

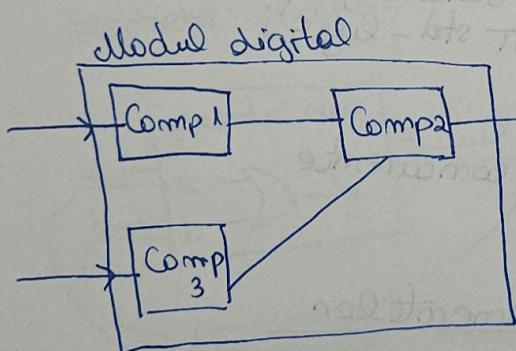
- partial pe 20 noiembrie 1h-

Descrierea structurală

Proiectarea modulelor VHDL prin descrierea structurală constă în specificarea conexiunilor dintre modulele constitutive, dar și reutilizarea acestora.

Descrierea structurală este utilizată în principal în organizarea proiectelor complexe pe module (componente).

Descrierea structurală → Declarația componentelor
→ Împletere componentelor



Declarația componentelor

Prima declarație se creează o copie a unei module deja proiectată pentru a fi reutilizat în proiecte viitoare.

Componentele sunt declarate în zona declarativă a arhitecturii.

```
COMPONENT nume-componentă  
PORT ( lista porturi )  
END COMPONENT;
```

Lista de porturi și numele componentei trebuie să fie identice cu entitățile modulului digital care a fost proiectat și se dorește să fie reutilizată.

ex COMPONENT poarta_sau
PORT (a,b : IN std_logic;
y : OUT std_logic);
END COMPONENT;

ex ARCHITECTURE ex_comp OF modul_electricic IS

```
COMPONENT poarta_si  
PORT ( a,b : IN std_logic;  
g : OUT std_logic );  
END COMPONENT;
```

BEGIN
- zonă specificării concurențe
END ex_comp;

Instanțierea componentelor

Prim inst. comp. se înțelege realizarea interconexiunilor dintre acestea, utilizând clauza "PORT MAP".

Eticheta: nume-componentă PORT MAP(lista nemai).

Lista de semnale constă în asocierea semnalelor interne declarate pt. interconectarea componentelor și porturile componentelor.

Asocierea se face prin 2 metode:

- asociere pozitională;
- asociere după nume;

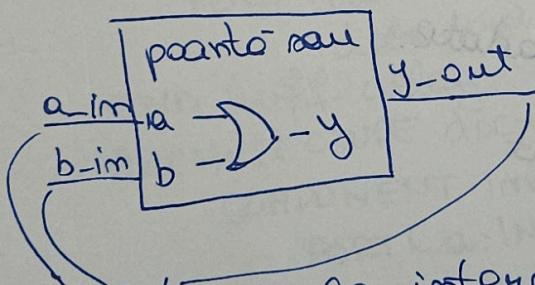
Asociere pozitională

Constituie intr-o metodă rapidă de asociere a porturilor componente cu semnalele interne de interconectare prin porturile interne a porturilor din componente în aceeași ordine ca și în clauze PORT MAP.

Etichetă: nume_componentă PORT MAP (semnal1, ..., ...);

Ex: Etichetă: poartă_sau PORT MAP (a_im, b_im, y_out);

Etichetă: poartă_si PORT MAP (a_im, b_im, g_out);



semnale interne de interconectare

Asocierea după nume

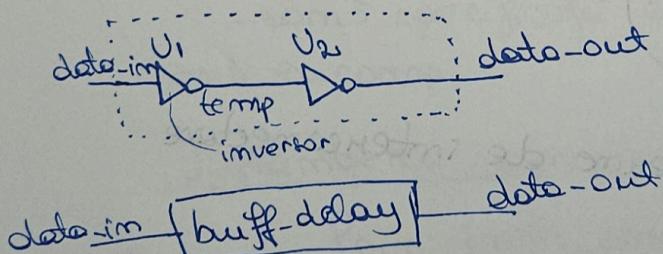
Metoda în care e mențină descrierea explicită a asocierii dintre portul componentei și semnalul intern de interconectare.

Etichetă: nume-componentă PORT MAP (nume-port => nume-semnal,
nume-port => nume-semnal-)

Această metodă se folosește la componente cu multe porturi și se doresc scrisoare explicită a relației dintre portul și semnalele de interconectare.

ex | Etichetă: poartă-ba1($a \Rightarrow a_im, b \Rightarrow b_im, y \Rightarrow y_out$);
PORT MAP

Să se implementeze primul descriere structură și un buffer de întârziere format din 2 inversare cascade.



