

# Digitalno načrtovanje

2017/18

Izvajalec: prof. dr. Patricio Bulić

Asistent: Rok Češnovar

# Potek vaj

- opisovanje strojne opreme z jezikom VHDL
- izgradnja sistema na čipu
  - izgradnja VGA krmilnika
  - izgradnja PS2 krmilnika
  - vključitev in povezovanje PicoBlaze CPE
  - ...
- Pogoji za opravljene vaje
  - sprotne naloge
  - seminar

# VHDL

- VHSIC Hardware Description Language
  - VHSIC = very-high-speed integrated circuits
- Jezik za opisovanje strojne opreme/digitalnega vezja ter modeliranje/simulacijo vezij
- VHDL opis bomo sintetizirali in programirali programabilno logično vezje (FPGA)

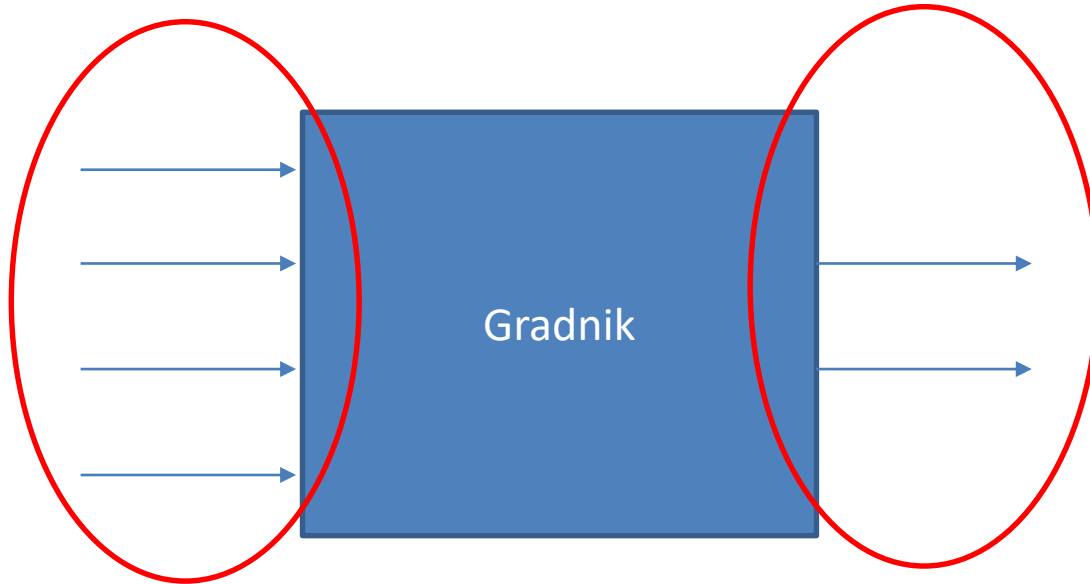
# Oprema

- Programska oprema
  - Xilinx ISE Webpack (Povezava na e-učilnici)
- Strojna oprema
  - Razvojna plošča Digilent Nexys2
    - Xilinx Spartan 3E-500 FPGA s 50 MHz uro
    - Vhod/izhod: VGA, PS2, RS232, gumbi, stikala, LED, 7 segmentni prikazovalnik
  - Razvojna plošča Digilent Nexys4
    - Xilinx Artix-7 s 100 MHz uro
    - Vhod/izhod: VGA, USB (PS2 emulator), LED, gumbi, stikala, 7 segmentni, 3barvne let, akcelerometer, ...

