

VHDL jezik za opis hardvera

Prof. Dr Predrag Petković

Doc. Dr Miljana Milić

Sadržaj

- 1. Šta je VHDL?**
- 2. Opis hardvera**
- 3. Signali**
- 4. Osnove VHDL pravopisa**
- 5. Opis osnovnih logičkih operacija**
- 6. Stilovi opisa**

1 (a) Šta je VHDL?

Very high speed integrated circuits

Hardware

Description

Language



1 (b) Šta je VHDL?

Zašto?

da se popravi komunikacija medju projektantima koji rade na razvoju istog integrisanog kola.

Standardizaciju je iniciralo i finansiralo Ministarstvo odbrane SAD-a (US DoD).

Kad je nastao i kako se razvija?

**VHDL je 1987. godine usvojen kao standard:
IEEE Std. 1076-1987**

VHDL-93,

**VHDL AMS (što dolazi od Analogue and
Mixed Signals) VHDL-99.**

1 (d) Šta je VHDL?

ŠTA OMOGUĆAVA?

VHDL može da se primenjuje u svim fazama projektovanja:

opis,

verifikaciju (simulaciju),

sintezu i

dokumentovanje.

Mogućnost jednostavnog opisa projekta na višim nivoima apstrakcije.

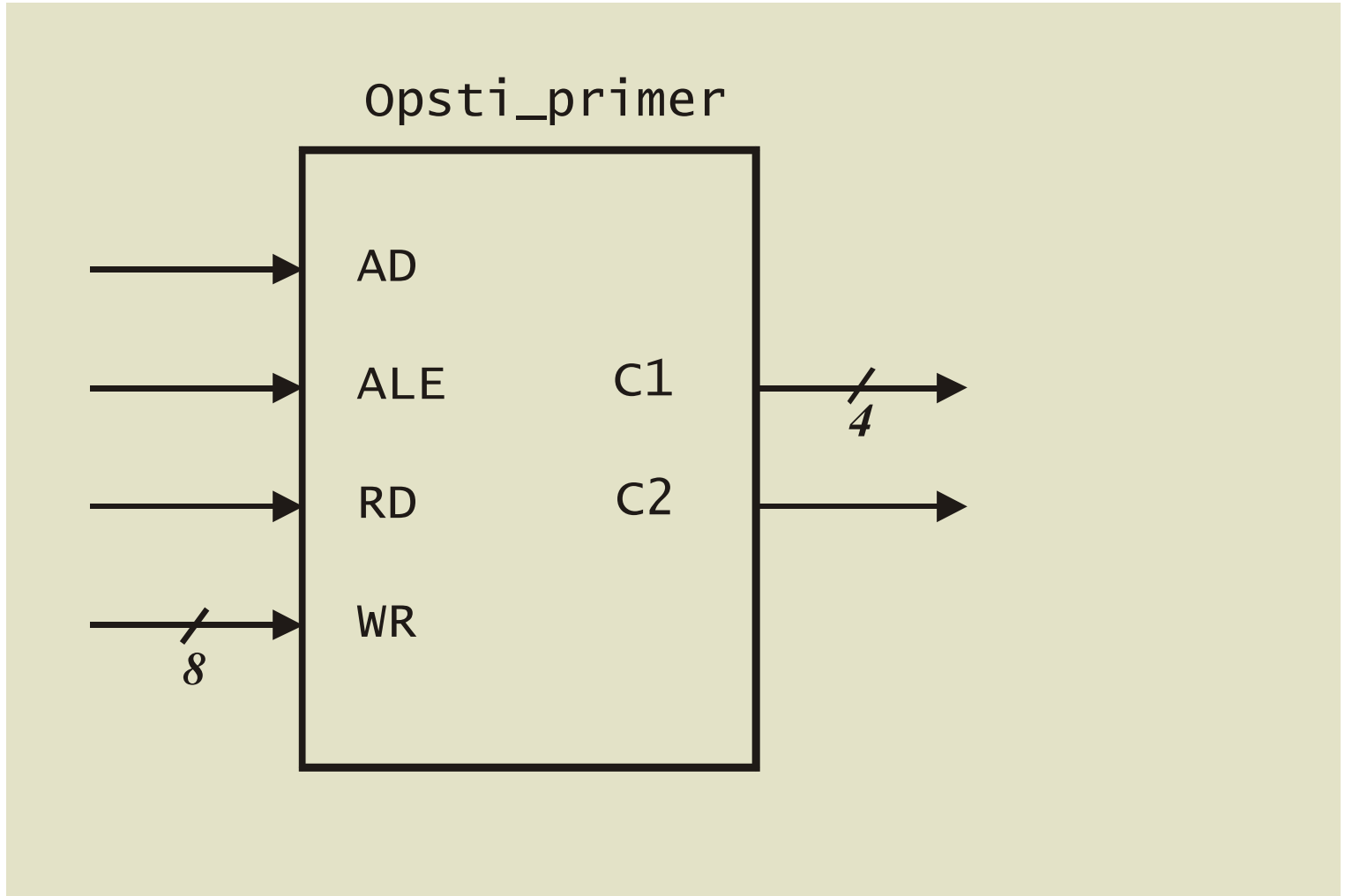
2 (a) Opis hardvera

VHDL opis AND kola

```
entity AND2 is
  port (a, b: in bit ;
        c : out bit);
end AND2;

architecture beh of AND2 is
begin
  c <= a and b;
end beh;
```

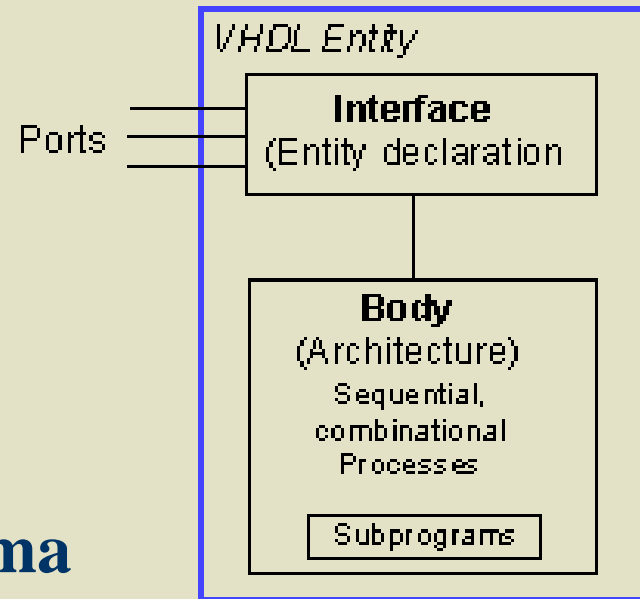
2 (a) Opis hardvera



2 (b) Opis hardvera

Svaki blok VHDL prepoznaje kao entitet koji se definiše preko ključne reči *entity*.

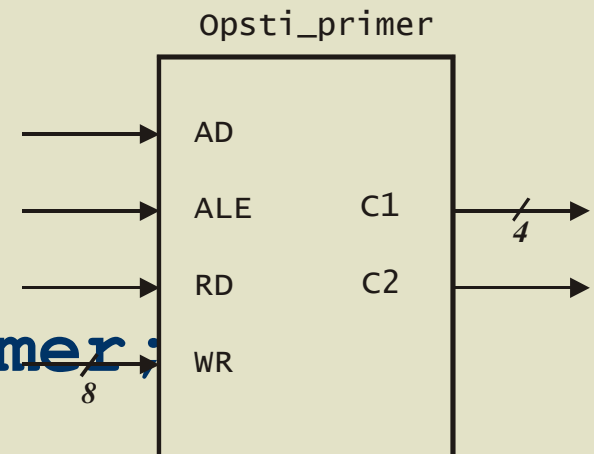
Arhitektura funkcionalno povezuje stanja na ulaznim i izlaznim portovima entiteta. Opis arhitekture počinje sa *architecture*.



