

## 8) Traitement Séquentiel : bases et syntaxe

### a) Domaine Séquentiel :

- \* A l'intérieur du domaine Concurrent, on peut effectuer des descriptions séquentielles  $\rightarrow$  Domaine Séquentiel.
- \* La description Séquentielle permet d'écrire des blocs de test.
- \* Mot clé débutant la description Séquentielle : **process**  
nom-du-process : **process** (liste-de-sensibilité)  
 $\rightarrow$  liste-de-sensibilité : liste des signaux dont le changement d'état sans le calcul du **process**

### b) Structure d'un process :

- \* nom-du-process : **process** (liste-de-sensibilité)
  - Déclaration des **variables** internes au process
  - Déclaration des constantes
  - begin**
    - Description Séquentielle }  $\rightarrow$ 
      - affectations
      - structures conditionnelles
      - structures répétitives
  - end process** nom-du-process;

Re

- Déclaration de variable :
  - avant de **begin** du process.
  - Variable nom-variable : type := valeur\_initiale;

### c) Structures de programmation dans les Process :

#### i) Affectation Simple :

- \* Pour les signaux :  $\leftarrow$
- \* Pour les variables :  $:=$



## ii) Structure Conditionals: "if ... then ... else ... end if;"

### → Sur une seule ligne

- if condition then action-du-then ; end if ;
- if condition then action ; else action-du-else ; end if ;

### → Sur plusieurs lignes

- if condition then  
    -- action  
end if ;
- if condition then  
    -- action  
else  
    -- action  
end if ;
- if condition then  
    -- action du then  
elseif condition-e then  
    -- action du elseif then  
else  
    -- action du else  
end if ;

### → Structure Case

- | case | Signal ou Variable | is  |
|------|--------------------|---|
| when | val                | ⇒ action-1 ; -- si Variable == val              |
| when | val1 to valn       | ⇒ action-2 ; -- si variable ∈ [val1, valn]      |
| when | val1 val2 ... valn | ⇒ action-3 ; -- si variable ∈ {val1, ..., valn} |
| when | others             | ⇒ action ; -- si rien                           |

  
end case ;



### iii) structures répétitives :

→ Boucle : For... loop :

```
for index in valeur-debut to valeur-fin loop
    -- actions de la boucle for... loop
end loop;
```

Rem : index de type : INTEGER, NATURAL, POSITIVE dans valeur-debut  
: valeur-debut et valeur-fin de même type

→ Boucle : while... loop :

```
while condition loop
    -- action
end loop;
```