CEG 3555: Systèmes Numériques II
(Automne 2023)
Prof. Rami Abielmona
Solutions Possibles pour Devoir #1:
Optimisation de Fonction Logique

3 Octobre, 2023

Question I

Cette question se concerne avec l'optimisation de fonction logique en utilisant la manipulation algébrique.

Partie a

`Commençant par les sommes-de-produits canoniques pour f, on obtient:`

$$f = \overline{x_1} \cdot \overline{x_2} \cdot x_3 + \overline{x_1} \cdot x_2 \cdot \overline{x_3} + \overline{x_1} \cdot x_2 \cdot x_3 + x_1 \cdot \overline{x_2} \cdot \overline{x_3} + x_1 \cdot \overline{x_2} \cdot x_3 + x_1 \cdot x_2 \cdot \overline{x_3} + x_1 \cdot x_2 \cdot x_3 \quad (1)$$

$$= x_1(\overline{x_2}\cdot\overline{x_3}+\overline{x_2}\cdot x_3+x_2\cdot\overline{x_3}+x_2\cdot x_3)+x_2(\overline{x_1}\cdot\overline{x_3}+\overline{x_1}\cdot x_3+x_1\cdot\overline{x_3}+x_1\cdot x_3)+x_3(\overline{x_1}\cdot\overline{x_2}+\overline{x_1}\cdot x_2+x_1\cdot\overline{x_2}+x_1\cdot x_2)$$

$$= x_1(\overline{x_2} \cdot 1 + x_2 \cdot 1) + x_2(\overline{x_1} \cdot 1 + x_1 \cdot 1) + x_3(\overline{x_1} \cdot 1 + x_1 \cdot 1)$$

$$= x_1(\overline{x_2} \cdot x_2) + x_2(\overline{x_1} \cdot x_1) + x_3(\overline{x_1} \cdot x_1)$$

$$= x_1 \cdot 1 + x_2 \cdot 1 + x_3 \cdot 1$$

$$= x_1 + x_2 + x_3 \quad (2)$$

Partie b

`Commençant par les produits-de-sommes canoniques pour f, on obtient:`

$$f = (x_1+x_2+x_3)(x_1+x_2+\overline{x_3})(x_1+\overline{x_2}+x_3)(x_1+\overline{x_2}+\overline{x_3})\cdot(\overline{x_1}+x_2+x_3)(\overline{x_1}+x_2+\overline{x_3})(\overline{x_1}+\overline{x_2}+x_3) \quad (3)$$

$$= ((x_1+x_2+x_3)(x_1+x_2+\overline{x_3}))((x_1+\overline{x_2}+x_3)(x_1+\overline{x_2}+\overline{x_3}))\cdot((\overline{x_1}+x_2+x_3)(\overline{x_1}+x_2+\overline{x_3}))((\overline{x_1}+\overline{x_2}+x_3)(\overline{x_1}+x_2+x_3))$$

$$= (x_1 + x_2 + x_3 \cdot \overline{x_3})(x_1 + \overline{x_2} + x_3 \cdot \overline{x_3})(\overline{x_1} + x_2 + x_3 \cdot \overline{x_3})(\overline{x_1} + \overline{x_2} \cdot x_2 + x_3)$$

$$= (x_1 + x_2)(x_1 + \overline{x_2})(\overline{x_1} + x_2)(\overline{x_1} + x_3)$$

$$
\begin{aligned}
&= (x_1 + x_2 \cdot \overline{x_2})(\overline{x_1} + x_2 \cdot x_3) \\
&= x_1(\overline{x_1} + x_2 \cdot x_3) \\
&= x_1 \cdot \overline{x_1} + x_1 \cdot x_2 \cdot x_3 \\
&= x_1 \cdot x_2 \cdot x_3 \qquad (4)
\end{aligned}
$$

Partie c

Une expression SOP de mimimum-coût pour la fonction $f(x_1, x_2, x_3) = \sum m(3, 4, 6, 7)$ est

$$f = x_1 \cdot x_2 + x_1 \cdot \overline{x_3} + \overline{x_1} \cdot x_3 \qquad (5)$$

Le circuit correspondant réalisé à l'aide des portes NON-ET est présenté dans le schéma 1.

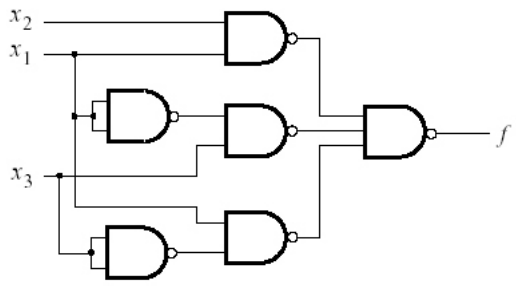

Figure 1: Circuit correspondant à la partie c

Partie d

Une expression POS de mimimum-coût pour la fonction $f(x_1, x_2, x_3) = \sum m(3, 4, 6, 7)$ est

$$f = (x_1 + x_3)(\overline{x_1} + x_2 + \overline{x_3}) \qquad (6)$$

Le circuit correspondant réalisé à l'aide des portes NON-OU est présenté dans le schéma 2.

