



20200646: HAUDIE Jean-samuel  
Celestin Inkpé  
L3: TD I / Com G23  
TE506V-DE (07/12/2021)  
C001

Prénom ..... Jean-Samuel .....  
Nom ..... HAUDIE .....  
Promotion ..... L3 .....  
Groupe .....

Ne rien inscrire dans ce cadre

15

**Promotion : L3**

**Module Conception de circuit numérique**

**TE506V**

**Devoir Ecrit Horaire : 08h00-09h50**

**Examen sur table**

Sujet rédigé par : ILKI (Didier MEIER)

**Supports autorisés :**

Calculatrice autorisée : **non**

Documents autorisés : **non**

Traducteur électronique : **non**

Dictionnaire : **non**

**Consigne :**

Merci de restituer : **SUJET**

**Rappel :**

- Tous les appareils électroniques (téléphones portables, ordinateurs, tablettes, montres connectées, accès à internet ...) doivent être éteints et rangés.
- Il est interdit de communiquer.
- Toute fraude ou tentative de fraude fera l'objet d'un rapport de la part du surveillant et sera sanctionnée par la note zéro, assortie d'une convocation devant le conseil de discipline. Aucune contestation ne sera possible. Tous les documents et supports utilisés frauduleusement devront être remis au surveillant.
- Aucune sortie de la salle d'examen ne sera autorisée avant la moitié de la durée de l'épreuve.

Ne rien inscrire dans ce cadre

begin

-- début de la description de l'architecture

MyProcess: Process (clk)

if (clk'event and clk = 1) then

  if (sel\_in = '0') then

    MyS\_out  $\leftarrow$  A\_in \* B\_in;

  else

    MyS\_out  $\leftarrow$  B\_in + B\_in;

  endif;

endif;

end Process;

S\_out  $\leftarrow$  MyS\_out;

end ma\_fonction\_arch;



NOM :	Audie'
PRENOM :	Jean-Samuel

## Conception de circuit numérique

### Introduction :

Le devoir surveillé de conception de circuit numérique se décompose en quatre parties distinctes et indépendantes :

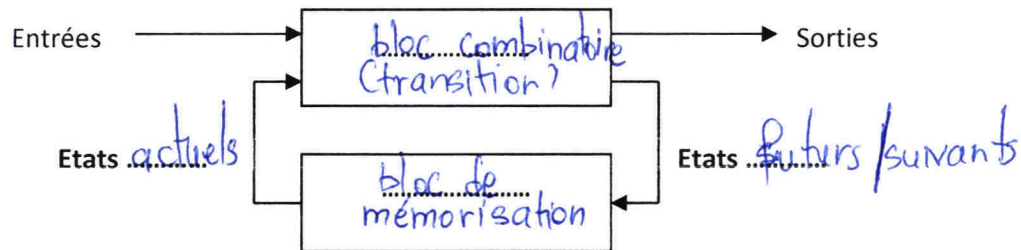
1. Questions de cours (6 points)
2. Exercices de conception (6 points)
3. Analyses de synthèses en VHDL (4 points)
4. Réalisation d'une synthèse en VHDL (4 points)

Vos réponses aux différentes questions sont à indiquer directement sur l'énoncé. L'ensemble du feuillet est à remettre obligatoirement au surveillant à la fin de l'examen.

Bon travail !

