

Vezja FPGA in IP

Digitalni elektronski sistemi 2024/25

Andrej Trost



UNIVERZA
V LJUBLJANI

FE

Fakulteta
za elektrotehniko

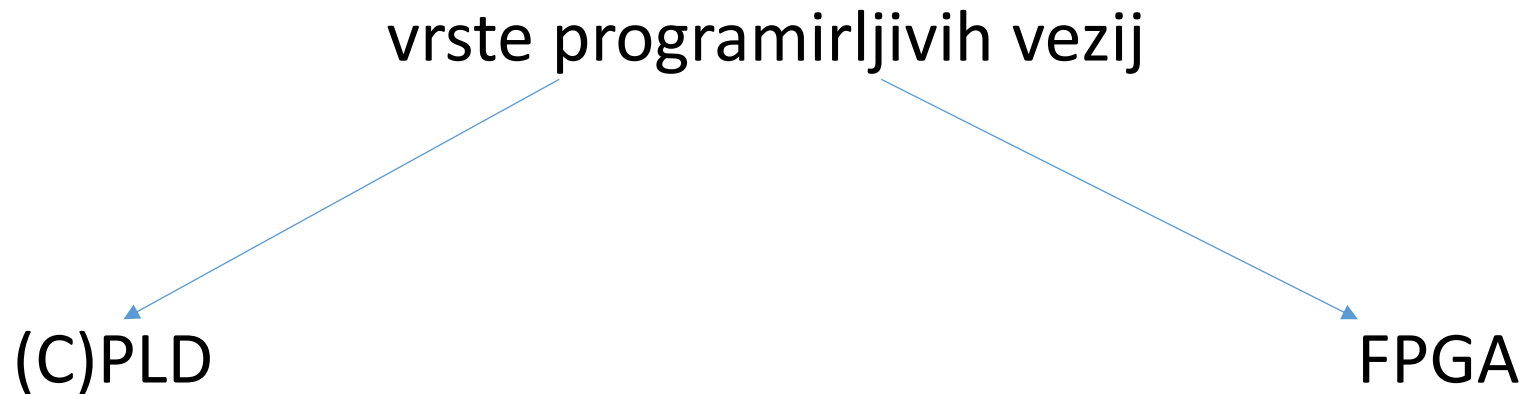
Programirljive naprave

- digitalna integrirana vezja, ki jih nastavimo za želeno funkcijo

Zakaj?

- hitrejši in cenejši razvoj, možnost popravkov

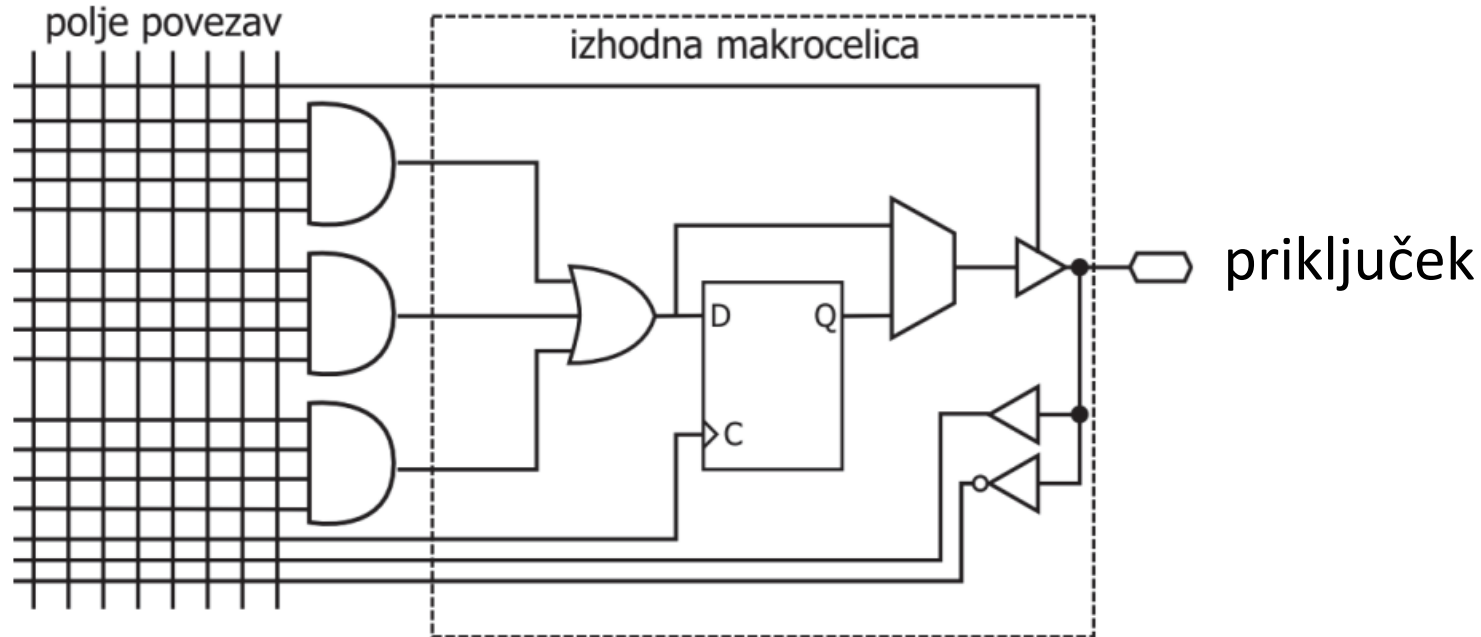
Vrste, zgradba, delovanje in uporaba



Programirljive logične naprave (PLD)

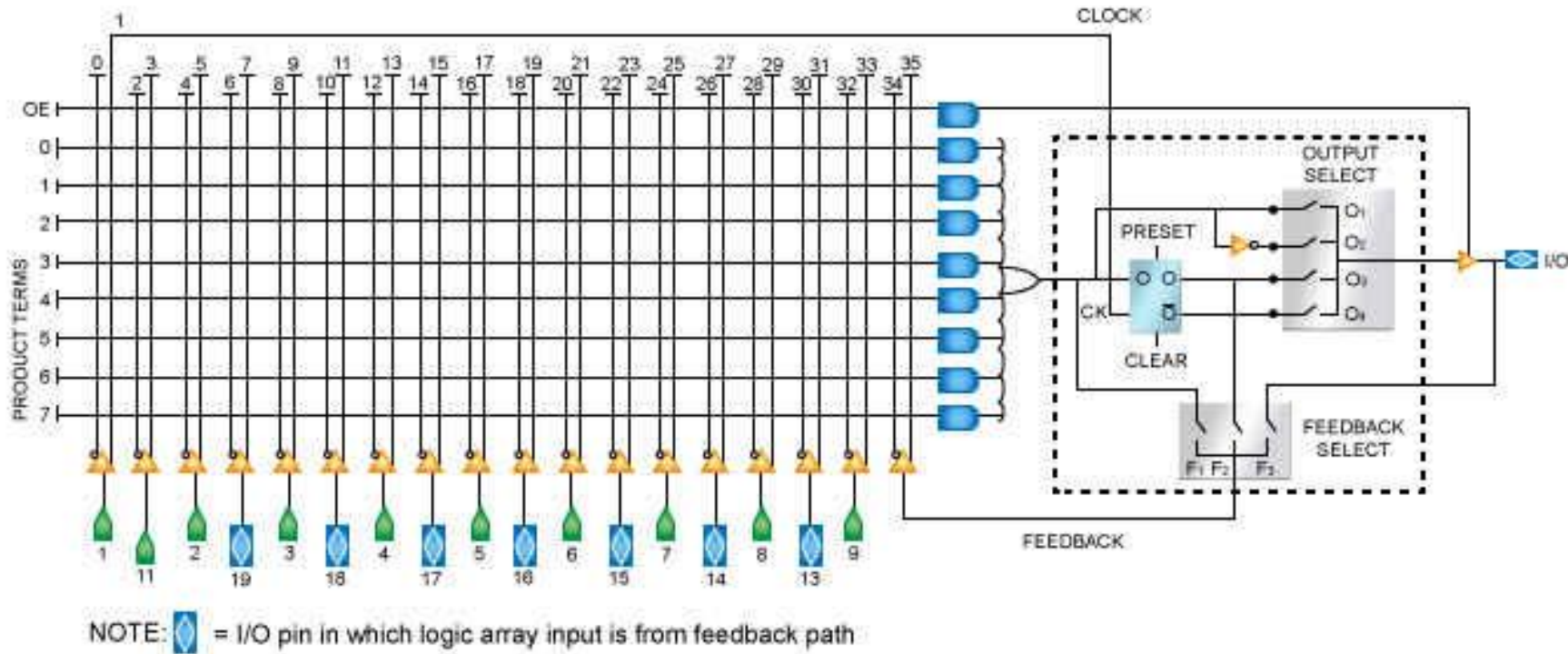
- vsebujejo logična vrata, povezave določimo med programiranjem
- AND-OR matrika, celice s flip-flopi in povratne povezave

kombinacijska in sekvenčna logika



Programirljive logične naprave (PLD)

- Altera (sedaj Intel) EP300, leta 1984

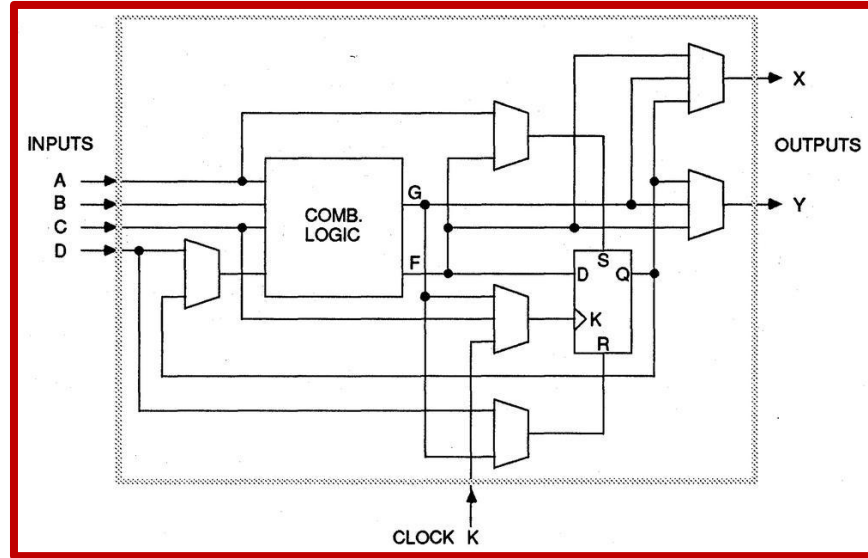
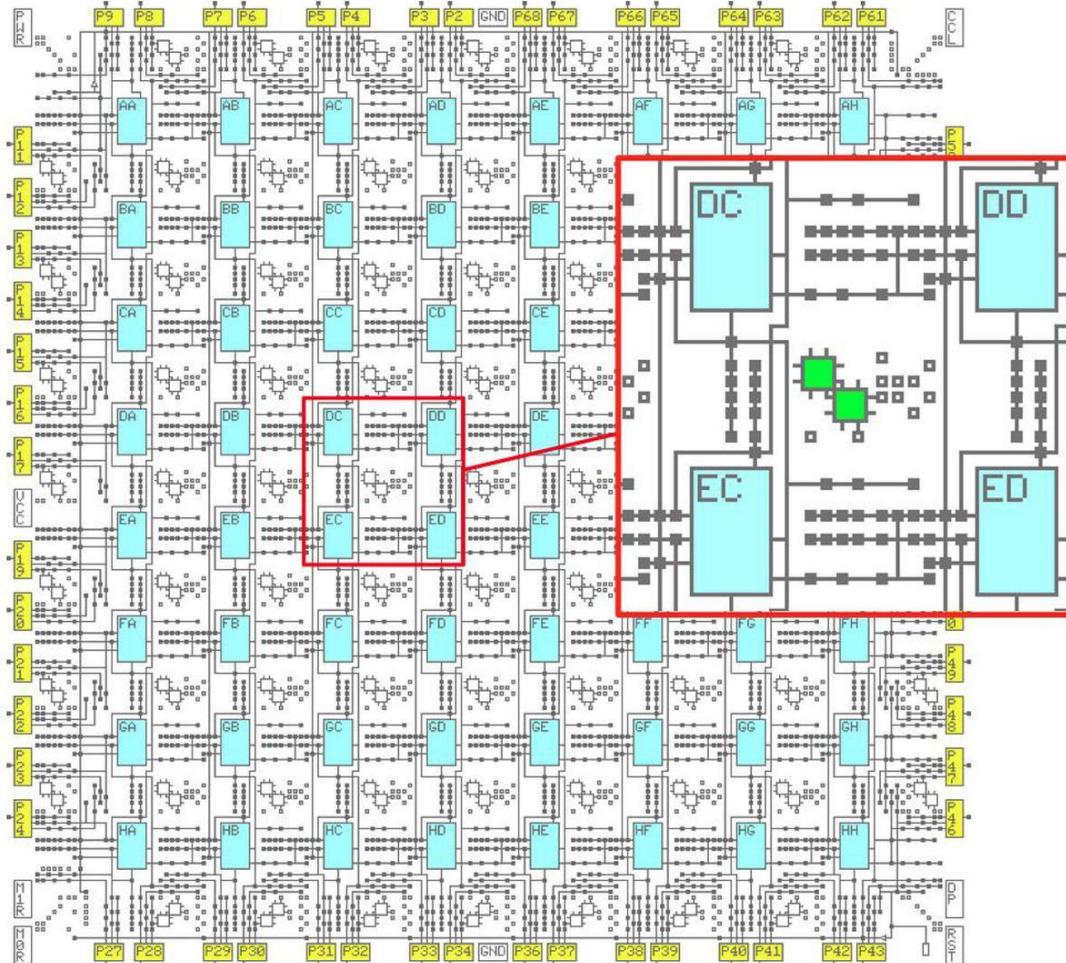


This diagram shows one of the eight macrocells within the EP300/EP310.

- minimizirani Boolovi izrazi
- UV EEPROM, brisanje s svetlobo
- Npr. 16-bitni ali 7-segm. dekoder, avtomat z 8 stanji...

Programirljiva matrika (FPGA)

- Xilinx (sedaj AMD), leta 1986



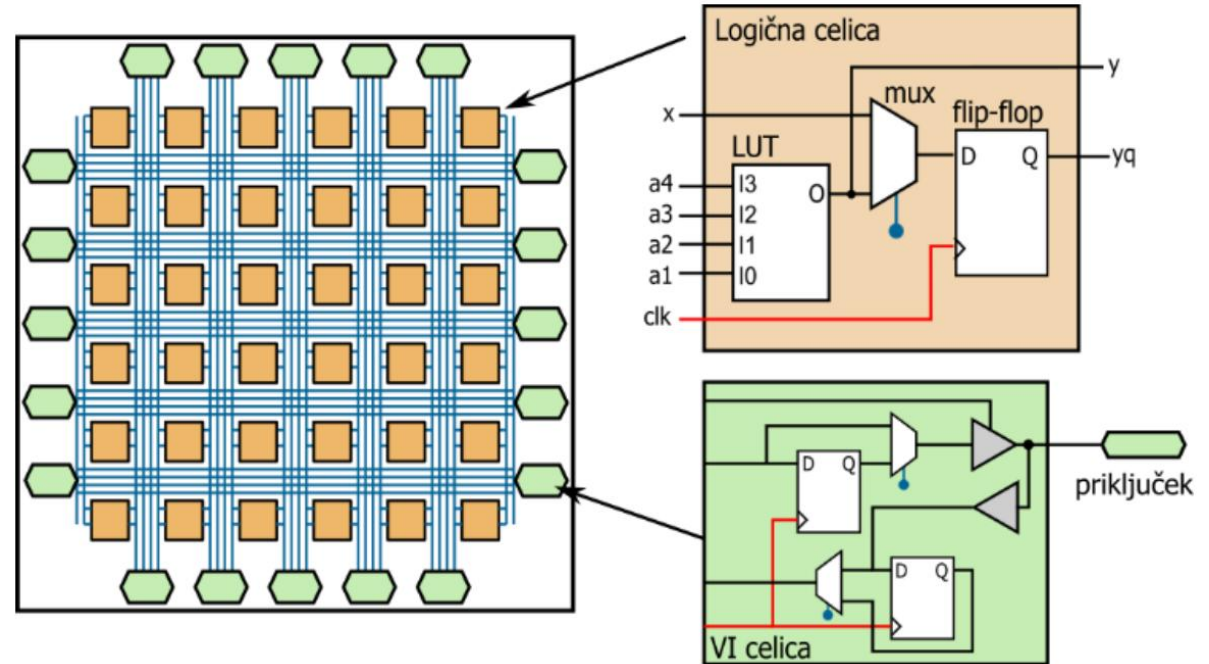
- logične celice s tabelami in flip-flopi
- programirljive povezave
- vhodno/izhodne celice



Zgradba in delovanje FPGA

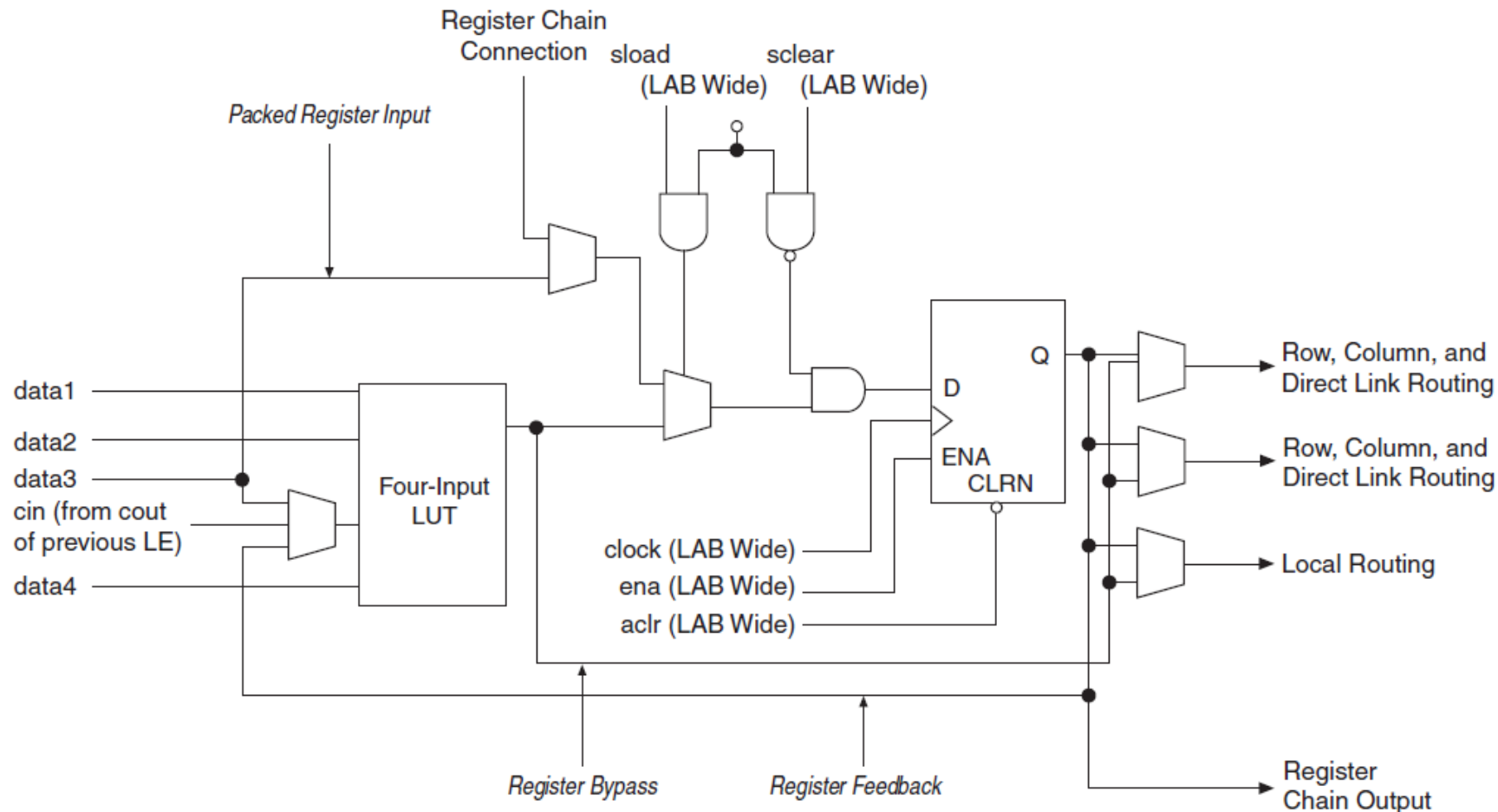
Field Programmable Gate Array

- zmogljiva vezja, veliko celic in zelo veliko povezav
- razdelitev na družine vezij posameznih proizvajalcev
 - znotraj družine enaka zgradba, različna velikost matrice, ohišje in hitrostni razred