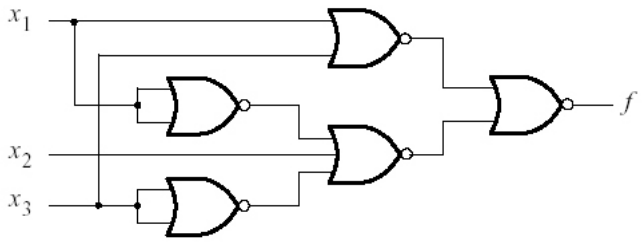

Figure 2: Circuit correspondant à la partie d

Partie e

Commençant avec les deux fonctions:

$$f(x,y,z) = x \cdot y + y \cdot z + \overline{x} \cdot z + \overline{x} \cdot \overline{y} \tag{7}$$
$$g(x,y,z) = x \cdot y + \overline{x} \cdot \overline{y} + \overline{x} \cdot y \cdot z \tag{8}$$

On peut les représenter dans leurs formes canoniques:

$$f(x,y,z) = x \cdot y \cdot (z + \overline{z}) + (x + \overline{x}) \cdot y \cdot z + \overline{x} \cdot (y + \overline{y}) \cdot z + \overline{x} \cdot \overline{y} \cdot (z + \overline{z})$$
$$= x \cdot y \cdot z + x \cdot y \cdot \overline{z} + \overline{x} \cdot y \cdot z + \overline{x} \cdot \overline{y} \cdot z + \overline{x} \cdot \overline{y} \cdot \overline{z} \tag{9}$$
$$g(x,y,z) = x \cdot y + \overline{x} \cdot \overline{y} + \overline{x} \cdot y \cdot z = x \cdot y \cdot (z + \overline{z}) + \overline{x} \cdot \overline{y} \cdot (z + \overline{z}) + \overline{x} \cdot y \cdot z$$
$$= x \cdot y \cdot z + x \cdot y \cdot \overline{z} + \overline{x} \cdot \overline{y} \cdot z + \overline{x} \cdot \overline{y} \cdot \overline{z} + \overline{x} \cdot y \cdot z \tag{10}$$

Puisque les deux formes canoniques des équations 9 et 10 sont identiques, elles représentent la même fonction booléenne.

Question II

Cette question se concerne avec la synthèse à sorties et niveaux multiples.

Partie a

La réalisation la plus simple pour la fonction $f(x_1, ..., x_4) = \sum m(0, 3, 4, 7, 9, 10, 13, 14)$, supposant que les portes logiques ont un fan-in maximum de deux est:

$$f = x_1 \cdot g + \overline{x_1} \cdot \overline{g} \tag{11}$$

où $g = \overline{x_3} \cdot x_4 + x_3 \cdot \overline{x_4}$.

Partie b

Si on laisse $D(0,20) = 0$ et $D(15,26) = 1$, la décomposition produit

$$g = x_5(\overline{x_1} + x_2)$$

$$f = (\overline{x_3} \cdot \overline{x_4} + x_3 \cdot x_4)g + \overline{x_3} \cdot x_4 \cdot g = x_3 \cdot x_4 \cdot g + \overline{x_3} \cdot \overline{x_4} \cdot g + \overline{x_3} \cdot x_4 \cdot g$$

Le coût de ce circuit est 9 + 18 = 27. Par conséquent, la forme SOP optimale est:

$$f = \overline{x_3} \cdot x_4 \cdot \overline{x_5} + \overline{x_1} \cdot x_3 \cdot x_4 \cdot x_5 + \overline{x_1} \cdot \overline{x_3} \cdot \overline{x_4} \cdot x_5 + x_2 \cdot \overline{x_3} \cdot \overline{x_4} \cdot x_5 + x_2 \cdot x_3 \cdot x_4 \cdot x_5$$

Le coût de ce circuit est 7 + 29 = 36, alors on a réduit le coût de la réalisation par 9 en utilisant la décomposition fonctionelle.

Partie c

La carte de Karnaugh à cinq variables est divisé en deux cartes de Karnaugh à quatre variables. La carte de gauche représenté en figure 3 contient les mintermes où $a = 0$ et la carte de droite représenté en figure 3 contient les mintermes où $a = 1$.

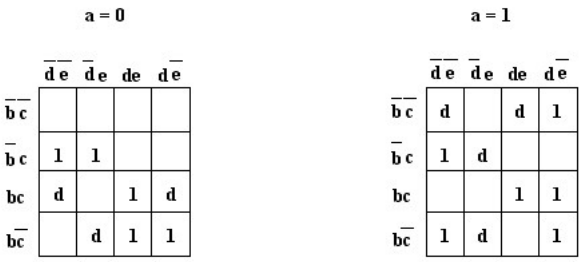

Figure 3: Cartes de Karnaugh à cinq variables

On peut grouper les deux cartes en une seule carte comme démontré en figure 4.

On peut alors réduire la carte en deux étapes: (i) grouper tous les terms avec des 1's et des d's (figure 5), puis (ii) grouper tous les termes avec des variablse avec des 1's et d's adjacents (figure 5).

La première étape nous donne $f(a,b,c,d,e) = \overline{b} \cdot c \cdot \overline{d} + b \cdot c \cdot d$, et la deuxième étape nous donne notre fonction minimisée finale $f(a,b,c,d,e) = a \cdot \overline{c} \cdot \overline{e} + \overline{a} \cdot b \cdot d + \overline{b} \cdot c \cdot \overline{d} + b \cdot c \cdot d$.

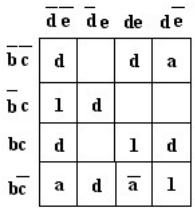

Figure 4: Carte de Karnaugh minimisé

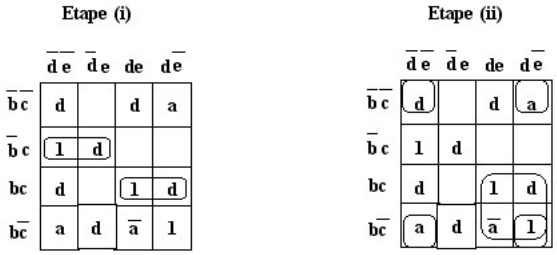

Figure 5: Deux étapes de minimisation

Partie d

En utilisant des cartes de Karnaugh pour chacune de ces fonctions, on obtient les expressions booléennes suivantes:

$$f_1 = \overline{a} \cdot \overline{c} \cdot \overline{d} + b \cdot \overline{c} \cdot d + a \cdot d + \overline{b} \cdot c \cdot \overline{d}$$

$$f_2 = a \cdot b + b \cdot \overline{c} \cdot d + a \cdot c + \overline{b} \cdot c \cdot \overline{d}$$

$$f_3 = \overline{a} \cdot \overline{b} \cdot \overline{c} + \overline{a} \cdot \overline{b} \cdot c + a \cdot b \cdot c + a \cdot c$$

