

NOM :

PRENOM :

Lundi 30 novembre 2015

PROMOTION :

**Contrôle de connaissances
Systèmes Numériques
Partie Électronique Numérique**

(Durée conseillée 1h)

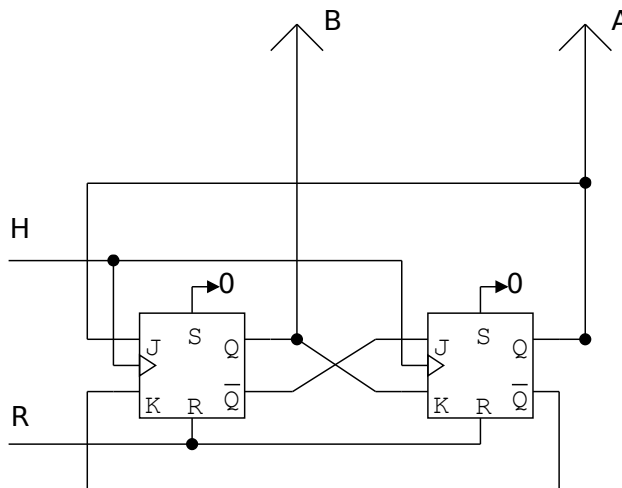
Aucun document autorisé – Calculatrice non programmable autorisée

Tous les exercices sont indépendants

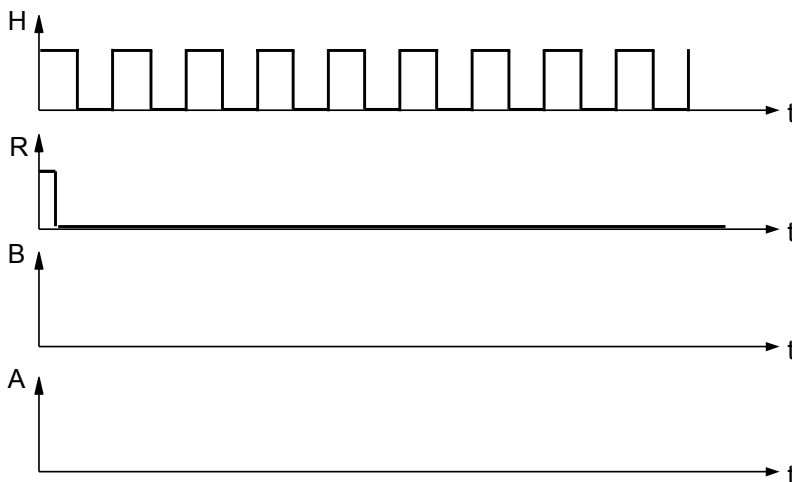
Vous trouverez en annexe un recto-verso résumant les structures VHDL abordées en cours

Vous devez répondre impérativement sur le sujet.

1. Soit le circuit de la figure suivante :



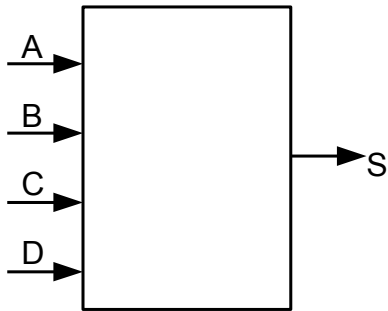
a) Complétez le chronogramme suivant en expliquant sur le côté votre démarche:



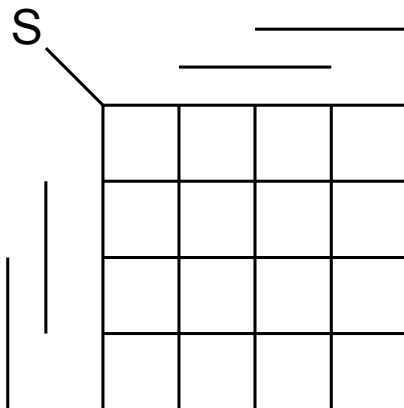
b) Quel dispositif a-t-on réalisé ?

2. Synthèse logique combinatoire.

On désire réaliser un dispositif dont la sortie sera à 1 si au moins deux des quatre entrées du système sont à 1.



