

OBIECTE IN LIMBAJUL VHDL:

În limb. VHDL datele sunt manipulate prin intermediul unor elemente numite obiecte. Acestea sunt următoarele:

- **semnalul** = sunt elemente fizice (fizice) care fac interconexiunile între circuite sau componente.
- **constantele** = reprez. identificatorii ce nu pot fi modificate pe parcursul operării
- **variabilele** = elemente prin care se realizează salvarea datelor și manipularea acestora în cadrul descrierilor secvențiale

SEMNALUL = un obiect folosit pt. interconectarea fizică a elementelor de circuit între ele. Semnalul e considerat a face parte din elem. concurente și interconectarea de asemenea componentă concurente între ele. Semnalele pot fi declarate atât global în pachetele librarilor, dar și local pînă la nivel de arhitectură.

PACKAGE ex_package is

SIGNAL A

USE librarie.ex_package.all;
ENTITY ex_entitate is
PORT (B: ...);

ARCHITECTURE ex_arch OF ex_entitate is
SIGNAL C
BEGIN

A

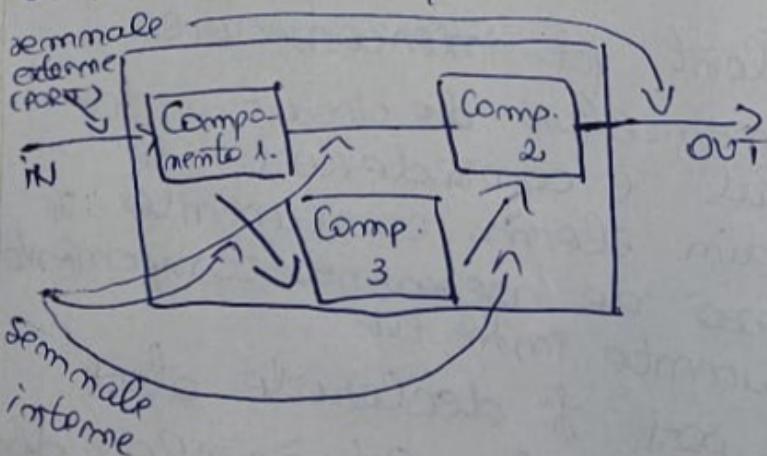
B

C

Semnalele [Externe (Porturi declarate în Entity)
Internă (Declarate în zonă declarativă a arhitecturii)]

Porturile sunt semnale care fac interacția cu mediul exterior și sunt declarate în Entity.

Semnalele interne sunt locale doar arhitecturii și vizibile doar în interiorul acesteia.



Semnalele interne se declară în zonă declarativă a arhitecturii cu sintaxa:

SIGNAL nume_semnal [nume_semnal]: tip_semnal (= valoare_imițială)

Prin aceeași declaratie "SIGNAL" pot fi declarate mai multe semnale de același tip.

Valoarea initială poate fi declarată optional în cazul în care se dorește cunoașterea valoții inițiale atunci când poartă circuitul.

ARCHITECTURE ex_arch OF ex_entitate IS

SIGNAL a,b: std_logic;

SIGNAL c: std_logic := '1';

SIGNAL d: std_logic_vector (4 DOWNTO 0);

SIGNAL e: std_logic_vector (4 DOWNTO 0) := "00001111";

BEGIN

Semnalele externe au și mod prim care să arate direcția informației în acesta (IN, OUT, INOUT...) în timp ce semnalele interne au decât tip.

În cazul magistratelor e necesară definirea nr. de semnale care fac parte din acestea. Definirea nr. de limii se face prin DOWNTO (ordinea descrescătoare) sau TO (ordinea crescătoare).

Definirea prim DOWNTO implică bitul cel mai semnificativ având indexul cel mai mare, iar bitul cel mai nesemnificativ având indexul cel mai mic.



+ DOWNT00.

b.c.m.s.

