

TD Langage VHDL

Exercice1 :

Dessinez le circuit correspondant au code VHDL suivant (tous les signaux sont du type std_logic) :

architecture synthesizable of test is

signal E : std_logic;

begin

process (A, B, C, D, E)

begin

if A'event and A = '0' then

if B = '0' then

E <= C;

else

E <= D;

end if;

F <= E;

end if;

end process;

end synthesizable;

Exercice 2 :

Considérez le circuit décrit par le VHDL ci-dessous :

```
ENTITY ent IS
    PORT (din   : IN   STD_LOGIC;
          t1    : IN   STD_LOGIC;
          t2    : IN   STD_LOGIC;
          dout  : OUT  STD_LOGIC);
END ent;

ARCHITECTURE arch OF ent IS
    a, b, c, d, e, f, g, h, sel : STD_LOGIC;
BEGIN
    dout <= h;
    d <= NOT din;
    c <= NOT b;
    f <= NOT e;
    sel <= g XOR h;
    a <= d WHEN sel = '1' ELSE g;
```

```

p1 : PROCESS (t1, a, c)
BEGIN
    IF t1 = '1' THEN
        b <= a;
    ELSE
        h <= c;
    END IF;
END PROCESS p1;

p2 : PROCESS (t2, din, f)
BEGIN
    IF t2 = '1' then
        e <= din;
    ELSE
        g <= f;
    END IF;
END PROCESS p2;
END ARCHITECTURE arch;

```

- a. Dessinez le schéma logique du circuit décrit par ce code.
- b. Ce circuit est-il combinatoire ou séquentiel? Justifiez votre réponse .

Exercice 3 :

- c. Ecrire en langage Vhdl l'entité **basculeD** et l'architecture **descbasD** de la bascule D avec bit de forçage RAZ (remise à zéro) en utilisant une description **comportementale**
- d. Ecrire en langage Vhdl l'entité **struct** et l'architecture **descstruct** de circuit ci-dessous (figure 1) en utilisant une description **structurelle**

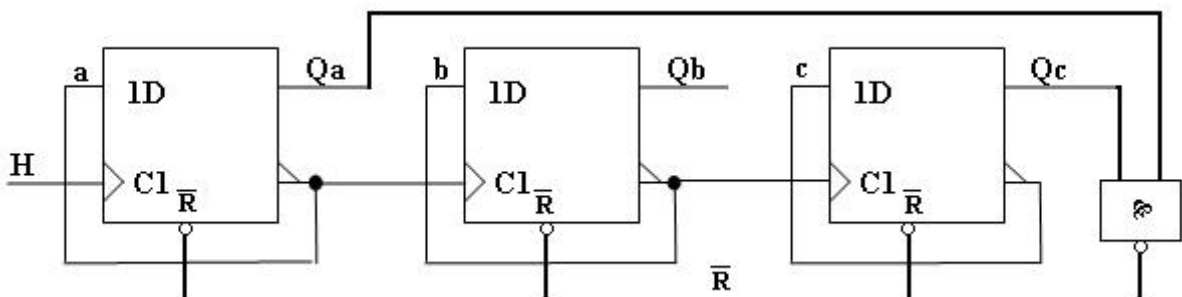
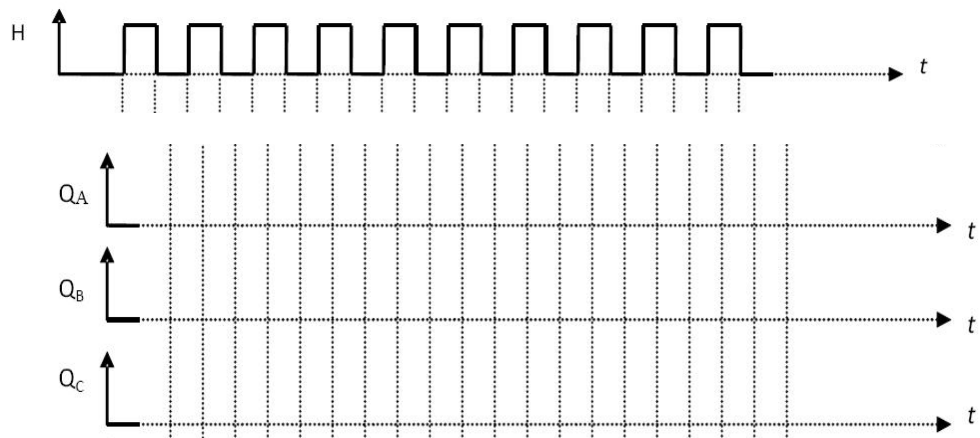
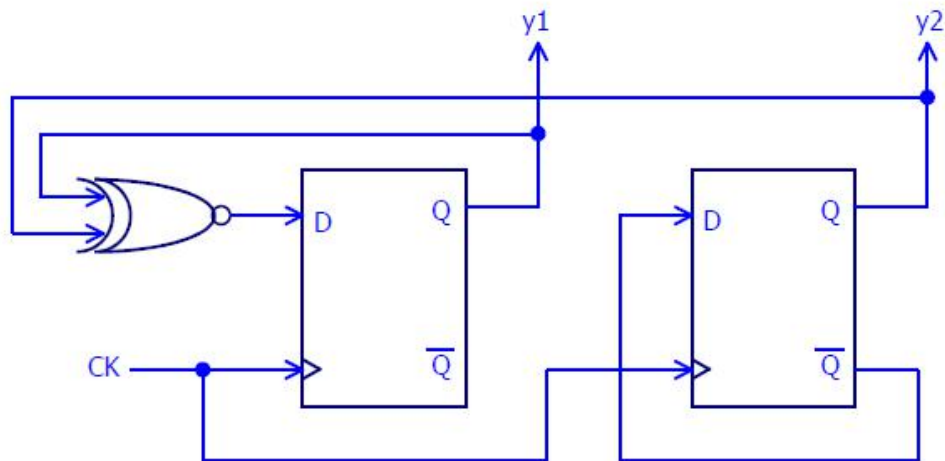


