



Laboratorij za načrtovanje integriranih vezij

Univerza *v Ljubljani*
Fakulteta *za elektrotehniko*



Andrej Trost

priprava na laboratorijske vaje 2012

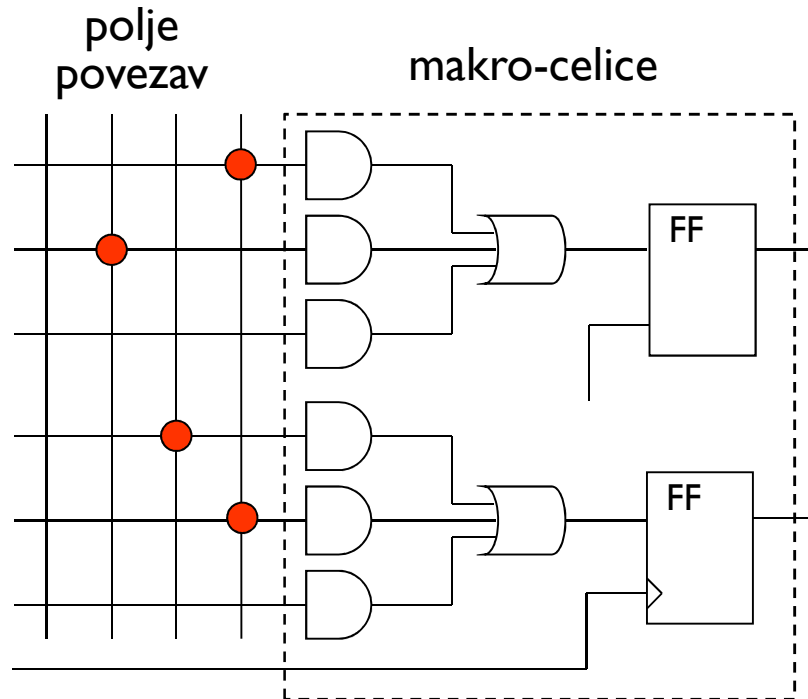
Integrirana vezja

Literatura: A. Trost: Načrtovanje digitalnih vezij v jeziku VHDL, FE 2011

Programirljiva vezja: CPLD, FPGA

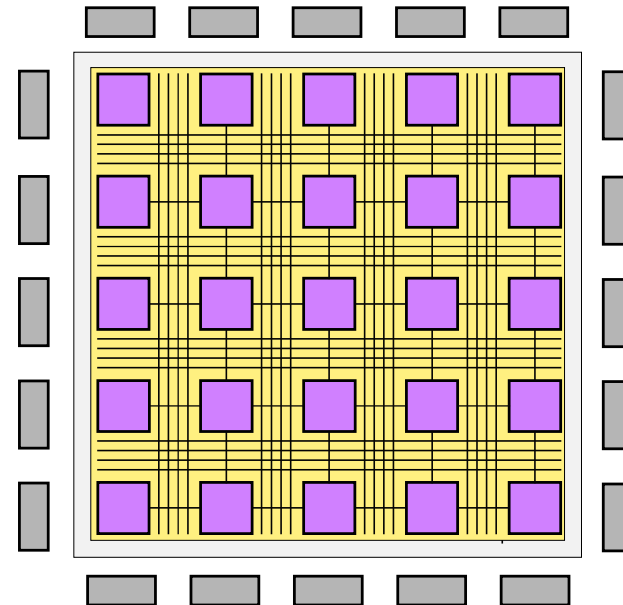
Complex Programmable Logic Device

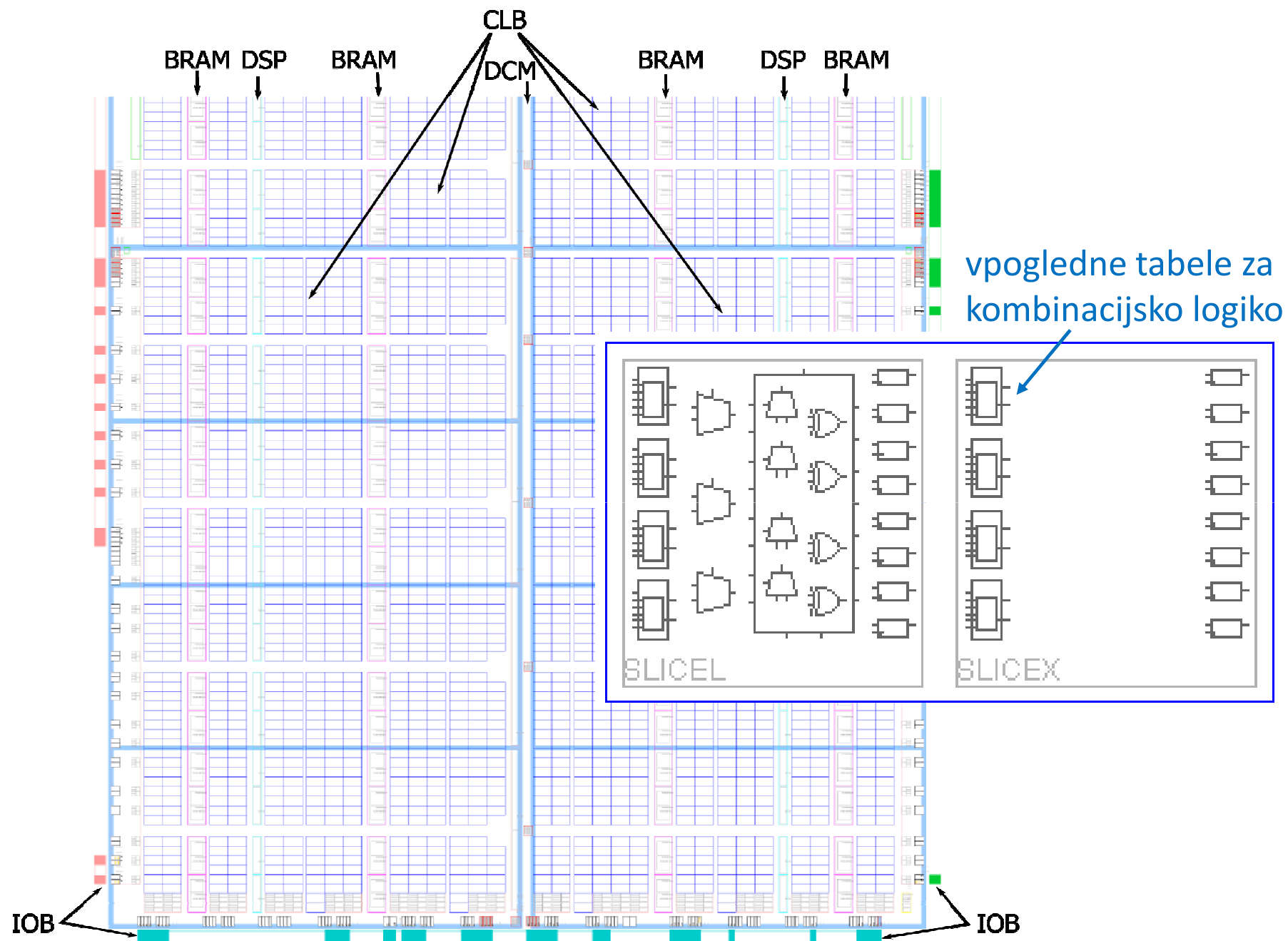
- ▶ Makro-celice s FF in povezovalno polje
- ▶ 1.000-20.000 log. vrat, 50-500 FF
- ▶ FLASH tehnologija, 1.8V



Field Programmable Gate Array

- ▶ Matrika log. celic in povezovalno polje
- ▶ 10.000-10.000.000 vrat, 100k RAM
- ▶ CMOS tehnologija, 1.2V





Razvoj vezij FPGA

► Družine vezij proizvajalca **Xilinx**

2001	2003	2004	2006	2009	2011
Virtex-II	Virtex-II pro	Virtex-4	Virtex-5	Virtex-6	Virtex-7
	Spartan-3			Spartan-6	Artix-7