# Opis osnovnega gradnika

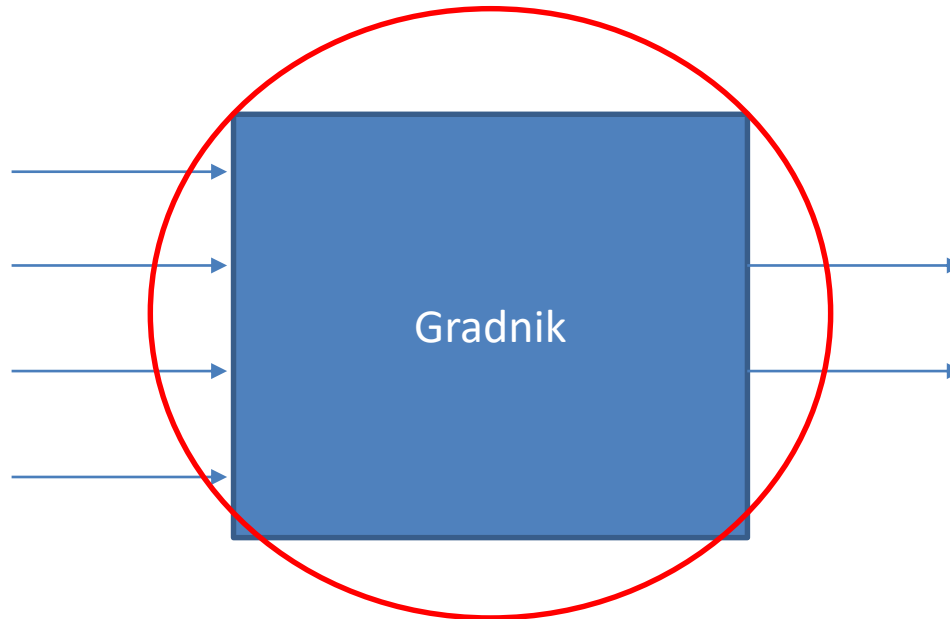


# Opis osnovnega gradnika



Definiramo zunanje signale gradnika

# Opis osnovnega gradnika



Definiramo delovanje gradnika – kaj "počne" s signali

# Opis gradnika v VHDL

```
entity ime_gradnika is  
    //deklaracija zunanjih signalov vezja  
    port( ... );  
end ime_gradnika;
```

```
architecture opis_vezja of ime_vezja is  
    //deklaracija notranjih signalov  
begin  
    //stavki za opis vezja  
end opis_vezja;
```



# Deklaracija zunanjih signalov

```
entity ime_vezja is
port (
    ime_signala : smer tip_signala;
    ime_signala_2 : smer tip_signala;
    ...
    ime_signala_n: smer tip_signala
);
end ime_vezja;
```

- Smer: in, out, inout, buffer
- Tip signala: std\_logic, std\_logic\_vector()

# Primera deklaracij

```
entity ime_vezja is
port (
    a: in std_logic;
    b: out std_logic
);
end ime_vezja;
```

```
entity ime_vezja is
port (
    a: in std_logic_vector(7 downto 0);
    b: out std_logic_vector(0 to 4)
);
end ime_vezja;
```

# Deklaracija notranjih signalov

```
architecture Behavioral of ime_vezja is
    signal ime_signala: tip_signala;
begin ...
```

# Prireditveni stavek

Sintaksa:

```
signal <= izraz;
```

Primeri:

```
a <= '0'; //prireditv konstante
```

```
b <= "01001"; //prireditv konstante za vektor
```

```
c(3 downto 0) <= "0111"; //prireditv delu  
vektorja
```

# Primer – dvovhodna logična vrata

```
entity and_xor_vrata is
    port(
        a: in std_logic;
        b: in std_logic;
        c: out std_logic;
        d: out std_logic
    );
end and_xor_vrata ;
```

# Opis parametrov in signalov

architecture Behavioral of and\_xor\_vrata is

begin

    //stavki za opis vezja

    c <= a and b;

    d <= a xor b;

end Behavioral ;

# Signali gradnika in FPGA

- Pred sintezo je za glavni (zunanji) gradnik potrebno določiti kam naj bodo povezani zunanji signali
  - npr.: a in b sta stikala, c & d led diode
- To počnemo v t.i. UCF datoteki ( user constraints file )
- Sintaksa

```
NET "a" LOC = "J14";
```

```
NET "b<1>" LOC = "H18";
```

```
NET "b<0>" LOC = "G18";
```

- Oznako fizičnega vhoda/izhoda najdete v reference manual-u razvojne plošče ali neposredno na razvojni plošči

# Projekt v Xilinx ISE Design Tools

## Nexys4

- Odprite Project Navigator
- File -> New Project
  - Izberite ime in direktorij
- Izberite naslednje nastavitve -> Next -> Finish

