**Entiteti:**

**entity** <ime\_entiteta> **is**

**generic**(<lista ulaznih parametara> : <tip>);

**port**(<lista portova> : <in / out> <tip>);

**end** **entity** <ime\_entiteta>

* ako nema tip onda je inout tip.

**Instanciranje entiteta**:

<име инстанце>: **entity** <библиотека>.<име ентитета>(<име арихтектуре>)

2 [**generic map**(<predefinisanje vrednosti generic konstanti>)]

3 **port map**(<mapiranje portova komponente na signale iz okruženja>);

**Arhitektura:**

**architecture** <ime\_arhitekture> **of** <ime> **is**

<signali>

<**signal** <ime\_signala>: <tip\_signala>;>

**begin**

**end architecture** <ime\_arhitekture>;

**Procesi (unutar arhitekture):**

<labela procesa>: **process** (<parametri pri cijoj promeni ce se izvrsiti proces>) **is**

<promenljive ili signali>

<variable> <ime\_promenljive>;

**begin**

**end process** <labela procesa>;

* Promenljive se mogu deklarisati samo unutar procesa i vidljive su samo unutar procesa u kojem su deklarisane
* Signali ne mogu biti deklarisani unutar procesa

Wait klauzula (unutar procesa):

**wait** **[on** lista\_signala**]** **[until** uslov**]** **[for** vreme**];**

**Kašnjenje:**

Delta kašnjenje: Sve dodele u procesu se izvršavaju odjednom kada se dodje do prvog suspendovanja procesa (wait ili na bilo koji drugi način...)

Transportno kašnjenje: clk **<=** not clk **AFTER** T\_pw**;**

Dodela se vrši nakon vremena T\_pw.

**Uslovna dodela:**

<signal ili promenljiva> <= <vredonst 1> **when** <uslov 1> **else**

<vredonst 2> **when** <uslov 2> **else**

.

.

**else** <podrazumevana vrednost>;

**Selektivna dodela:**

**with** <izraz> **select** <signal kome se dodeljuje> <=

<vrednost 1> **when** <izbor 1>,

<vrednost 2> **when** <izbor 2> | <izbor 3>, --’ili’

<vrednost 3> **when** <izbor 4> **to** <izbor 5>, -- opseg

<podrazumevana vrednost> **when others**;

Pr:

**with** a **select** b <=

"1000" **when** "00",

"0100" **when** "01",

"0010" **when** "10",

"0001" **when** "11";

ILI

b <= "1000" **when** a = "00" **else**

"0100" **when** a = "01" **else**

"0010" **when** a = "10" **else**

"0001" **when** a = "11";

Ovo se koristi van procesa.

**Upravljačke strukture (samo unutar tela procesa):**

**IF:**

1 **[**labela ifa**:]** **IF** logicki izraz **THEN**

2 sekvencijalni izraz

3 **ELSIF** logicki izraz **THEN**

4 sekvencijalni izraz

5 **ELSE**

6 sekvencijalni izraz

7 **END** **IF** **[**labela ifa**];**

**CASE:**

**case** izraz **is**

02 **when** vrednost\_1 **=>** --vrednost - само статичке вредности (константе)

03 klauzula\_1\_1**;**

04 klauzula\_1\_2**;**

05 -- ...

06 **when** vrednost\_2 **=>** -- више вредности се повезује са |

07 klauzula\_2\_1**;**

08 klauzula\_2\_2**;**

09 -- ...

10 **when** **others** **=>**

11 klauzula\_o\_1**;**

12 -- ...

13 **end** **case;**

**LOOP:**

1 **EXIT;** -- kraj izvršenja loop klauzule

2 **EXIT** **WHEN** uslov -- kraj izvršenja ako je uslov ispunjen

3 **NEXT** **WHEN** uslov -- prekida tekuću iteraciju i prelazi na iduću

**Izlaz iz petlje:**

1 **EXIT;** -- kraj izvršenja loop klauzule

2 **EXIT** **WHEN** uslov -- kraj izvršenja ako je uslov ispunjen

3 **NEXT** **WHEN** uslov -- prekida tekuću iteraciju i prelazi na iduću

**exit** ime\_labele **when** uslov

**FOR:**

**for** i **in** diskretni\_opseg **loop**

klauzula\_1**;**

klauzula\_2**;**

**end** **loop**

Diskretni opseg:

**(\_, \_, \_, \_)**

**\_ to \_**

**\_ downto \_**

**WHILE:**

labela**:** **while** izraz **loop**

2 klauzula\_1

3 klauzula\_2

4 -- ...

5 **end** **loop** **[**labela**]**

**SLOŽENI TIPOVI (polja):**

**deklaracije tipova polja:**

**type** memory\_word\_type **is** **array** **(**natural **range** **<>)** **of** bit**;**

**variable** memorijska\_rec**:** memory\_word\_type**(**127 **downto** 0**);**

**----isto kao i-----**

**type** memory\_word\_type **is** **array** **(**127 **downto** 0**)** **of** bit**;**

**variable** memorijska\_rec**:** memory\_word\_type**;**

**tipovi nabrajanja:**

**type** ime\_tipa **is** **(**ime\_1**,** ime\_2[**,** **…**]**)**

**-------**

1 **type** color **is** **(**red**,** green**,** blue**)**

2 **variable** boja**:** color **:=** red

3 **signal** S**:** color **<=** blue

Opseg indeksa vektora može biti I tipa nabrajanja.

01 **TYPE** controller\_state **IS** **(**initial**,** idle**,** **active,** error**);**

02 **TYPE** state\_counts **IS** **ARRAY** **(**idle **TO** error**)** **OF** NATURAL**;**

03 **variable** state\_counter**:** state\_counts**;**

Dimenzije višedimenzionalnih polja ne moraju da budu istog tipa.

1 **type** symbol **is** **(**'a'**,**'f'**,**'d'**,**'h'**,**digit**,**cr**,**error**);**

2 **type** state **is** **range** 0 **to** 6**;** -- nije naveden tip, podrazumeva se bilo koji ugrađeni tip koji ima literale 0 i 6. To je integer.

3 **type** transition\_matrix **is** **array** **(**state**,** symbol**)** **of** state**;**

4 ---------------------------

5 **variable** transition\_table **:** transition\_matrix**;**

6 transition\_table**(**5**,**'d'**);**

U ovom primeru, indeksi prve dimenzije (vrsta) promenljive transition\_table su celobrojnog tipa, a indeksi druge dimenzije(kolone) su nabrajačkog tipa symbol. Sami elementi matrice su celobrojnog tipa state.

**Pristup poljima:**

U VHDL-u se može pristupati celom polju istovremeno, ili jednom delu (slice).

coef**(**0**)** **:=** 0.0**;** -- jednom elementu, niz indeksiran celobrojnim tipom

2 counters**(active)** **:=** counters**(active)+**1**;** -- elementu niza indeksinanog sa enum

3

4 data\_register **:=** buffer\_register**; --celom nizu se može pristupati odjednom!**

5 -- ili delovima nizova**:**

6 buffe\_register**(**31 **downto** 16**)** **:=** data\_register**(**15 **downto** 0**); --ograničenje: da su slices sa leve i desne strane dodele istih dimenzija**