Spartan-3	XC3S50	XC3S200	...	XC3S5000
matrika CLB	16 x 12	24 x 20		104 x 80
flip-flopov	1536	3840		66560
BRAM	4 (9kB)	12 (27kB)		104 (234kB)
vh-izh	124	173		633
cena	\$12	\$15		\$160

- Osnovni paket razvojnih orodij je brezplačen
 - Xilinx ISE WebPACK 12.3 (www.xilinx.com, 10GB)

The diagram illustrates the AT91SAM7S64 microcontroller system architecture. At the center is the **MICROPROCESSOR**. It is connected to an **LCD DISPLAY** through an **LCD CONTROLLER** (labeled XAPP904). A **KEYPAD SCANNER** (labeled XAPP512) is also connected to the microprocessor. For communication with other devices, the microprocessor uses an **I²C PORT EXPANDER** (labeled XAPP799) and an **SMB/I²C** interface (labeled XAPP385). A central **BUS** connects the microprocessor to five main peripheral interfaces: **NAND FLASH INTERFACE** (XAPP354) for **NAND FLASH** storage, **COMPACT FLASH INTERFACE** (XAPP398) for **COMPACT FLASH** storage, **MMC/SD INTERFACE** (XAPP906) for **MMC/SD** storage, **DDR INTERFACE** (XAPP384) for **DDR SDRAM**, and **MEMORY INTERFACE** (XAPP394) for **MOBILE SDRAM**.

The diagram illustrates the SPARTAN-3E SoC architecture, centered around a blue box labeled **SPARTAN-3E** at the bottom. The internal components are organized into three columns:

- Left Column (Red Boxes):** 32-Bit CPU, SDRAM (two blocks), RF Tuner, AC97 CODEC, and Drive Unit.
- Middle Column (Blue Boxes):** Processor Interface, DDR Memory Controller, Filtering & Formatting, User Interface, AC97 I/F, and ATAPI / IDE.
- Right Column (Blue Boxes):** VGA Controller, PCI Bridge, CAN 2.0B Controller, PCMCIA, and Discrete Logic.

External components and their connections are shown on the left and right:

- Left Side:** A satellite icon connects to the RF Tuner. Speakers connect to the AC97 CODEC. A DVD icon connects to the Drive Unit. External Control (represented by three arrows) connects to the User Interface.
- Right Side:** A TFT LCD connects to the VGA Controller. An Entertainment Network (MOST) connects to the PCI Bridge. Body Electronics connects to the CAN 2.0B Controller. A Plug-In Card connects to the PCMCIA interface.

Internal connections are shown with blue double-headed arrows between adjacent components in the same row and between the CPU and the Processor Interface.

- ▶ digitalni osciloskop, DMX konzola, GPS, VGA video igrice, MIDI vmesniki, mikroprocesorji, audio spektralni analizator
- ▶ **Tekmovanje študentskih projektov 2012**

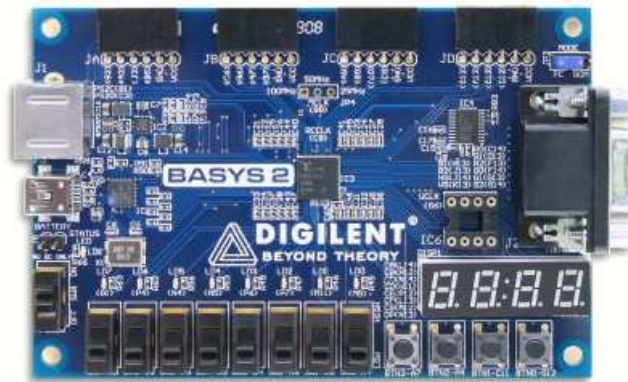
- ▶ digitalni osciloskop, DMX konzola, GPS, VGA video igrice, MIDI vmesniki, mikroprocesorji, audio spektralni analizator
- ▶ **Tekmovanje študentskih projektov 2012**

Razvojni sistemi



Digilent CPLD
CoolRunner II

- ▶ razvojni sistem za lab. vaje in projekte
XC3S100E ali XC3S250E (Digilent)