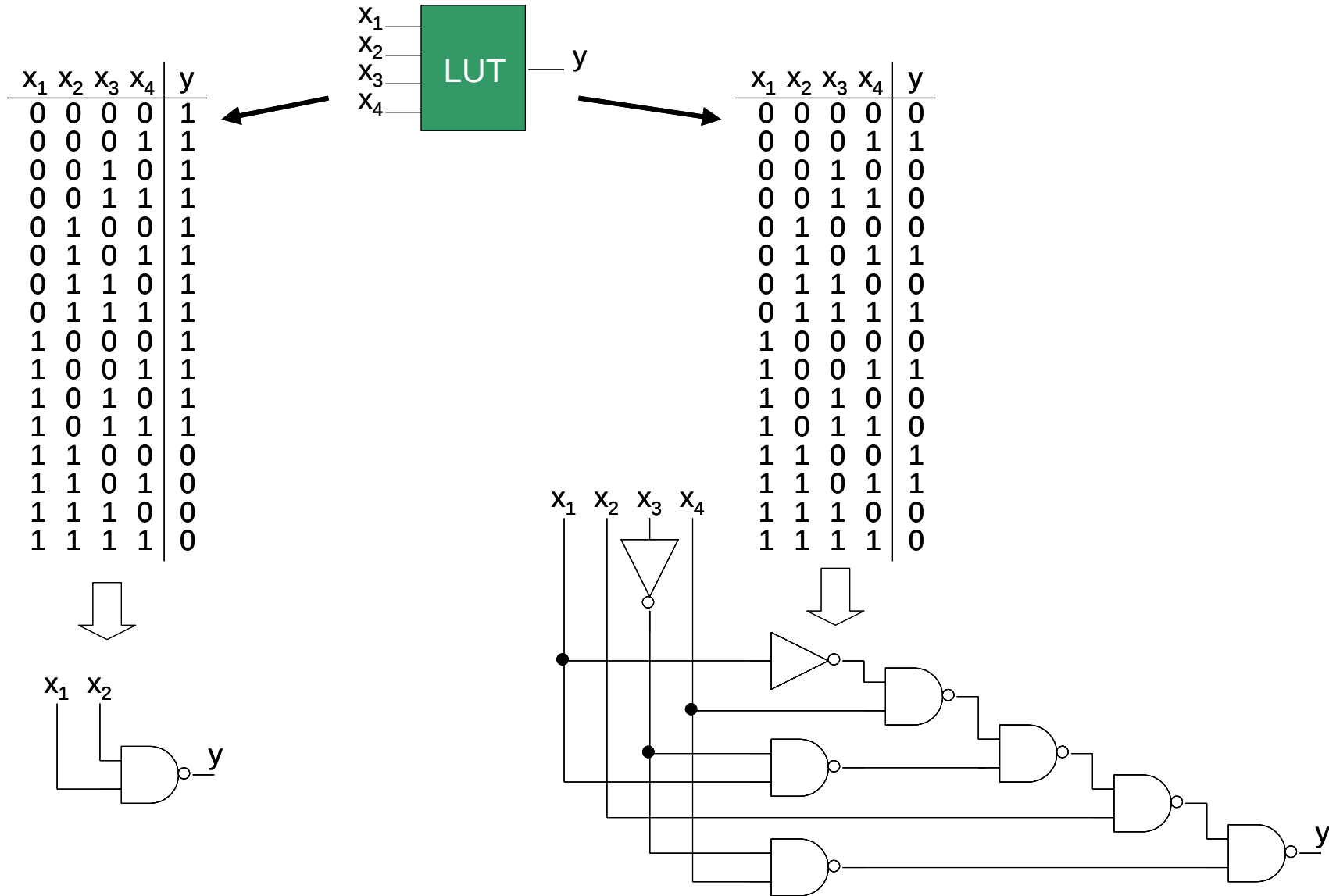


Zgradba logične celice

npr. družina Altera Cyclone IV (FPGA na naši razvojni plošči)
logična celica: LE (**logic element**)

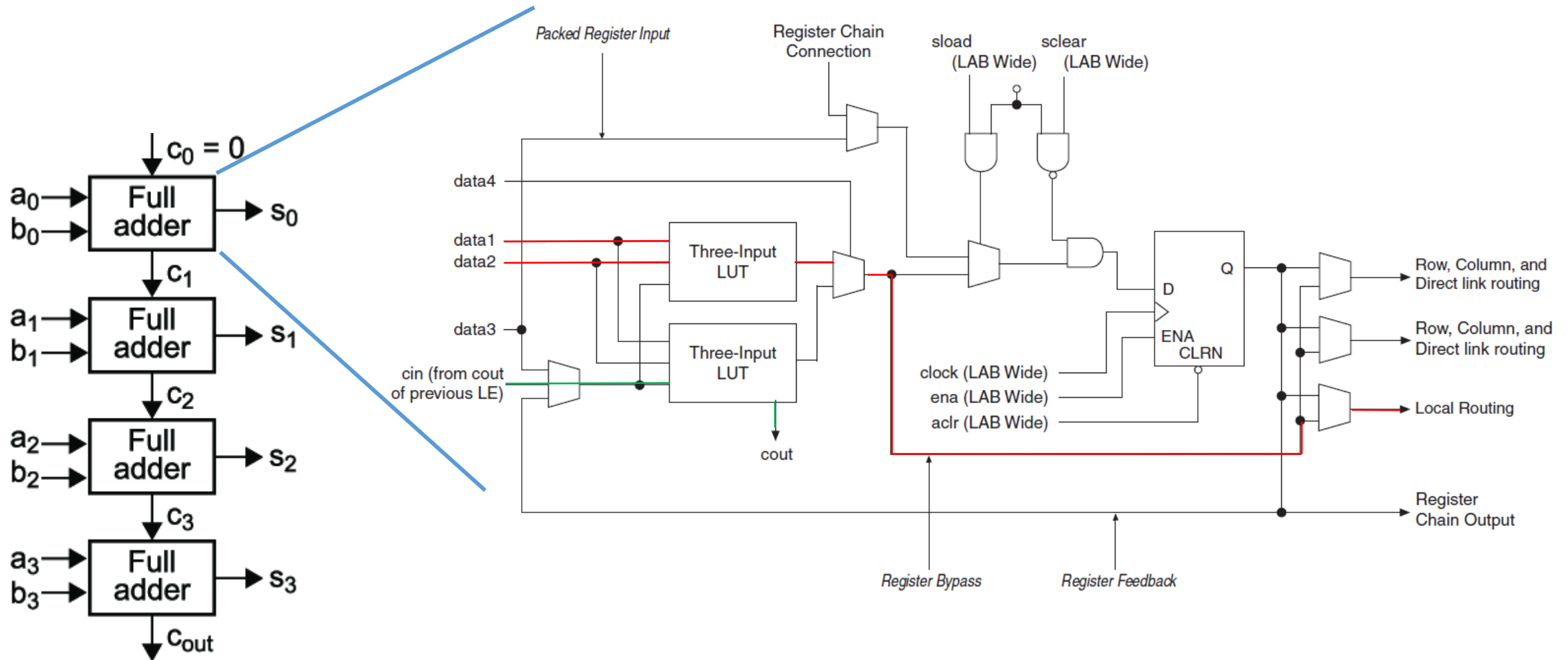


Delovanje vpogledne tabele



Logična celica s prenosno logiko

- prenos (**carry**) za seštevavnike, števnice,... povezan med sosednjimi celicami

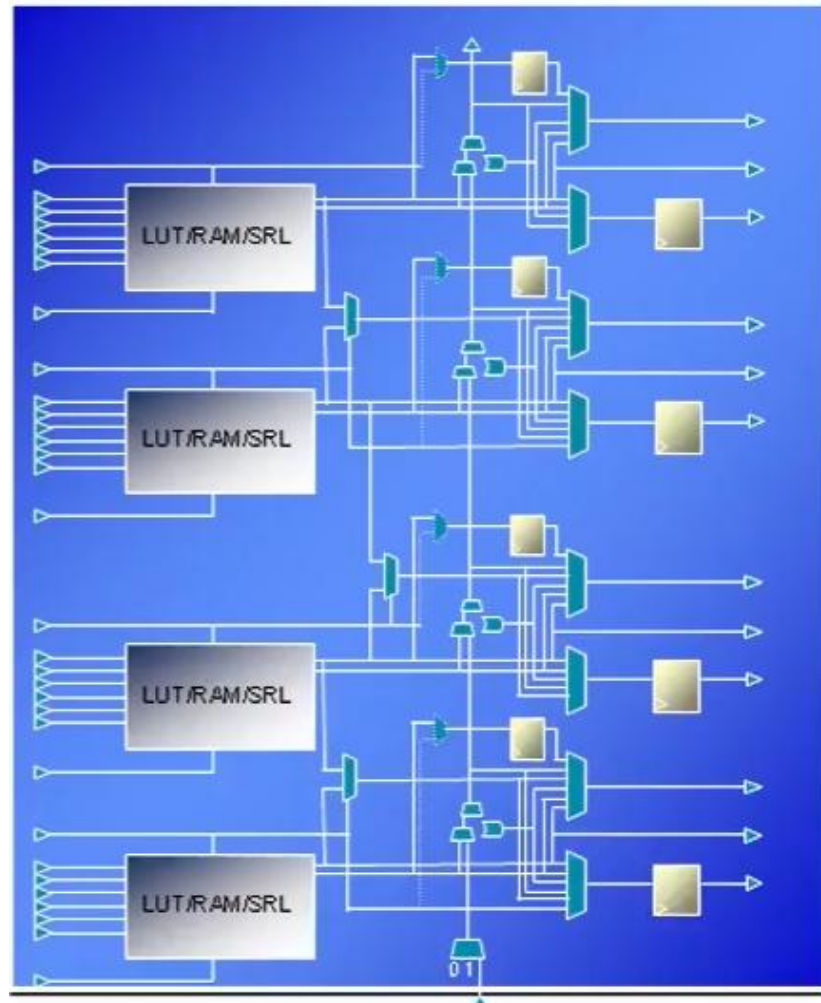


Alternativne funkcije vpoglednih tabel

- LUT lahko uporabimo kot majhen pomnilnik ali pomikalni register

Npr. logične celice v 7. družini FPGA Xilinx (AMD)

- 6-vhodne LUT

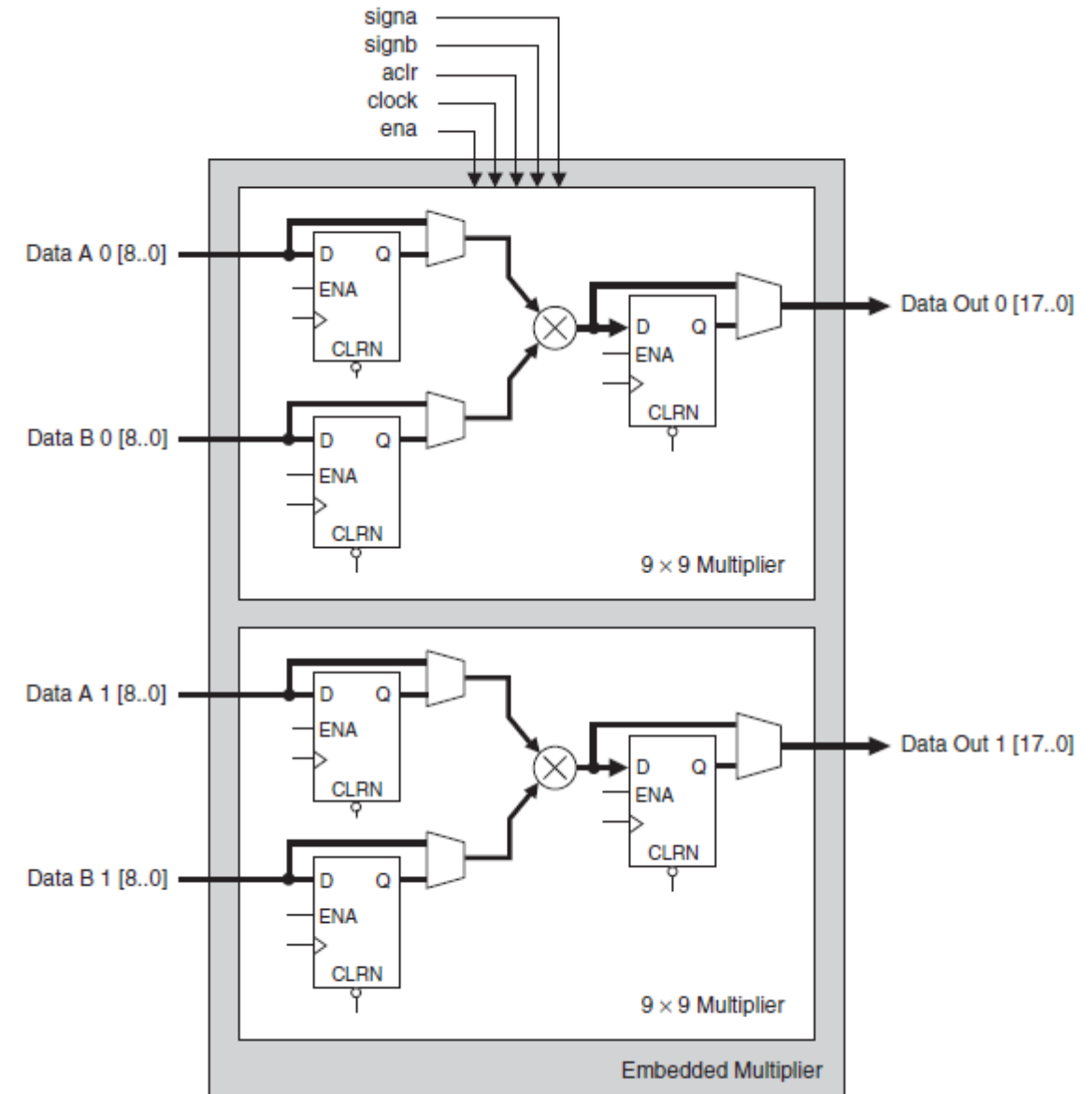


Dodatni gradniki v sodobnih vezjih FPGA

RAM - pomnilniki

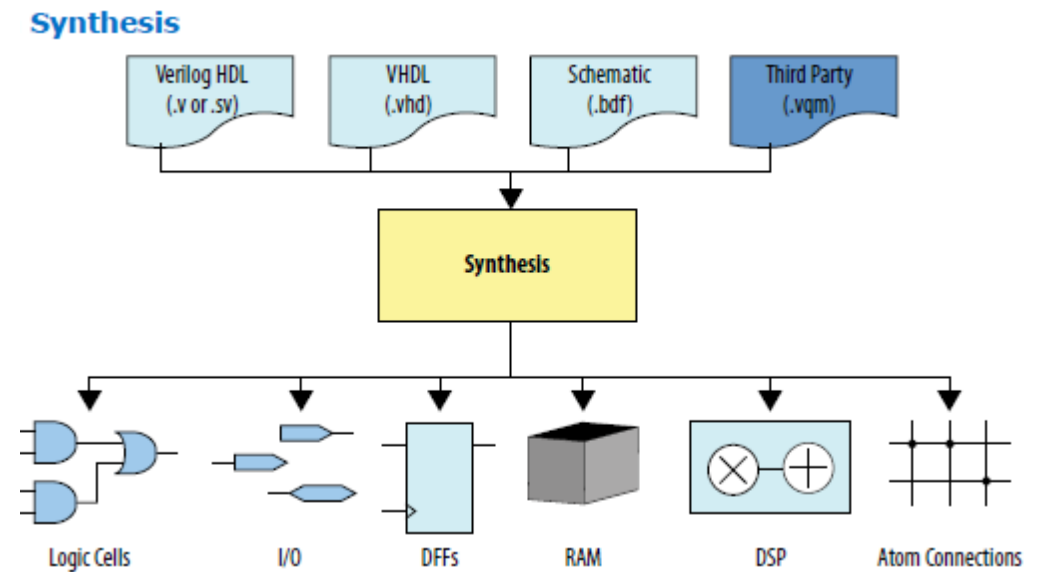
Cyclone IV: M9K

- ▶ 8192 x 1 bit,
- ▶ 4092 x 2 ...
- ▶ 256 x 36 bitov

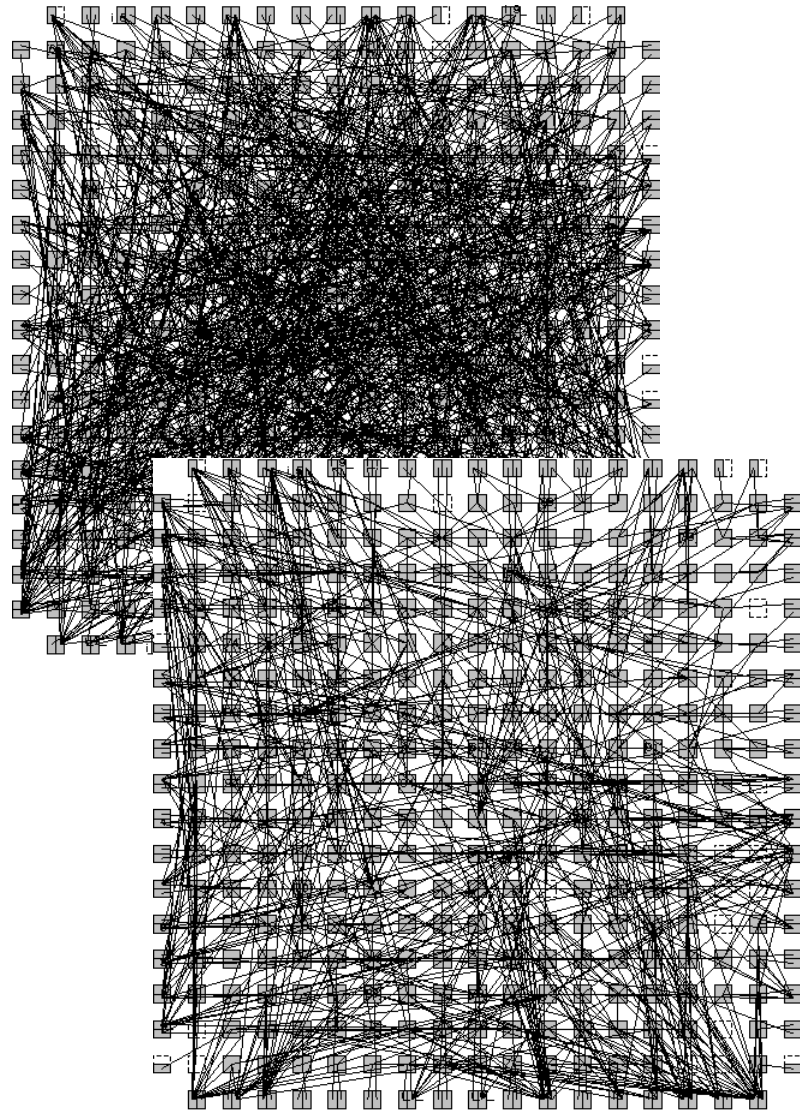


Prevajanje vezja

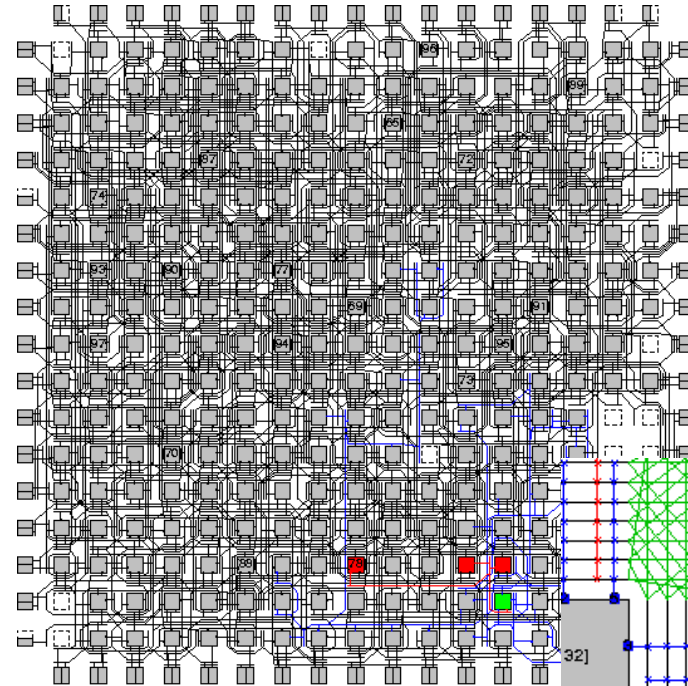
✓	▼ ▶ Compile Design
	> ▶ Analysis & Synthesis
	> ▶ Fitter (Place & Route)
✓	> ▶ Assembler (Generate programming files)
✓	> ▶ Timing Analysis
✓	> ▶ EDA Netlist Writer



