

UTILISATION DES PROCESS VHDL POUR LES BASCULES, LES VERROUS, LES COMPTEURS, LES REGISTRE A DÉCALLAGE, ...

Liste de sensibilité des process

- Rappels
 - Un process s'exécute exclusivement si un des signaux listés dans la liste de sensibilité a évolué
- Pour une bascule D
 - CLK
 - CLEAR – RESET
 - Synchrone
 - Asynchrone
- Pour un verrou D
 - LE (*Latch Enable*)
 - D (entrée)
 - CLEAR – RESET
 - Synchrone
 - Asynchrone
- Besoin : Coder la sensibilité à des fronts

39

Logique séquentielle et process

- La logique séquentielle est nécessairement décrite dans les process
- La logique combinatoire peut être décrite hors process (instructions concurrentes) ou dans un process (l'ordre des instructions est important)
- Structure d'un process :


```

process (liste de signaux)
  -- partie déclarative (si besoin)
begin
    if SEL = ...
  -- suite du code
end process;
```

Exemple de bascule D

```

■ Pour une bascule D
Library ieee;           -- en-tête
USE ieee.std_logic_1164;

Entity Bascule_D is   --entité
port(
    D, CLK : in std_logic;
    Q : out std_logic);
End Bascule_D ;

Architecture ARCHI of Bascule_D is -- architecture
  Begin
    Process(CLK)
  Begin
    if (CLK = '1') then
      Q <= D;
    end if ;
  End process
End ARCHI ;
```

Exemple de bascule D

- Pour une bascule D

```
Architecture ARCHI of Bascule_D is -- architecture
Begin
  Process(CLK)
  Begin
    wait until (CLK='1')
      Q <= D;
    End process
End ARCHI ;
```

- Intérêt : meilleure lisibilité
- Inconvénient : un seul signal testable ainsi

Notion d'attributs

- Exemples

CLK'event : front d'horloge

Q'right : bit de droite du mot Q

Integer'range : gamme de variation des entiers

- Application

CLK'event and CLK = '1' → détecte un front montant d'horloge

Bascule D avec Clear asynchrone

- Exemples

```
Entity Basc_D is --entité
port(
  D, CLK, CLEAR : in std_logic;
  Q : out std_logic);
End Basc_D ;
```

```
Architecture ARCHI of Basc_D is -- architecture
Begin
  Process(CLK,CLEAR)
  Begin
    if CLEAR = '1' then Q <= '0'
    elsif (CLK'event and CLK = '1') then Q <= D;
    end if ;
  End process
End ARCHI ;
```

← Clear asynchrone
← Mettre un else

Bascule D avec Clear synchrone

- Exemples

```
Entity Basc_D is --entité
port(
  D, CLK, CLEAR : in std_logic;
  Q : out std_logic);
End Basc_D ;
```

```
Architecture ARCHI of Basc_D is -- architecture
Begin
  Process(CLK,CLEAR)
  Begin
    if (CLK'event and CLK = '1') then
      if CLEAR = '1' then Q <= '0'
      else Q <= D;
    end if ;
  End process
End ARCHI ;
```

Registre à décalage

■ Exemples

```
Entity REGISTRE is --entité
port(
    D_IN, CLK : in std_logic;
    Q_OUT : out std_logic);
End EXEMPLE ;

Architecture ARCHI of REGISTRE is -- architecture
Begin
Signal S : std_logic
Process(CLK)
Begin
    if (CLK'event and CLK = '1') then
        S1 <= D_IN ;
        Q_OUT <= S1;
    end if ;
End process
End ARCHI ;
```

!! Les signaux ne sont mis à jour qu'a la fin du process

46

DESCRIPTION DU LANGAGE VHDL (SUITE) : VHDL STRUCTUREL

VHDL structurel

■ Principe

- Écrire des blocs de code déclarés comme composants
- Réutiliser ces composants
- On parle d'instanciation de composant lorsqu'on fait appel à un module
- Le code des composants peut être dans le même fichier, dans un autre fichier, ou déjà compilé.

■ Intérêt

- Design hiérarchisé
- Appel à des éléments déjà mis au point

VHDL structurel

```
Entity MA_FCT is
Port( A,B,C,D,E,F,G,H,I : in std_logic_vector;
      S : out);
End MA_FCT;

Architecture ARCHIT of MA_FCT is
Signal J,K,L : bit_vector -- Declarations
Component : OR3 port ( A1,B1,C1 : in std_logic; S1: out std_logic);
Component : AND3 port( A2,B2,C2 : in std_logic; S2: out std_logic); -- declaration composant

UN : OR3 port map(A,B,C,J); -- description structurelle
DEUX : OR3 port map(D,E,F,K);
TROIS: OR3 port map(G,H,I,L);
QUATRE: AND3 port map(J,K,L,S); -- Ici S est générée

End ARCHIT;
```



VHDL structurel

■ Intérêt

- Réutiliser des fonctions
- Clarté d'écriture

■ Inconvénient

- Fichiers « lourds »
 - Création de package qui contiennent les composants

■ Avec Quartus

- L'aspect structurel → Graphique
- L'aspect algorithmique → textuel Vhdl