Descrierea struct. se face în 2 etape:

1. Descrierea tuturor componentelor prin programe VHDL separate.

În casul de față se realizează descrierea componentelor invizion.



```
LIBRARY IEEE,  
USE IEEE std_logic_1164.all;  
ENTITY invizion IS  
    PORT (a: IN std_logic,  
          y: OUT std_logic);
```

END invizion;

ARCHITECTURE desc OF invizion IS.

BEGIN

$$y \leftarrow \text{NOT } a;$$

END desc.

~~Se realiz. modulul digital prim~~
~~interconectarea componentelor descrise în~~
~~paralel anterior.~~

```
LIBRARY IEEE;  
USE IEEE std_logic_1164.all;  
ENTITY buff_delay IS
```

```
    PORT (data_im: IN std_logic,  
          data_out: OUT std_logic);
```

END buff_delay;

ARCHITECTURE desc OF buff_delay IS

COMPONENT invizion

```
    PORT (a: IN std_logic;  
          y: OUT std_logic);
```

END COMPONENT;

SIGNAL temp: std_logic;

BEGIN

U1: invizion PORT MAP(data_im,temp);

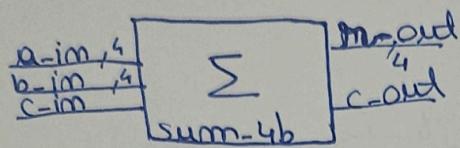
U2: invizion PORT MAP(temp,data_out);

END desc;

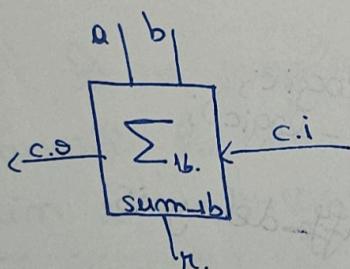
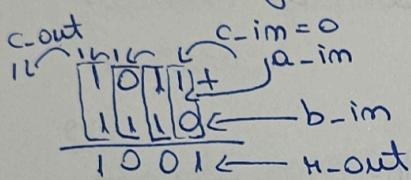
(depuții-memne)

U1: inversor PORT MAP ($a \Rightarrow \text{data_in}$, $y \Rightarrow \text{data_out}$);
 U2: inversor PORT MAP ($a \Rightarrow \text{temp}$, $y \Rightarrow \text{data_out}$);

Să se descrie în l. VHDL un sumator complet cu operație pe 4 biti printr-un interconectare a 4 sumatoare complete pe un singur bit.



$a_{\text{im}}, b_{\text{im}} \Rightarrow$ operații de intrare
 $c_{\text{im}} \Rightarrow$ carry - input



| a | b | c_{i} | Σ | c_{o} |
|-----|-----|----------------|----------|----------------|
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 0 | 1 | 0 |
| 0 | 1 | 1 | 0 | 1 |
| 1 | 0 | 0 | 1 | 0 |
| 1 | 0 | 1 | 0 | 1 |
| 1 | 1 | 0 | 0 | 1 |
| 1 | 1 | 1 | 1 | 1 |

$$\begin{aligned}
 \Sigma &= \overline{a} \overline{b} c_i + \overline{a} b \overline{c}_i + a \overline{b} \overline{c}_i + a b c_i = \\
 &= \overline{a} (\overline{b} c_i + b \overline{c}_i) + a (\overline{b} \overline{c}_i + b c_i) = \\
 &= \overline{a} (b \oplus c_i) + a (b \oplus c_i) = a \oplus b \oplus c_i
 \end{aligned}$$

$$\begin{aligned}
 r &= \overline{a} b c_i + a \overline{b} c_i + a b \overline{c_i} + a b c_i = \\
 &= c_i (\overline{a} b + a \overline{b}) + a b (\overline{c_i} + c_i) = \\
 &= c_i (a \oplus b) + a b .
 \end{aligned}$$

LIBRARY

~~LIBRARY~~ IEEE;

USE IEEE STD_LOGIC_1164.ALL;

ENTITY sum_1b IS

PORT (a, b, ci : IN STD_LOGIC,
~~r~~ → ~~o~~ : OUT STD_LOGIC);

END sum_1b;

ARCHITECTURE descr OF sum_1b IS

BEGIN

$r = a \text{XOR } b \text{ XOR } ci$;

$co = (ci \text{ AND } (a \text{XOR } b)) \text{OR } (a \text{ AND } b)$

END descr;