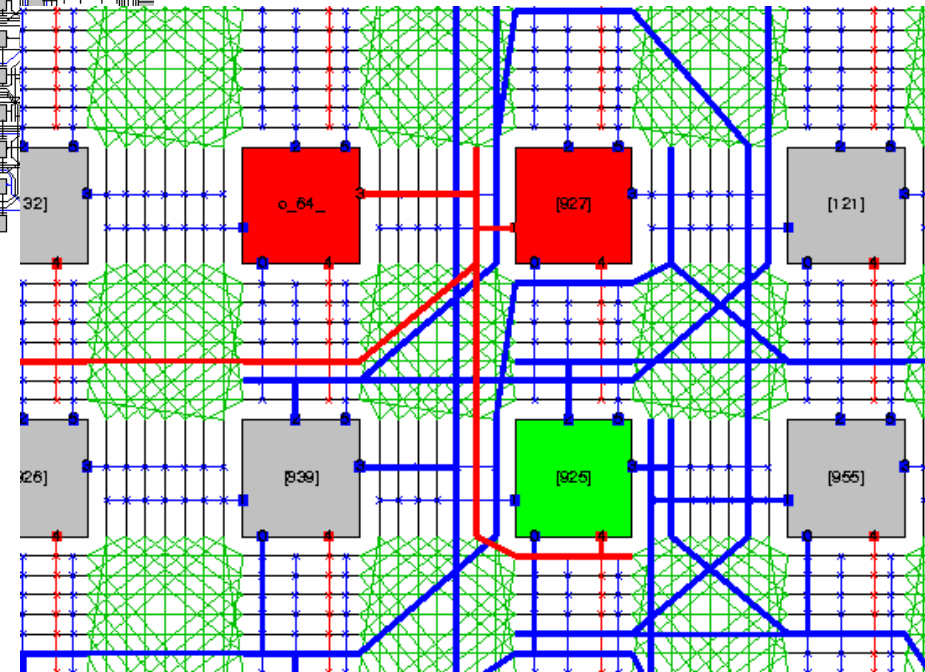
Place & Route

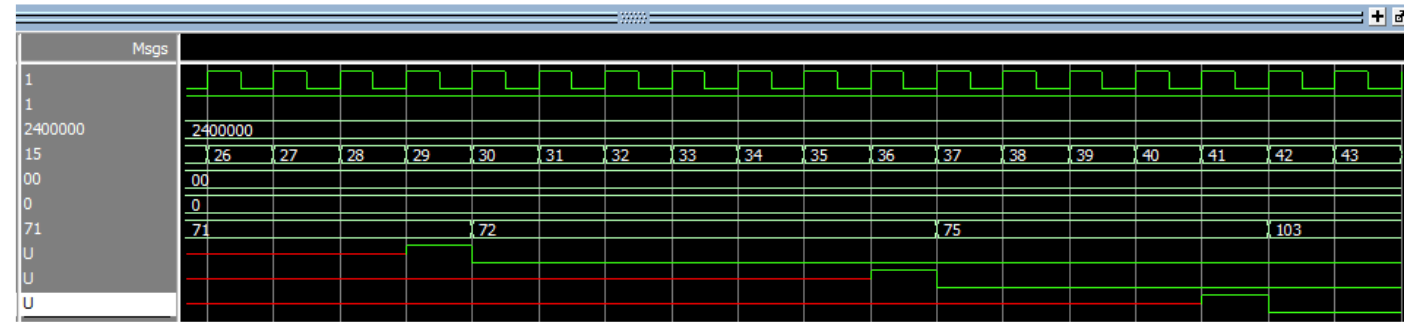
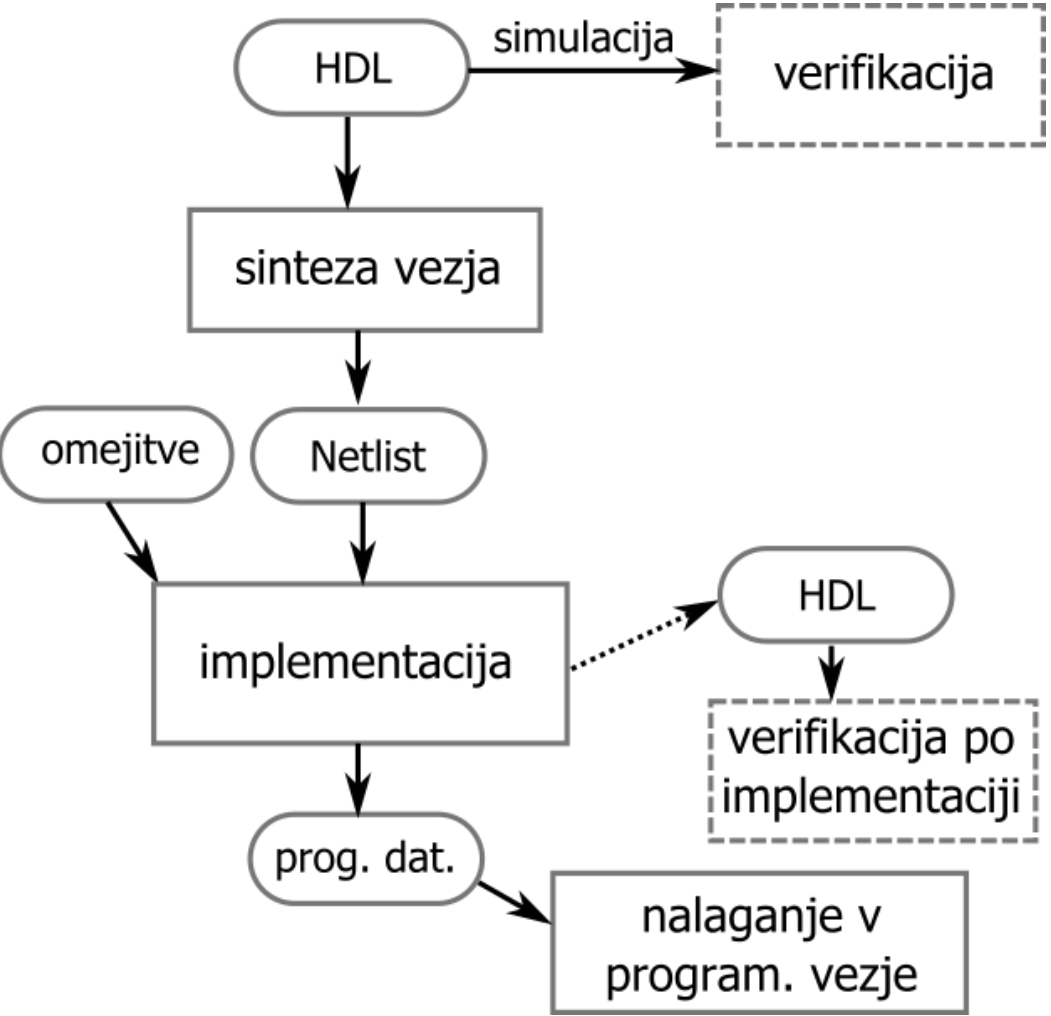


Final Placement. Cost: 28.5384. Channel Factor: 100



Routing succeeded with a channel width factor of 7.

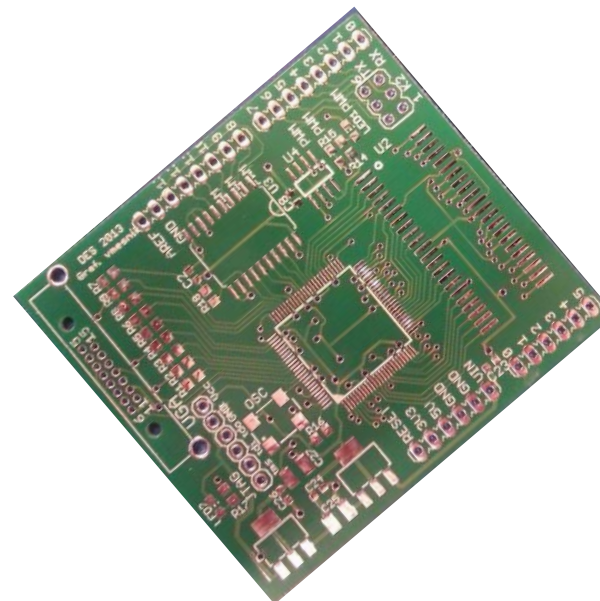
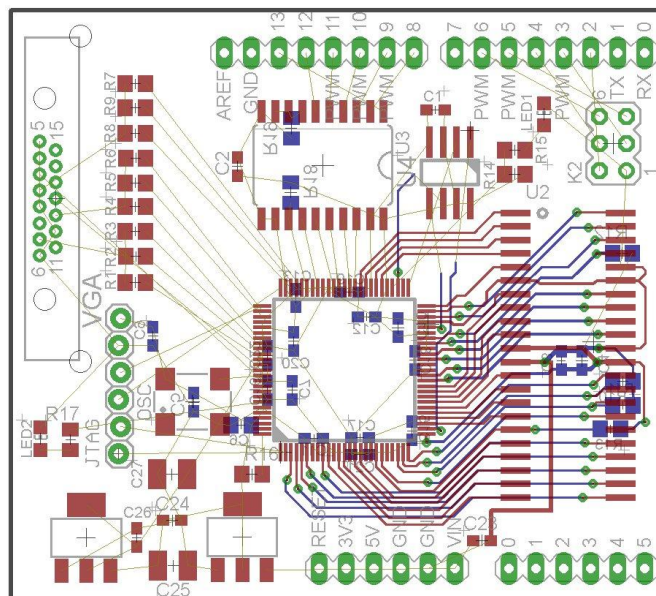
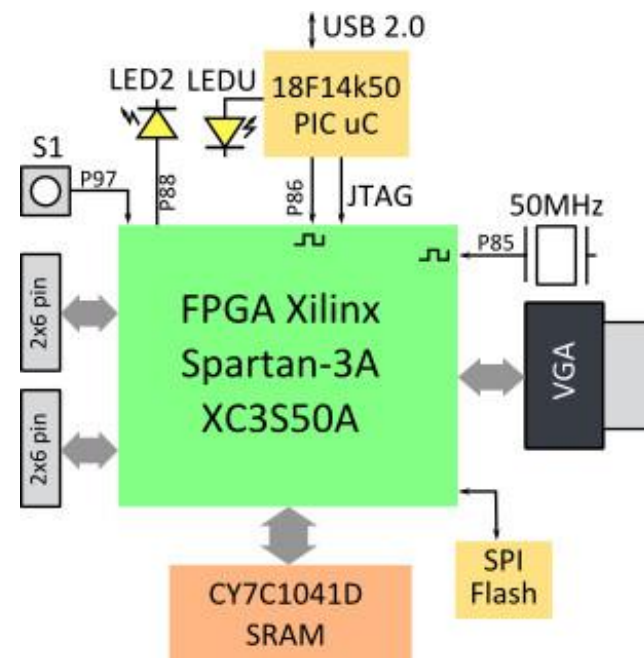


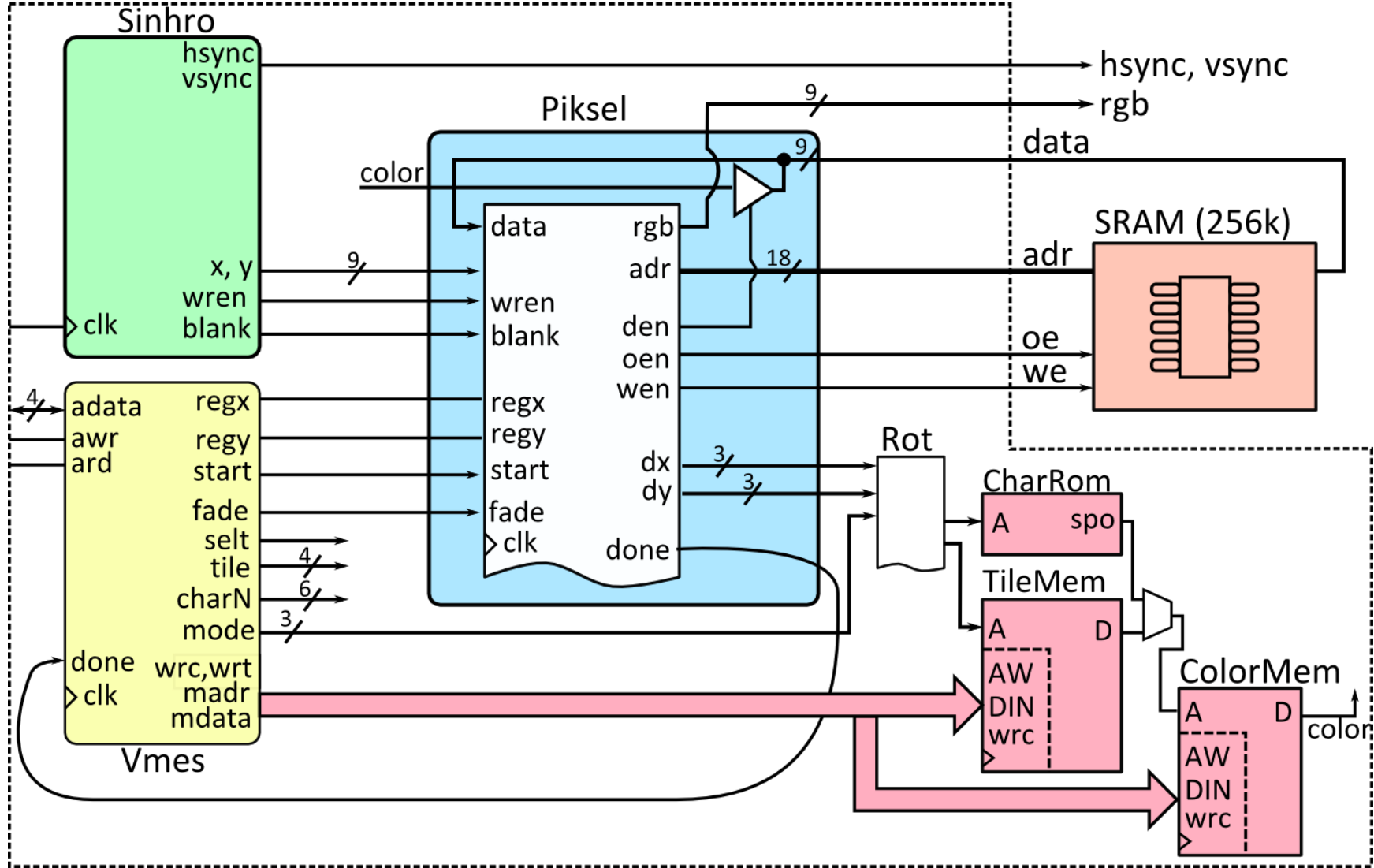


Področja uporabe FPGA

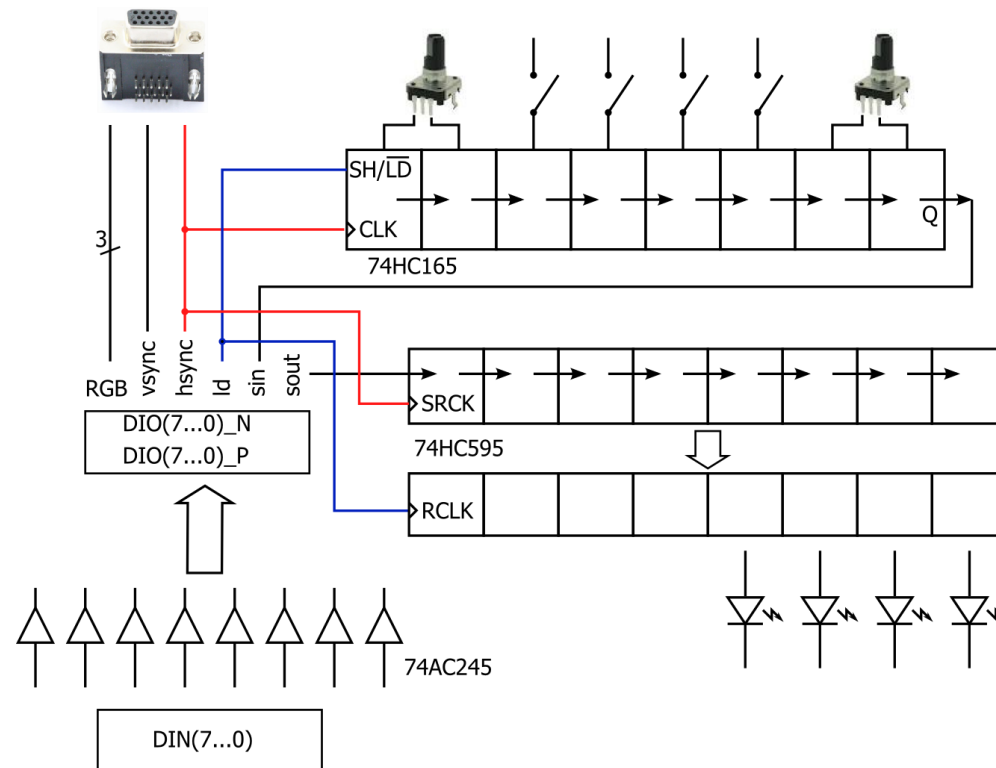
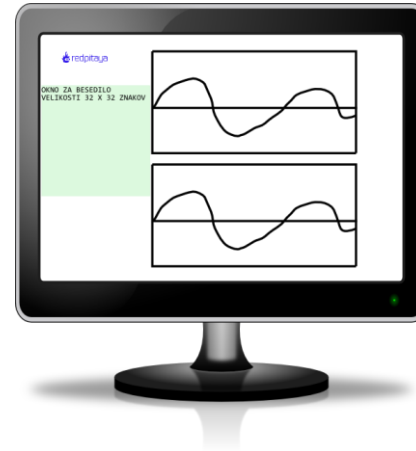
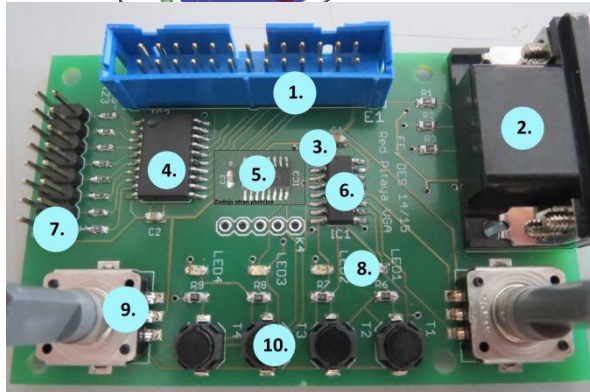
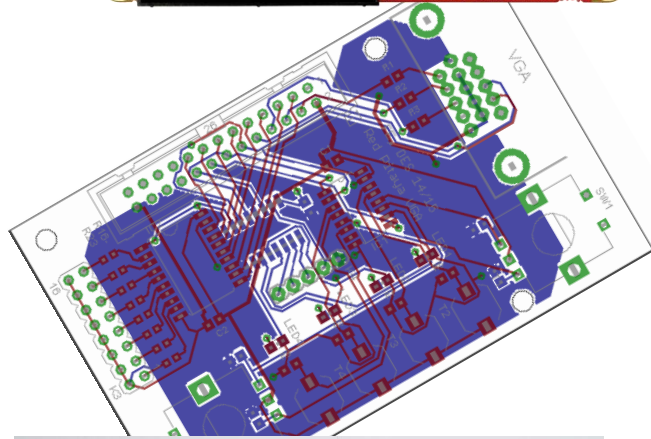
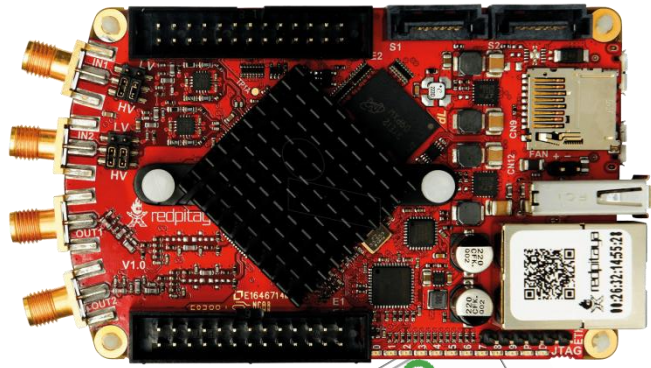
- telekomunikacije
 - paketna obdelava, 5G, radiodifuzija
- industrijska
 - strojni vid, krmilniki, robotika, prototipi in posnemovalniki
- avtomobilska
 - avtonomni asistenčni sistemi, infotainment, fuzija senzorjev
- uporabniška elektronika, medicinska elektronika, vojaška, letalska, vesoljska tehnika
- podatkovni centri
 - vmesniki, strojno pospeševanje algoritmov

