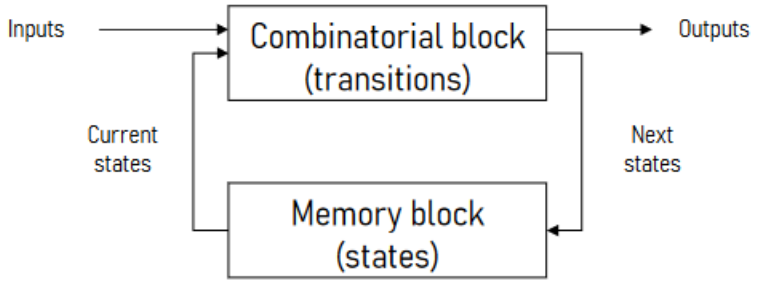
Conception de circuit numérique

DE 2022 S5 corrigé

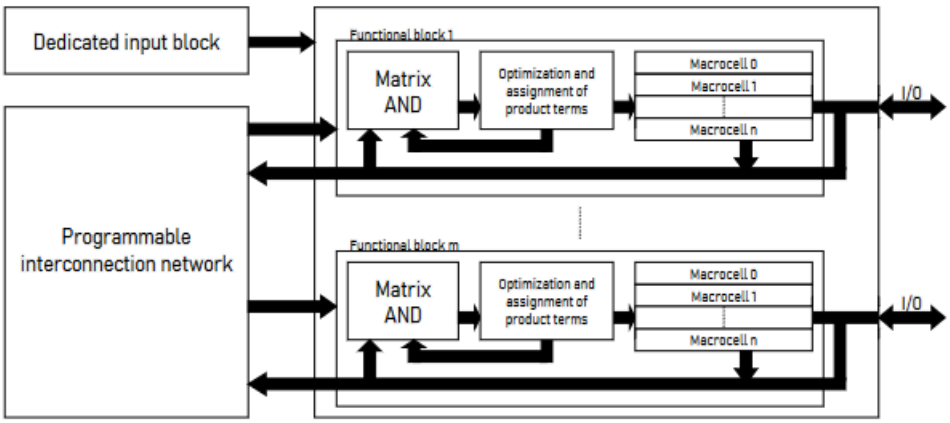
1 QUESTIONS DE COURS

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



Page 20/203

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.



Page 44/203

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)* ?

Irreversible (fuses or anti-fuses) => Anti-fuse FPGA

- Faster
- Not easy to use

Reversible (EEPROM or SRAM) => FPGA to SRAM

- Slower
- Easy to use

Type	Anti-fuse	EEPROM	SRAM
Speed	+	-	-
Density	+	-	--
Ease of use	-	+	+
Re-programmability	-	+	++

Page 63/203

1.4. Que signifie l'acronyme VHDL ?

"*VHSIC (Very High-Speed Integrated Circuit) Hardware Description Language*"

Page 92/203

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

The sensitivity list (signals separated by commas) allows you to define the signals allowing to "wake up" the process (signals to monitor...)

Page 135/203

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

The entity "entity" allows to describe the external view of a conception unit.

Page 107/203

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

The architecture "architecture" allows to describe the internal view of a conception unit.

Page 107/203

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

Function bodies and procedures are composed of sequential statements.

Page 121/203

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

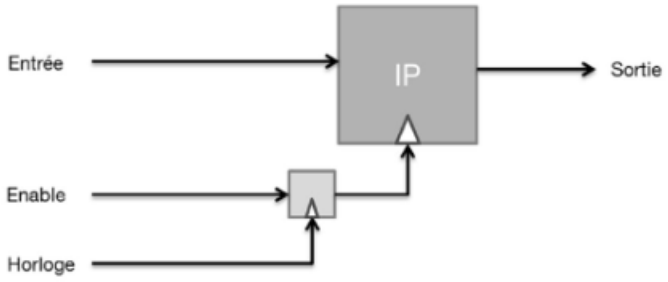
From the point of view of execution, it can be said that within an architecture that "Everything is PROCESS"
The processes are the basic computational elements of the simulator:

- Explicit processes: using the process statement
- Implicit processes: competitive instructions

Page 131/203

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.