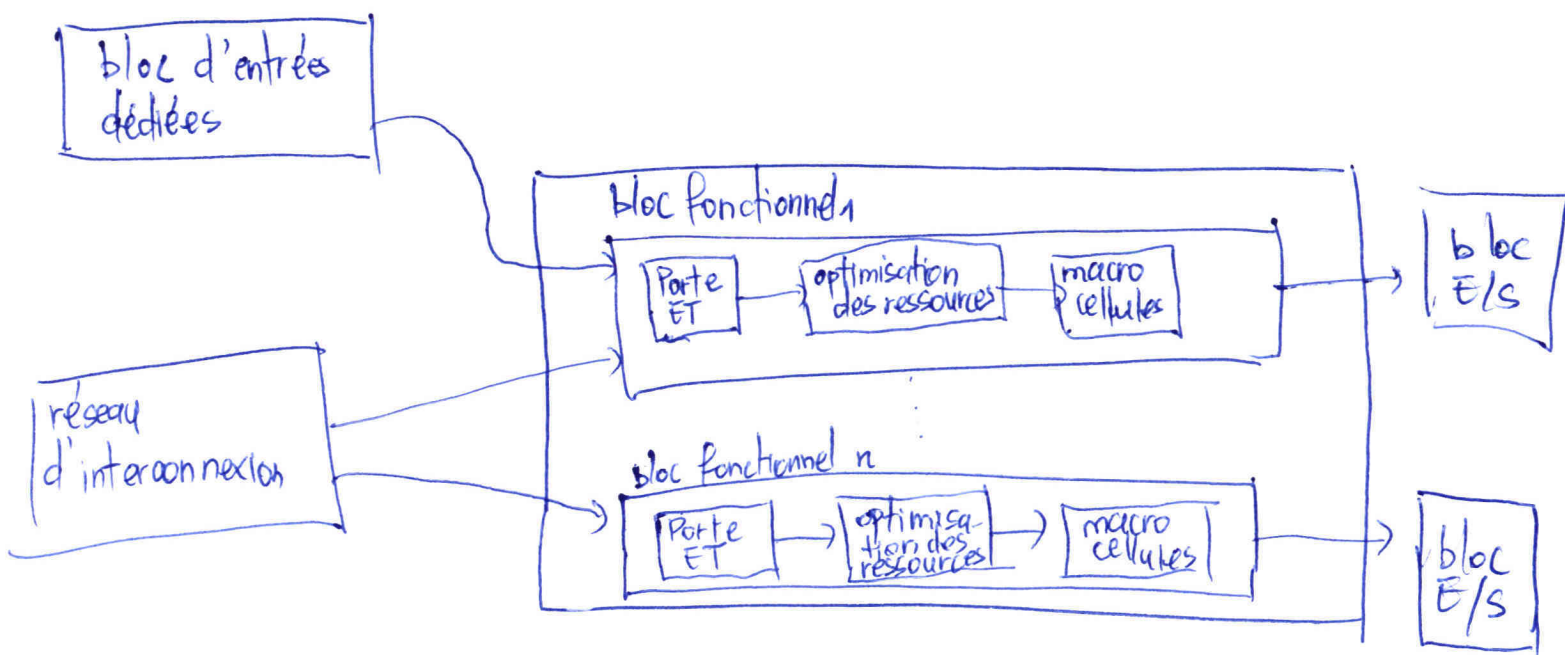
## 1. Questions de cours (6 points)

1.1. Complétez le schéma générique d'une structure séquentielle permettant de synthétiser un automate à l'aide de bascules.



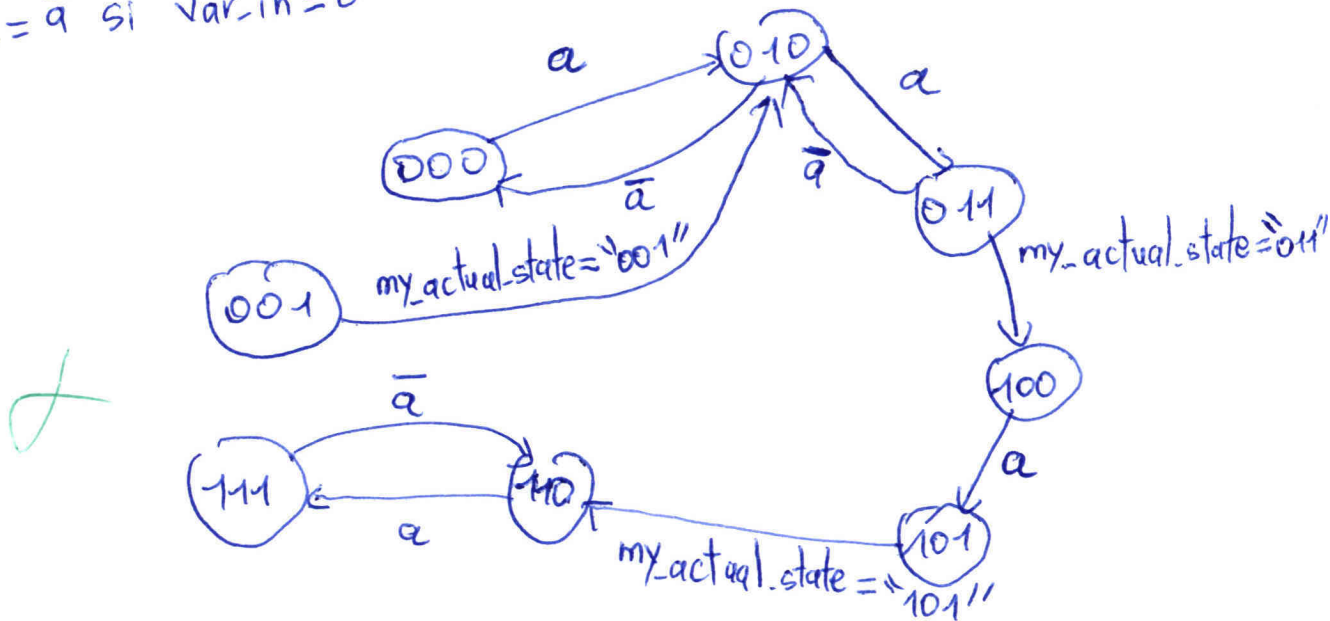
1.2. Décrivez, à l'aide d'un schéma, l'architecture « globale » interne d'un CPLD.

