

**Correction de l'examen de la matière :
Electronique Numérique Avancée FPGA & VHDL (ESE72)**

Exercice n°1 : (04 points) Types de PLDs

Indiquez, pour chacune des expressions suivantes, quelle technologie cible lui correspond le mieux dans la liste suivante: FPGA, CPLD, PAL, PLA, EPLD ou FPGA-plateforme(PSOC).

1. Dispositifs logiques programmables ayant la plus haute densité : **CPLD** (1 pt)
2. Dispositifs logiques programmables complexes le plus souvent configurables via une mémoire vive statique : **FPGA EPLD** (1pt)
3. Dispositifs logiques programmables complexes plutôt combinatoires que séquentiels : **PLA** (1pt)
4. Dispositifs logiques programmables simples avec matrice-ET et matrice-OU programmables : **EPLD** (0,5 pt)
5. Dispositifs logiques programmables complexes dont les délais entrée-sortie sont prévisibles : **EPLD** (0,5 pt)

Exercice N°2 : (5 points)

Il y'a plusieurs façons d'écrire la description VHDL d'un multiplexeur. On cite deux méthodes :

Première méthode (la plus simple):

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity MUX is port(
X0, X1, X2, X3: in std_logic;
A, B : in std_logic_vector(1 downto 0) ;
Y: out std_logic);
End MUX;
architecture DESCRIPTION of MUX is
begin
S <= ((not A) and (not B) and X0) OR
(A and (not B) and X1) OR
((not A) and B and X2) OR
(A and B and X3);
end DESCRIPTION ;

```

Deuxième méthode :

```

library ieee;
use ieee.std_logic_1164.all;
use ieee.std_logic_arith.all;
use ieee.std_logic_unsigned.all;
entity MUX is port(
X0, X1, X2, X3: in std_logic;
A, B : in std_logic_vector(1 downto 0) ;
Y: out std_logic);
end;
architecture DESCRIPTION of MUX is
begin
Y <= X0 when (A='0' and B='0') else
X1 when (A='1' and B='0') else
X2 when (A='0' and B='1') else
X3 when (A='1' and B='1') else
'-';
end DESCRIPTION ;

```

Exercice N°3 : (5 points)

Description vhdl d'un compteur modulo 16 synchrone :

```
Library IEEE;
Use ieee.std_logic_1164.all;
Use ieee.std_logic_arith.all;
Use ieee.std_logic_unsigned.all;
entity CMP4BITS is
PORT (CLOCK : in std_logic;
Q :out std_logic_vector(3 downto 0));
end CMP4BITS;
architecture DESCRIPTION of CMP4BITS is
SIGNAL S : STD_LOGIC_VECTOR (3 DOWNTO 0);
begin
process (CLOCK)
begin
if (CLOCK ='1' and CLOCK'event) then
Q <= Q + 1;
end if;
end process;
q<=s;    -- affectation du bus interne au
         -- signal de sortie Q
end DESCRIPTION;
```

Exercice N°4 : (6 points)

On considère le programme ci-dessous (écrit en VHDL) :

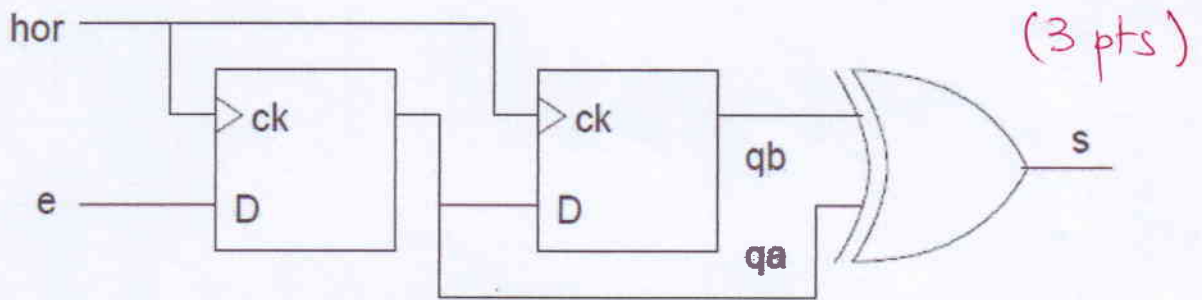
```
entity transitm is
port ( hor, e : in bit ;
s : out bit );
end transitm ;
architecture quasi_struct of transitm is
signal qa, qb : bit ;
begin
s <= qa xor qb ;
```

```

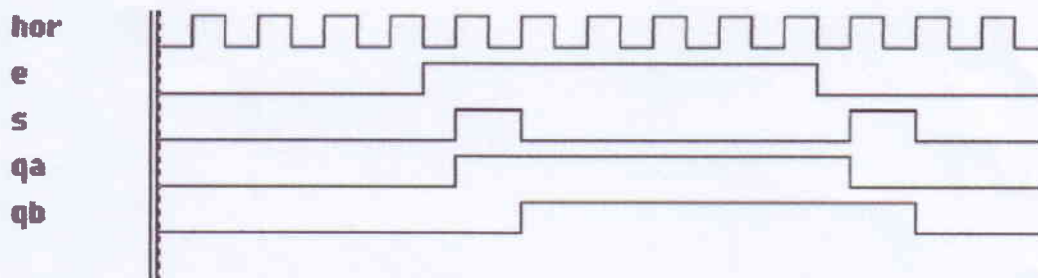
schem : process
begin
wait until hor = '1' ;
qa <= e ;
qb <= qa ;
end process schem ;
end quasi_struct ;

```

a. Dédurre de ce programme, par une construction méthodique, un schéma (bascule et portes logiques).



• Chronogramme :



Le chargé de la matière :

Dr.SAHRAOUI kamel