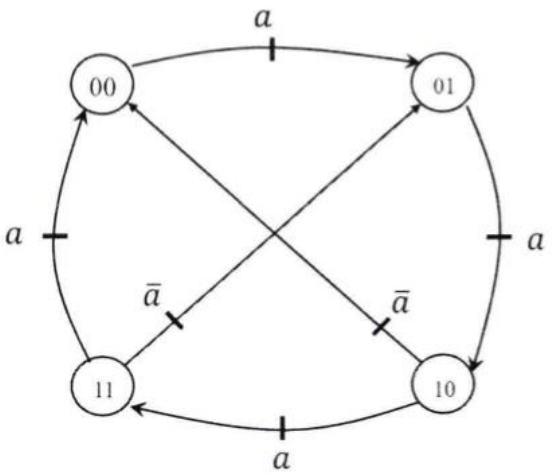
Clock gating is a circuit clock network control technique that consists of stopping the propagation of the clock on a part of the circuit when it is not needed.



Page 178/203

2 EXERCICES DE CONCEPTION

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



Current States		Condition	Next States	
Q_1	Q_2	a	D_1	D_2
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	0
1	0	1	1	1
1	1	0	0	1
1	1	1	0	0

D_0 :

Q_1Q_0 a	00	01	11	10
0	0	1	1	0
1	1	0	0	1

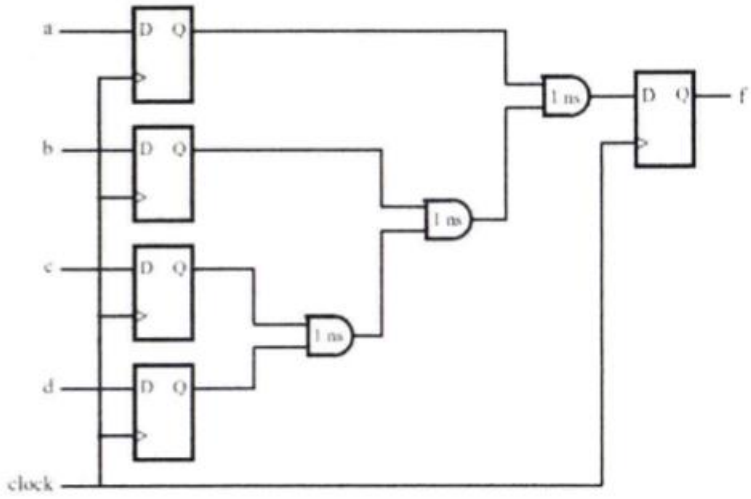
$D0 = Q0.\bar{a} + \overline{Q0}.a$
$D0 = Q0 \oplus a$

D_1 :

Q_1Q_0 a	00	01	11	10
0	0	0	0	0
1	0	(1)	0	(1)