Digilent Spartan-3E Board

- ▶ pri nas razviti FPGA razvojni moduli
 - ▶ <http://lniv.fe.uni-lj.si/Spartan3Modul.html>
 - ▶ XC3S50E + SRAM + VGA

Spartan-3 XC3S200
razvojni modul (Lab)



Preverjanje delovanja razvitega vezja

- ▶ Simulacija na osebnem računalniku

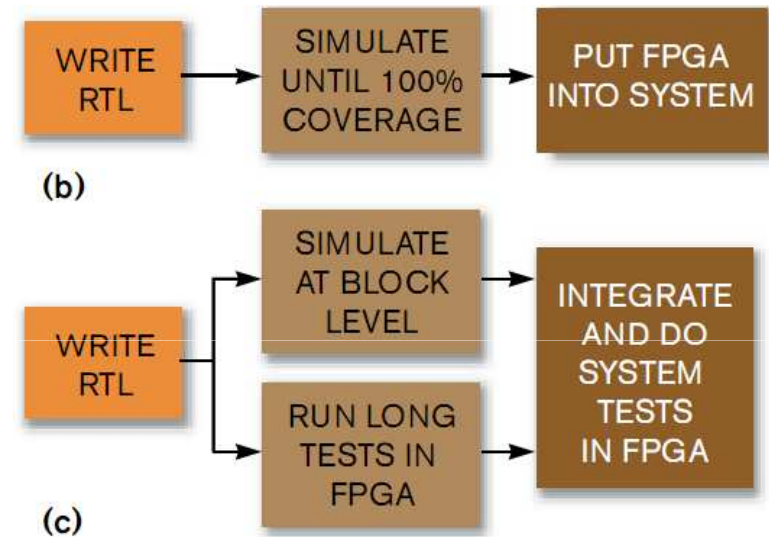
- ▶ funkcionalna in časovna simulacija
- ▶ ASIC100% simulacijo > cena napake!

- ▶ Programirljiva vezja

- ▶ vsaj funkcionalna simulacija blokov
 - ▶ problem je simulacijski čas
- ▶ emulacija vezja
 - ▶ problem je vidnost signalov, čas priprave in možnost priprave kompleksnih testov

- ▶ Mikroprocesorji

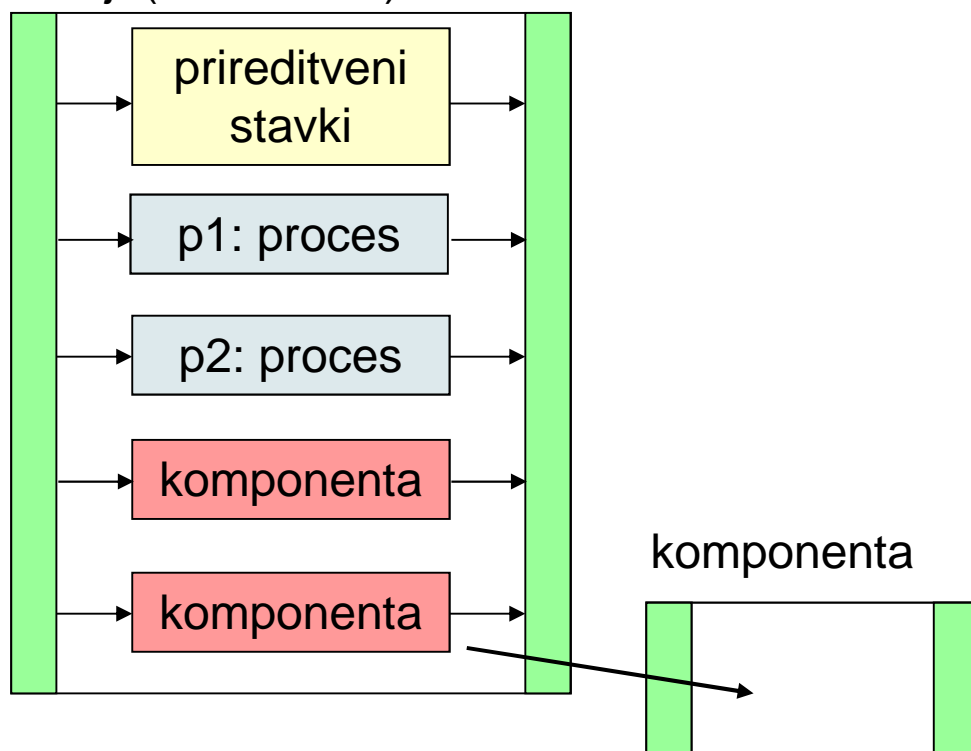
- ▶ debugiranje v sistemu ali emulatorju