Il y a 12 termes de produit dans les expressions originales, dont 4 peuvent être partagés, alors on peut récrire les fonctions comme:

$$f_1 = W + X + a \cdot d + Z \tag{12}$$

$$f_2 = a \cdot b + X + Y + Z \tag{13}$$

$$f_3 = W + \overline{a} \cdot \overline{b} \cdot c + Y + a \cdot b \cdot b \tag{14}$$

où $W = \overline{a} \cdot \overline{b} \cdot \overline{d}, X = b \cdot \overline{c} \cdot d, Y = a \cdot c,$ et $Z = \overline{b} \cdot c \cdot \overline{d}$.

Question III

Cette question se concerne avec les composants de circuit combinationnel.

Partie a

L'expansion de Shannon en termes de w_2 produit

$$f_{w_2} = \overline{w_2}(1 + \overline{w_1} \cdot \overline{w_3} + w_1 \cdot w_3) + w_2(\overline{w_1} \cdot \overline{w_3} + w_1 \cdot w_3)$$

$$= \overline{w_1} \cdot \overline{w_2} \cdot \overline{w_3} + w_1 \cdot \overline{w_2} \cdot w_3 + \overline{w_2} + \overline{w_1} \cdot w_2 \cdot \overline{w_3} + w_1 \cdot w_2 \cdot w_3$$

L'expansion successive de Shannon en termes de w_1 produit

$$f_{w_1} = \overline{w_1}(w_2 \cdot \overline{w_3} + \overline{w_2} \cdot \overline{w_3} + \overline{w_2}) + w_1(w_2 \cdot w_3 + \overline{w_2} \cdot w_3 + \overline{w_2})$$

$$= \overline{w_1} \cdot w_2 \cdot \overline{w_3} + \overline{w_1} \cdot \overline{w_2} \cdot \overline{w_3} + \overline{w_1} \cdot \overline{w_2} + w_1 \cdot w_2 \cdot w_3 + w_1 \cdot \overline{w_2} \cdot w_3 + w_1 \cdot \overline{w_2}$$

L'expansion successive de Shannon en termes de w_3 produit

$$f_{w_3} = \overline{w_3}(\overline{w_1} \cdot w_2 + \overline{w_1} \cdot \overline{w_2} + \overline{w_1} \cdot \overline{w_2} + w_1 \cdot \overline{w_2}) + w_3(w_1 \cdot w_2 + w_1 \cdot \overline{w_2} + w_1 \cdot \overline{w_2} + \overline{w_1} \cdot \overline{w_2})$$

$$= \overline{w_1} \cdot w_2 \cdot \overline{w_3} + \overline{w_1} \cdot \overline{w_2} \cdot \overline{w_3} + w_1 \cdot \overline{w_2} \cdot \overline{w_3} + w_1 \cdot w_2 \cdot w_3 + w_1 \cdot \overline{w_2} \cdot w_3 + \overline{w_1} \cdot \overline{w_2} \cdot w_3$$

Partie b

L'expansion de Shannon en termes de w_1 produit

$$f_{w_1} = \overline{w_1} \cdot w_2 + \overline{w_1} \cdot \overline{w_3} + w_1 \cdot w_2$$

L'expansion successive de Shannon en termes de w_2 produit

$$f_{w_2} = \overline{w_2}(\overline{w_1} \cdot \overline{w_3}) + w_2(w_1 + \overline{w_1} + \overline{w_1} \cdot \overline{w_3})$$

$$= \overline{w_1} \cdot w_2 + \overline{w_1} \cdot w_2 \cdot \overline{w_3} + \overline{w_1} \cdot \overline{w_2} \cdot \overline{w_3} + w_1 \cdot w_2$$

L'expansion successive de Shannon en termes de w_3 produit

$$f_{w_3} = \overline{w_3}(\overline{w_1} \cdot \overline{w_2} + w_1 \cdot w_2 + \overline{w_1} \cdot w_2 + \overline{w_1} \cdot w_2) + w_3(w_1 \cdot w_2 + \overline{w_1} \cdot w_2)$$

$$= \overline{w_1} \cdot \overline{w_2} \cdot \overline{w_3} + w_1 \cdot w_2 \cdot \overline{w_3} + \overline{w_1} \cdot w_2 \cdot \overline{w_3} + \overline{w_1} \cdot w_2 \cdot w_3 + w_1 \cdot w_2 \cdot w_3$$

Partie c

D'abord, laissez-nous définir un ensemble de variables intermédiaires:

$$i_0 = \overline{w_7} \cdot \overline{w_6} \cdot \overline{w_5} \cdot \overline{w_4} \cdot \overline{w_3} \cdot \overline{w_2} \cdot \overline{w_1} \cdot w_0$$

$$i_1 = \overline{w_7} \cdot \overline{w_6} \cdot \overline{w_5} \cdot \overline{w_4} \cdot \overline{w_3} \cdot \overline{w_2} \cdot w_1$$

$$i_2 = \overline{w_7} \cdot \overline{w_6} \cdot \overline{w_5} \cdot \overline{w_4} \cdot \overline{w_3} \cdot w_2$$

$$i_3 = \overline{w_7} \cdot \overline{w_6} \cdot \overline{w_5} \cdot \overline{w_4} \cdot w_3$$

$$i_4 = \overline{w_7} \cdot \overline{w_6} \cdot \overline{w_5} \cdot w_4$$

$$i_5 = \overline{w_7} \cdot \overline{w_6} \cdot w_5$$

$$i_6 = \overline{w_7} \cdot w_6$$

$$i_7 = w_7$$

Maintenant, un encodeur binaire traditionnel peut être utilisé pour réaliser l'encodeur de priorité:

$$y_0 = i_1 + i_3 + i_5 + i_7$$

$$y_1 = i_2 + i_3 + i_6 + i_7$$

$$y_2 = i_4 + i_5 + i_6 + i_7$$

Partie d

Premièrement, les fonctions doivent être écrites en forme canonique.