Property Name	Value
Top-Level Source Type	HDL
Evaluation Development Board	None Specified
Product Category	All
Family	Artix7
Device	XC7A100T
Package	CSG324
Speed	-1
Synthesis Tool	XST (VHDL/Verilog)
Simulator	ISim (VHDL/Verilog)
Preferred Language	VHDL
Property Specification in Project File	Store all values
Manual Compile Order	<input type="checkbox"/>
VHDL Source Analysis Standard	VHDL-93
Enable Message Filtering	<input type="checkbox"/>



# Dodajanje novega modula

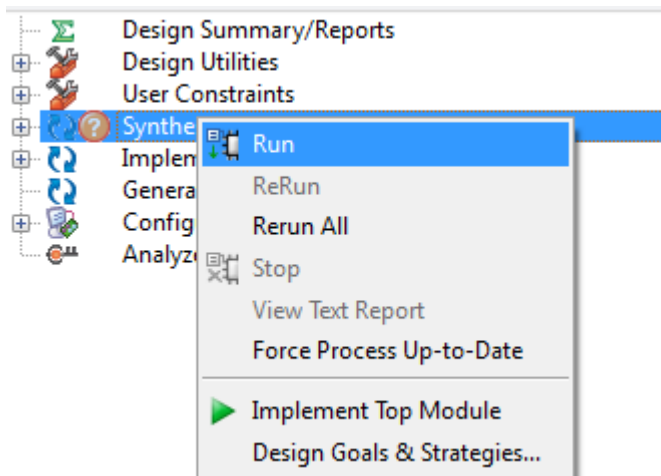
- Project -> New Source
- Izberite VHDL Module
- Izberite ime -> Next -> Next -> Finish

# Dodajanje UCF datoteke

- Project -> New Source
- Izberite Implementation Constraints File
- Izberite ime -> Next -> Finish

# Preverjanje sintakse & sinteza

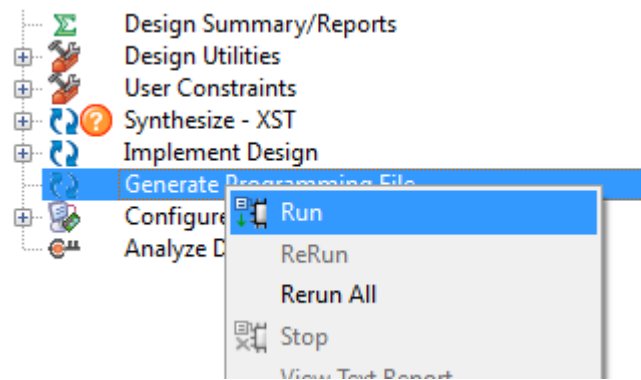
- Desni klik na Synthesize -> Run



- Če želite samo preveriti sintakso, poženite Check Syntax

# Programiranje FPGA čipa

- Desni klik na Generate Programming File



- Če se pojavi zelena kljukica, lahko programiramo FPGA čip

# Programiranje FPGA čipa

- Dvoklik na Configure Target Device
- ISE iMPACT
  - Boundary Scan
  - Desni klik v desnem prostoru -> Initialize Chain
  - Yes
  - V mapi projekta poiščite datoteko s končnico .bit
  - No
  - Izberite Bypass in OK
  - Desni klik na XC3S500E (ali XC7A100T) in Program

# Pogojni prireditveni stavek

```
signal <= izraz1 when pogoj1 else izraz2;
```

```
signal <= izraz1 when pogoj1 else  
        izraz2 when pogoj2 else  
        izraz3;
```

Pogojni operatorji:

- enako, ni enako                   =, /=
- večje, manjše, ...               >, <, >=, <=

# Naloge

- Realizirajte sledeče gradnike, ki jih testirajte s pomočjo stikal in LED diod
  - 2/1 multiplekser
  - primerjalnik dveh 4bitnih števil
    - izhod=2, ko je prvo število večje
    - izhod=1, ko je prvo število manjše
    - izhod=0, ko sta števili enaki