

Partie II :TD-VHDL

Exercices

LOGIQUE COMBINATOIRE

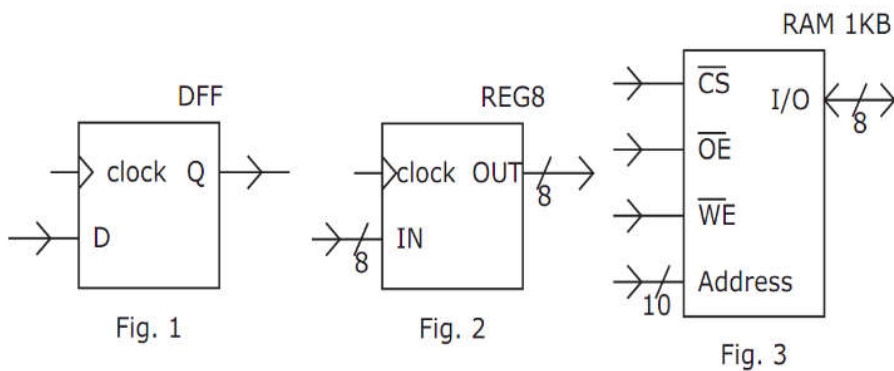
Exo 1

Dessiner la « boîte noire » correspondante à l'entité suivante.

```
library ieee ;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all ;
entity counter is
    port ( clock , raz : in std_logic ; q : out unsigned (7 downto 0 ) ) ;
end counter ;
```

Exo 2

Écrire les entités correspondant aux boîtes noires de la figure 1, 2 et 3



Exo 3

Soit la description VHDL suivante :

```
entity exer
port( x1, x2, x3, sel: in std_logic; y: out std_logic);
end entity exer;
architecture archi of exer is
    signal a, b, c, d, e, f: std_logic;
begin
    a <= x1 or x3; b <= x1 and x3; c <= x2 and a; d <= b or c; e <= x1 xor x2; f <= x3 xor e;
    Pro: process (d, f, sel)
    begin
        if sel='0' then y <= d;
        else y <= f;
        end if;
    end process Pro;
end architecture archi;
```

Questions:

- Tracez le schéma correspondant (à partir des éléments de base).
- Dire que fait ce circuit ?
- Quel est le type de cette architecture ? Justifiez.
- Le processus 'Pro' est-il combinatoire ou séquentiel ? Justifiez.

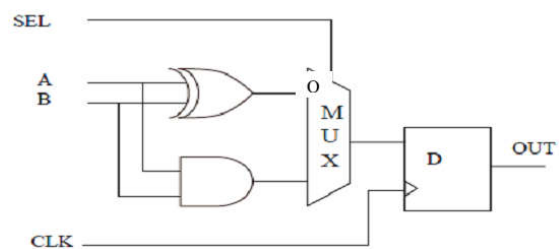
Exo 4

Quelle est la fonction du programme VHDL ci-dessous.

```
process(clk)
begin
if ck'event and clk='1' then front <= '0' ; ex_entree <= ENTREE ;
if ex_entree = '0' and ENTREE = '1' then front <= '1' ;
end if ;
end if ;
end process ;
```

Exo 5

Ecrire un code VHDL correspondant au circuit suivant :



LOGIQUE SEQUENTIELLE

Exo 6

Nous voulons réaliser un compteur (4bits) qui incrémente seulement quand un bouton (E) est appuyé. Quand on relâche le bouton le compte arrête automatiquement (refaire le calcul si le compte =15). Ecrire le programme VHDL complet de ce circuit.

Exo 7

Expliquer le déroulement de code suivant :

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_unsigned.all;
entity compteur is
port ( clk : in std_logic; entree : in std_logic; compt_sortie : out std_logic_vector(4
downto 0) );
end compteur;
architecture rtl of compteur is
signal s_compt_sortie : std_logic_vector(4 downto 0);
signal s_jaipese : std_logic;
```

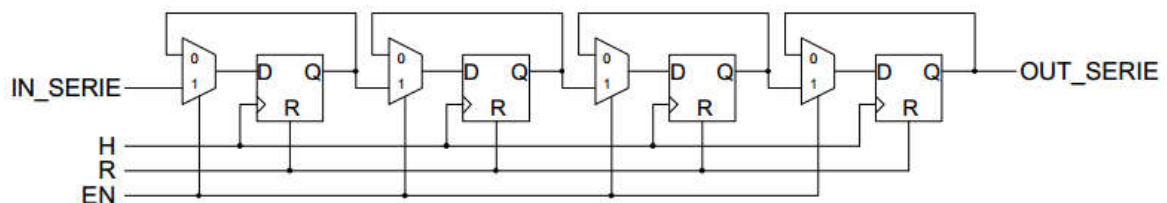
```