$D1 = Q1.\overline{Q0}.a + \overline{Q1}.Q0.a$

2.2. On considère le montage ci-dessous :



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

- **TSU = 1 ns**
- **TCO = 1 ns**
- **skew négligeable**

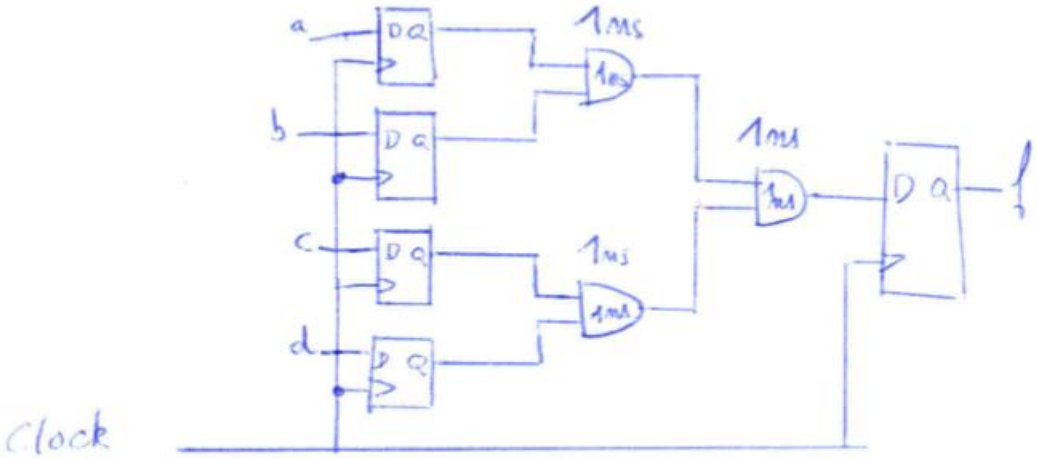
$$Fmax = \frac{1}{(Tco + Tpd + Tsu - skew)}$$

With :
T_{co} clock to output delay : time put by the gate to refresh its exit
T_{pd} propagation delay : propagation time between 2 gates
T_{su} setup time : minimum data stability time before the clock front

$$Fmax1 = \frac{1}{(1+3+1)\times 10^{-9}} = \frac{1}{5\times 10^{-9}} = \boxed{200\ MHz}$$

For more information -> Page 169/203

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} ?



$$Fmax1 = \frac{1}{(1+2+1)\times 10^{-9}} = \frac{1}{4\times 10^{-9}} = \boxed{250\ MHz}$$

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.

Myprocname1 : process (clock, reset)
Begin
if (reset = '1') then
Qout <= '0';
elsif (clock'event and clock = '1') then
Qout <= Din;
end if;
End process ;
	This process is a D flip-flop.
	If reset is at '1', the output will be 0.
	Else, the output will be the input value.

Myprocname2 : process (selection)
Begin
if (Ain > Bin) then
gre <= '1'; equ <='0'; low <= '0';
elsif (Ain = Bin) then
gre <= '0'; equ <='1'; low <= '0';
else
gre <= '0'; equ <='0'; low <= '1';
end if;
End process ;
	This process allows to compare two inputs.
	If the input A is greater than input B, "gre" takes the value '1'.
	If the input A is equal to input B, "equ" takes the value '1'.
	If the input A is lower than input B, "lower" takes the value '1'.

Myprocname3 : process (clock, reset)
Begin
if (clock'event and clock = '1') then
If (reset = '1' or Cntout = "1001") then
Cntout <= "0000";
else
Cntout <= CntOut+1;
end if;
end if;
End process ;
	This process is a synchronous counter, from 0 to 9 in binary.
	If reset is at '1' or the counter reaches "1001", the counter is reset to "0000"
	Else, the counter is incremented

3 ANALYSES DE SYNTHÈSES EN VHDL

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) ?

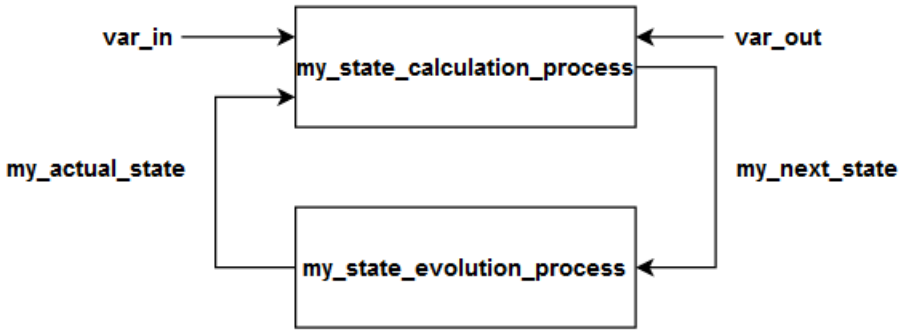
In this process, the reset is used regardless of the clock. The automata is initialized in an asynchronous way.