*Vous ferez, au minimum, apparaître les blocs des E/S, le PIA, les LAB ou FB et porterez une attention particulière à la représentation des interconnexions de ces blocs sur votre schéma.*

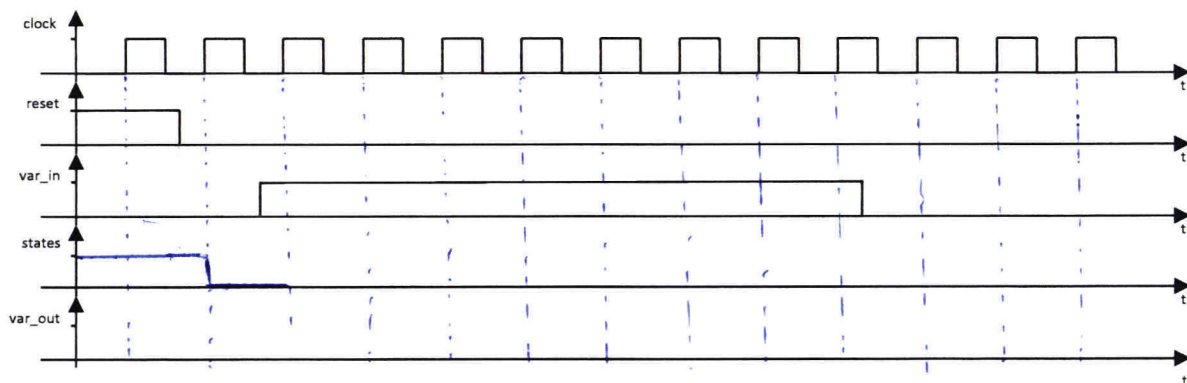


3.1.4. A l'aide d'un schéma, décrivez l'automate du code VHDL précédent en respectant le codage binaire des états et en faisant apparaître les transitions entre états, les entrées et les sorties.

Soit  $var_{in} = a$  si  $var_{in} = 1$   
 $var_{in} = \bar{a}$  si  $var_{in} = 0$



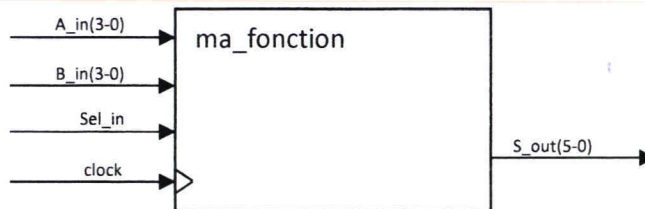
3.1.5. Complétez, à l'aide du code VHDL précédent, le chronogramme suivant pour les sorties  $var\_out$  et  $states$  en fonction des évolutions des différentes entrées.