Projekti: VGA Arduino



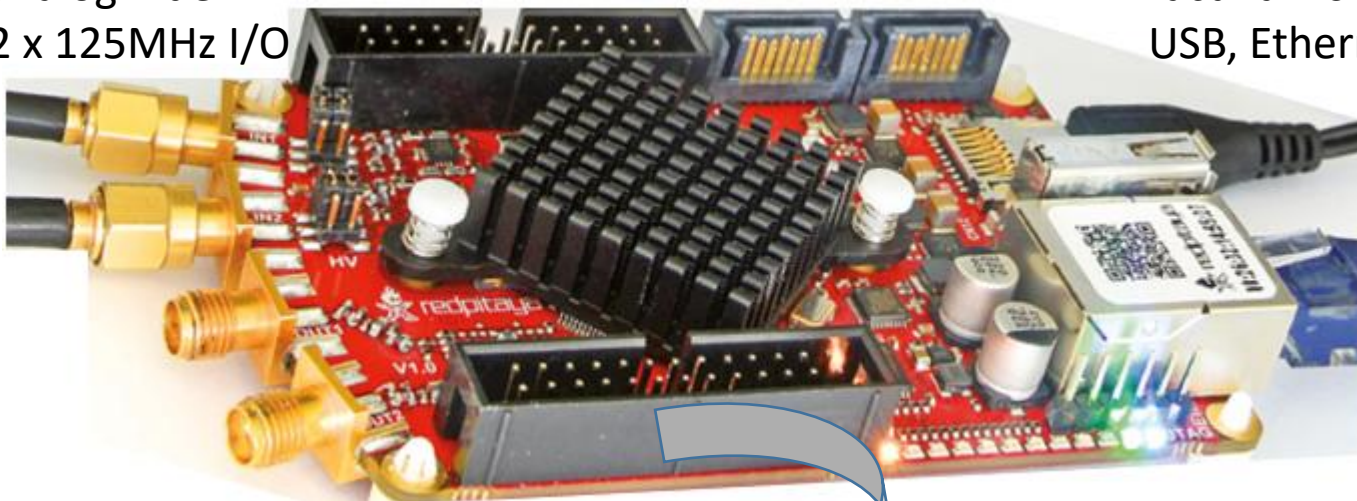


Red Pitaya



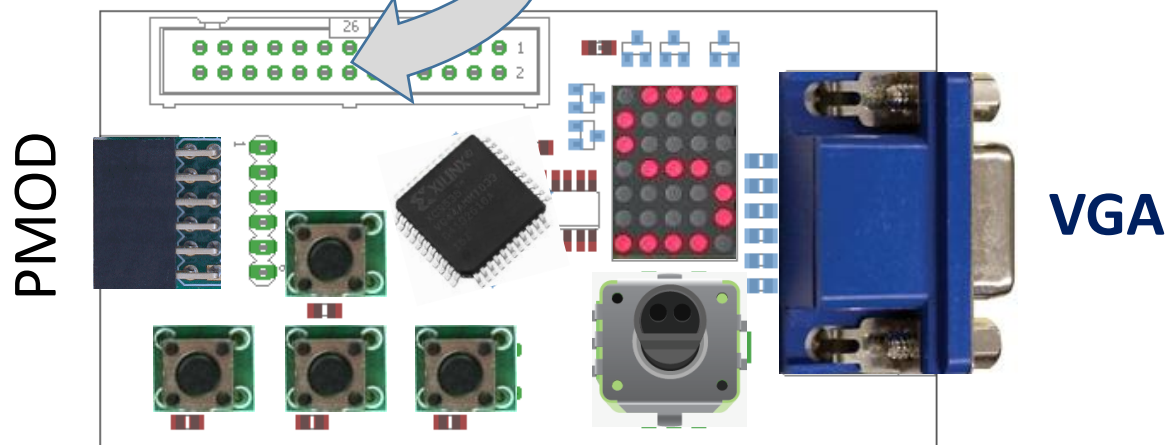
analogni del
2 x 125MHz I/O

računalniški del
USB, Ethernet, SD

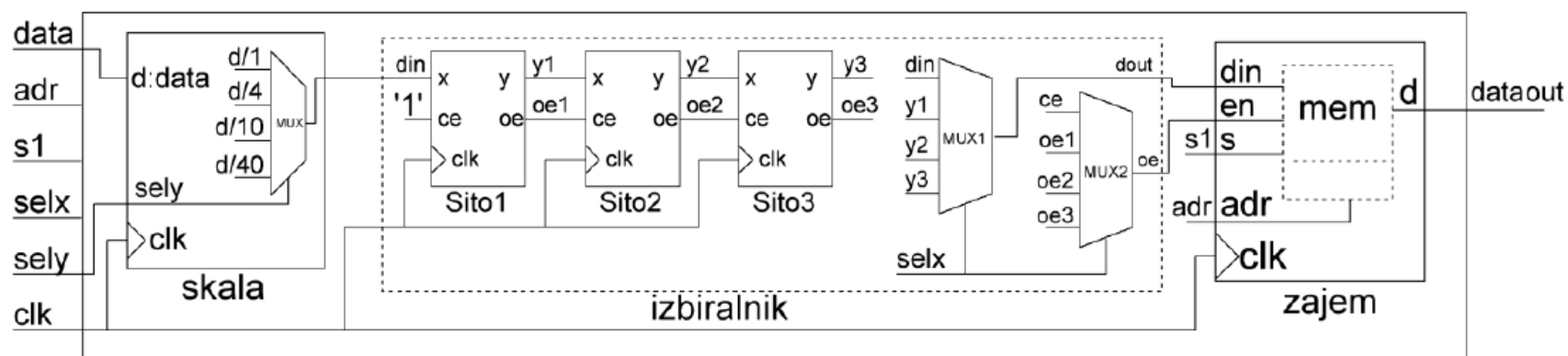
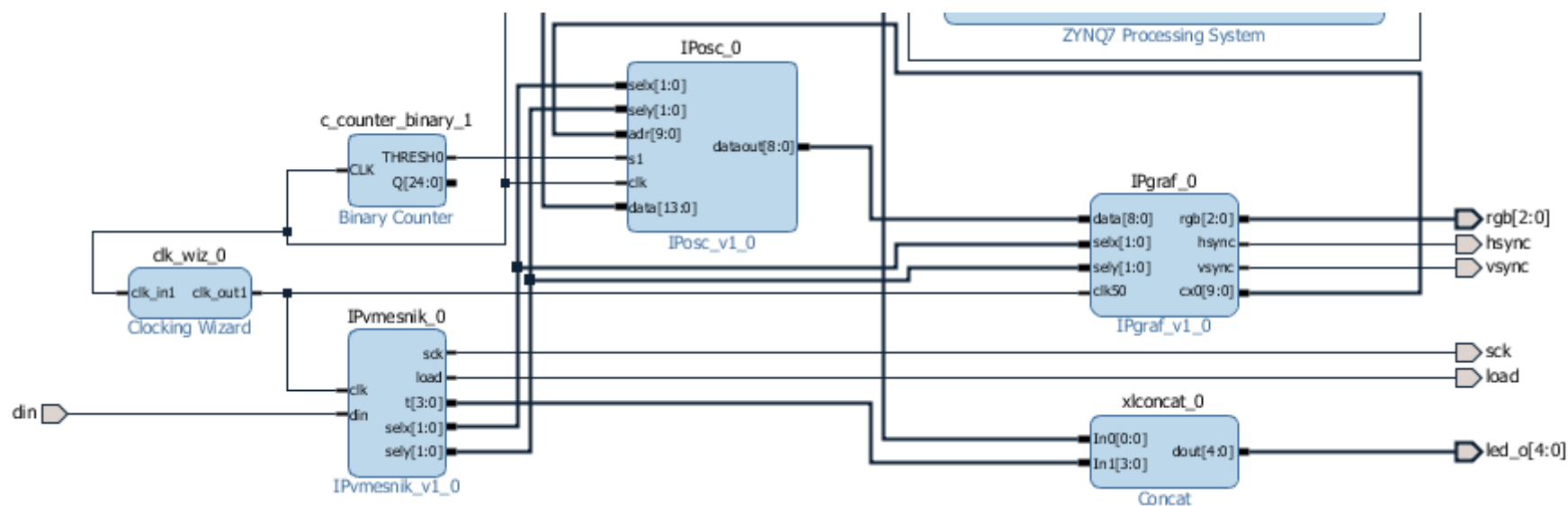


16 I/O

prikaz



tipke in rotacijski kodirnik

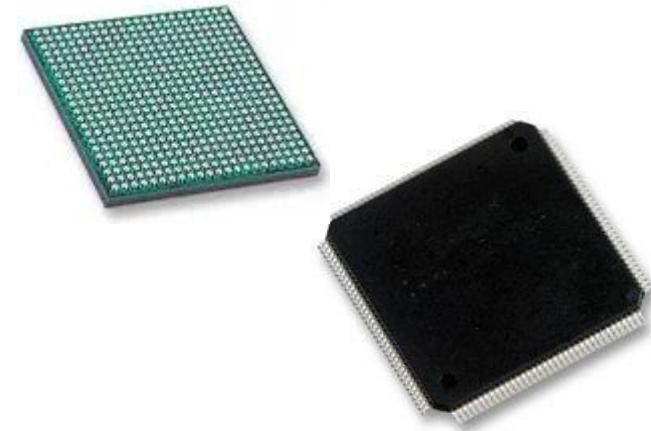


IP-komponenta: IPosc - osciloskop

Lastnosti FPGA

Zmogljivost za izvedbo logičnih vezij:

- število logičnih celic (LC) ali elementov (LE) in flip-flopov
- frekvenca- odvisna od vezja, gradnikov
- število in vrsta zunanjih priključkov, ohišje: **BGA, QFP**
V/I standardi: 1.2-3.3 V LVCMOS, 3.0-3.3 V LVTTL, PCI, SSTL, HSTL, diferencialni SSTL, HSTL, LVDS, RSDS, LVPECL...
- količina hitrega notranjega pomnilnika
- posebni gradniki: DSP, SerDes, CPU...



Primerjava FPGA z namenskimi čipi (ASIC)

- veliko krajši čas razvoja
- ni stroškov priprave proizvodnje, NRE >100.000 €
- cena programirljivosti
 - veliko večja površina vezja (cena), razmerje lahko tudi 40:1
 - zakasnitve programirljivih povezav (3:1)
 - višja poraba energije, 7x slabša zmogljivost/W kot ASIC

Kuon and Rose, *Measuring the Gap Between FPGAs and ASICs*, FPGA 2006

E. Nurvitadhi et al (Intel), *Accelerating Recurrent Neural Networks in Analytics Servers: Comparison of FPGA, CPU, GPU & ASIC*, FPL 2016

Primerjava FPGA z mikrop procesorji

FPGA

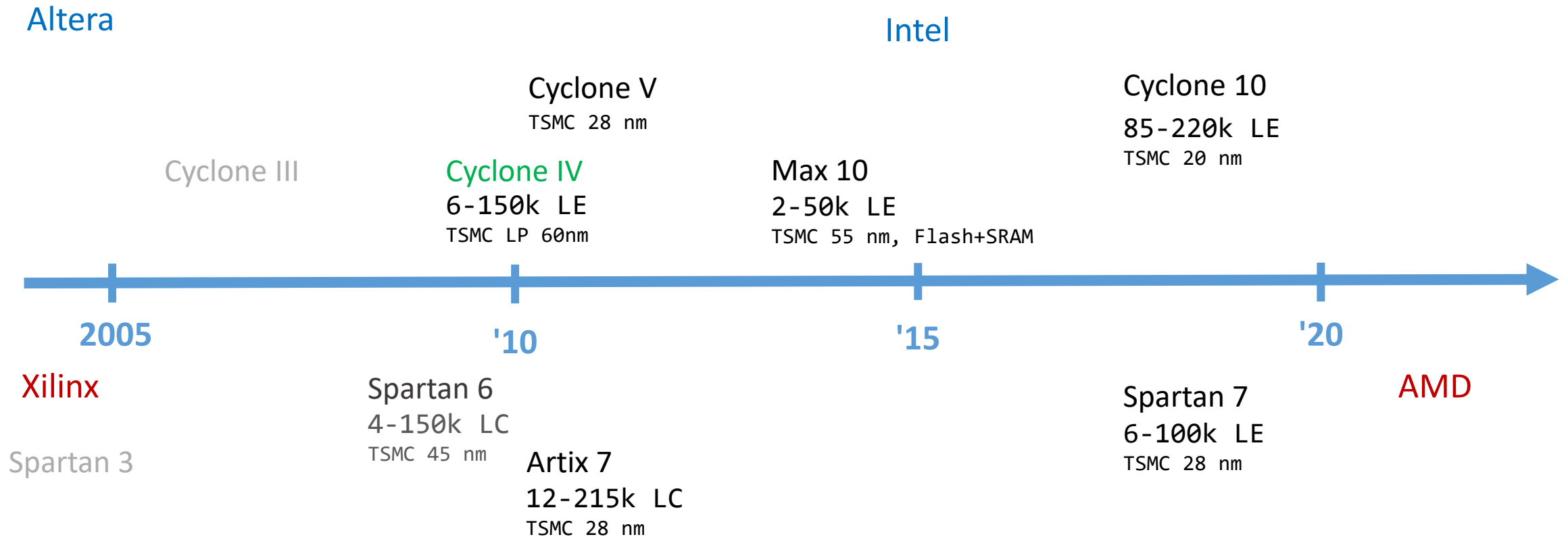
- energijsko bolj učinkoviti
 - do 10x bolj kot CPU, ker so podatki in računske enote skupaj
- paralelno (hkratno) izvajanje operacij, hiter odziv
- fleksibilni vmesniki
 - visoke hitrosti in nizke latence pri prenosu podatkov, več 100 GB/s

Mikrop procesorji

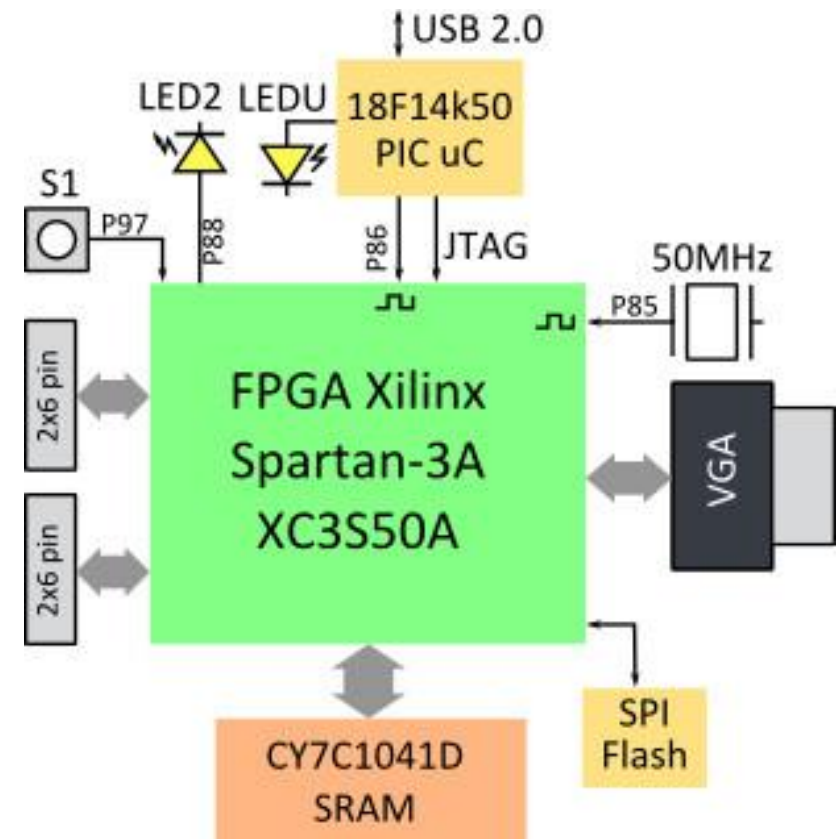
- prilagodljivi in enostavni za uporabo
 - kratek čas prevajanja
- energijsko manj učinkoviti
 - odvisno od zahtevnosti obdelave podat.
- zaporedno izvajanje operacij
 - počasnejša obdelava podatkov in odziv

Izbira FPGA

- najbolj zmogljivi: Altera: Arria,Stratix / Xilinx: Kintex,Virtex
- manjši, cenejši in nižja poraba:



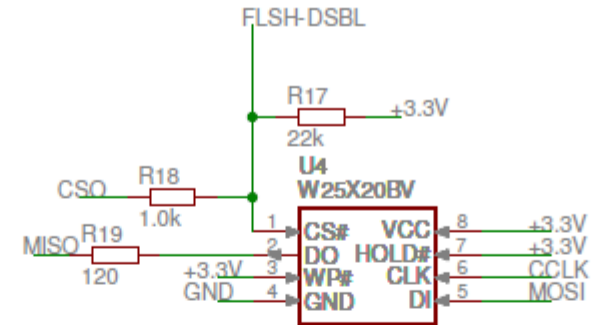
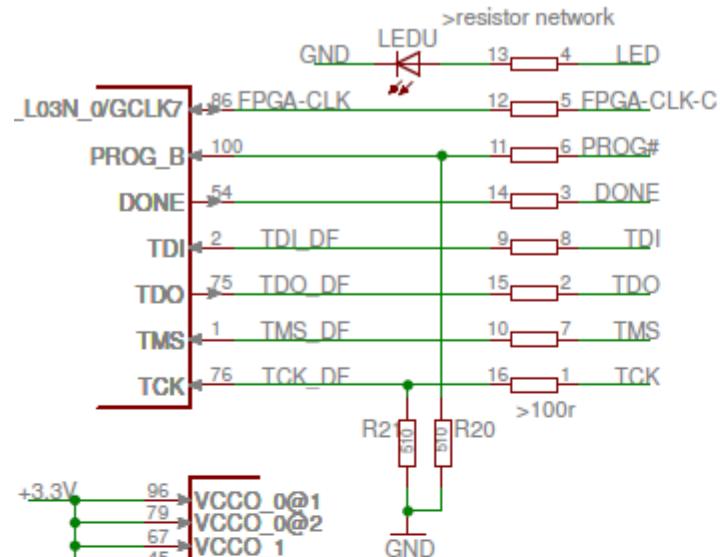
Primer tiskanega vezja z FPGA