2 (c) Opis hardvera

```
entity Opsti_primer is port(  
  AD ,  
  ALE , RD ...  
  WR ...  
  C1 ...  
  C2 ...  
);  
end entity Opsti_primer;
```

deklaracija portova



3 (a) Signali

Jedan entitet povezuje se sa drugim preko portova.

Kroz portove putuju signali, a kako se radi o opisu digitalnih kola, reč je o digitalnim signalima.

Zato i definicija signala u VHDL-u ima sve attribute koji se javljaju u realnim digitalnim kolima:

Vrednost

Tip

Mod



3 (b) Signali

Svaki signal nosi *informaciju o logičkom stanju* ili logičkoj *vrednosti* `std_logic` (standard IEEE 1164):

vrednost	značenje
U	neinicirani signal (<i>Uninitialized</i>)
X	jako nepoznato stanje
0	jaka nula
1	jaka jedinica
Z	visoka impedansa
W	slabo nepoznato stanje (<i>Weak</i>)
L	slaba nula (<i>Low</i>)
H	slaba jedinica (<i>High</i>)
-	nebitno stanje (<i>don't care</i>)

3 (c) Signali

Tip signala:

kroz jednu *žicu*, prolazi signal tipa **bit**
std_logic,

kroz *magistralu*, signal je tipa **vektor bitova**
std_logic_vector.

Mod signala:

ulazni ⇔ *in*

Entitet može samo da čita sadržaj ovog signala, a ne može mu dodeljivati novu vrednost

izlazni ⇔ *out*

vrednost mu se dodeljuje unutar entiteta, a njegov sadržaj ne može da se koristi za pobudu drugih ulaza istog entiteta

bidirekcionni ⇔ *inout*

3 (d) Signali

Redosled bitova u signalu tipa *std_logic_vector*:

MSB levo:

```
std_logic_vector(3 downto 0)
```

0101 ⇔ 5

MSB desno:

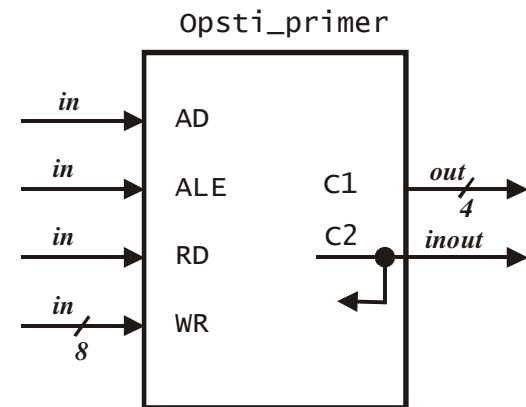
```
std_logic_vector(0 to 3)
```

0101 ⇔ 10



3 (e) Signali

Redosled navođenja :
naziv porta : mod tip



```
entity Opsti_primer is port
( AD: in std_logic;
  ALE, RD: in std_logic;
  WR: in std_logic_vector (7 downto 0);
  C1: out std_logic_vector (3 downto 0);
  C2: inout std_logic);
end entity Opsti_primer;
```


2 (d) Opis hardvera

```
architecture proba of Opsti_primer  
is
```

-
- **deklaracija signala**
-

```
begin
```

-
- **telo arhitekture**
-

```
end proba;
```

2 (e) Opis hardvera

Kao što se u elektronici jedna ista funkcija može realizovati na više načina, tako i jednom entitetu može da se pridruži više arhitektura.

Obrnuto ne važi jer jedna arhitektura može da obavlja samo funkciju zbog koje je projektovana.

Očigledno je struktura VHDL jezika prilagodjena osnovnim strukturama koje se javljaju tokom projektovanja elektronskih kola.