$$f_1(a, b, c) = a \cdot b \cdot c + a \cdot b \cdot \overline{c} + \overline{a} \cdot \overline{b} \cdot \overline{c}$$

$$f_2(a, b, c) = \overline{a} \cdot b \cdot c + \overline{a} \cdot b \cdot \overline{c} + a \cdot \overline{b} \cdot \overline{c} + a \cdot \overline{b} \cdot c$$

Les variables d'entrée $a, b,$ et c sont reliées aux entrées d'addresse du décodeur; les sorties du décodeur correspondantes aux mintermes des fonctions données sont reliées aux entrées des deux portes NON-ET. Le circuit résultant est démontré dans la figure 6. Notez les expressions suivantes pour ce décodeur: $D_0 = \overline{\overline{a} \cdot \overline{b} \cdot \overline{c}}$, $D_1 = \overline{\overline{a} \cdot \overline{b} \cdot c}$, $D_2 = \overline{\overline{a} \cdot b \cdot \overline{c}}$, $D_3 = \overline{\overline{a} \cdot b \cdot c}$, $D_4 = \overline{a \cdot \overline{b} \cdot \overline{c}}$, $D_5 = \overline{a \cdot \overline{b} \cdot c}$, $D_6 = \overline{a \cdot b \cdot \overline{c}}$, et $D_7 = \overline{a \cdot b \cdot c}$.

Partie e

Le décodeur 6-à-64 est démontré dans la figure 7 ci-dessous.

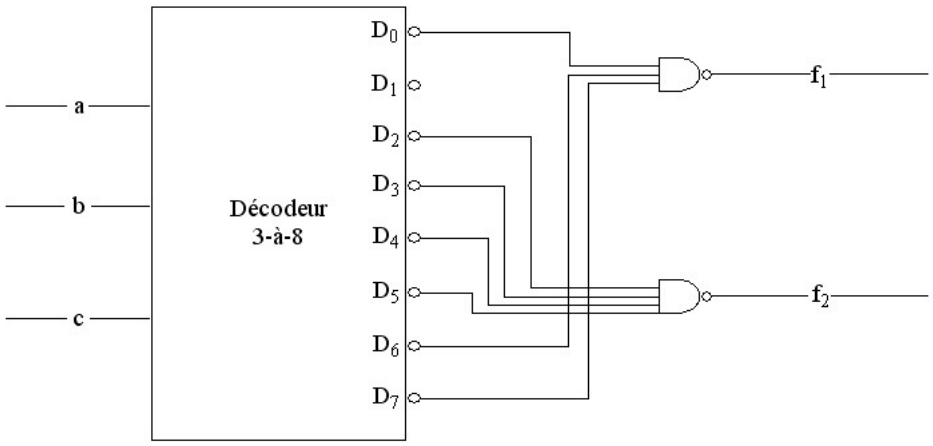

Figure 6: Réalisation de f_1 et f_2

Question IV

Cette question se concerne avec le VHDL au niveau structurel.

Partie a

Les entités *if2a4* et *h3a8* sont codées en VHDL comme ci-dessous:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY if2a4 IS
PORT (w : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
      En : IN STD_LOGIC;
      y : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
END if2a4;

ARCHITECTURE Behavior OF if2a4 IS
BEGIN
    PROCESS (En, w)
    BEGIN
        IF En = '0' THEN
            y <= "0000";
        ELSE
            IF w = "00" THEN
                y <= "0001";
            ELSIF w = "01" THEN
```

```
                y <= "0010";
            ELSIF w = "10" THEN
                y <= "0100";
            ELSE
                y <= "1000";
            END IF;
        END IF;
    END PROCESS;
END Behavior;

LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY h3a8 IS PORT (
       w : IN STD_LOGIC VECTOR(2 DOWNTO 0);
       En : IN STD_LOGIC;
       y : OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END h3a8;

ARCHITECTURE Structure OF h3a8 IS
    SIGNAL EnableTop, EnableBot : STD_LOGIC ;
    COMPONENT if2a4
        PORT (w : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
              En : IN STD_LOGIC;
              y : OUT STD_LOGIC_VECTOR(3 DOWNTO 0));
    END COMPONENT;

BEGIN
    EnableTop <= w(2) AND En;
    EnableBot <= (NOT w(2)) AND En;

    Decoder1: if2a4 PORT MAP (w(1 DOWNTO 0), EnableBot, y(3 DOWNTO 0));
    Decoder2: if2a4 PORT MAP (w(1 DOWNTO 0), EnableTop, y(7 DOWNTO 4));
END Structure;
```

Partie b

L'entité *h6a64* est codée en VHDL comme ci-dessous:

```
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY h6a64 IS PORT (
       w : IN STD_LOGIC_VECTOR(5 DOWNTO 0);
       En : IN STD_LOGIC;
       y : OUT STD_LOGIC_VECTOR(63 DOWNTO 0));
```

```
END h6a64;

ARCHITECTURE Structure OF h6a64 IS
    SIGNAL Enables : STD_LOGIC_VECTOR(7 DOWNTO 0);
    COMPONENT h3a8
        PORT (w : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
              En : IN STD_LOGIC;
              y : OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
    END COMPONENT;