Déterminez une équation simplifiée de la sortie S en fonction des entrées A,B,C,D. Pour cela vous utiliserez le tableau de Karnaugh ci-dessous en le complétant.



3.

a) Le listing VHDL suivant est utilisé pour réaliser un compteur modulo 16 :

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

entity il_est_tot is
  port (
    H,RESET : in std_logic;
    Q : out std_logic_vector(3 downto 0));
end il_est_tot;

architecture exam of il_est_tot is
begin --reveille_toi...
  process (H,RESET)

  begin

    if RESET='1' then Q<= "0000" ;
      elsif rising_edge(H) then Q <= Q+1 ;

    end if ;

  end process;

end exam;
```

Une erreur subsiste et il est du coup impossible d'avoir un résultat de simulation.
Corriger le listing **en utilisant une couleur différente du rouge** de façon à éliminer cette erreur en expliquant ci dessous d'où vient le problème.

b) Modifiez le listing précédant en utilisant **une autre couleur (mais différente du rouge...)** pour obtenir cette fois-ci un compteur modulo 12.

4. Soit les 2 listings suivants décrivant en VHDL les entités rad1 et rad2

```

library ieee;
use ieee.std_logic_1164.all;

entity rad1 is
  port (
    E,H : in std_logic;
    Q : out std_logic_vector(3 downto 0));
end rad1;

architecture archi_rad1 of rad1 is
begin -- archi_rad
  process (H)
    variable QQ : std_logic_vector(3 downto 0);
    begin -- process
      if rising_edge(H) then
        QQ(0):=E;
        QQ(1):=QQ(0);
        QQ(2):=QQ(1);
        QQ(3):=QQ(2);
      end if;
      Q<=QQ;
    end process;
end archi_rad1;

```

```

library ieee;
use ieee.std_logic_1164.all;

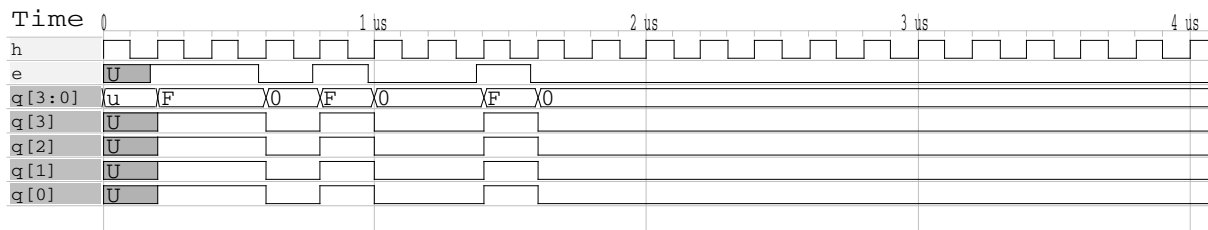
entity rad2 is
  port (
    E,H : in std_logic;
    Q : out std_logic_vector(3 downto 0));
end rad2;

architecture archi_rad2 of rad2 is
  signal QQ : std_logic_vector(3 downto 0);
  begin -- archi_rad
    process (H)
      begin -- process
        if rising_edge(H) then
          QQ(0)<=E;
          QQ(1)<=QQ(0);
          QQ(2)<=QQ(1);
          QQ(3)<=QQ(2);
        end if;
      end process;
      Q<=QQ;
    end archi_rad2;

