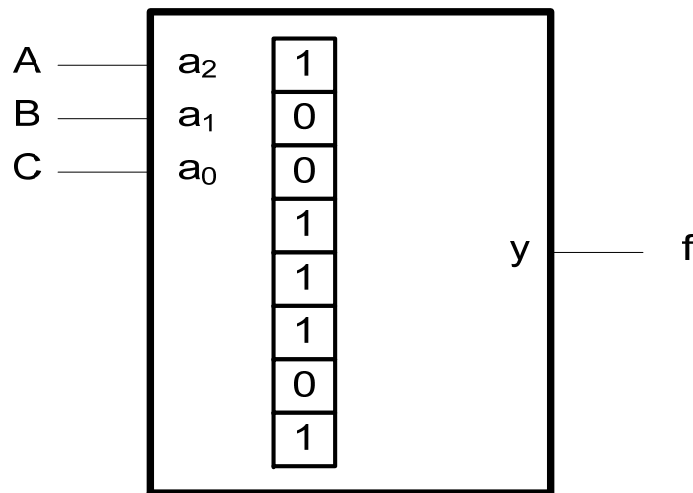
$$f = f_1 + f_2$$

$$= x_1 \cdot x_2 + \overline{x_2} \cdot x_3$$

# Programirljivo polje logičkih blokova

*Primjer:*  $f(A, B, C) = B \cdot C + \overline{B} \cdot \overline{C} + A \cdot C$

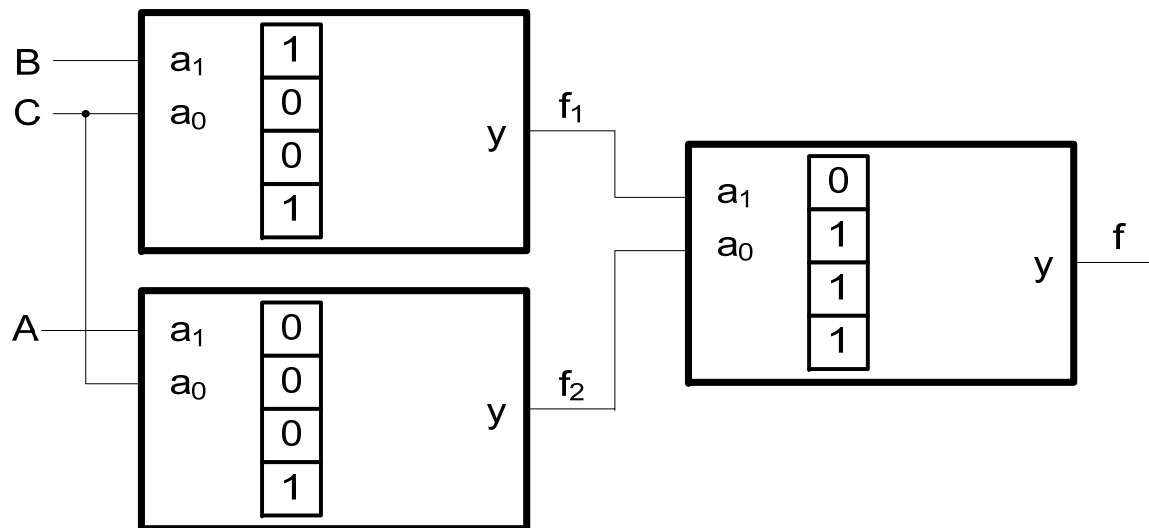
- ostvarenje LUTom s 3 ulaza  
~ standardno "programiranje" multipleksora



# Programirljivo polje logičkih blokova

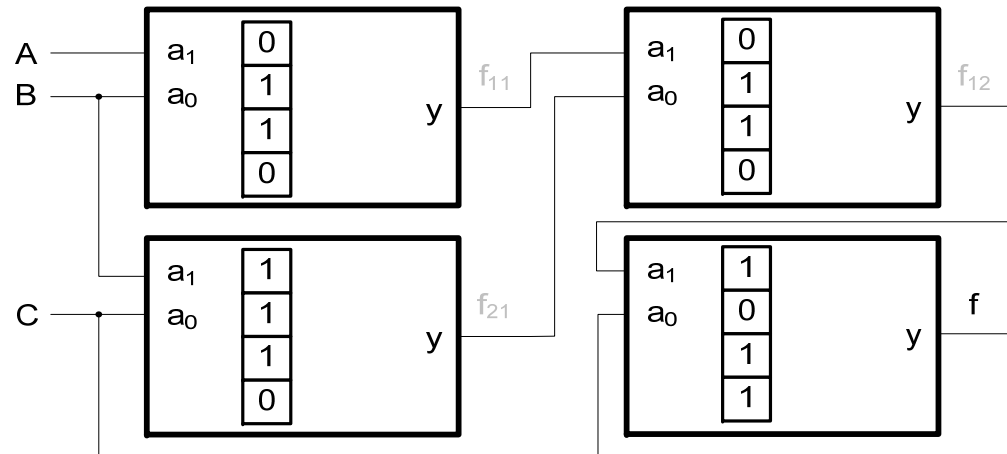
- ostvarenje LUTovima s 2 ulaza  
~ kaskadiranje multipleksora

$$\begin{aligned} f(A, B, C) &= B \cdot C + \overline{B} \cdot \overline{C} + A \cdot C \\ &= (B \cdot C + \overline{B} \cdot \overline{C}) + A \cdot C \\ &= f_1 + f_2 \end{aligned}$$



# Programirljivo polje logičkih blokova

*Primjer:*  $f = ?$



$$f_{11}(A, B) = A \oplus B$$

$$f_{21}(B, C) = \overline{BC}$$

$$f_{12}(f_{11}, f_{21}) = f_{11} \oplus f_{21} = A \oplus B \oplus \overline{BC}$$

$$f(f_{12}, C) = f_{12} + \overline{C} = A \oplus B \oplus \overline{BC} + \overline{C} = A \oplus B \oplus (\overline{B} + \overline{C}) + \overline{C}$$

$$f = A + BC + \overline{BC}$$



U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 7: Standardni kombinacijski moduli.

- programirljivo logičko polje: str. 276-281
- poluprogramirljivo logičko polje: str. 281-282
- složeni programirljivi moduli: str. 283-284
- programirljivo polje logičkih blokova: str. 285-286



# Zadaci za vježbu (1)

---

U. Peruško, V. Glavinić: *Digitalni sustavi*, Poglavlje 7:  
Standardni kombinacijski moduli.

- programirljivo logičko polje: 7.2, 7.30-7.32, 7.34
- programirljivo polje logičkih blokova: 7.33



## Zadaci za vježbu (2)

---

M. Čupić: *Digitalna elektronika i digitalna logika. Zbirka riješenih zadataka*, Cjelina 6: Standardni programirljivi moduli.

- programirljivo logičko polje
  - riješeni zadaci: 6.1-6.4, 6.7, 6.13
  - zadaci za vježbu: 1-3
- poluprogramirljivo logičko polje
  - riješeni zadaci: 6.5-6.6
- programirljivo polje logičkih blokova
  - riješeni zadaci: 6.8-6.11