Spartan 3

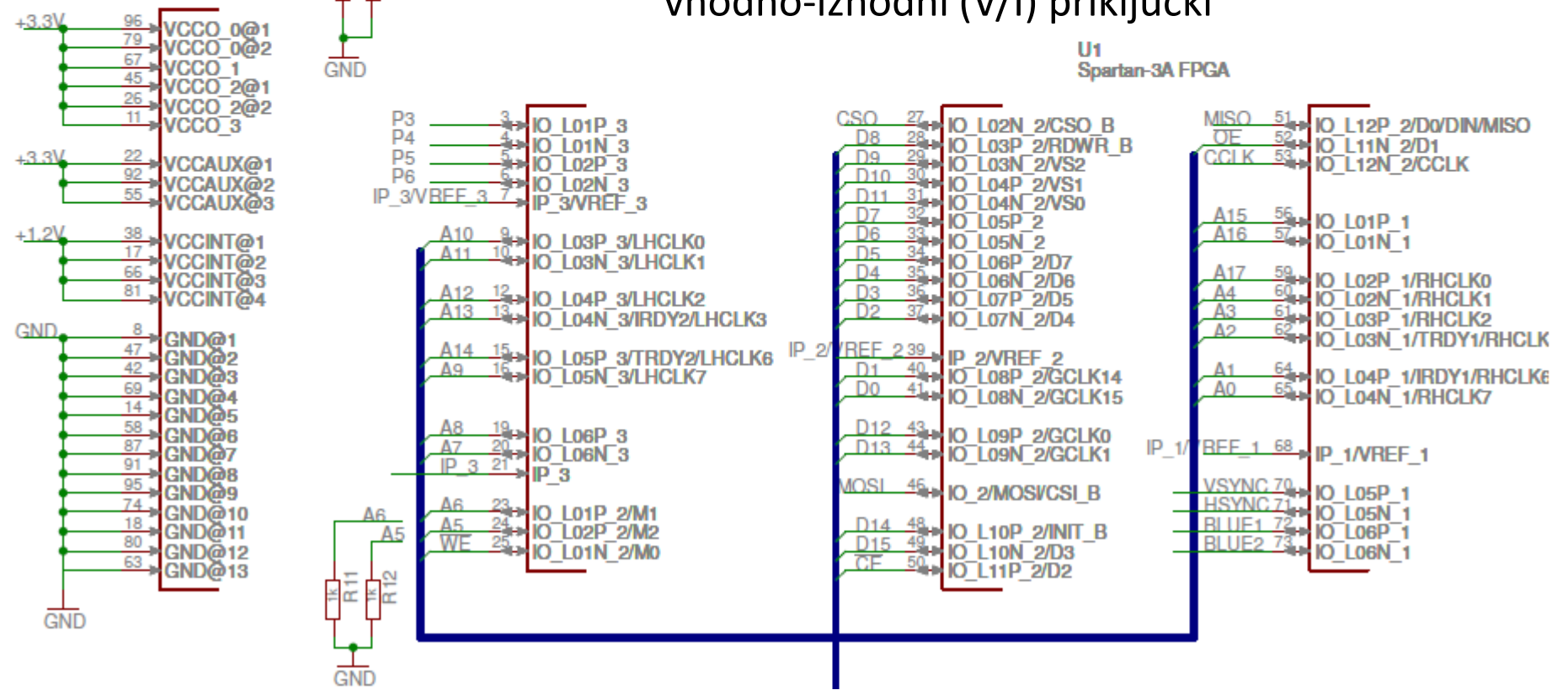
nastavitveni
priključki

napajalni
priključki



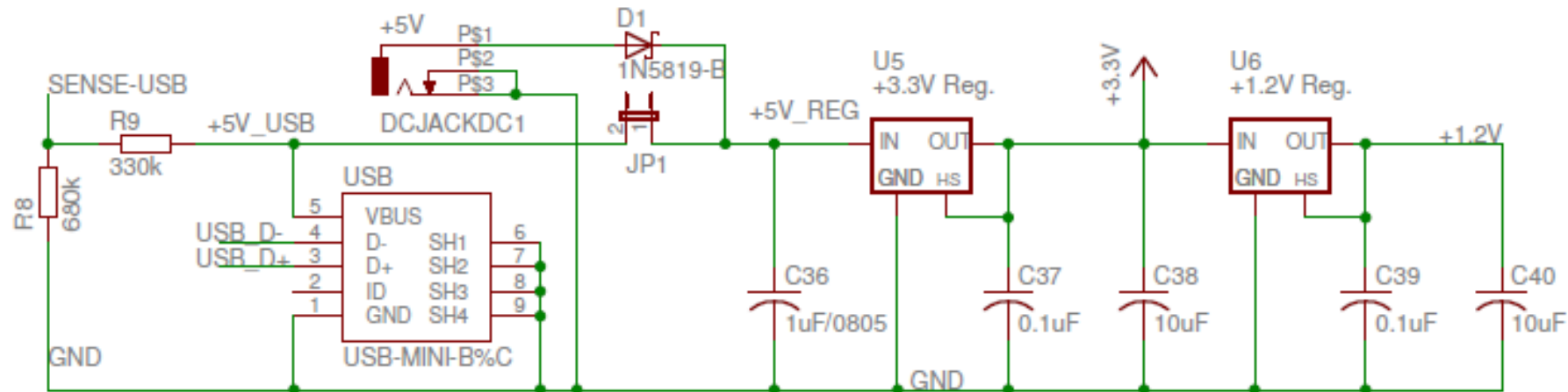
nastavitveni
serijski Flash

vhodno-izhodni (V/I) priključki

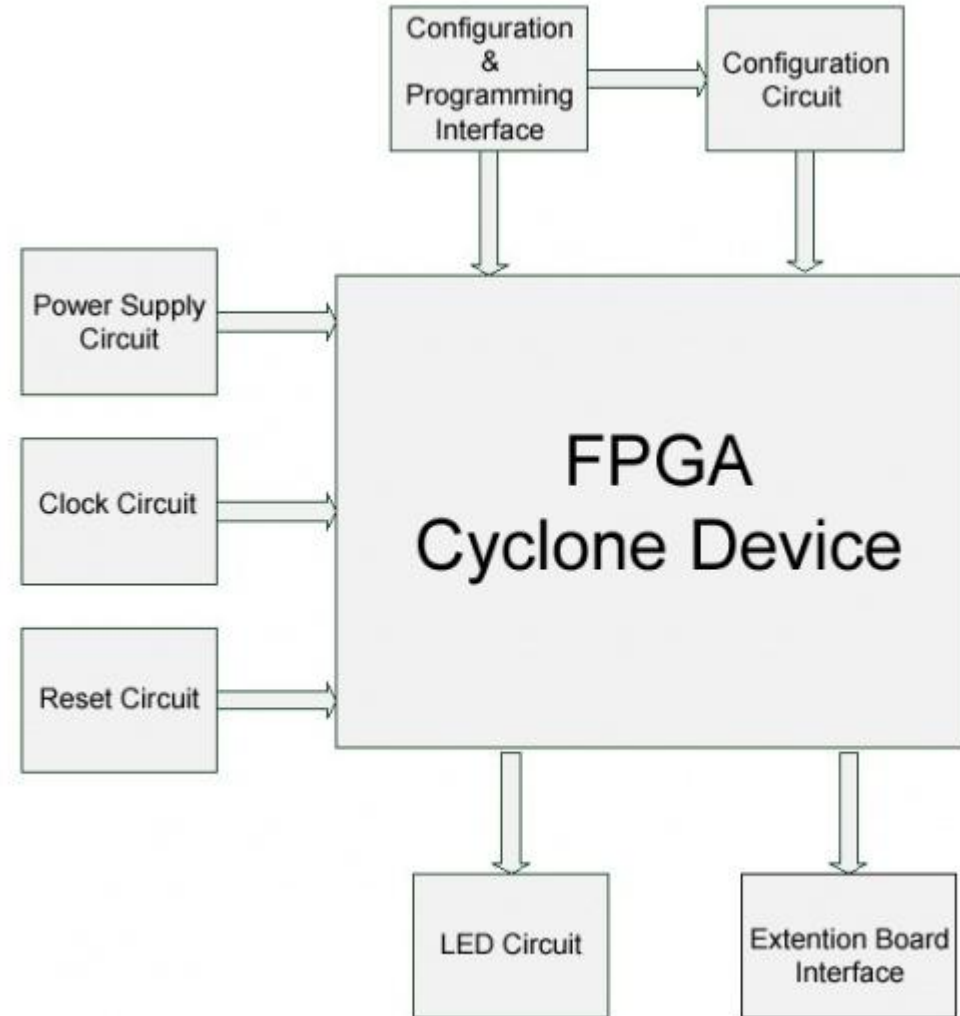


Napajalni del

- ločeno napajanje za jedro vezja FPGA in V/I priključke

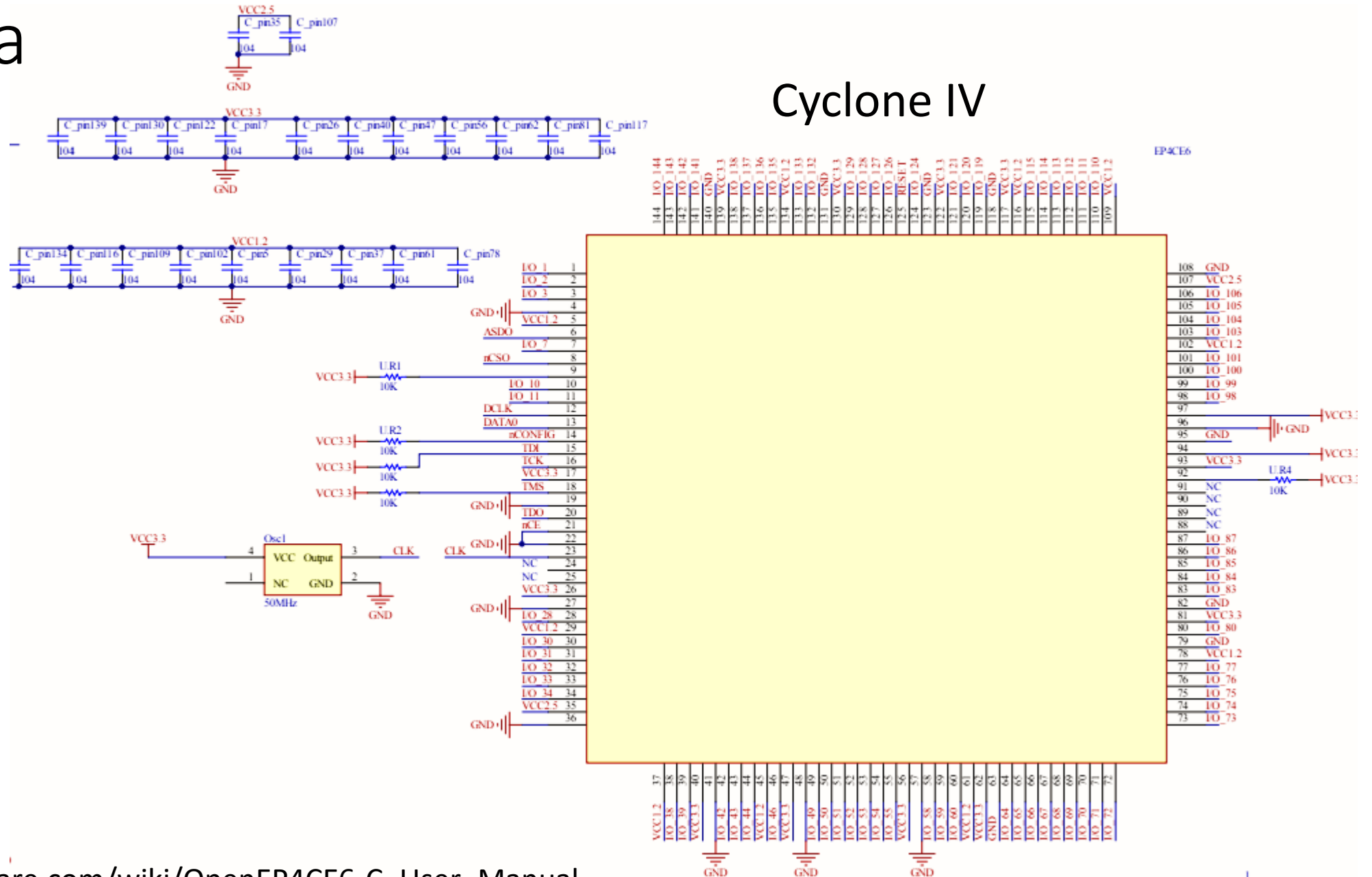


Primer TIV z vezjem Cyclone IV



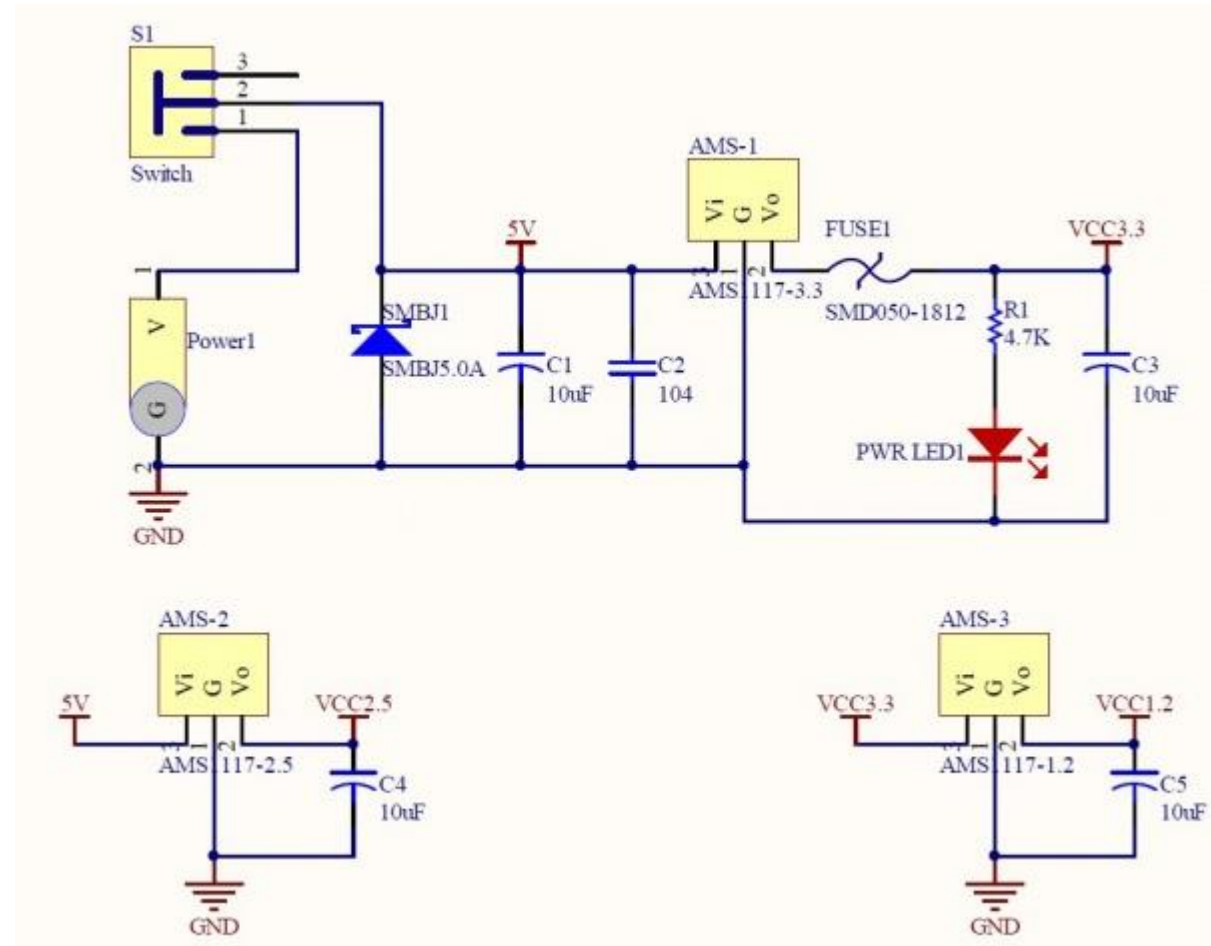
Shema

Cyclone IV



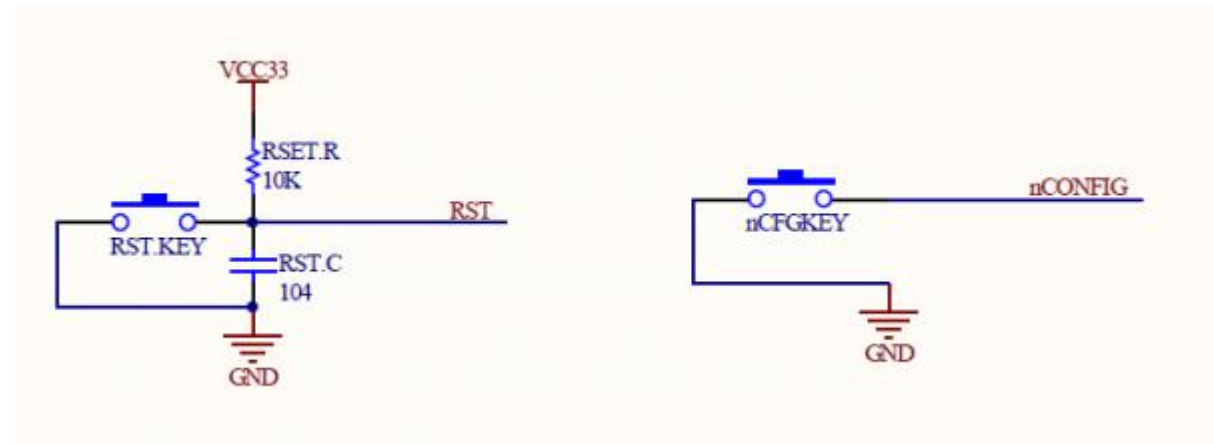
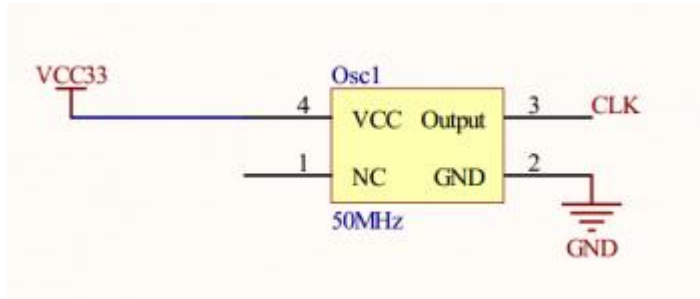
Napajanje

- napajenje V/I
- napajanje jedra
- napajanje prog. dela FPGA



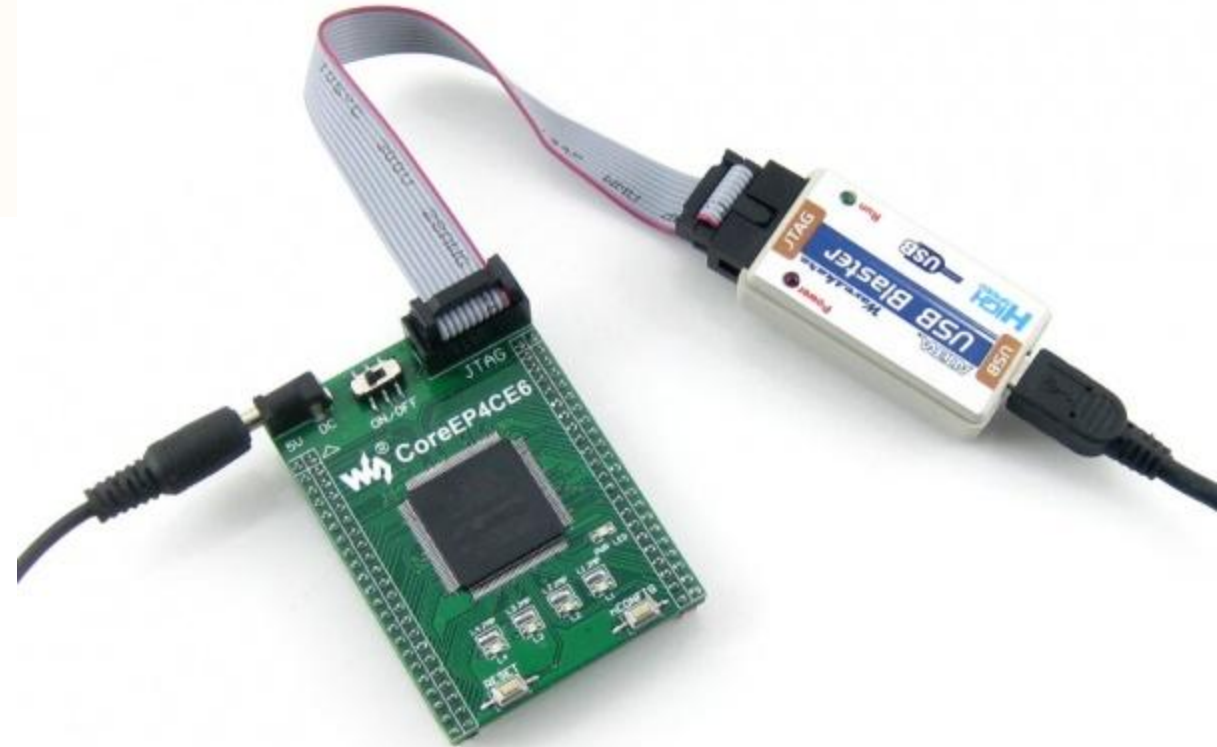
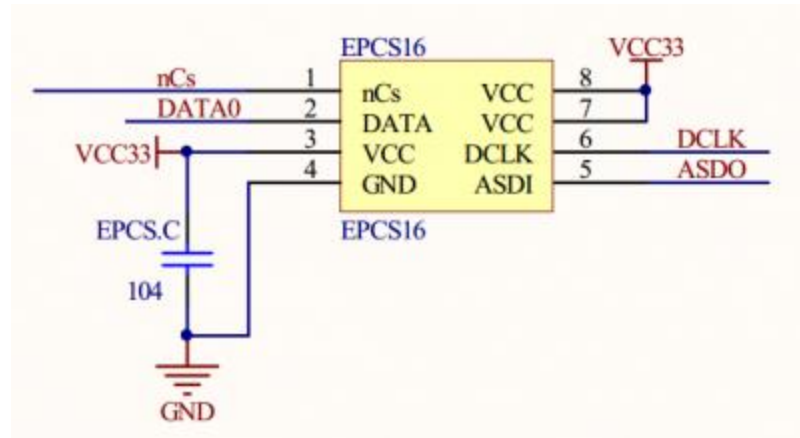
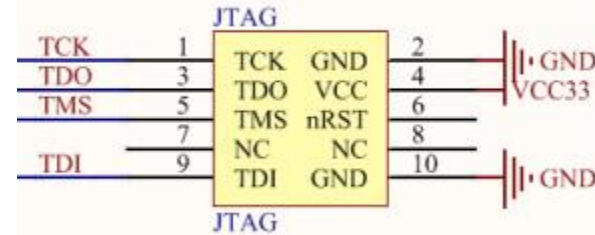
Ura in reset