2 (e) Opis hardvera

Razliciti stilovi opisa kola

4(a) Osnove VHDL pravopisa

Svaki jezik karakteriše skup sintaksnih pravila za pisanje kojih se korisnici moraju pridržavati. Ta pravila mogu da se podele u dve osnovne kategorije:

- **način označavanja pojmova**
- **format navodjenja pojmova.**

Način označavanja pojmova

- a. Ne pravi se razlika između malih i velikih slova;
- b. Svi nazivi moraju da počnu slovnim znakom
- c. Koristite samo slova (a-z i A-Z), brojeve (0-9) i podvučenu crtu (_)
- d. NE koristite znakove interpunkcije
!, ?, ., ,, i.t.d.
- e. NE koristite sukcesivno dva simbola (__)

4(c) Osnove VHDL pravopisa

Ispravne oznake

Prvi_primer

Opsti_Primer0

OPSTI_PRIMER_9

Opsti_Primer_0

Neispravne oznake

1_Primer

Opsti_#1_primer

Opsti!_primer

Opsti__primer

Razlog

prekršeno pravilo b

prekršeno pravilo c

prekršeno pravilo d

prekršeno pravilo e

4(d) Osnove VHDL pravopisa

Način označavanja pojmova (nastavak)

f. NE mogu dva različita pojma u okviru istog entiteta ili arhitekture imati ista imena, odnosno oznake

g. Postoje rezervisane reči koje se NE SMEJU koristiti za označavanje pojmova (date u svakom priručniku za VHDL)

4(e) Osnove VHDL pravopisa

rezervisane reči

abs	downto	library	procedure	subtype
access	else	linkage	process	then
after	elsif	literal*	pure*	to
alias	end	loop	range	transport
all	entity	map	record	type
and	exit	mod	register	units
architecture	file	nand	reject*	unaffected*
array	for	new	rem	until
assert	function	next	report	use
attribute	generate	nor	return	variable
begin	generic	not	rol*	wait
block	group*	null	ror*	when
body	guarded	of	select	while
buffer	if	on	severity	with
bus	impure*	open	shared*	xor
case	in	others	signal	xnor*
component	inertial*	out	sla*	
configuration	inout	package	sll*	
constant	is	port	sra*	
disconnect	label	poostponed*	srl*	

4(f) Osnove VHDL pravopisa

Način označavanja pojmova (nastavak)

h. Linijski komentari počinju dvostrukim minus znakom “_”

```
entity Opsti_primer is port
(
  AD: in std_logic;           -- svaki port moze da se deklarisu u
                              -- posebnoj liniji
  ALE, RD: in std_logic;     -- portovi istog tipa mogu da se
                              -- deklarise u istoj liniji
  WR: in std_logic_vector (7 downto 0); -- tip bit-vektor mora
                              -- da sadrzi duzinu i
                              -- redosled, 7. je MSB

  C1: out std_logic_vector (0 to 3); -- 0. je MSB, 3. - LSB
  C2: inout std_logic         --ovde nema ';'
);                             --jer se ';' stavlja iza zaokružene
                              --logičke celine koja sadrži
                              --i zatvorenu zagradu
end entity Opsti_primer;
```



4(g) Osnove VHDL pravopisa

Način označavanja pojmova (nastavak)

i. Ne postoji ograničenje u broju karaktera kojim se može označiti neki pojam **IPAK** ne treba preterivati!!!

Međutim neki programi za sintezu prepoznaju najviše 32 karaktera. Zato se preporučuje da broj karaktera u jednoj oznaci ne prelazi 32. **Dobro je da oznake budu dovoljno duge da ukažu na pravo značenje, ali da ne budu i predugačke.**

Dobra je praksa da se signali nazovu

clock, **data** ili **global_input**, a ne **c**, **d**, ili **g**.



4(h) Osnove VHDL pravopisa

Format pisanja treba sagledati sa stanovišta opisa pojedinih celina.

Celinu čini:

- jedna naredba,**
- struktura naredbi ili**
- grupa naredbi.**

Pod jednom *naredbom* podrazumevamo

- opis koji počinje nekom od rezervisanih reči ;
- opis aktivnosti koja označava dodeljivanje vrednosti nekom signalu.

4(j) Osnove VHDL pravopisa

strukturu naredbi čini više povezanih naredbi sa jasnim semantičkim značenjem.