Figure 1

- c. Etablir les chronogrammes de fonctionnement des sorties QA, QB et QC de ce circuit (a l'état initial, QA=QB=QC='0')



- d. Supposez le compteur de la figure ci-dessous :



- Ecrire les équations de récurrence de ce compteur
- En déduire le programme vhdl (entité et architecture) correspondant (description **comportementale**).
- Ecrire un programme en vhdl permettant de décrire le circuit ci-dessus de façon **structurelle**

TD 4

Langage VHDL (LISI 2)

Exercice 1 :

Le langage VHDL peut également servir afin de programmer des composants pouvant réaliser des fonctions décrites dans un grafcet.

En fait, nous n'allons pas programmer un grafcet mais un **graphe d'état** ce qui ressemble beaucoup. En apparence du moins, mais nous ne rentrerons pas dans les détails, ce que l'on souhaite c'est une méthode simple, rapide et graphique de montrer ce que le programme doit faire.

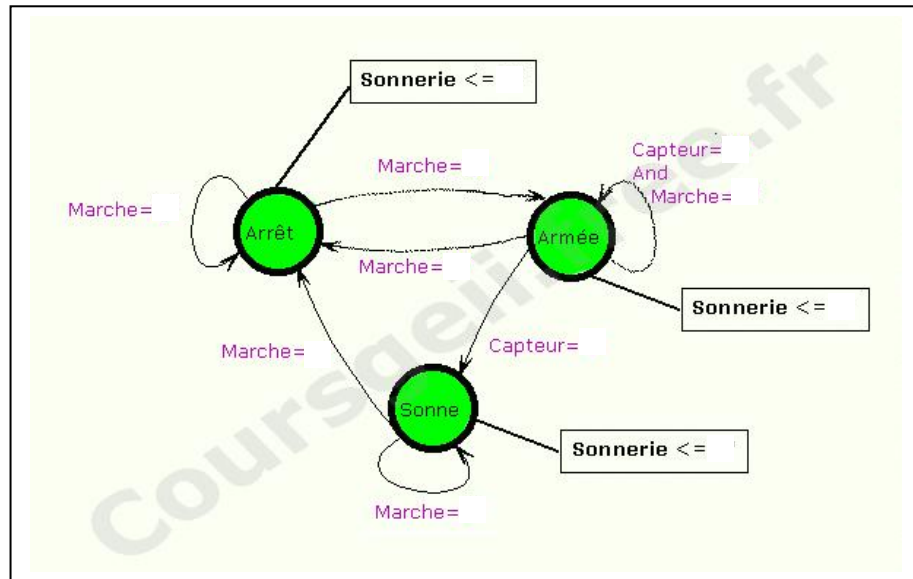
Nous allons prendre l'exemple d'une alarme simple:

- Nous avons l'entrée 'marche' c'est la touche qui permet d'armer l'alarme.
- Il y a également l'entrée 'capteur' c'est lui qui va dire si l'alarme doit sonner.
- Et puis l'entrée clock qui représente l'horloge interne de l'appareil.
- Il y a une seule sortie 'sonnerie'.

Traduisons en français le comportement de cette alarme. Afin de simplifier, on va dire qu'une entrée active a un niveau haut ('1') et inversement.