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

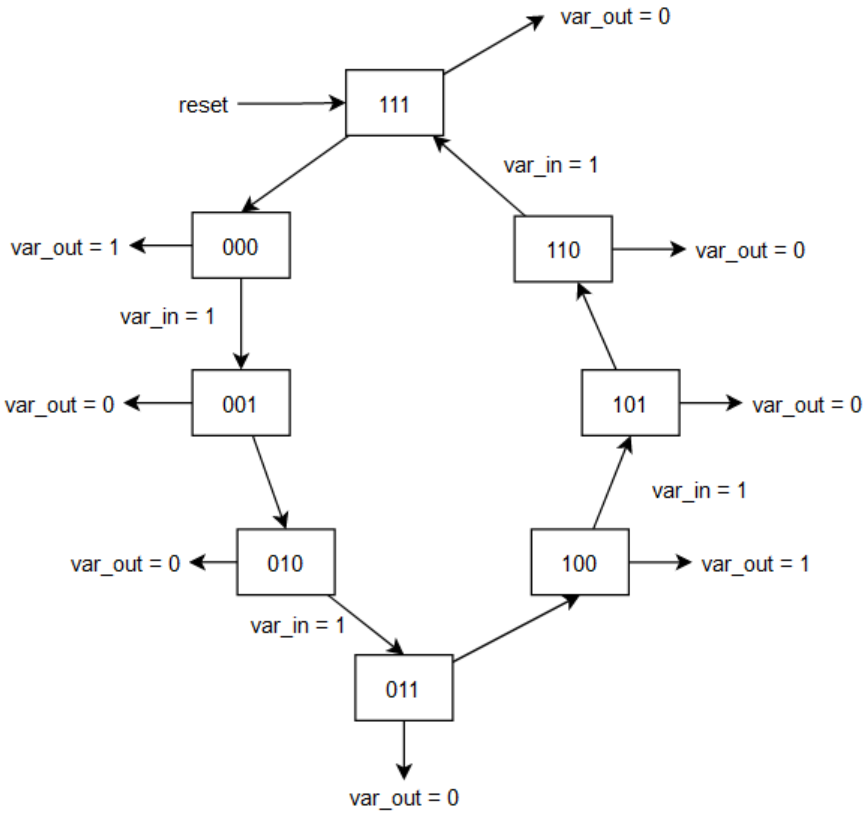
By looking at the condition *clock='1'*, the instructions are executed on clock rising edge.

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**



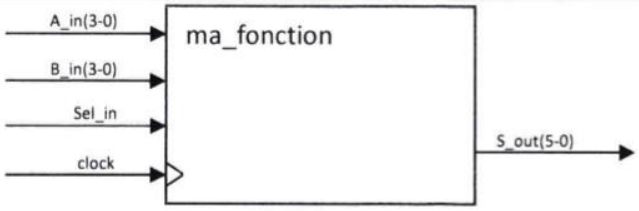
3.1.4. A l'aide d'un schéma, décrivez l'automate du code VHDL précédent en respectant le codage binaire des états et en faisant apparaître les transitions entre états, les entrées et les sorties.



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

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) :

begin

-- début de la description de l'architecture

```
ma_fonction_arch : process(clock, Sel_in, A_in, B_in)
   begin
      if(rising_edge(clock)) then
         if Sel_in = '0' then
            S_out(5 downto 3) <= "00"
            S_out(2 downto 0) <= A_in * B_in;
         else
            S_out(5 downto 3) <= "00"
            S_out(2 downto 0) <= B_in + B_in;
         end if;
      end if;
   end process;
end ma_fonction_arch;
```

end ma_fonction_arch;