

TD N°2
Initiation à VHDL

Bref rappel.

Comme précisé dans le cours il existe 3 styles pour décrire des circuits numériques :

-Flot de données.

Réservé aux circuits de petite taille. On implémente les fonctions sous forme d'équations logiques (fonctions booléennes).

- Structurel.

Réservé aux circuits de taille moyenne à grande. Le circuit est décrit comme un ensemble de boîtes noires interconnectées au moyen de **signaux**.

- Comportementale.

Une suite d'instructions de contrôles (tests, boucles, etc.) précise le fonctionnement voulu dans un **process**. Un peu à la manière d'un langage procédural.

A) Logique combinatoire.

Des combinaisons d'entrées à l'instant t produisent des combinaisons de sorties, sans que ces dernières dépendent des entrées présentes à l'instant $t-1$.

Sans process

Exercice N°1.

Donner la description par **flot de données** du multiplexeur 4 vers 1 abordé en cours et dont on vous donne l'entité :

```
library ieee;
use ieee.std_logic_1164.all;

entity MUX_4to1 is
port
(
    E0, E1, E2, E3, SEL0, SEL1 : in std_logic;
    S: out std_logic
);
end MUX_4to1;
```

```
architecture flot_MUX4to1 of MUX_4to1 is
begin
```

```
end flot_MUX4to1
```

Compléter
le code



Exercice N°2.

Proposer une description **comportementale** d'un comparateur d'égalité 4 bits.

```
library ieee;
```

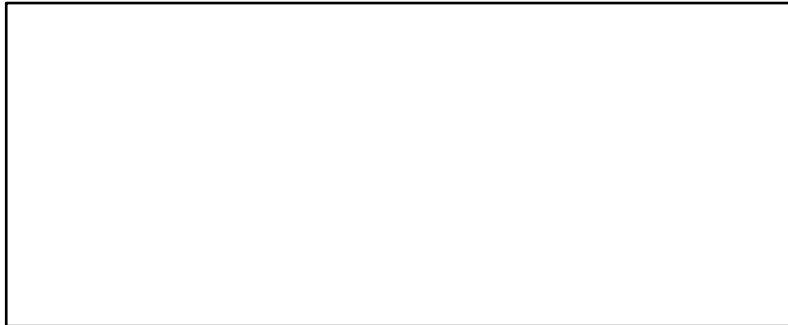
```
use ieee.std_logic_1164.all;
```

```
entity eqcomp4 is
port
```


```
(
    a,b      : in std_logic_vector(__ downto __);
    equal    : out std_logic
);
```

```
end eqcomp4;
```

```
architecture behav_ eqcomp4 of eqcomp4 is
begin
```



```
end behav_ eqcomp4
```



Compléter
le code

B) Logique séquentielle.

Se caractérise principalement par la présence d'un signal d'horloge en entrée du circuit à décrire. Cette horloge plus exactement un front (montant ou descendant) « lance » l'exécution d'un bout de code. Les sorties du circuit dépendront (état futur) des entrées à l'instant t (état présent) ainsi que de leurs valeurs à l'instant $t-1$ (état passé).

Exercice N°3.

Analyser le code page suivante et donner le nom de la fonction réalisée par ce circuit nommé bidule. Représenter sa « boîte noire » complète.

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;

entity bidule is
generic(bidule_Width : INTEGER := 8); -- paramètre de configuration
de bidule
port
(
  CLK : in std_logic;
  RST : in std_logic; -- Reset _____
  EN  : in std_logic; -- ENable
  UD  : in std_logic;
  Q   : out unsigned(bidule_Width-1 downto 0)
);
end bidule;

architecture arch_bidule of bidule is
signal bid : unsigned(Q'range);
-- idem std_logic_vector(bidule_Width-1 downto 0)
begin
  process(CLK,RST)
  begin
    if (RST='1') then
      bid <= (others => '0');
    elsif rising_edge(CLK) then
      if EN='1' then
        if UD='1' then
          bid <= bid + 1; -- _____
        else
          bid <= bid - 1; -- _____
        end if;
      end if;
    end if;
  end process;
  Q <= unsigned(bid);
end arch_bidule;

```

Compléter tous
les
commentaires

Écriture plus
condensée

Exercice N°4

Reprendre l'exercice du TD N°1 « détecteur de séquence » et le coder en VHDL en utilisant 1 process. On nommera l'entity « detect_seq » et l'architecture « fsm_TD1 »