Primer

if...then...elseif...else struktura:

```
if uslov1 then akcija1;  
elseif uslov2 then akcija2;  
else akcija3;  
end if;
```

4(k) Osnove VHDL pravopisa

Grupa naredbi obično počinje:

- ključnom rečju,
- imenom bloka ili
- labelom (oznakom),

može da sadrži:

- naredbe kojima se deklarišu promenljive koje se javljaju u toj grupi,
- ključnu reč za početak opisa tela grupe “begin”,
- telo i
- ključnu reč “end” kojom se opis grupe naredbi završava



4(I) Osnove VHDL pravopisa

- Svaka logička celina završava se znakom “;”.
- U okviru jedne logičke celine VHDL koristi “slobodni format” za pisanje koda.

Primer:

```
if uslov1 then akcija1; --nacin 1
```

```
if uslov1 then akcija1; --nacin 2
```

```
if uslov1  
    then akcija1; -- nacin 3
```



5 (a) Opis osnovnih logičkih operacija

- Pored ostalog VHDL podržava korišćenje osnovnih logičkih operatora nad signalima tipa *std_logic*.

operator	značenje
NOT	Logička operacija komplementa
AND	Logička I operacija
NAND	Logička NI operacija
OR	Logička ILI operacija
NOR	Logička NILI operacija
XOR	Logička Ekskluzivno ILI operacija
XNOR	Logička Ekskluzivno NILI operacija

5 (b) Opis osnovnih logičkih operacija

- Dodeljivanje logičke vrednosti nekom signalu ili portu označava se simbolom:

<=

Operacije sa većim brojem signala, npr. a i b može da se iskaže na sledeći način:

```
rezultat <= a AND b;
```

5 (c) Opis osnovnih logičkih operacija

- Operator NOT ima najveći prioritet, dok su svi ostali operatori istog prioriteta. Redosled njihovog izvršavanja određuje se redosledom navođenja u naredbi. Tako u naredbi:

```
rezultat <= a AND b OR c XOR d;
```

izvrši se najpre AND operacija, zatim OR i na kraju XOR, što može da se iskaže kao

```
(( (a AND b) OR c) XOR d) .
```

5 (c) Opis osnovnih logičkih operacija

Međutim u opisu:

```
rezultat <= a AND NOT b OR c XOR d;
```

redosled izvršavanja operacija jeste

```
(( (a AND (NOT b)) OR c) XOR d) .
```

5 (c) Opis osnovnih logičkih operacija

Redosled izvršavanja operatora kontroliše se upotrebom zagrada:

```
rezultat <= (a AND b) OR (c XOR d) ;
```

Najpre se obave AND i XOR operacija, a zatim OR. Da bi se povećala čitljivost koda, preporučuje se korišćenje zagrada.

6(a) Stilovi opisa projekta

VHDL podržava opis projekata na:

- algoritamskom nivou i
- nivou logičkih jednačina.

S obzirom da je VHDL i razvijan sa ciljem da se efikasno opišu složena kola, rezultujući kôd karakteriše jezgrovitost koja proističe iz hijerarhijskog pristupa dekompoziciji projekta.

6(b) Stilovi opisa projekta

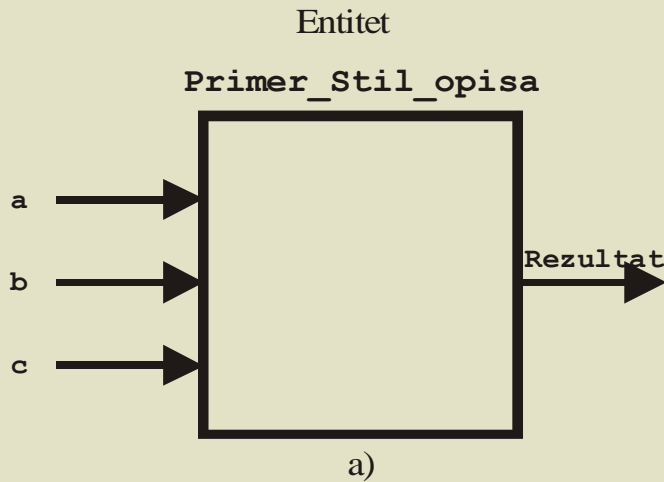
Postoje tri stila opisa projekta u VHDL-u. To su