- kvarčni oscilator za stabilno frekvenco
- znotraj FPGA posebne povezave za uro in možnosti reguliranja frekvence



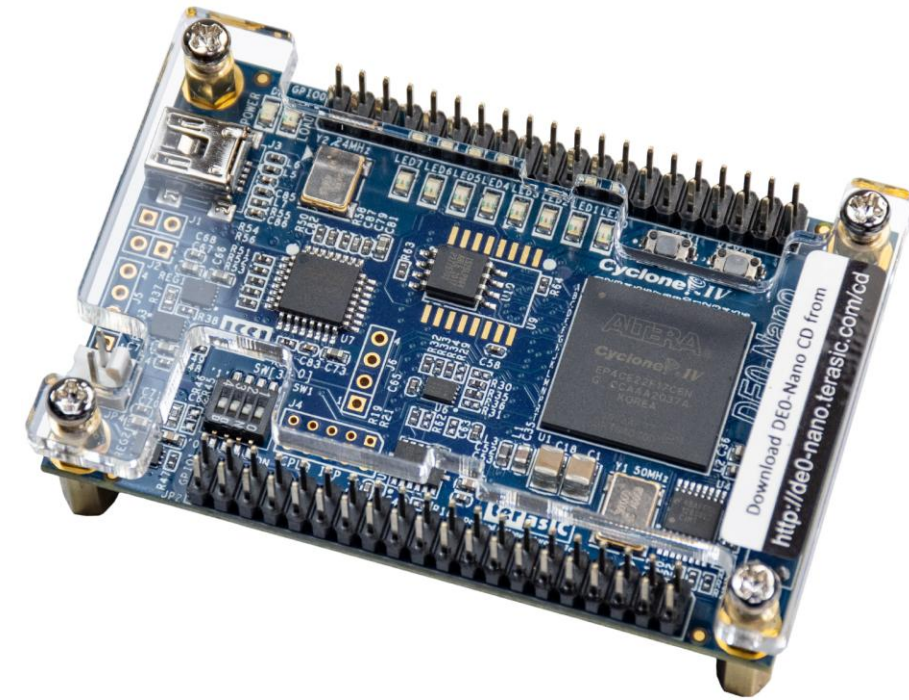
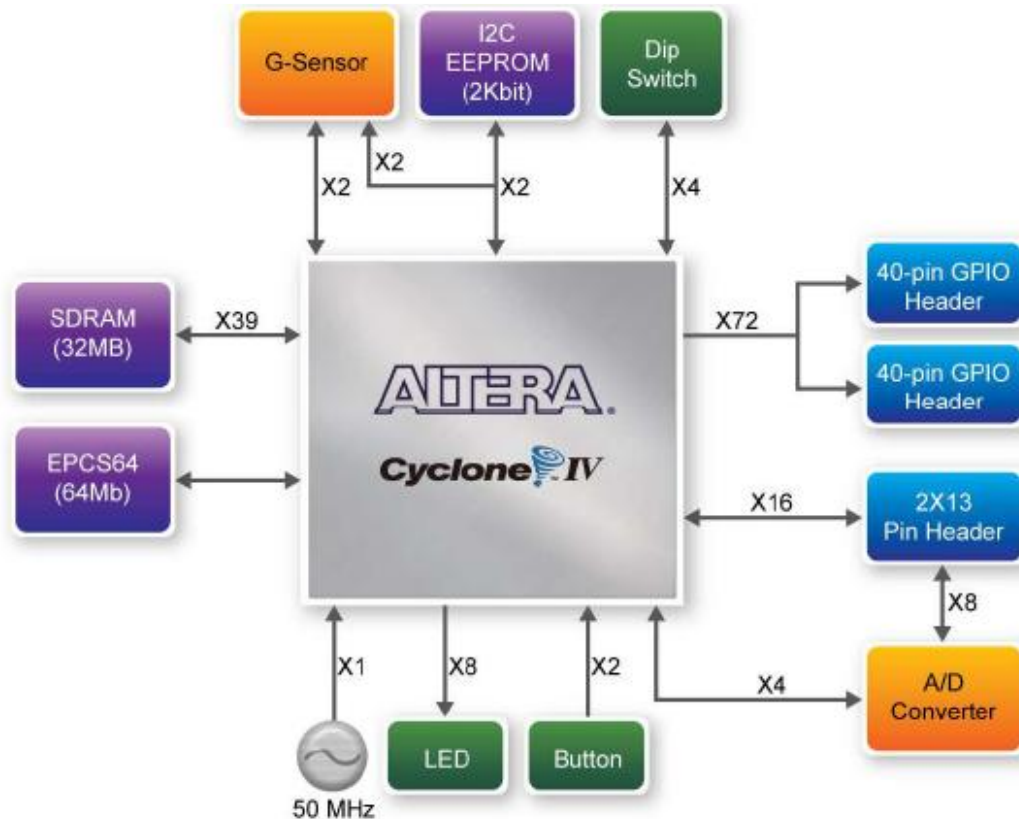
Nalaganje (programiranje) vezja

- JTAG priključek
- serijski Flash

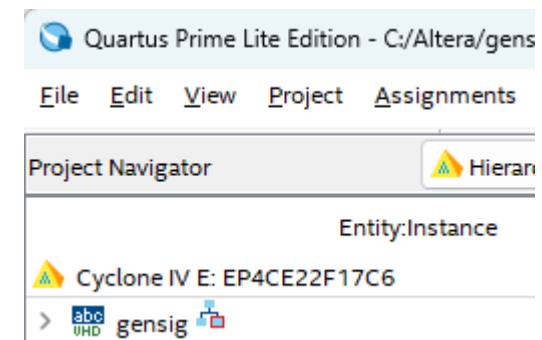


Razvojna plošča DE0-Nano

- FPGA Altera Cyclone IV E
- 22 320 LE, 594 kbit RAM, 66 MUL



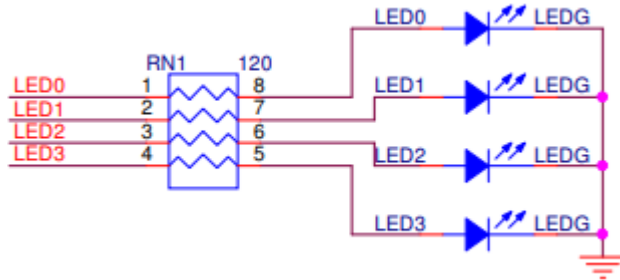
- Quartus projekt



Tipke in LED

- RC in prožilnik odstrani vpliv odskakovanja tipk

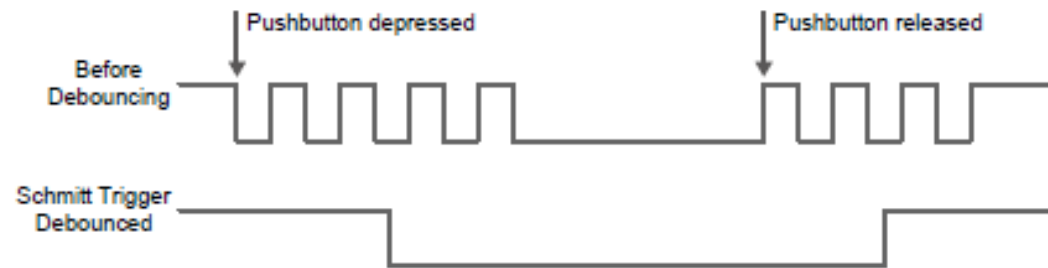
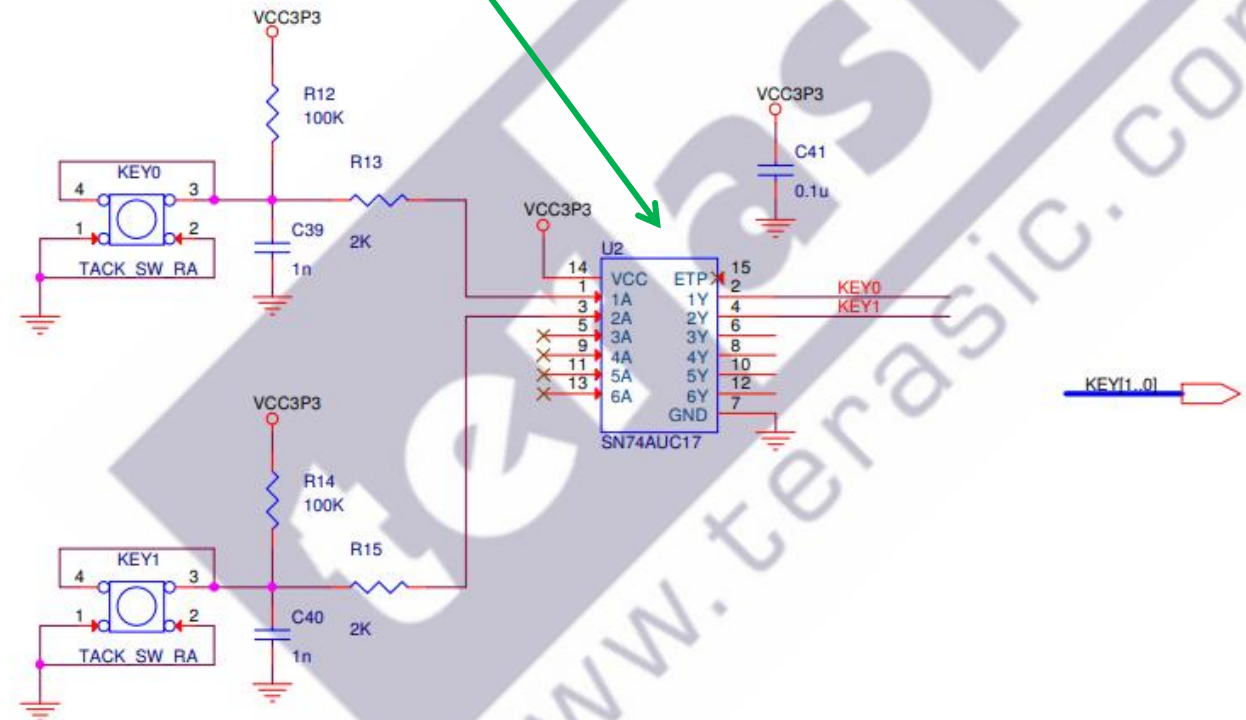
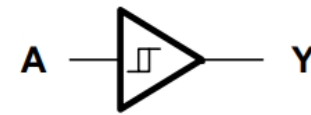
- LED



- lokacije v datoteki *.qsf

```
set_location_assignment PIN_A15 -to led[0]
set_location_assignment PIN_A13 -to led[1]
set_location_assignment PIN_B13 -to led[2]
set_location_assignment PIN_A11 -to led[3]
set_location_assignment PIN_D1 -to led[4]
set_location_assignment PIN_F3 -to led[5]
set_location_assignment PIN_B1 -to led[6]
set_location_assignment PIN_L3 -to led[7]
set_location_assignment PIN_J15 -to key[0]
set_location_assignment PIN_E1 -to key[1]
```

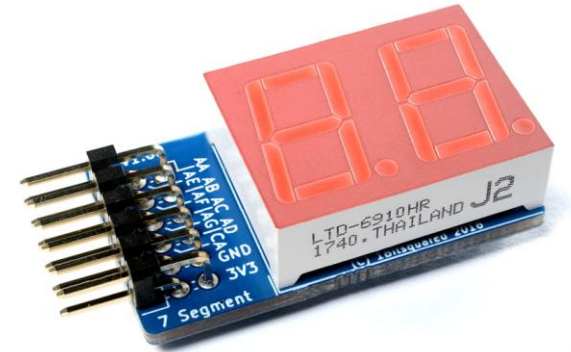
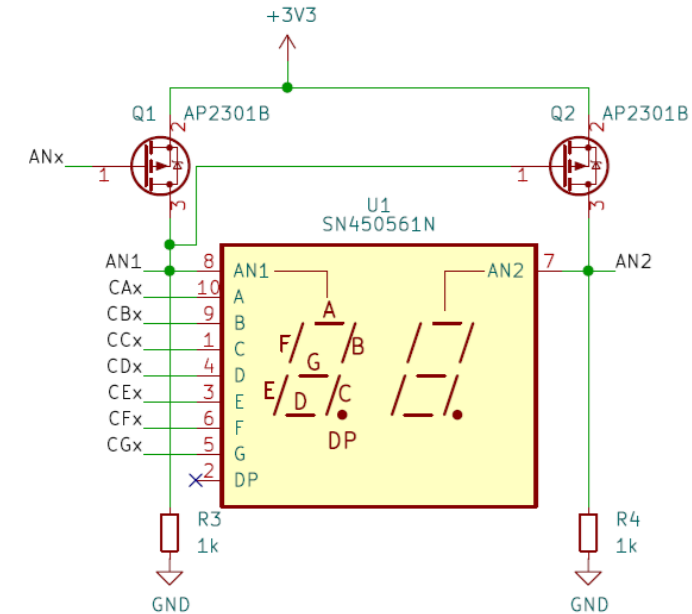
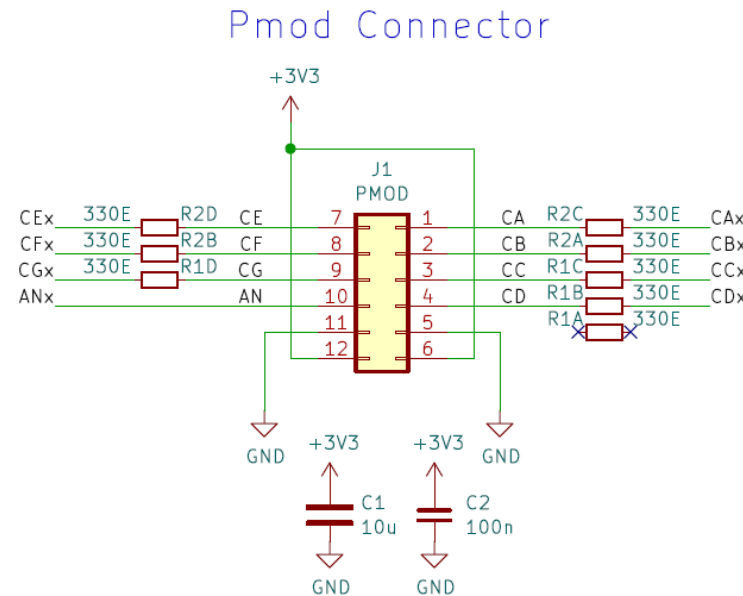
Schmitov
prožilnik



LED prikazovalnik

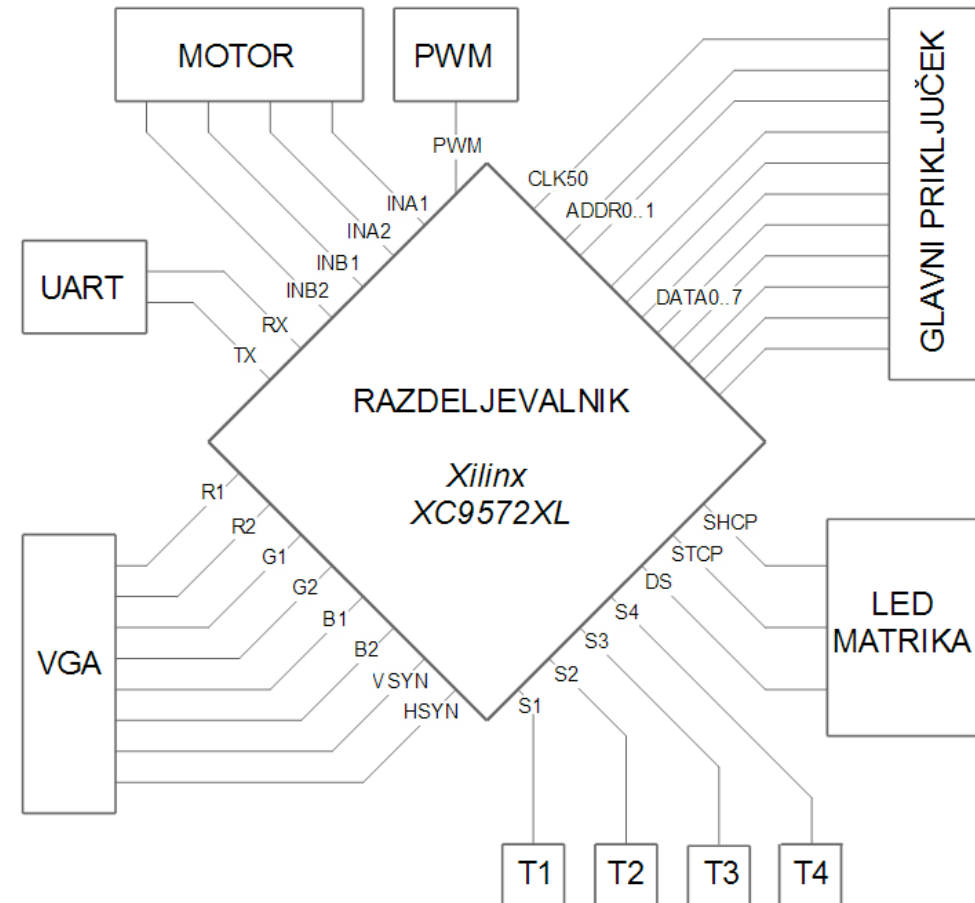
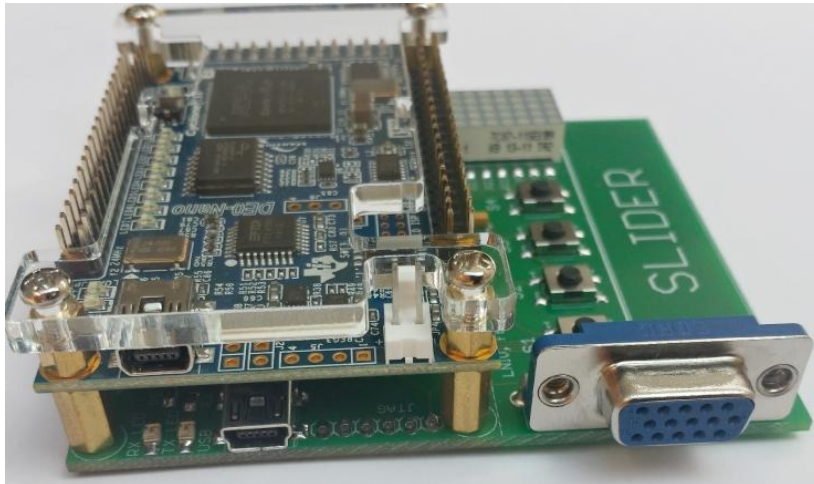
- npr. 7-segmentni: 7-8 signalov/števko
- dekoder bcd -> segment in časovno preklapljanje

```
process(bcd)
begin
  case bcd is
    when "0000" => seg <= "0000001"; -- "0"
    when "0001" => seg <= "1001111"; -- "1"
    when "0010" => seg <= "0010010"; -- "2"
    when "0011" => seg <= "0000110"; -- "3"
    when "0100" => seg <= "1001100"; -- "4"
    when "0101" => seg <= "0100100"; -- "5"
    when "0110" => seg <= "0100000"; -- "6"
    when "0111" => seg <= "0001111"; -- "7"
    when "1000" => seg <= "0000000"; -- "8"
    when "1001" => seg <= "0000100"; -- "9"
    when "1010" => seg <= "0000010"; -- a
    when "1011" => seg <= "1100000"; -- b
    when "1100" => seg <= "0110001"; -- C
    when "1101" => seg <= "1000010"; -- d
    when "1110" => seg <= "0110000"; -- E
    when "1111" => seg <= "0111000"; -- F
  end case;
end process;
```