#### 4. Réalisation d'une synthèse en VHDL (4 points)

On souhaite décrire en VHDL le fonctionnement « hardware » du bloc suivant :



Le bloc fonctionnel réalise un multiplieur / additionneur synchrone. Son fonctionnement est le suivant :

- A chaque front montant de l'horloge clock :
  - si l'entrée Sel\_in est égale à '0' : la sortie S\_out est égale  $A\_in * B\_in$  (multiplication)
  - si l'entrée Sel\_in est égale à '1' : la sortie S\_out est égale  $B\_in + B\_in$  (addition)

Pour la synthèse VHDL, les types de signaux utilisés sont STD\_LOGIC et STD\_LOGIC\_VECTOR.

##### 4.1.1. Complétez alors la description ci-dessous de l'entité.

entity ma\_fonction is

Port (

A\_in: in std\_logic\_vector(3 downto 0);  
 B\_in: in std\_logic\_vector(3 downto 0);  
 sel\_in: in std\_logic;  
 clock: in std\_logic;  
 S\_out: out std\_logic\_vector(5 downto 0)

);

end ma\_fonction;

##### 4.1.2. Complétez l'architecture nommée ma\_fonction\_arch de l'entité ma\_fonction permettant de décrire puis synthétiser ce bloc fonctionnel.

architecture ma\_fonction\_arch of ma\_fonction is

-- définition des signaux (si nécessaire) :

Signal Mys\_out: std\_logic\_vector(5 downto 0);



1.3. Quels sont les deux principaux types de FPGA (en termes de technologies d'interconnexions) ? Quelles en sont les principales différences (vous indiquerez au moins 2 différences) ?

- FPGA programmable une fois (fusible, anti-fusible)
- Programmable plusieurs fois

1.4. Que signifie l'acronyme VHDL ?

VHSIC Hardware ~~language~~ Description language

1.5. En VHDL, qu'appelle-t-on « liste de sensibilité » d'un process ? Que contient-elle ? Quelle est son rôle ?

la liste de sensibilité est la liste d'élément qui permet de réveiller un process, elle contient des broches d'entrées

- 1.6. L'entité « entity » permet-elle de décrire la vue externe ou interne d'une description matérielle ?

Elle permet de décrire la vue externe

- 1.7. L'architecture « architecture » permet-elle de décrire la vue externe ou interne d'une description matérielle ?

Elle permet de décrire la vue interne

- 1.8. Les corps des fonctions et les procédures contiennent-ils des instructions séquentielles ou concurrentielles ?

les corps des fonctions et des procédures contiennent des instructions séquentielles

- 1.9. Pourquoi peut-on dire « au sein d'une architecture, tout est process ! » ? Justifier votre réponse.

Parce qu'elles se font de manière concurrentielle

### 3. Analyses de synthèses en VHDL (4 points)

#### 3.1. On considère le code VHDL suivant représentant un automate :

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.STD_LOGIC_UNSIGNED.ALL;

entity DS_Auto_1 is
  Port (
    reset : in STD_LOGIC;
    clock : in STD_LOGIC;
    var_in : in STD_LOGIC;
    states : out STD_LOGIC_VECTOR (2 downto 0);
    var_out : out STD_LOGIC);
end DS_Auto_1;

architecture my_auto_arch of DS_Auto_1 is
  signal my_actual_state : STD_LOGIC_VECTOR (2 downto 0) := "000";
  signal my_next_state : STD_LOGIC_VECTOR (2 downto 0) := "000";
begin

  my_state_evolution_process : process (reset, clock)
  begin
    if (reset = '1') then
      my_actual_state <= "111";
    elsif (clock'event and clock = '1') then
      my_actual_state <= my_next_state;
    end if;
  end process;

  my_state_calulation_process : process (my_actual_state, var_in)
  begin
    case (my_actual_state) is
      when "000" =>
        if (var_in = '1') then my_next_state <= "001";
        else my_next_state <= "000";
        end if;
      when "001" =>
        my_next_state <= "010";
      when "010" =>
        if (var_in = '1') then my_next_state <= "011";
        else my_next_state <= "010";
        end if;
      when "011" =>
        my_next_state <= "100";
      when "100" =>
        if (var_in = '1') then my_next_state <= "101";
        else my_next_state <= "100";
        end if;
      when "101" =>
        my_next_state <= "110";
      when "110" =>
        if (var_in = '1') then my_next_state <= "111";
        else my_next_state <= "110";
        end if;
      when others =>
        my_next_state <= "000";
    end case;
  end process;