BEGIN
    root: h3a8 PORT MAP (w(5 DOWNTO 3), En, Enables);
    leaf0: h3a8 PORT MAP (w(2 DOWNTO 0), Enables(0), y(7 DOWNTO 0));
    leaf1: h3a8 PORT MAP (w(2 DOWNTO 0), Enables(1), y(15 DOWNTO 8));
    leaf2: h3a8 PORT MAP (w(2 DOWNTO 0), Enables(2), y(23 DOWNTO 16));
    leaf3: h3a8 PORT MAP (w(2 DOWNTO 0), Enables(3), y(31 DOWNTO 24));
    leaf4: h3a8 PORT MAP (w(2 DOWNTO 0), Enables(4), y(39 DOWNTO 32));
    leaf5: h3a8 PORT MAP (w(2 DOWNTO 0), Enables(5), y(47 DOWNTO 40));
    leaf6: h3a8 PORT MAP (w(2 DOWNTO 0), Enables(6), y(55 DOWNTO 48));
    leaf7: h3a8 PORT MAP (w(2 DOWNTO 0), Enables(7), y(63 DOWNTO 56));
END Structure;
```

Partie c

En utilisant un arrangement comme celui la de la figure 6.56 dans votre manuel, le circuit désiré peut être spécifié par la table de vérité suivante:

Gauche	*Droite*	h	y_3	y_2	y_1	y_0	k
0	0	0	w_3	w_2	w_1	w_0	0
0	1	1	w_0	w_3	w_2	w_1	w_0
1	0	w_3	w_2	w_1	w_0	w_3	0

En utilisant des multiplexeurs, la table de vérité peut être réalisé comme dans la figure 8.

Question V