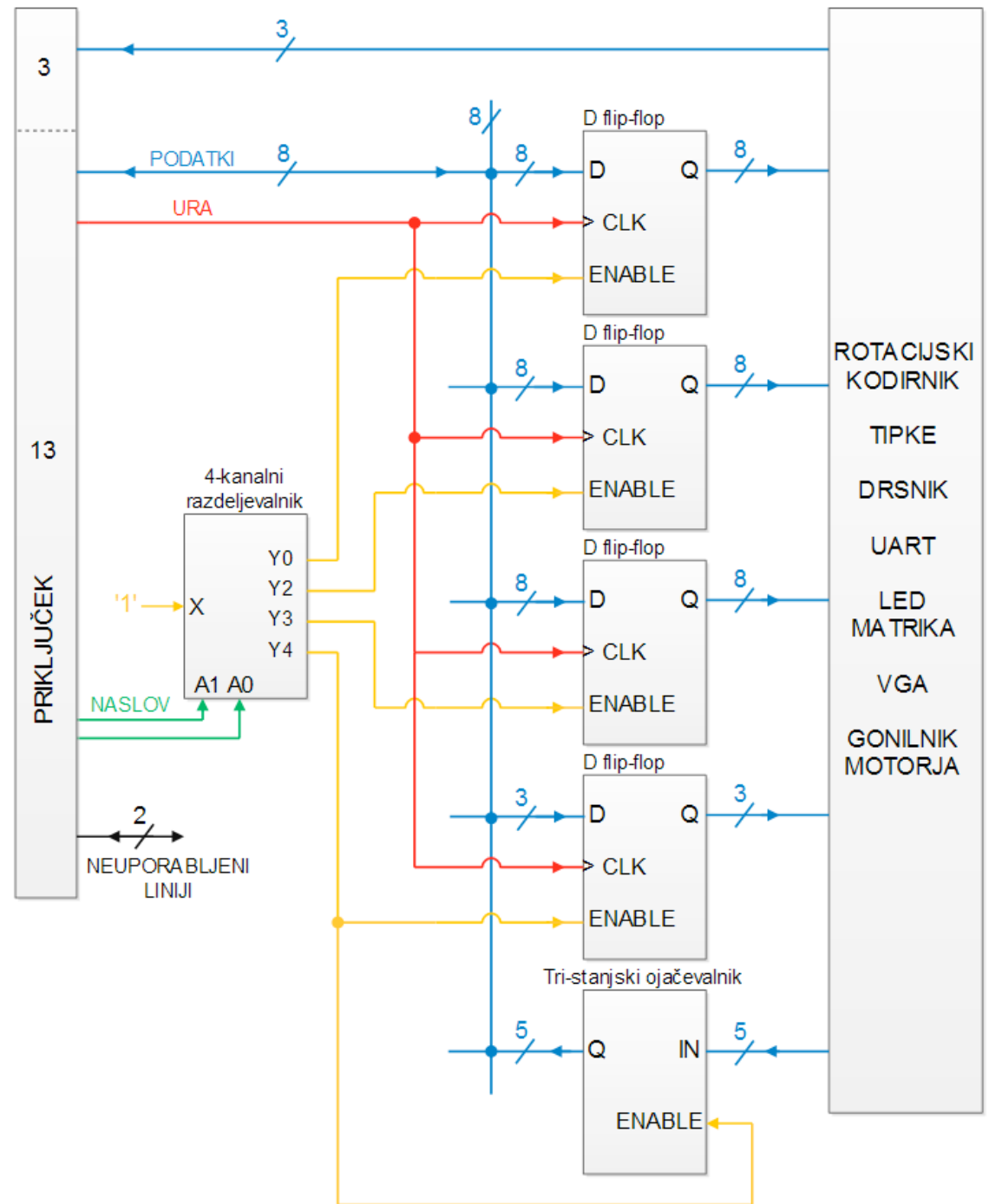


Razširitvena plošča za razvojni sistem

- na razvojnem sistemu imamo razširitveno ploščo s perifernimi enotami
- v vezju CPLD je razdeljevalnik za krmiljenje vseh enot



- 8-bitno sinhrono vodilo
- 2-bitni naslov



Specifikacija vmesnika za sinhrono vodilo

- Vmesnik za razširitveno ploščo na vezju FPGA
- Nadrejena enota (master) na 8-bitnem vodilu
- Vmesnik naj samodejno (aktivno) prenaša podatke do/od perifernih enot
 - vsebuje izbiralnik, ki sekvenčno naslavlja periferne enote
- Dodatne funkcije:
 - dekodirnik za matriko LED
 - ... serijski vmesnik UART,
 - sinhronizacijski generator za VGA,

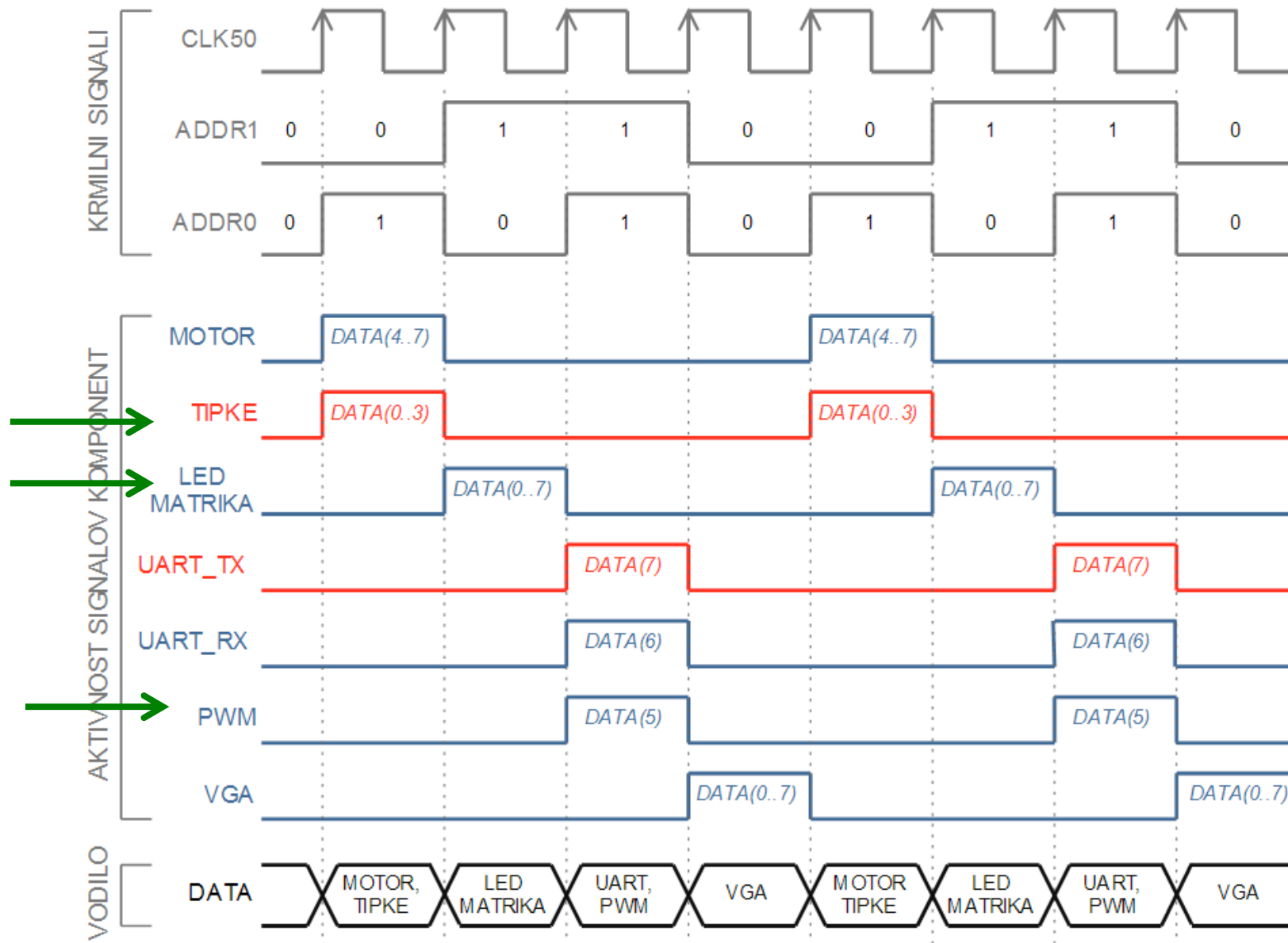
Naslovi perifernih enot

VODILO	NASLOV (<i>ADDR1</i> , <i>ADDR0</i>)			
	»00«	»01«	»10«	»11«
<i>DATA0</i>	R1	S1	<i>buff</i> (0)	-
<i>DATA1</i>	R2	S2	<i>buff</i> (1)	-
<i>DATA2</i>	G1	S3	<i>buff</i> (2)	-
<i>DATA3</i>	G2	S4	<i>buff</i> (3)	-
<i>DATA4</i>	B1	INA1	<i>buff</i> (4)	-
<i>DATA5</i>	B2	INA2	<i>buff</i> (5)	PWM
<i>DATA6</i>	VSYN	INB1	<i>buff</i> (6)	RX
<i>DATA7</i>	HSYN	INB2	<i>buff</i> (7)	TX

VGA

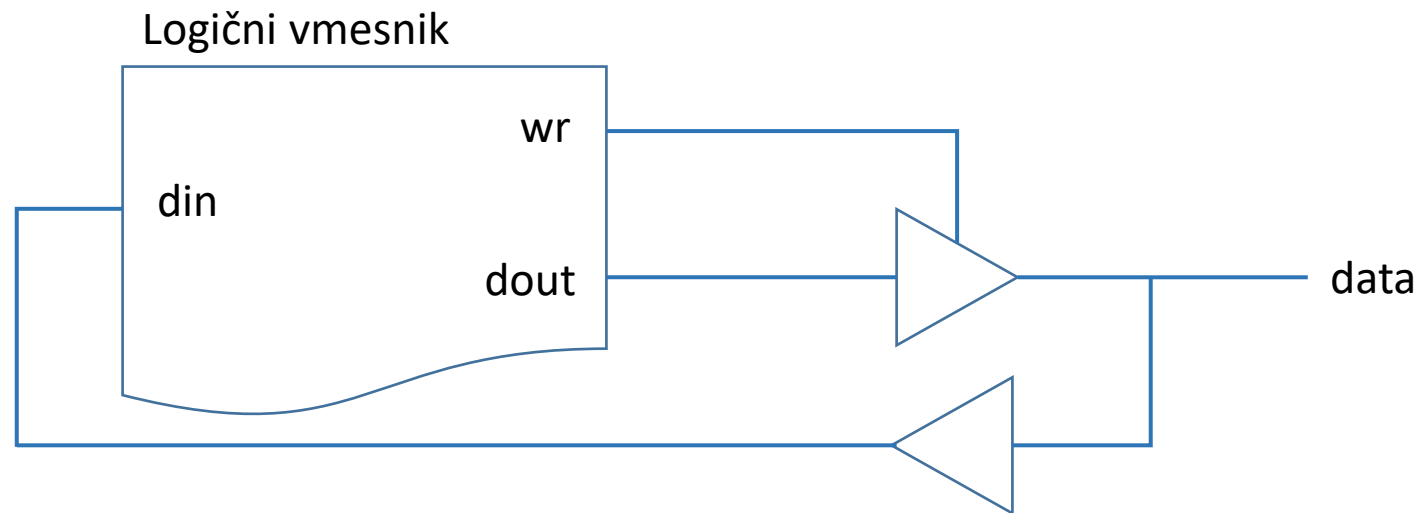
buff(7:5) izbrana vrstica matrike LED,
buff(4:0) posamezne LED v vrstici

Časovni potek signalov



Komunikacija po dvosmernem vodilu

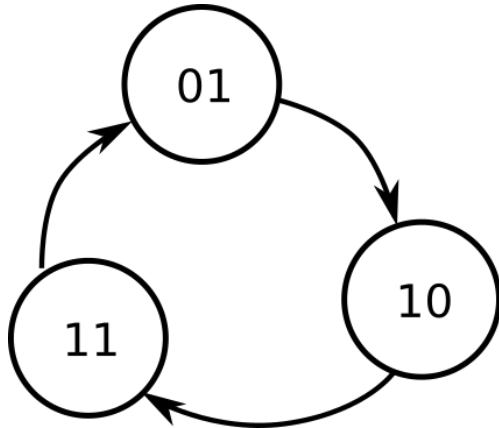
- logični vmesnik, ki ima le enosmerne signale (primerno za vezje znotraj FPGA in za model v SHDL)
- fizični del vmesnika s tri-stanjskimi ojačevalniki za vodilo data, ki so v vhodno/izhodnih blokih FPGA



```
data(7 downto 0) <= "ZZZZZZZZ" when wr='0' else dout;  
din <= data(7 downto 0);
```

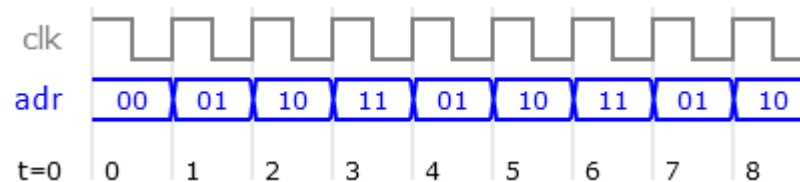
Sekvenčni stroj za naslavljanje

- Naslov ciklično spreminja stanja



```
if adr="01" then adr<="10"  
elsif adr="10" then adr<="11"  
else adr <= "01";  
end
```

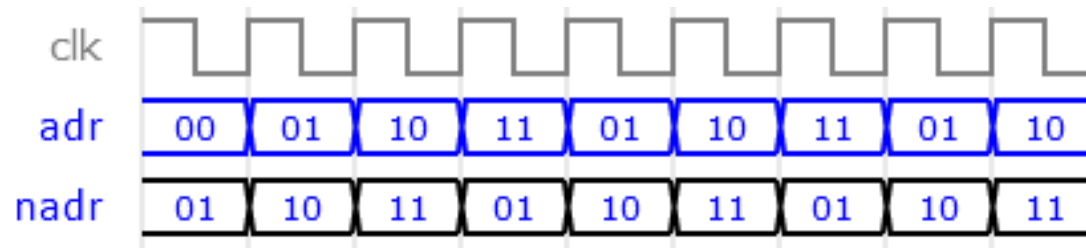
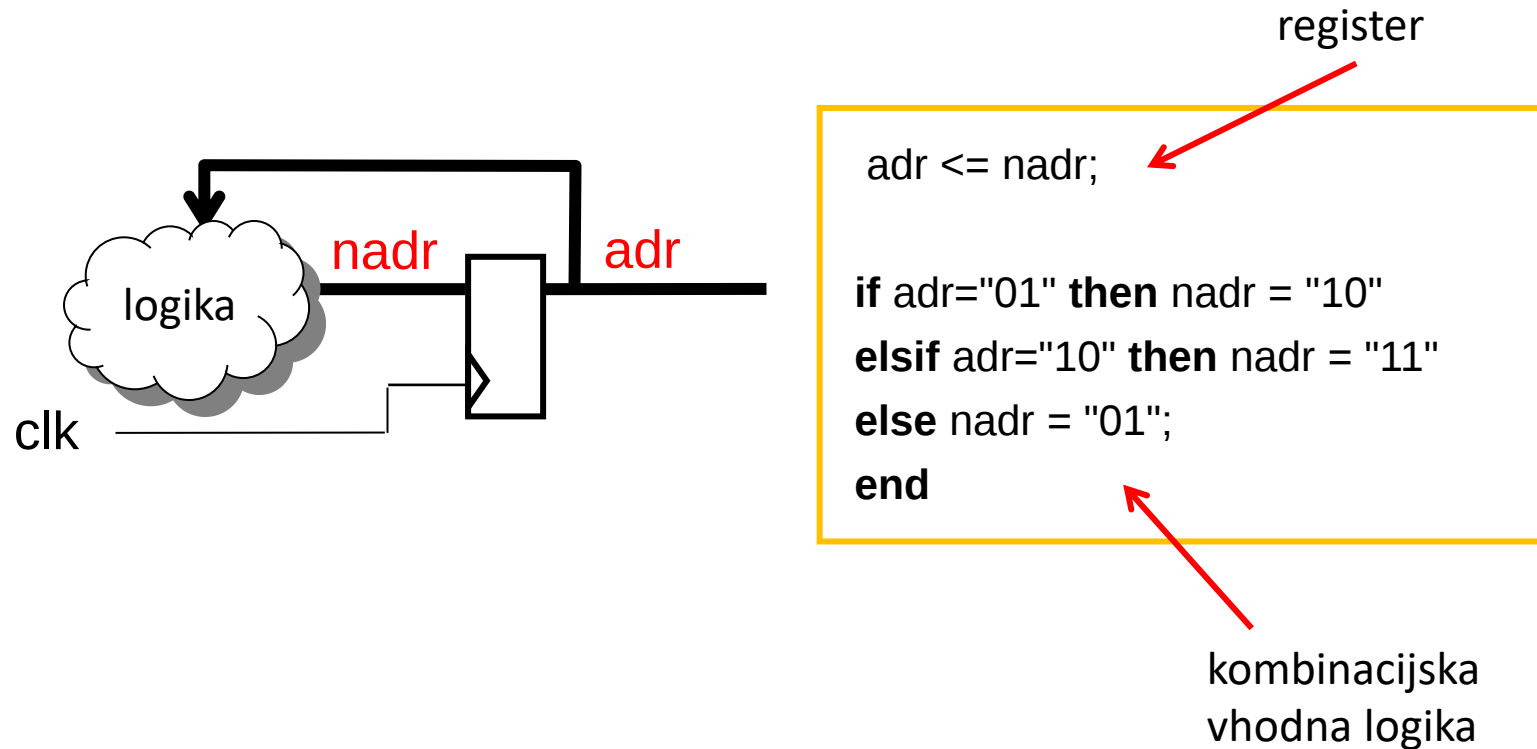
končni else namesto elsif adr= "11"
poskrbi za nedefinirana stanja



- ▶ npr. 2-bitni števec, ki ima štiri stanja

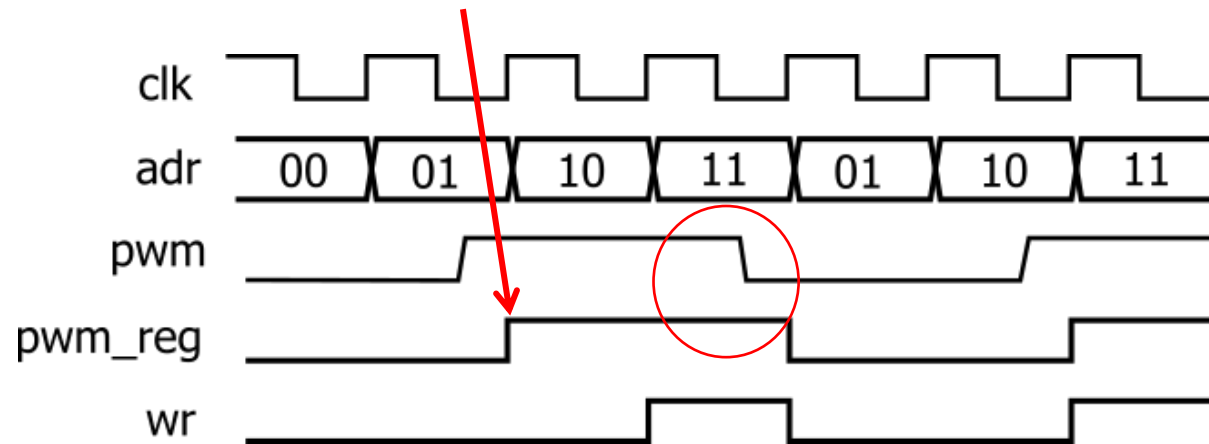
```
adr <= adr + 1
```

Sekvenčni stroj za naslavljanje



Izbiralnik za PWM

- Izbiralnik prenese vsaj tretji (četrti) cikel ure vrednost PWM na razširitveno ploščo
- signale sinhroniziramo, ker ne želimo da se podatek spreminja znotraj cikla ure in ne želimo motenj na wr



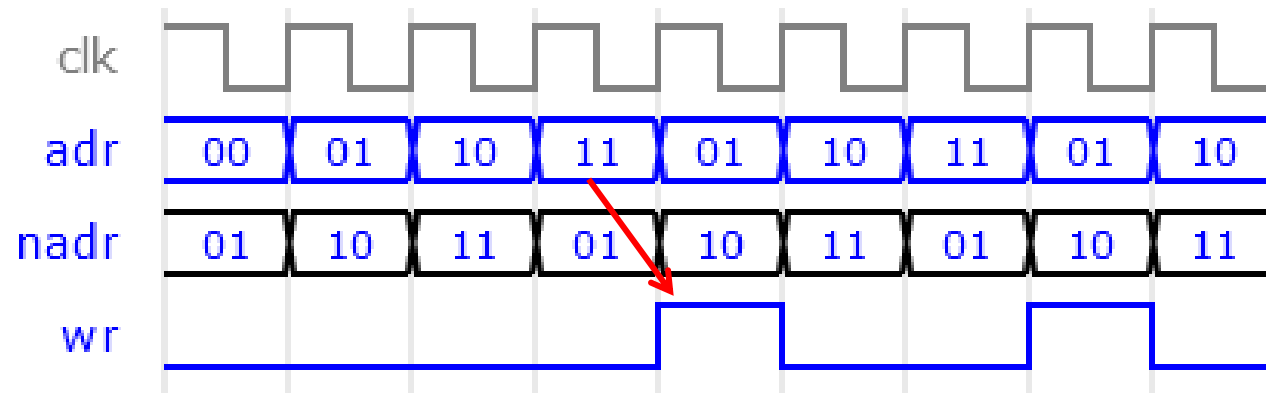
- ▶ Sinhronizacijo naredimo tako, da pošljemo signal na register ali D flip-flop

Izbiralnik za PWM

Kaj je narobe s kodo ?

```
if adr="11" then wr <= 1 else wr <= 0 end
```

- ▶ sekvenčna prireditev (flip-flop) povzroči zakasnitev!