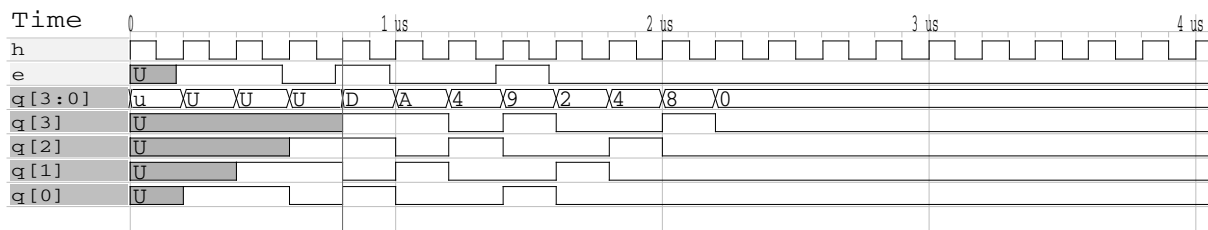
```

On donne les chronogrammes résultant de la simulation de chacune de ces 2 entités mais sans spécifier lequel des chronogrammes correspond à la simulation de rad et rad2 :

Résultat 1 :



Résultat 2 :



- a) Indiquez pour chacun des résultats 1 et 2 à la simulation de quel entité il correspond en expliquant impérativement pourquoi **sinon la réponse ne sera pas prise en compte**.

- b) On désire en fait réaliser un registre à décalage. Indiquez lequel des 2 résultats de simulation est correct : résultat 1 ou résultat 2.

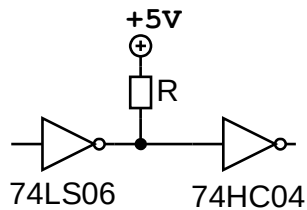
- c) Expliquez la présence de symboles **U** au début du chronogramme des sorties $Q(0)$, $Q(1)$, $Q(2)$ et $Q(3)$ dans les résultats de simulation.
- d) Modifiez à l'aide d'un stylo d'une couleur différente du noir, le code qui donne le résultat de simulation erroné pour un registre à décalage **sans modifier ni la nature de QQ ni sa déclaration**.
Expliquez impérativement votre raisonnement en quelques lignes **sinon la réponse ne sera pas prise en compte**.
- e) Donner un schéma logique de ce registre à décalage en utilisant des bascules D. Expliquez en quelques mots son fonctionnement.

5. On donne le tableau suivant résumant les caractéristiques statiques des circuits logiques intégrés (pour une tension d'alimentation de 5V) en fonction de leur technologie :

	-----CMOS-----			-----TTL-----			
	4000B	74 HC	74 HCT	74	74LS	74 AS	74ALS
V_{IHmin}	3.5V	3.5V	2V	2V	2V	2V	2V
V_{ILmax}	1.5V	1V	0.8V	0.8V	0.8V	0.8V	0.8V
V_{OHmin}	4.95V	4.9V	4.9V	2.4V	2.7V	2.7V	2.7V
V_{OLmax}	0.05V	0.1V	0.1V	0.4V	0.5V	0.5V	0.4V
I_{IHmax}	1μA	1μA	1μA	40μA	20μA	200μA	20μA
I_{IImax}	1μA	1μA	1μA	1.6mA	0.4mA	2mA	100μA
I_{OHmax}	0.4mA	4mA	4mA	0.4mA	0.4mA	2mA	400μA
I_{OLmax}	0.4mA	4mA	4mA	16mA	8mA	20mA	8mA

a) Combien d'entrées 74 LS une sortie 74HCT peut-elle piloter ?

Soit le schéma suivant :



Paramètres technologiques de la porte 74LS06 :

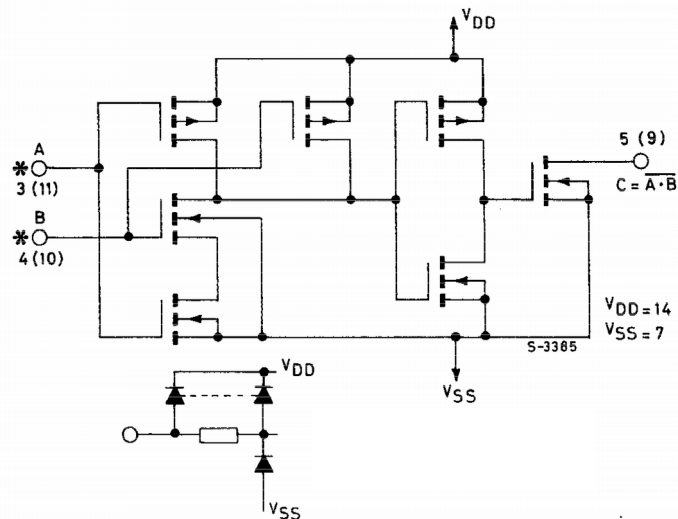
$$|I_{OL}|_{MAX} = 16 \text{ mA} \quad \text{et} \quad |I_{OH}|_{MAX} = 0,25 \text{ mA}$$

Tous les autres paramètres sont identiques aux portes de la technologie TTL LS

b) Justifiez la présence de R sur le schéma

c) Déterminez la limite sur R permettant un bon fonctionnement du système pour un état logique 0 en sortie de la porte 74ls06

Soit le schéma interne d'une porte logique :



d) A quelle famille technologique appartient cette porte logique (TTL ou CMOS) ? **Justifiez impérativement sinon votre réponse ne sera pas prise en compte.**

e) Quel est le type de sortie de cette porte logique (collecteur/drain ouvert ou totem/pôle) ? **Justifiez impérativement sinon votre réponse ne sera pas prise en compte.**

Page volontairement laissée blanche

Structure d'une description VHDL

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.all;

ENTITY nomentite IS
    PORT (
        .....
        .....
    );
END ;

ARCHITECTURE nomarchitecture OF
nomentite IS
BEGIN
.....
.....
.....
.....
END ;
```