- **strukturni opis (structural)**
- **opis ponašanja (behavioral)**
- **opis toka podataka (dataflow)**

6(c) Stilovi opisa projekta

Primer:

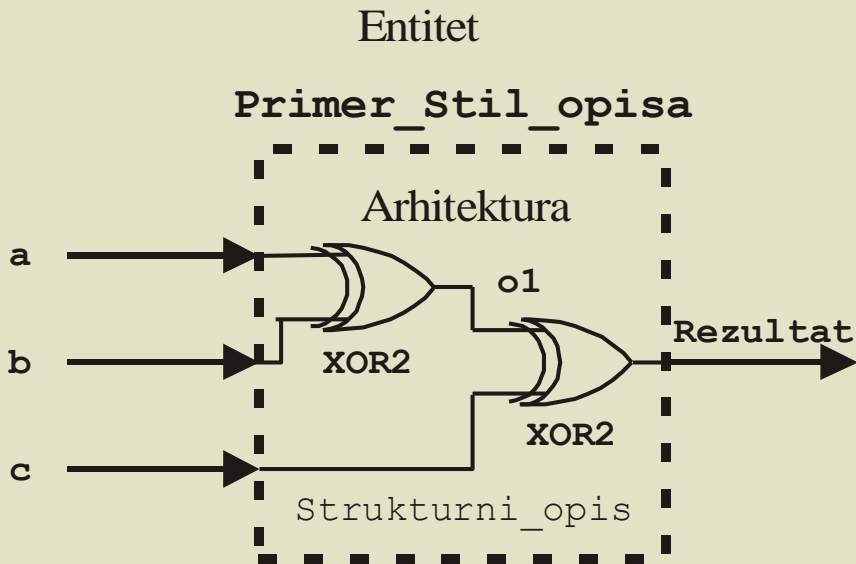
$$\text{Rezultat} = (a \otimes b) \otimes c$$



```
entity Primer_Stil_opisa is
port
(
  a: in std_logic;
  b: in std_logic;
  c: in std_logic;
  Rezultat: out std_logic
);
end Primer_Stil_opisa;
```

6(d) Stilovi opisa projekta

Strukturni opis:



```
architecture Strukturni_opis of
    Primer_Stil_opisa is
    signal o1: std_logic; --deklaracija
                        -- internih signala
    begin
    u1:xor2 port map ( a => I1,
                      b => I2,
                      o1 => Y); -- telo
    u2:xor2 port map ( o1 => I1,
                      c => I2,
                      Rezultat => Y);
    end architecture Strukturni_opis;
```


6(e) Stilovi opisa projekta

Opis ponašanja:

```
architecture Opis_Ponasanja of
  Primer_Stil_opisa is
  XOR_od_3: process (a, b, c)
    -- imenovanje (labeliranje) procesa
    -- nije neophodno, ali je korisno
  begin
    if ((a XOR b) XOR c) = '1' then
      Rezultat <= '1';
    else
      Rezultat <= '0';
    end if;
  end process XOR_od_3;
end architecture Opis_Ponasanja;
```

6(f) Stilovi opisa projekta

Opis toka podataka:

```
architecture Dataflow of Primer_Stil_opisa
is
    signal o1: std_logic; -- deklaracija
                          -- internog signala
begin
    o1 <= a XOR b;
    Rezultat <= o1 XOR c;
end architecture Dataflow;
```

Šta treba da znamo?

Elementarno (za potpis)

Šta omogućava VHDL?

Osnovna (za 6)

- 1. Koji par ključnih reči opisuje osnovne osobine hardvera u VHDLu?**
- 2. Stilovi opisa kola u VHDLu.**

Šta treba da znamo?

Ispitna pitanja

- a) Šta je i kako se definiše port?
- b) Šta su i kako se definišu signali?
- c) Oznaka redosleda bitova kod tipa `std_logic_vector`?
- d) Pravila za označavanje pojmova.