Cette question se concerne avec la méthode d'ASM. Revisez les leçons et le laboratoire #1 pour avoir une idée plus profonde sur cette technique de concevoir des circuit numériques systématiquement.

Un compteur à quatre bits doit être conçu pour compter dans l'order suivant: $0000 \rightarrow 0001 \rightarrow 0011 \rightarrow 0010 \rightarrow 0110 \rightarrow 0111 \rightarrow 0101 \rightarrow 0100 \rightarrow 1100 \rightarrow 1101 \rightarrow 1111 \rightarrow 1110 \rightarrow 0000 \rightarrow ....$ Le compteur commence à l'état 0000 et compte quand le signal de commande $COMMENCE$ est allumé.

Partie a

Il y a beaucoup de différents pseudo-codes qui résoudront ce problème, dont un est présenté ci-dessous. Ce pseudo-code utilise le concept qu'une porte EXCLUSIF-OU peut comparer deux nombres et produire la différence en bit entre les deux.

1. $CGC \leftarrow 0000$; Étape $\leftarrow 0$
2. if (COMMENCE)
   - if (Étape = 3)
     - $CGC \leftarrow CGC$ XOR 0100;
   - else if (Étape = 7)
     - $CGC \leftarrow CGC$ XOR 1000;
   - else if (Étape = 11)
     - Retournez à l'étape #1;
   - else if (Étape = 1) or (Étape = 5) or (Étape = 9)
     - $CGC \leftarrow CGC$ XOR 0010;
   - else