begin
process (clk)
begin
if clk'event and clk = '1' then
if s_jaipese = '1' then
if s_compt_sortie >= 23 then s_compt_sortie <= "00000";
else
s_compt_sortie <= s_compt_sortie + 1;
end if;
end if;
end if;
end process;
process (clk)
begin
if clk'event and clk = '1' then
if entree = '1' then
s_jaipese <= '1';
elsif s_compt_sortie >= 23 then s_jaipese <= '0';
end if;
end if;
end process;
compt_sortie <= s_compt_sortie;
end;

```

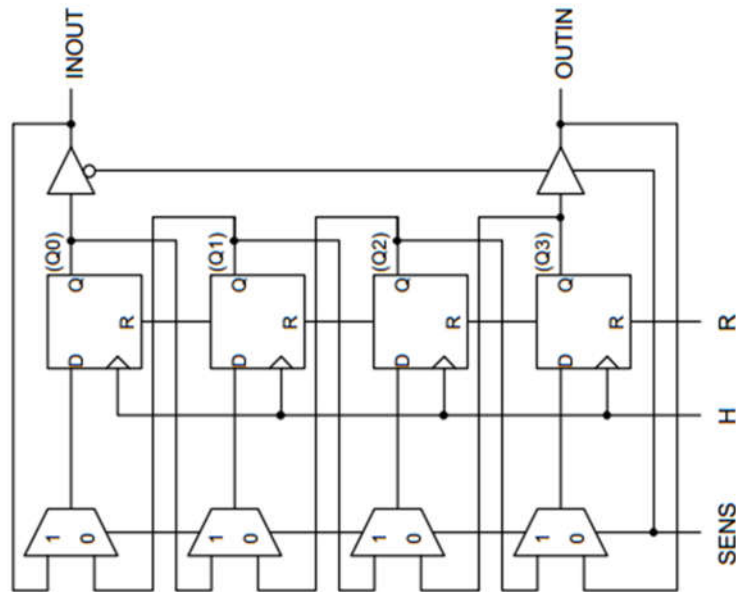
Exo 8

Ecrire un programme VHDL qui permet de simuler le fonctionnement d'un registre à décalage à droite à autorisation d'horloge.



Exo 9

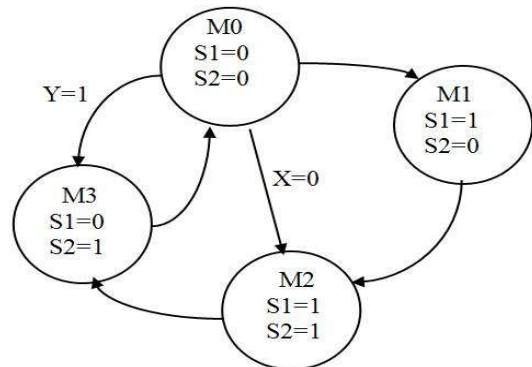
Ecrire un programme VHDL qui permet de simuler le fonctionnement d'un registre à décalage à droite ou à gauche



MACHINE D'ETAT

Exo 10

La description d'état est utilisée pour décrire des systèmes séquentiels quelconques (machine d'état). La description du système se fait par un nombre fini d'états. Ci-dessous la représentation schématique d'un système à 4 états (M0 à M3), 2 sorties (S1 et S2), 2 entrées X et Y, sans oublier l'entrée d'horloge qui fait avancer le processus, et celle de remise à zéro qui permet de l'initialiser :



L'état initial est M0. Les 2 sorties sont à 0. Au coup d'horloge on passe inconditionnellement à l'état M1 sauf si la condition Y=1 a été vérifiée, ce qui mène à l'état M3 ou si X=0 a été validé ce qui mène à M2. De M3 on revient au coup d'horloge à M0. De M1 on passe à M2, et de M2 on passe à M3...

Dans chaque état on définit les niveaux des sorties.

Exo 11

Ecrire la description VHDL (Entité et Architecture) de ce système en se basant sur le diagramme d'état. Le système est actif sur front montant.

Le système à concevoir dispose de trois entrées et de deux sorties. Les entrées sont l'horloge clk, RAZ et la commande x ; les sorties sont S0 et S1. Les entrées et les sorties sont de type std_logic sauf x est de type bit. La description du système se fait par un nombre fini d'états. Ci-dessous la représentation schématique d'un système à 4 états (A, B, C et D) :

Donner la description en langage VHDL (Entité et Architecture) de ce système en se basant sur le diagramme d'état. Le système est actif sur front montant.