R. Wilson, Verifying FPGA designs, EDN, feb 2009

- ▶ VHDL je standardni jezik za opis vezja
 - ▶ VHDL ni programski jezik !
 - ▶ stavki za opis vezja se izvajajo paralelno
- ▶ Opis zgradbe vezja

vezje(arhitektura)



VHDL

Very high-speed IC

Hardware

Description

Language

- ▶ opis postopkov, obnašanje vezja

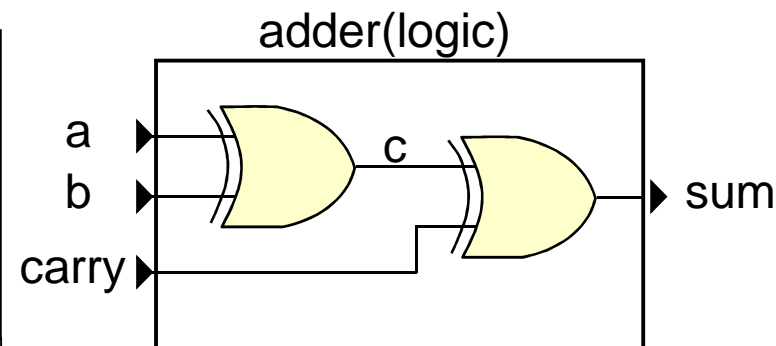
```
p1: process(pretok)
begin
  ventil <= '0';
  if pretok > 10 then
    ventil <= '1';
  end if;
end process;
```


Funkcijski opis vezja v jeziku VHDL

- ▶ stavki opisujejo gradnike vezja
 - ▶ vrstni red stavkov ni pomemben (sočasni stavki)

```
entity adder is  
  port ( a, b : in std_logic;  
         carry : in std_logic;  
         sum : out std_logic);  
end adder;
```

```
architecture logic of adder is  
  signal c : std_logic;  
begin  
    sum <= c xor carry;  
    c <= a xor b;  
end one;
```



deklaracija notranjega signala

Podatkovni tipi: vektorji

- ▶ Območje indeksov običajno deklariramo od MSB proti LSB

```
signal a, b, c: std_logic_vector(7 downto 0);  
signal high, low: std_logic_vector(3 downto 0);
```

- ▶ Podvektor

```
high <= a(7 downto 4);  -- 4 bitni podvektor  
low  <= a(3 downto 0);  -- 4 bitni podvektor  
sign <= a(7);           -- sign je tipa std_logic
```

- ▶ Predznačeni ali nepredznačeni vektorji: IEEE.numeric_std
- ▶ Definirane osnovne aritmetične operacije

```
use IEEE.numeric_std.all;  
  
architecture one of test is  
  signal a, b, sum: unsigned(7 downto 0);  
begin  
  sum <= a + b;      -- 8 bitni seštevalnik
```

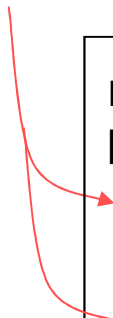
Sekvenčni elementi in sinteza vezja

- ▶ Program za sintezo zna narediti register ali flip-flop, če uporabimo ustrezno obliko opisa vezja

I. oblika: popolnoma sinhrono vezje

```
reg: process (clk)
begin
  if rising_edge(clk) then
    q <= d;
  end if;
end process;
```

II. oblika: asinhroni reset



```
reg: process (clk, reset)
begin
  if reset='1' then
    q <= "00000000";
  elsif rising_edge(clk) then
    q <= d;
  end if;
end process;
```