Manuel Ardouin

Déclaration d'une architecture

```
ARCHITECTURE a OF nomentite IS
    SIGNAL nomsignal : STD_LOGIC;
    SIGNAL nomsignal : STD_LOGIC;
BEGIN
.....
.....
.....
.....
END a;
```

← Déclaration des signaux internes

Manuel Ardouin

Affectation d'un signal Opérateurs logiques et arithmétiques

- L'opérateur d'instanciation pour un signal est '<='
- Les opérateurs logiques en VHDL sont :
AND OR XOR NOT NOR NAND XNOR
- Les opérateurs arithmétiques sont :
+, -, *, /, mod et abs

Manuel Ardouin

Affectation conditionnelle d'un signal

- structure WHEN...ELSE :
Exemple

```
A<='1' WHEN (B='1' OR C='0') ELSE '0';
```
- Structure SELECT (choix multiple)
Exemple

```
with toto select
m <=c when "00",
d when "01",
e when others;
```

Manuel Ardouin

VHDL : Opérateurs relationnels

- Les opérateurs relationnels utilisés à l'intérieur des conditions en VHDL sont :
- | | | | |
|---|-----------|----|-------------------|
| > | supérieur | >= | supérieur ou égal |
| < | inférieur | <= | inférieur ou égal |
| = | égal | /= | différent |

Manuel Ardouin

Librairies arithmétiques IEEE

- Librairies supplémentaires à utiliser pour pouvoir faire de l'arithmétique (opérations ou comparaisons) avec des *STD_LOGIC_VECTOR* :
- ☞ IEEE.STD_LOGIC_ARITH
 - ☞ IEEE.STD_LOGIC_UNSIGNED ou IEEE.STD_LOGIC_SIGNED

Manuel Ardouin

Déclaration d'un process

```
PROCESS (signal1, signal2, ...)
  VARIABLE nomvariable : STD_LOGIC;
  VARIABLE nomvariable : STD_LOGIC;
BEGIN
  -- Affectation d'un signal
  -- Affectation d'une variable
  -- Structure conditionnelle if then else
  -- Structure conditionnelle case
  -- boucle.
END PROCESS;
```

Déclaration des variables

Manuel Ardouin

IF ... THEN... ELSE

```
IF __expression THEN
  ...
ELSIF __expression THEN
  ...
ELSE
  ...
END IF;
```

Manuel Ardouin

Case

```
CASE expression IS
  WHEN constante =>
    ...
  WHEN constante =>
    ...
  WHEN OTHERS =>
    ...
END CASE;
```

Manuel Ardouin

VHDL : Variable

- L'opérateur d'affectation pour une variable est ':='
☞ Attention à ne pas confondre avec le '<=' pour un signal..

Manuel Ardouin

Boucles en VHDL

- La structure d'une boucle *FOR* est :

```
FOR index IN range LOOP
  ...
END LOOP
```

« range » 2 possibilités:

- ☞ X TO Y
- ☞ Y DOWNTO X

Manuel Ardouin

Boucles en VHDL

- Boucle *While*

```
while <condition> loop
  ...
end loop;
```

Manuel Ardouin