     - $CGC \leftarrow CGC$ XOR 0001;

   else
   - Retournez à l'étape #1;
3. Étape++; Retournez à l'étape #2;

Partie b

Vérifiez la figure 9 pour le diagramme d'ASM, la figure 10 pour le datapath, la figure 11 pour le diagramme détaillé d'ASM, et finalement la figure 12 pour le controlpath. Notez qu'on pouvait résoudre ce problème en utilisant une machine à états finis (comme on va voir dans des leçons suivantes), et qu'on pouvait utiliser les bascules d'états comme la sortie de notre module, CGC, mais on a pris la sortie du*compteur de Gray Code (CGC)* comme la sortie d'un ensemble de registres à 1-bit.

Question de Boni

Un débordement se produit quand deux nombres du complément de 2 sont ajoutés, si le report d'entrée dans le bit de signe est différent du report de sortie du bit de signe. Le bit de signe, dans ce cas-ci, est le bit le plus significatif (MSB). Par conséquent, l'exclusif-ou du report d'entrée et le report de sortie du bit le plus significatif (MSB), produit le bit de débordement. Référez-vous à la table 1 pour une explication plus claire des caisses impliquées.

Signe A	*Signe B*	*Report d'entrée*	*Report de sortie*	*Signe du résultat*	*Signe correcte du résultat*	*Débordement ?*	*Report d'entrée XOR Report de sortie*	*Notes*
0	0	0	0	0	0	Non	0	
0	0	1	0	1	0	Oui	1	Reports différent
0	1	0	0	1	1	Non	0	$\|A\| < \|B\|$
0	1	1	1	0	0	Non	0	$\|A\| > \|B\|$
1	0	0	0	1	1	Non	0	$\|A\| > \|B\|$
1	0	1	1	0	0	Non	0	$\|A\| < \|B\|$
1	1	0	1	0	1	Oui	1	Reports différent
1	1	1	1	1	1	Non	0	

Table 1: Table expliquante la détection de débordement

Remerciements

Les réponses et figures liées aux questions du manuel sont tirées du manuel d'accompagnement de l'instructeur pour *Fundamentals of Digital Logic with VHDL Design* par Stephen Brown et Zvonko Vranesic.

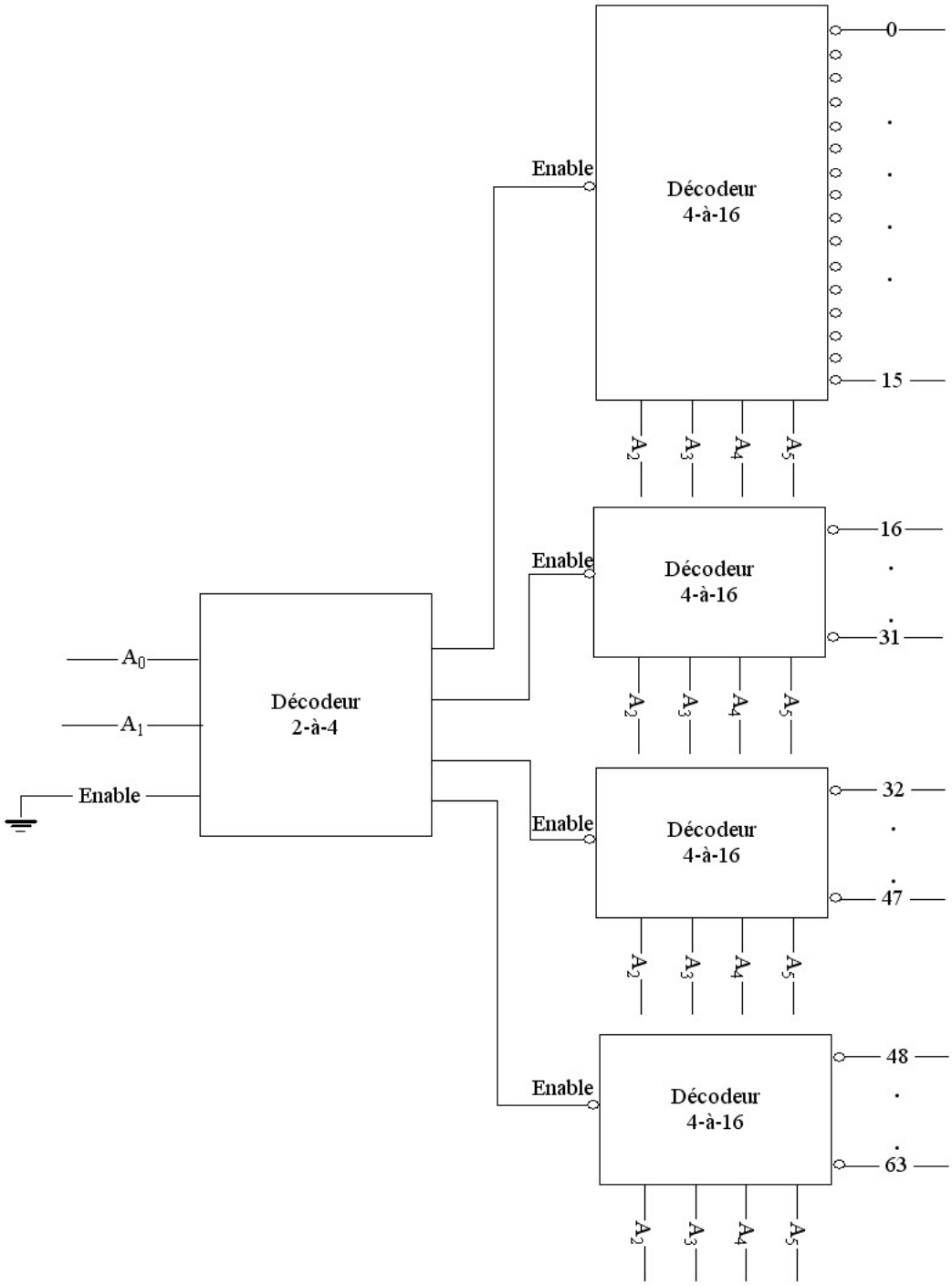

Figure 7: Réalisation d'un décodeur 6-à-64

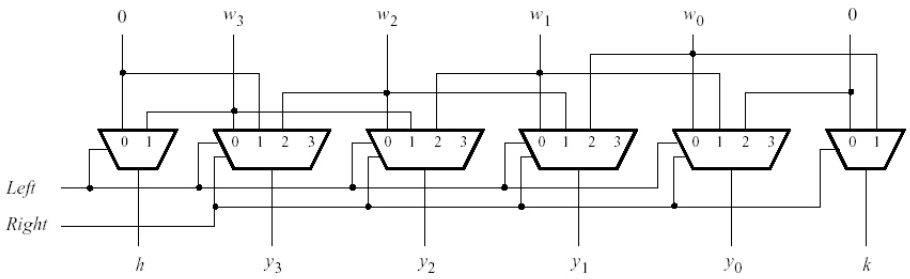

Figure 8: Réalisation d'un circuit de décalage

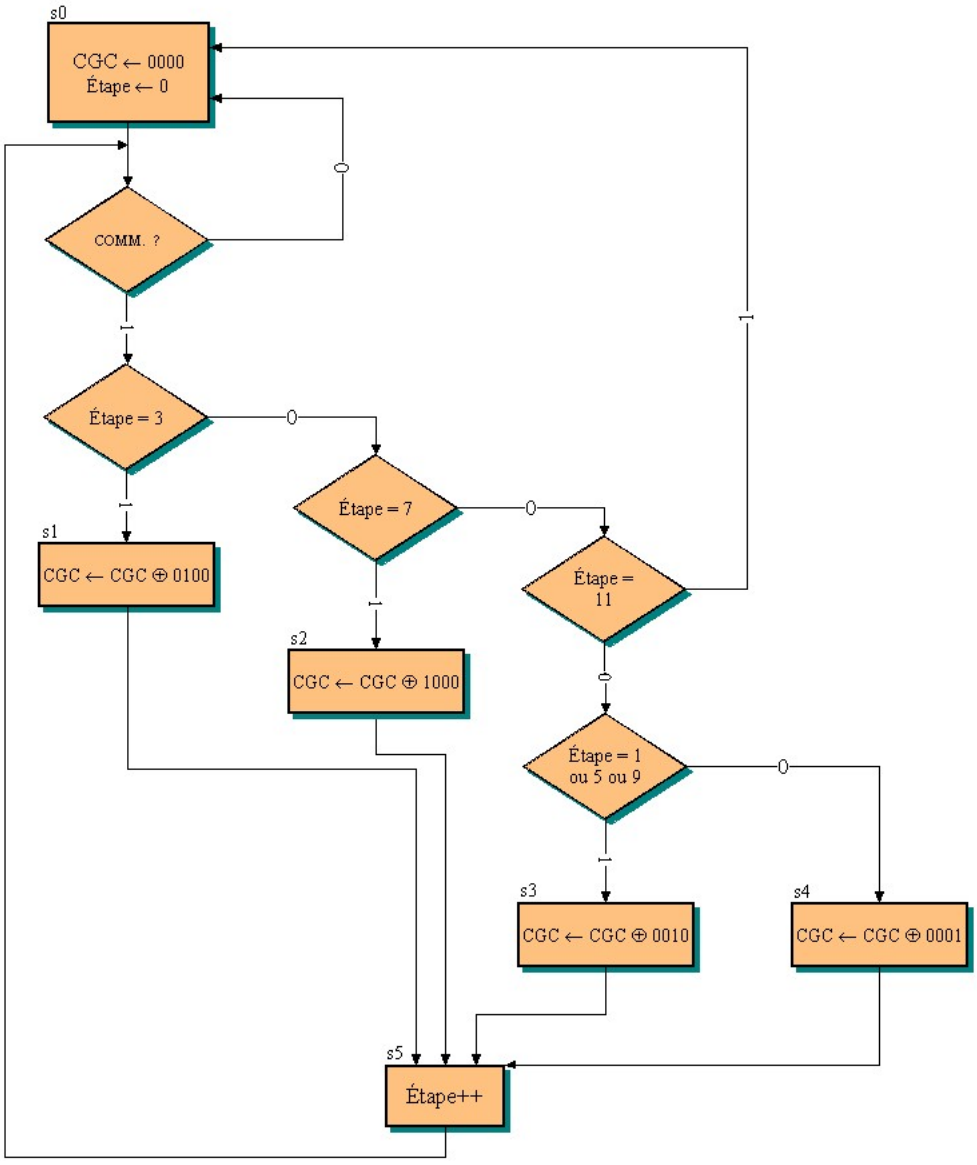

Figure 9: Diagramme d'ASM

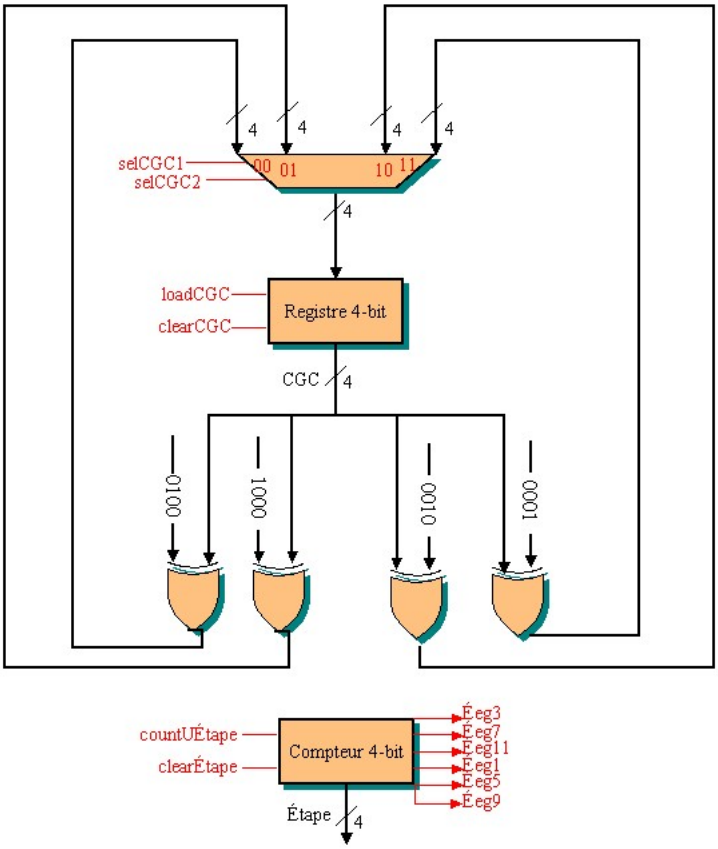

Figure 10: Datapath d'ASM

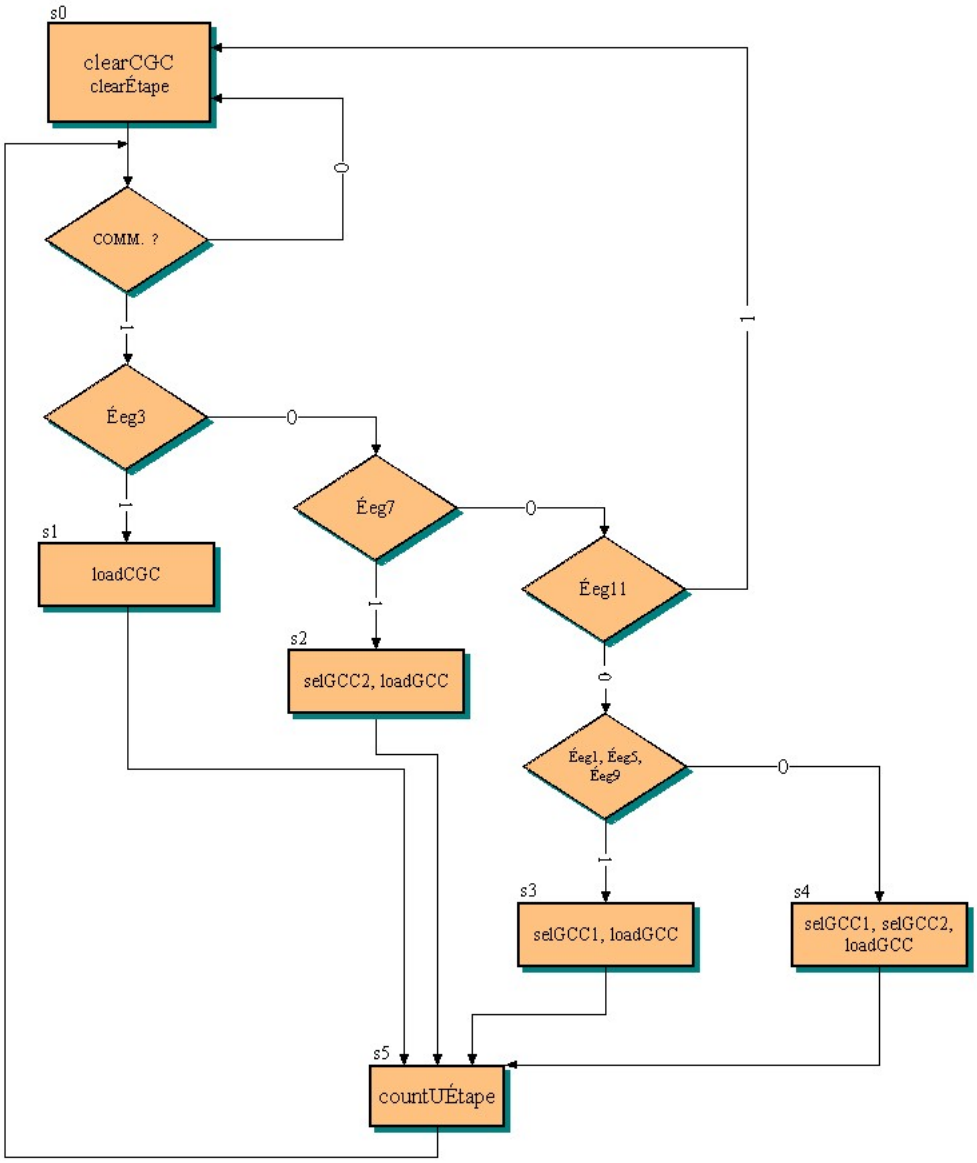

Figure 11: Diagramme détaillé d'ASM

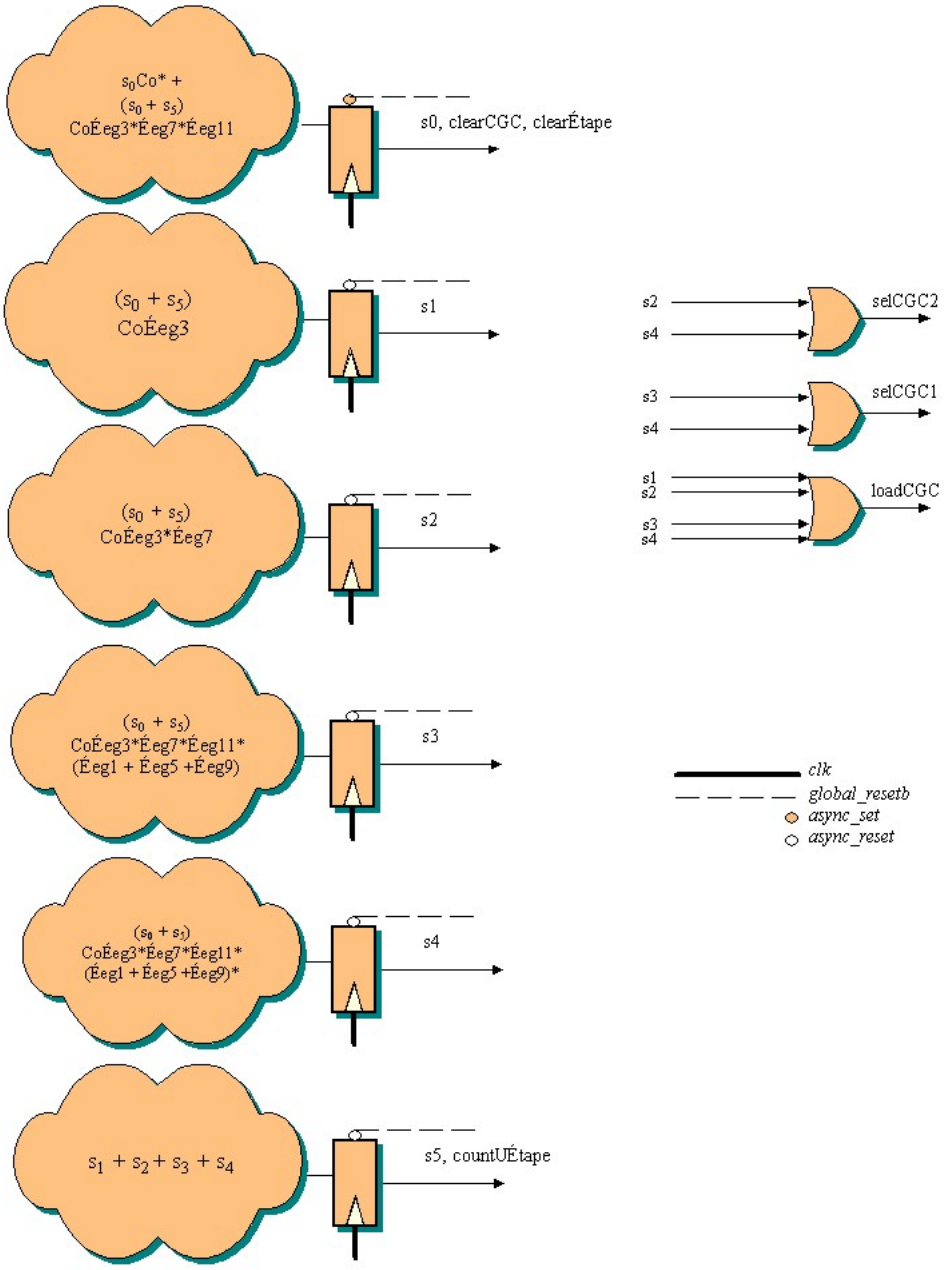

Figure 12: Controlpath d'ASM