



## Devoir surveillé d'informatique industrielle Circuits programmables et VHDL

### Questions de cours (8 points)

On considère le programme ci-dessous :

```
entity pot_pourri is
    port ( a,b,c : in bit ;
          d1,d2,d3,d4 : out bit ) ;
end pot_pourri ;

architecture ww of pot_pourri is
begin
ww1 : process(a,b,c)
begin
    if a = '0' then
        d1 <= b ;
    else
        d1 <= c ;
    end if ;
end process ww1;

ww2 : process(a,b)
begin
    if a = '0' then
        d2 <= '0' ;
    elsif b = '1' then
        d2 <= '1' ;
    end if ;
end process ww2;

ww3 : process(a)
begin
    if a = '1' and a'event then
        if b = '0' then
            d3 <= '0' ;
        elsif c = '1' then
            d3 <= '1' ;
        end if ;
    end if ;
end process ww3;

ww4 : process(a,b)
begin
    if b = '0' then
        d4 <= '0' ;
    elsif a = '1' and a'event then
        d4 <= '1' ;
    end if ;
end process ww4 ;

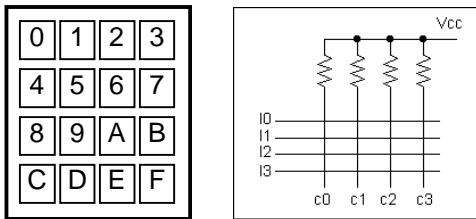
end ww ;
```

Chacun des processus de ce programme décrit un opérateur logique.

1. Pour chacun de ces opérateurs précisez, en justifiant votre réponse, s'il s'agit d'un opérateur combinatoire, d'un opérateur séquentiel asynchrone ou d'un opérateur séquentiel synchrone.
2. Quelle est la fonction (décrivez la en quelques lignes d'explications, une expression logique etc.) réalisée par chacun de ces opérateurs ?
3. Malgré la simplicité des équations générées par le programme, ce programme ne rentre ni dans un circuit du type 16V8, ni dans un circuit du type 22V10. Pourquoi ?

### Explorateur séquentiel de clavier hexadécimal. (12 points)

Derrière un clavier hexadécimal, comme ceux qui sont utilisés dans les digicodes se cache le câblage décrit ci-dessous :



le clavier est une grille de fils (4 lignes l0, l1, l2 et l3 et 4 colonnes c0, c1, c2, c3) qui se croisent sans se toucher. Quand une touche est enfoncée on met en contact une ligne et une colonne.

L'idée est de mettre les colonnes à '1' via une résistance de « pull-up » connectée à l'alimentation ('1' logique), les lignes sont pilotées par un

automate : toutes à 'Z' sauf une qui est à '0' et change au rythme d'une horloge.

L'automate reçoit en entrée les niveaux des colonnes : '1' ou '0'.

Quand une touche est enfoncée l'automate détecte un zéro quand la ligne qu'il pilote est celle de la touche, et à la colonne qui correspond.

Une possibilité de réalisation de l'encodeur de clavier est décrite par le programme de la page suivante.

1. Donner un schéma de principe du câblage à réaliser, l'explorateur de clavier sera représenté par une « boîte noire ».
2. Donner un schéma fonctionnel (une boîte noire par processus implicite ou explicite) de l'explorateur. On précisera pour chaque bloc sa nature combinatoire ou séquentielle.
3. Combien de bascules faut-il pour réaliser l'explorateur ?
4. L'explorateur ne rentre pas dans un circuit du type 22V10, pourquoi ?
5. Décrire le processus « autom » par un diagramme de transitions.
6. Si aucune touche n'est enfoncée décrire par un chronogramme les valeurs des sorties l0 à l3 en fonction du temps.
7. Quel est le rôle du signal « touche » ? Quel est le schéma logique qui le génère ? Est-il visible en sortie du montage ?
8. Expliquez comment est construit le code d'une touche.
9. Quel est le code généré si l'utilisateur appuie simultanément sur les touches 5 et 7 ?
10. Un clavier est un organe mécanique, qui génère des faux contacts. On souhaite réaliser un automate qui ne donne l'indication de touche enfoncée que si le contact est détecté pendant au moins quatre périodes d'horloges consécutives. Proposer un petit automate qui réalise cette fonction ; il reçoit en entrée touche\_in et fournit en sortie un signal touche\_in\_s débarrassé de toutes les impulsions de durée inférieures à 4 périodes d'horloge :



11. Compléter le programme VHDL fourni pour y inclure l'automate précédent.

```

library ieee ;
use ieee.std_logic_1164.all, ieee.numeric_std.all ;

entity explorateur_clavier is
port (hor,c3,c2,c1,c0 : in std_logic;
      l3,l2,l1,l0,strobe: out std_logic ;
      sort : out std_logic_vector(3 downto 0)) ;
end explorateur_clavier;

architecture simple of explorateur_clavier is
  signal touche : std_logic ;
  signal etat, colnum : integer range 0 to 3 ;
begin

l0 <= '0' when etat = 0 else 'Z' ;
l1 <= '0' when etat = 1 else 'Z' ;
l2 <= '0' when etat = 2 else 'Z' ;
l3 <= '0' when etat = 3 else 'Z' ;

strobe <= touche ;

strob : process
begin
  wait until rising_edge(hor);
  touche <= c3 and c2 and c1 and c0 ;
end process strob ;

autom : process
begin
  wait until rising_edge(hor);
  if touche = '1' then
    etat <= (etat + 1) mod 4 ;-- mod est l'opérateur modulo
  end if ;
end process autom ;

code_gen : process
begin
  wait until rising_edge(hor) ;
  if touche = '0' then
-- l'expression ci-dessous convertit un nombre entier en vecteur
    sort <= std_logic_vector(to_unsigned(colnum + 4*etat,4)) ;
  end if ;
end process code_gen ;

encodeur : process(c0,c1,c2,c3)
begin
  if c0 = '0' then
    colnum <= 0 ;
  elsif c1 = '0' then
    colnum <= 1 ;
  elsif c2 = '0' then
    colnum <= 2 ;
  elsif c3 = '0' then
    colnum <= 3 ;
  else
    colnum <= 0 ;
  end if ;
end process encodeur ;
end simple ;

```