Il y a **3 états** possibles : '**arret**', '**armee**' et '**sonne**' :

- Si on est dans l'état 'arret' , on reste dans cet état si **marche** n'est pas activé et on quitte 'arret' pour aller dans 'armee' si on a **marche** activé.
 - Si on est dans 'armee' on reste dans cet état si **marche** est activé et **capteur** inactif on quitte 'armee' pour aller dans l'état 'sonne' si on a **capteur** est actif. On peut également quitter 'armé' pour aller dans 'arret' si on a **marche** désactivé.
 - Si on est dans l'état 'sonne', on continue de sonner tant que **marche** est activé, pour arrêter de **sonner** il faut uniquement que **marche** devienne inactive et quelque soit l'état de 'capteur'.
- a) Compléter le schéma en graphe d'état représentant cette alarme (les valeurs des transitions et la sortie de chaque état) :



b) Compléter le programme suivant décrivant le model comportementale de système d'alarme :

Entity alarme **is**

Port (

Capteur,marche,clock : in bit;

Sonnerie : out bit);

End alarme;

Architecture ala **of** alarme **IS**

-- on crée un type de donnée pouvant prendre 3 états.

Type state is (.....,,);

Signal etat :.....;

Begin

Procces (clock) **begin**

IF clock='1' **AND** clock'event **then**

case **is**

When arret

=> IF marche='1' then

etat <=;

else

etat <=;

```

end IF;

<=.....;

sonnerie

When armee => IF marche = ..... then
    etat <=.....;
    elsif capteur ='1' then
        etat <= .....;
    else etat .....;
    End if;
    sonnerie <=.....;

When sonne => IF marche=..... then
    etat <= arret;
    else
        etat <= sonne ;
    end if;
    sonnerie <= .....;

End case;
End if;
End process;
End ala;

```

Exercice 2 (Analyse de code)

Dessiner le graphe d'état correspondant au morceau de code suivant :

```

etat_suivant: process (reg_etat,d,g)
begin
--next_etat<=reg_etat;
    case reg_etat is
        when S1=>
            if d='1' and g='0' then
                next_etat<=S2;
            elsif g='1' and d='0' then
                next_etat<=S3;
            else
                next_etat<=S1;
            end if;
    end case;
end process;

```

```

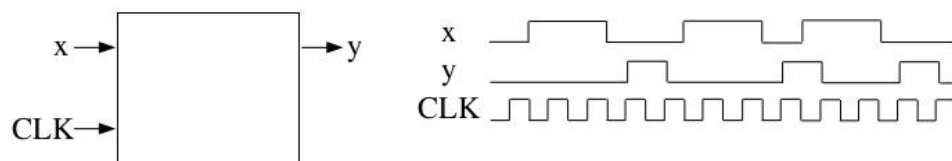
when S2=>
    if g='1' then
        next_etat<=S4;
    else
        next_etat<=S2;
    end if;
when S3=>
    if d='1' then
        next_etat<=S5;
    else
        next_etat<=s3;
    end if;
when S4=>next_etat<=S6;
when S5=>next_etat<=S7;
when S6=>
    if g='0' then
        next_etat<=S1;
    else
        next_etat<=S6;
    end if;
when S7=>
    if d='0' then
        next_etat<=S1;
    Else
        next_etat<=S7;
    end if;
end case;
end process etat_suivant;

```

Exercice 3: Synthèse : Détecteur de séquence

Question 1: Concevez une machine capable de détecter la séquence 0-1-1-0 sur son unique entrée (x).

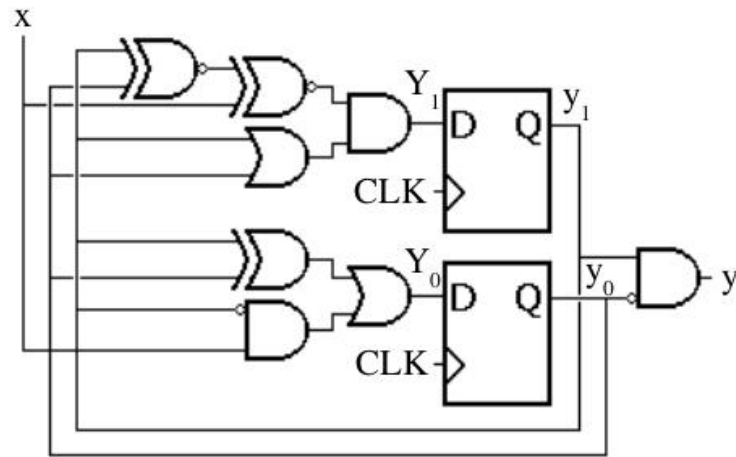
Quand la séquence se produit, il devrait émettre un 1 sur son unique sortie (y). Une diagramme temporelle du comportement attendu est donné. Notez que dans la séquence **0-1-1-0-1-1-0**, la séquence se produit deux fois ! Utilisez **une machine de Moore**.



Question 2 : Ecrire un programme en VHDL de cette machine

Exercice 4: Analyse : Détecteur de séquence

Le circuit suivant est un détecteur de séquence. En suivant les étapes de l'analyse, déterminez quelle séquence ce circuit détecte.



Question :

1. Déterminer la table de transition
2. Deducire le diagramme d'état
3. Ecrire le code vhd de la machine d'état