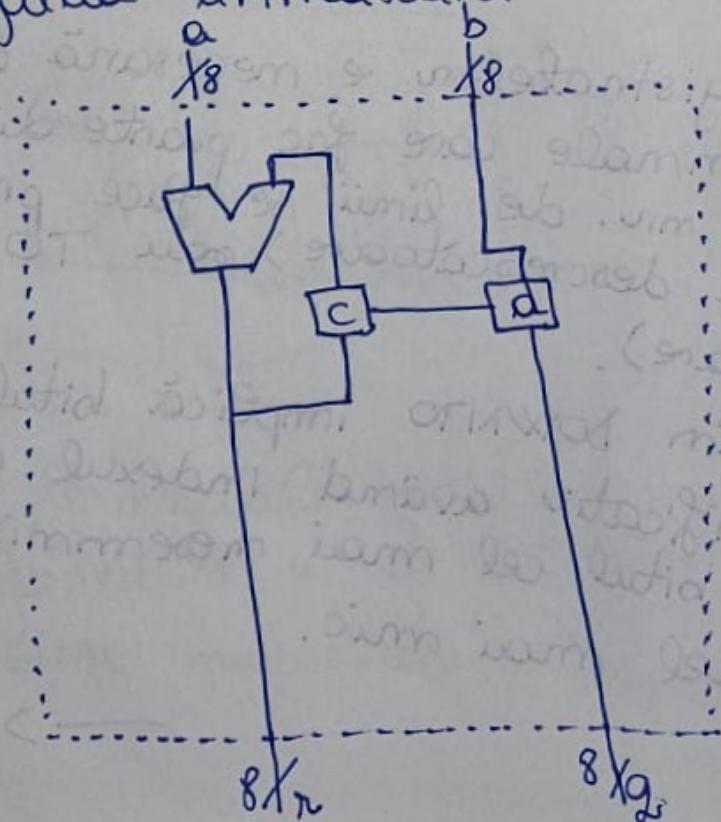
B₇ B₆ B₅ B₄ B₃ B₂ B₁ B₀ b.c.m.p.s.

În proiectarea digitală se recomandă ca magistralele să fie definite prin intervalul DOWNT0.

Cuvântul cheie "TO" se recomandă în general pt. definirea unui interval numeric SIGNAL a:INTEGER OT0255;

Nu se recomandă atribuirea două magistrale declarate diferenț (x cu DOWNT0, și alta cu TO) pt că se produce invinsarea bitelor.

ex) Să se descrie un circuit cu schemele din figura următoare.



ENTITY ex_ent is.
 PORT (a, b: IN std_logic_vector(4 DOWNTO 0);
 sem- externe { r: BUFFER std_logic_vector(4 DOWNTO 0);
 l: OUT std_logic_vector(4 DOWNTO 0);
 END ex_ent
 ARCHITECTURE ex_arch OF ex_ent IS
 SIGNAL temp: std_logic; seminal interne
 BEGIN

 END ex_arch;

CONSTANTE

Sunt obiecte atribuite o valoare statică cu un anumit tip de date.

CONSTANT nume_constanta [nume_constanta]; tip:=valoare;
ex:

ARCHITECTURE ex_arch OF ex_ent IS.

CONSTANT a, b, c: std_logic := '0';

CONSTANT d: std_logic_vector(5 DOWNTO 0) := x"4A2C";

CONSTANT pi: REAL := 3,14;

În limb. VHDL sunt identificate 3 tipuri de descriere:

- Flux de date
- Structurală
- Secvențială

Flux de date = realizarea circuitelor combinatoriale
 În timp ce descrierea secvențială e utilizată pt realizarea de circuite
~~acumulative~~ secvențiale.

Structurală: e fol. pt organizarea pe
componente a circuitelor complexe
dând și reutilizarea de componente
în cadrul unor aplicații complexe.

SPECIFICATII CONCURENTE

În cadrul programelor VHDL specificații concurente sunt pt a realiza circuite concurente care operează în paralel și acestea sunt plasate în zonă de descriere concurență a arhitecturii (între „BEGIN” și „END”).

Atribuirea concurență a semnalelor:

At. concurență constă în atribuirea de valori semnalelor și se realizează în specific al în zonă concurență a arhitecturii.

Descrierile concurente sunt utilizate în descrierile de tip flux de date, iar în urma sintezei rezulta circuite combinatoriale.

Flux de date → Circuit combinatorial
Secvențiale: [→ Circuit secvențial
] → Circuit combinatorial

At concurență se face cu $a <= "1"$ și se realizează
doar asupra semnalelor.

$a <= '1';$ (apostrof) pt că are un singur bit ('1'; '0'; '2')
 $b <= "0000";$ (ghilimele) pt că are mai multi biti.
 $c <= x"0";$ (x"4F")

Când se reprezintă în același său e (OTHERS => '0')