  var_out <= '1' when (my_actual_state(1 downto 0) = "00") else '0';

  states <= my_actual_state;

end my_auto_arch;
```

3.1.1. L'automate est-il initialisé (reset) de manière synchrone ou asynchrone (justifiez votre réponse) ?

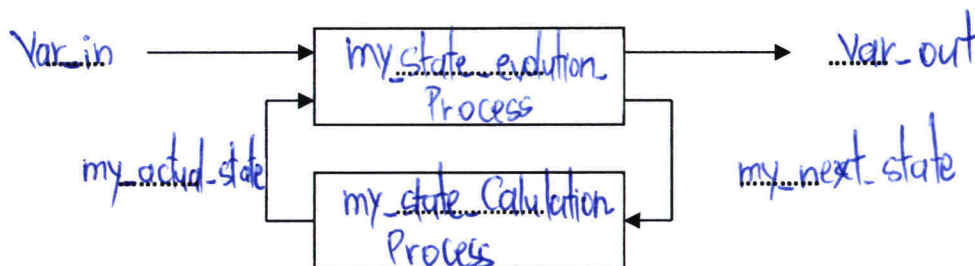
L'automate est initialisé de manière synchrone.  
En raison du reset dans la liste de sensibilité. Et il réagit à chaque front montant

3.1.2. L'évolution de l'automate s'effectue-t-elle sur front montant ou descendant de l'horloge d'entrée (clock) ?

L'évolution de l'automate se fait sur front montant d'horloge

3.1.3. Complétez le schéma suivant d'une structure séquentielle permettant de résoudre un automate à l'aide de bascules en y indiquant, au bon emplacement :

- my\_state\_evolution\_process
- my\_state\_calulation\_process
- my\_actual\_state
- my\_next\_state
- var\_in
- var\_out



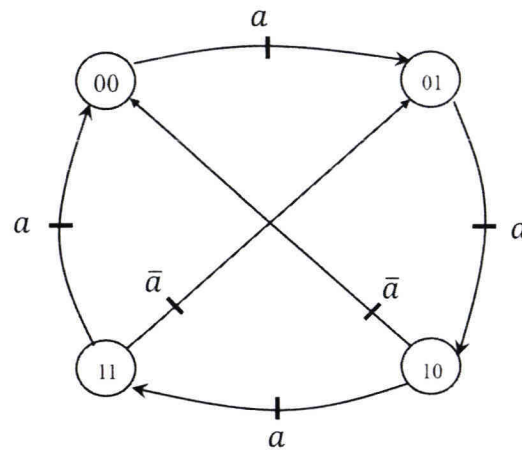
**1.10. Décrivez, de manière synthétique, le principe de fonctionnement de la technique du « clock gating » permettant de réduire la consommation d'énergie dans un composant programmable.**



**2. Exercices de conception (6 points)**

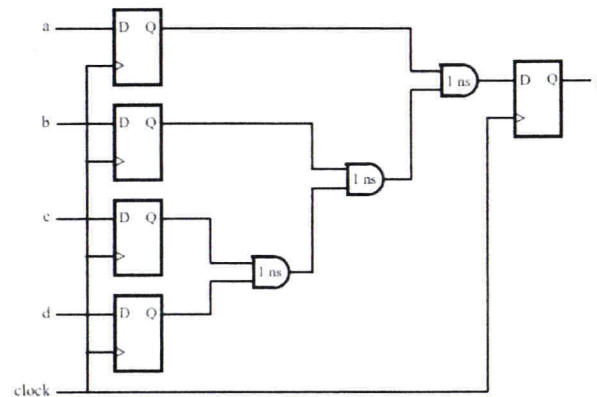
**2.1. Effectuer la synthèse du graphe d'état suivant à l'aide de deux bascules D avec la méthode du tableau.**

**Donner les expressions de D1 et D0**



2

## 2.2. On considère le montage ci-dessous :



### 2.2.1. Calculer la fréquence maximale $F_{MAX1}$ du circuit en considérant :

- $TSU = 1 \text{ ns}$
- $TCO = 1 \text{ ns}$
- skew négligeable

on sait que  $T_{max} = T_{CO} + 3T_{pd} + T_{SU}$

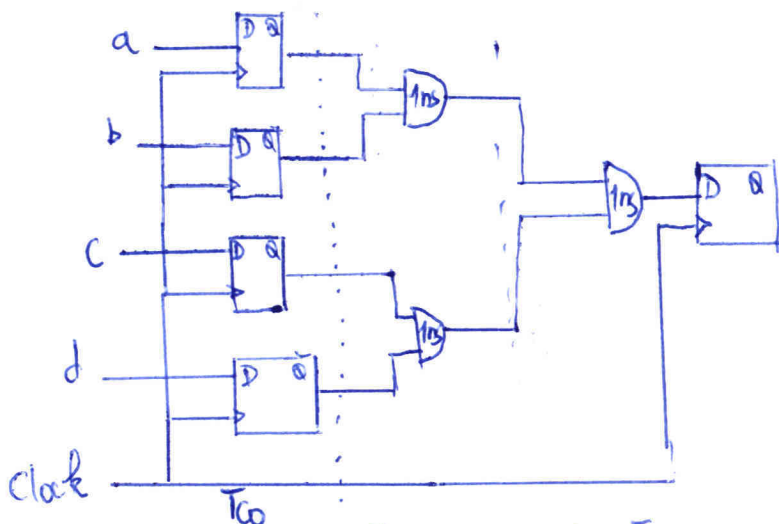
$$T_{max} = 1 + 3 + 1 = 5 \text{ ns}$$

$$F_{max1} = \frac{1}{T_{max}} = \frac{1}{5 \times 10^{-9}} = 0,2 \times 10^9 \text{ Hz}$$