- ▶ Rešitev:

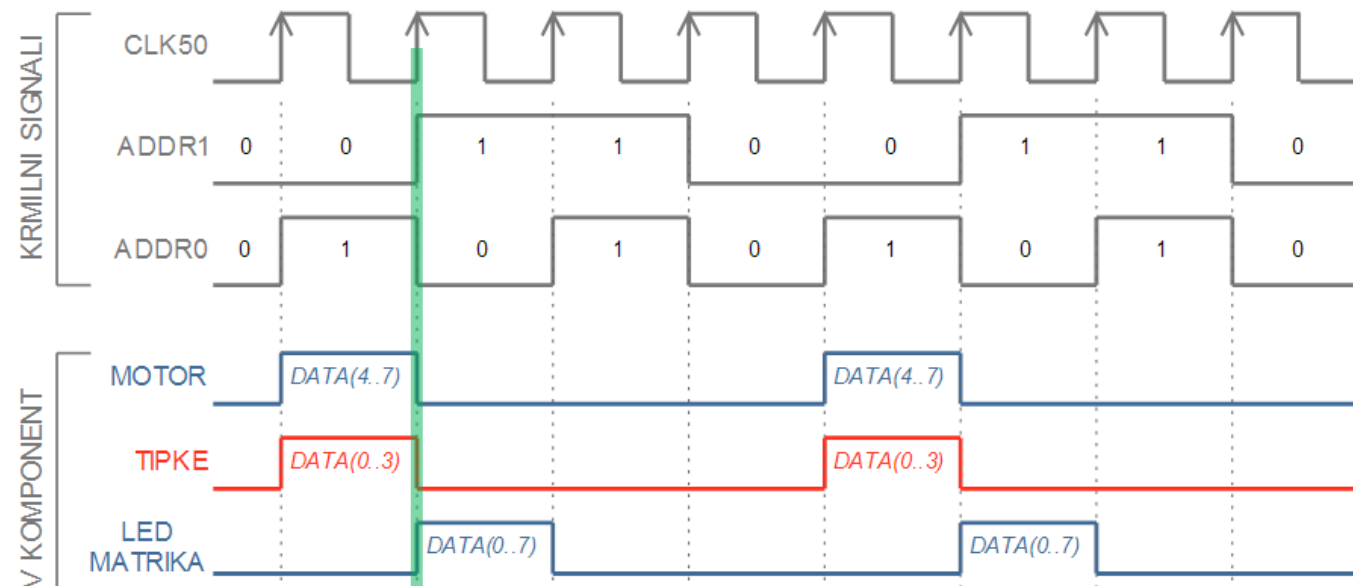
```
if adr="10" then wr <= 1 else wr <= 0 end
```

ali

```
data(7 downto 0) <= "ZZZZZZZZ" when adr="11" else dout;
```

Register za tipke in matriko LED

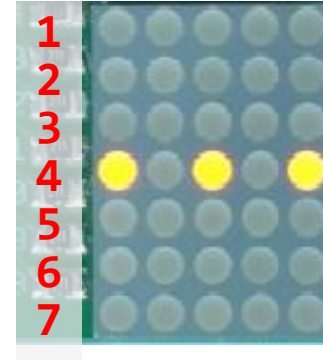
- Branje tipk in nastavljanje matrike ob isti fronti ure
 - ob adr=01 dobimo stanje tipk, ki ga preberemo na koncu
 - ob prehodu na adr=10 nastavimo stanje matrike LED



```
if adr="01" then
  btn <= din(3:0); dout <= ledm
end
```

Matrika LED

- Matriko LED 7x5 krmilimo z naslavljanjem vrstic
 - ledm(7:5) izbrana vrstica matrike LED,
 - ledm(4:0) posamezne LED v vrstici



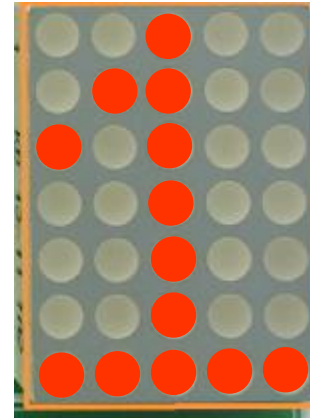
- ▶ Delilnik ure 50 MHz za osveževanje prikazovalnika
 - ▶ Npr. 16-bitni števec, $50 \text{ MHz} / (2^{16} * 7) = 108 \text{ Hz}$
 - ▶ osvežujemo tako hitro, da ne opazimo utripanja

```
u16 → d <= d + 1  
      if d=0 then  
        if (line=6) line <= 0 else line <= line + 1;  
      end
```

Grafični znak na matriki

- Grafiko shranimo v pomnilniku ROM, ki ga naslovimo s števcem vrstic

```
znak: 7u5 = "00100",  
            "00110",  
            "00101",  
            "00100",  
            "00100",  
            "00100",  
            "11111";
```



ledm(7:5) & ledm(4:0)

```
if d=0 then  
  ledm <= (line+1) & znak(line);  
end
```

korekcija, ker so vrstice matrike 1..7, v ROM pa 0..6

Prikazovanje več znakov

- Znake shranimo v ROM in sestavimo naslov iz št. znaka (num) in naslova vrstice (line)
- Če imamo za vsak znak 7 vrstic, naslov določimo z enačbo:

$$\text{romadr} = \text{num} * 7 + \text{line}$$

```
znak:7u5="00100",  
         "01100",  
         "10101",  
         "00100",  
         "00100",  
         "00100",  
         "11111",  
         "01110",  
         "10001",  
         "00010",
```

- ▶ Če dodamo v ROM eno prazno vrstico, ne potrebujemo množenja ampak naslov razdelimo na dva dela

$$\text{romadr} = \text{num} \& \text{line}$$

Dokumentacija vmesnika

- Vmesnik bomo uporabili v več projektih, ga dali v prosto uporabo ali prodali v obliki načrta z dokumentacijo
- Takšno vezje imenujemo komponenta intelektualne lastnine (**IP – Intellectual Property**)
 - IP je splošen izraz za nekaj, kar je narejeno v nefizični obliki, kot npr. načrt ali model vezja
 - nekatera podjetja se ukvarjajo le z izdelavo IP
 - podobno kot za programsko opremo obstajajo za digitalna vezja strani s prosto dostopnimi komponentami IP
npr. <https://opencores.org/>

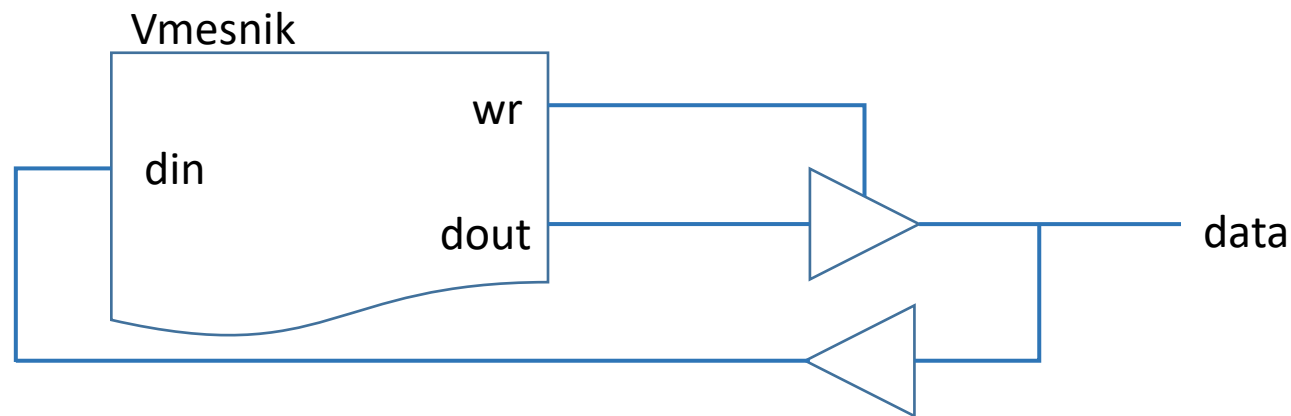


Dokumentacija komponente IP

- opis digitalne komponente je v obliki podatkovnega lista, ki predstavi izdelano komponento
 - opis je dokument namenjen za razvojnega inženirja, ki bi rad uporabil vašo komponento v svojem projektu
 - vsebuje naj blokovne sheme, časovnice, tabele...
 - inženirji razumemo sheme in neradi beremo dolga besedila
 - vsaka slika mora imeti kratko pojasnilo
- elementi dokumentacije komponente
 - specifikacija
 - blokovna shema
 - opis stanj in načina delovanja
 - lastnosti komponente
 - povezavo na datoteke, literaturo

Specifikacija vmesnika

- Vmesnik za razširitveno ploščo na razvojnem sistemu FPGA Altera DE0-Nano
- Deluje kot nadrejena enota na 8-bitnem vodilu
- Vmesnik vsebuje dekodirnik za prikaz 4-bitne BCD številke na matriki LED
- Uporabljamo ga skupaj s fizičnim vmesnikom:

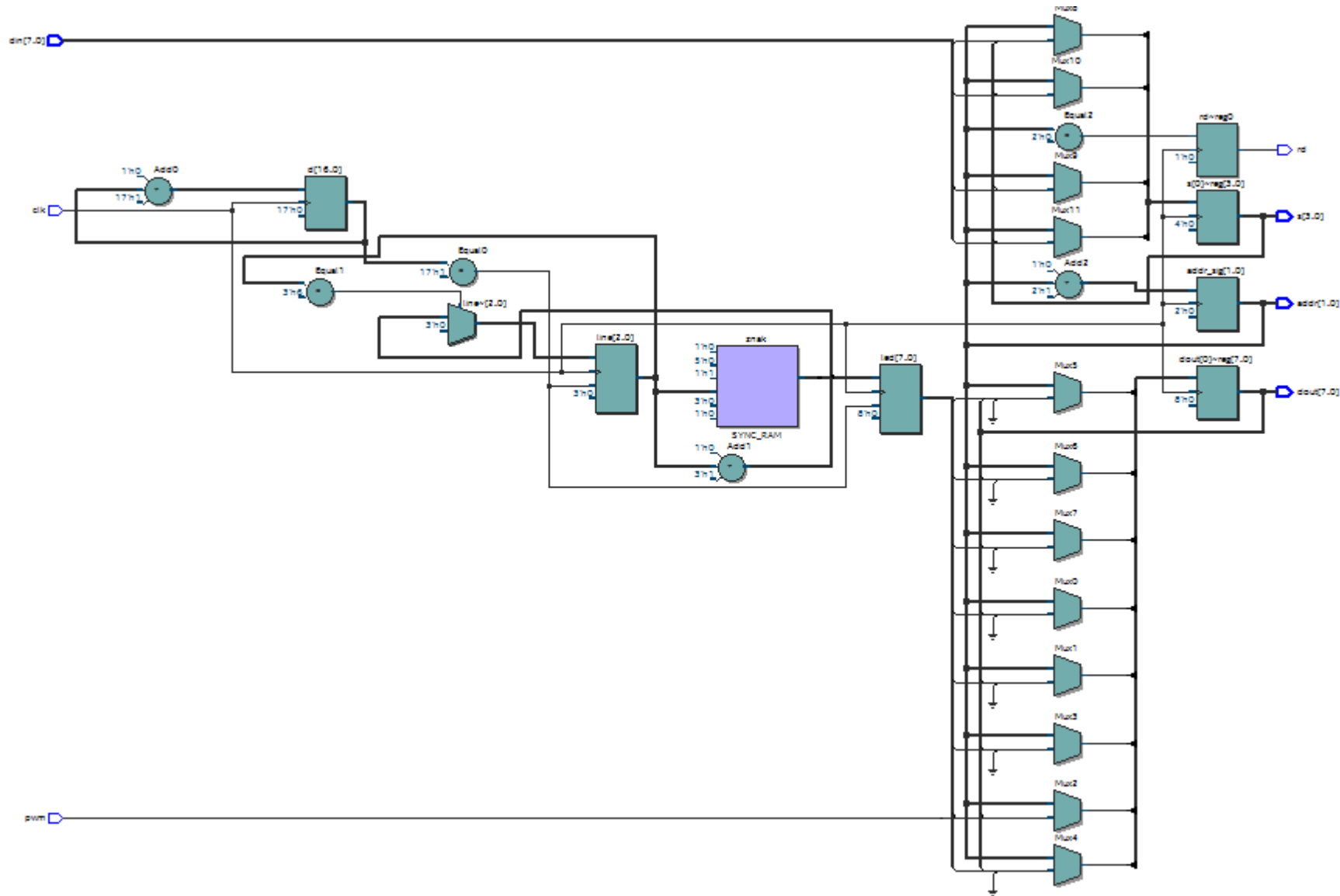


Seznam priključkov

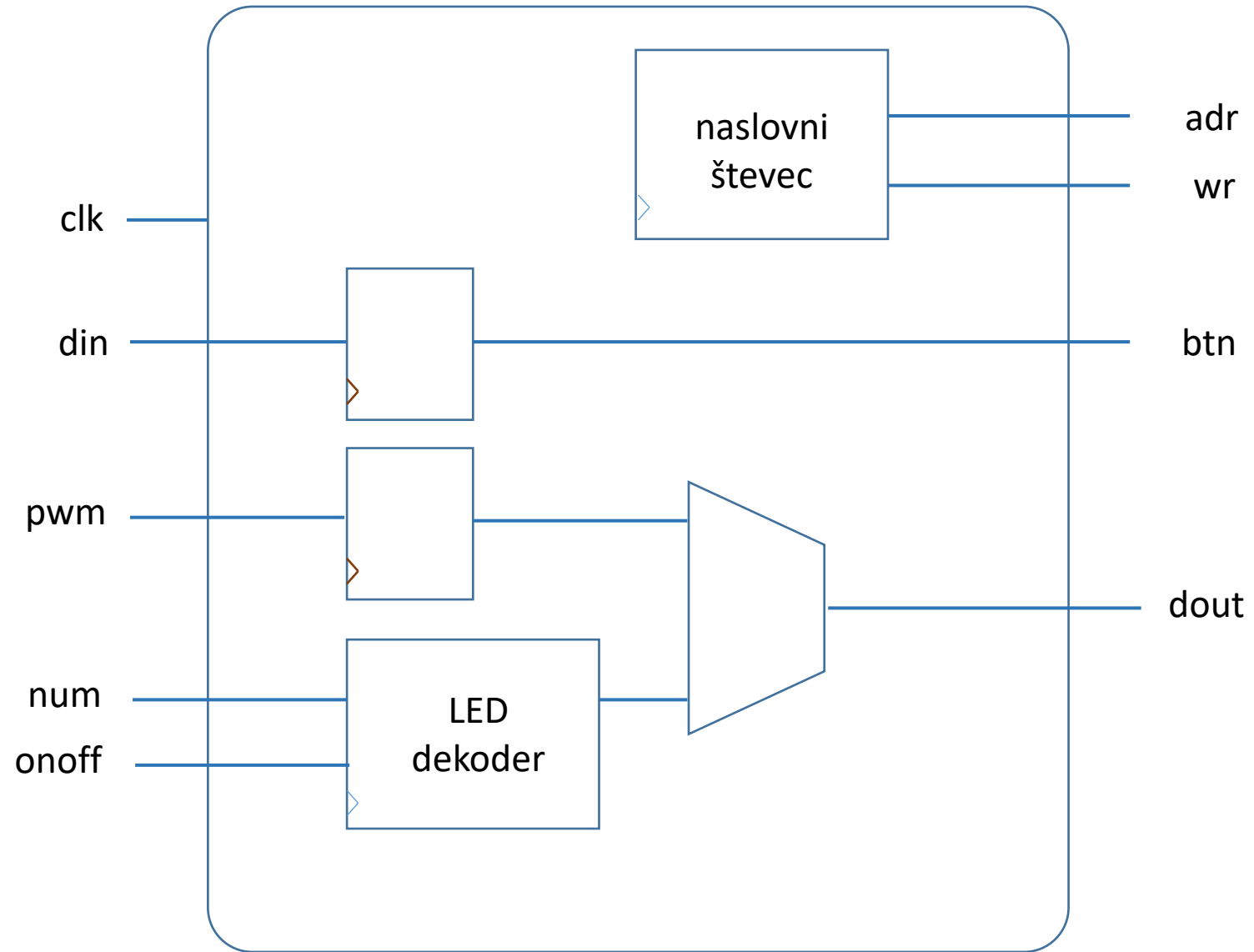
ime	vrsta	pomen
clk	1-bit, in	ura frekvence 50 MHz
pwm	1-bit, in	pulzno-širinski modulacijski signal
num	4-bit, in	BCD vrednost za prikaz na matriki LED
onoff	1-bit, in	vklop (onoff=1) ali izklop matrike LED
din	8-bit, in	vhodno vodilo
dout	8-bit, out	izhodno vodilo
wr	1-bit, out	krmiljenje 3-stanjskega vodila (wr=1, zapis na vodilo)
adr	2-bit, out	izhodni naslov
btn	4-bit, out	stanje tipkovnice

Shema

celotna shema na ravni RTL je preveč podrobna...

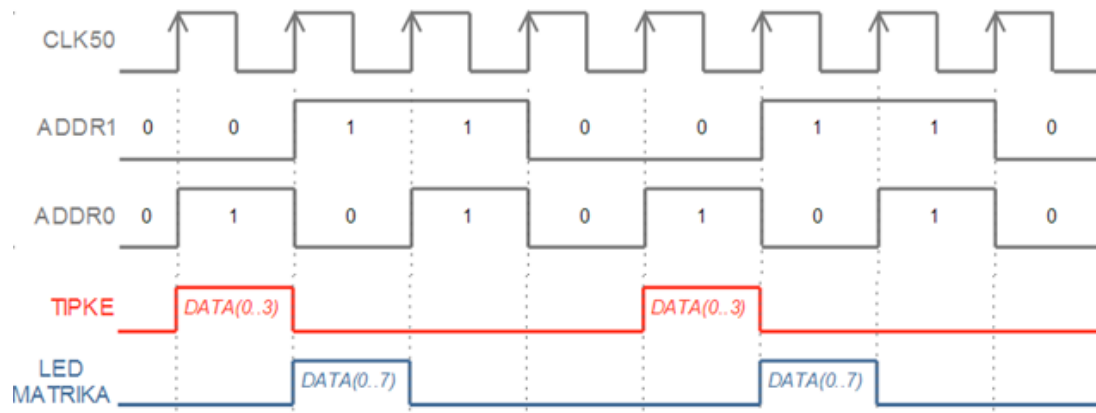


Blokovna shema



Način delovanja

- Vmesnik samodejno prenaša podatke do/od perifernih enot
 - vsebuje izbiralnik, ki sekvenčno naslavlja periferne enote
 - izhodno stanje vmesnika (signal adr):
 - 01 prenos stanja tipk
 - 10 nastavljanje matrike LED
 - 11 prenos PWM



- 4-bitni vhod num prikaže na matriki v grafični obliki
 - kadar je onoff=0 je prikaz izključen

Lastnosti vmesnika

- deluje pri vhodni uri 50 MHz
- prenaša 4-bitno stanje tipk in signal PWM
 - frekvenca PWM naj bo vsaj 4x nižja od frekvence ure
- prikazuje številke od 0 do 9
 - vhod num v obliki BCD
 - pri num>9 prikaže ...
- rezultat sinteze vmesnika v vezje Altera Cyclone IV E
 - 46 logičnih elementov (LE)
 - 40 flip-flopov
 - 29 priključkov
 - 0 množilnih blokov, 0 pomnilnih blokov

Reference

- Model vmesnika je v datotekah:
 - IOvmes.shdl, model v SHDL
 - IOvmes.vhd, model v jeziku VHDL
 - IOvmes_tb, testna struktura za simulacijo (opiši kaj preverjamo s to simulacijo)
- Literatura:
 - S. Rudolf, Razširitven plošča, magistrsko delo, FE, 2015
 - Digitalni elektronski sistemi, gradivo za predavanja, FE, 2024