

VLSI dizajn

Osnove jezika VHDL

# Tipovi podataka

VHDL je tipski jezik: Tipovi podataka VHDL portova, signala i promenljivih moraju biti određenog tipa.

- Podržani su skalarni I kompozitni tipovi.
- Skalarni tipovi su: Integer, Real, Enumerated, Physical
- Kompozitni tipovi su nizovi i slogovi.

# Tip integer

- Celobrojni tip
- Zauzima 4 bajta
- Vrednosti su u opsegu  $[-2^{31}, +2^{31}-1]$
- Primer upotrebe:

```
ARCHITECTURE explain_integer OF expl IS
BEGIN
    PROCESS(A)
        VARIABLE a, b:INTEGER;
    BEGIN
        a := 1;
        b := a + 10;
    END PROCESS;
END explain_integer
```

# Tip real

- Služi za čuvanje realnih brojeva (sa zarezom)
- Zauzima 4 bajta
- Vrednosti su u okvirnom opsegu od  $(-10^{38}, +10^{38})$
- Primer upotrebe:

```
ARCHITECTURE explain_real OF expl IS
```

```
BEGIN
```

```
    PROCESS(A)
```

```
        VARIABLE a, b:REAL;
```

```
    BEGIN
```

```
        a := 1.0;
```

```
        b := a + 10e-1;
```

```
    END PROCESS;
```

```
END explain_real
```

# Tip enumerated

- Omogućava korisniku da definiše moguće vrednosti promenljivih
- Uglavnom je dovoljan 1 bajt, odnosno uglavnom korisnik ne navodi više od 255 mogućih stanja
- Primer upotrebe:

```
TYPE action_enabled IS (TRUE, FALSE);  
ARCHITECTURE explain_enumerated OF expl IS  
BEGIN  
    PROCESS(A)  
        VARIABLE a, b:REAL;  
    BEGIN  
        a := TRUE;  
        b := a;  
    END PROCESS;  
END explain_enumerated
```

# Fizički tipovi

- Omogućavaju korisniku da definiše fizičke veličine
- Zahteva se fizička veličina pored vrednosti promenljive
- Opseg mora biti definisan
- Primer upotrebe:

```
TYPE mass IS RANGE 0 to 1000000;
```

```
UNITS
```

```
g; – gram
```

```
kg = 1000 g;
```

```
Ton = 1000 kg;
```

```
END UNITS;
```

# Kompozitni tipovi: Nizovi

- Omogućavaju korisniku da definiše nizove kao jedan tip
- Često se koristi kod magistrala podataka
- Primer upotrebe:

```
TYPE address IS ARRAY (0 to 31) OF BIT;
```

```
...
```

```
VARIABLE x1 : address;
```

```
VARIABLE pin : bit;
```

```
...
```

```
pin = x1(10); -- vraćanje 10. bita adrese
```

# Kompozitni tipovi: slogovi

- Omogućavaju korisniku da definiše složene tipove

- Primer upotrebe:

```
TYPE field IS
```

```
  RECORD
```

```
    size : integer;
```

```
    speed : real;
```

```
  END RECORD;
```

```
...
```

```
VARIABLE a : field;
```

```
...
```

```
a.size = 10;
```

```
a.speed = 10.1;
```



# VHDL objekti

Postoje sledeće 4 vrste objekata u jeziku VHDL:

- Konstante
- Promenljive
- Signali
- Fajlovi

# Konstante

- Predstavljaju nazive za određene vrednosti
- Primer upotrebe:

```
CONSTANT PI : REAL := 3.14;  
CONSTANT SIZE : REAL;
```

# Promenljive

- Predstavljaju konvencionalan mehanizam za skladištenje vrednosti
- Primer upotrebe:

```
VARIABLE a : REAL := 3.14;
```

# Signali

- Predstavljaju konvencionalan mehanizam za komunikaciju između komponenti
- Obično se pravi signali hardvera mapiraju na VHDL signale
- Primer upotrebe:

```
SIGNAL a : BIT;
```

```
...
```

```
a <= '0' AFTER 10ns;
```

# Fajlovi

- Omogućavaju da VHDL komunicira sa host računarom
- Mogu da se otvore za čitanje i upis
- Paket STANDARD definiše osnovne rutine vezane za I/O

# Entiteti

- Definišu šta će komponenta raditi

- Primer:

```
ENTITY uredjaj IS
```

```
    PORT (a,b : IN BIT; c : OUT INTEGER);
```

```
END uredjaj;
```

# Arhitekture

- Definišu kako će komponenta raditi

- Primer:

```
ARCHITECTURE primer OF uredjaj IS
```

```
BEGIN
```

```
    PROCESS (a, b, c)
```

```
        VARIABLE d : INTEGER;
```

```
BEGIN
```

```
    if a then
```

```
        d := 4;
```

```
    else
```

```
        if b then
```

```
            d := 6;
```

```
        else
```

```
            d := 10;
```

```
        endif;
```

```
    endif;
```

```
    c <= d;
```

```
END primer;
```