$$F_{max1} = \frac{1}{T_{max1}} = \frac{1}{5} = 0,2 \text{ nHz}$$

### 2.2.2. Proposer un circuit équivalent fonctionnant à une fréquence maximale plus élevée avec les mêmes considérations pour TSU, TCO et skew.

Quelle est cette nouvelle valeur de fréquence maximale  $F_{MAX2}$  ?



$$T_{max2} = T_{CO} + 2T_{pd} + T_{SU} = 1 + 2 + 1 = 4 \text{ ns}$$

$$F_{max2} = \frac{1}{T_{max2}} = \frac{1}{4} = 0,25 \text{ nHz}$$



**2.3. On considère les extraits de codes VHDL du tableau suivant. Indiquez les fonctions logiques décrites par ces extraits (soyez précis dans vos descriptions).**

Tous les signaux utilisés sont de types STD\_LOGIC ou STD\_LOGIC\_VECTOR.

Extraits de codes VHDL	Fonctions logiques décrites ?
<pre> Myprocname1 : process (clock, reset) Begin     if (reset = '1') then         Qout &lt;= '0';     elsif (clock'event and clock = '1') then         Qout &lt;= Din;     end if; End process ; </pre>	<p>Il s'agit d'une bascule D. Elle est <del>sequentielle</del> et synchrone <del>com</del> et combinatoire, pas d'effet mémoire</p>
<pre> Myprocname2 : process (selection) Begin     if (Ain &gt; Bin) then         gre &lt;= '1'; equ &lt;= '0'; low &lt;= '0';     elsif (Ain = Bin) then         gre &lt;= '0'; equ &lt;= '1'; low &lt;= '0';     else         gre &lt;= '0'; equ &lt;= '0'; low &lt;= '1';     end if; End process ; </pre>	
<pre> Myprocname3 : process (clock, reset) Begin     if (clock'event and clock = '1') then         if (reset = '1' or Cntout = "1001") then             Cntout &lt;= "0000";         else             Cntout &lt;= CntOut+1;         end if;     end if; End process ; </pre>	<p>Il s'agit d'un compteur modulo <u>9</u>. Elle réagit au front montant, donc elle est synchrone. Il y a de l'effet mémoire donc elle est séquentielle</p>