Chapitre 4 : La chaîne d'information des systèmes

4.4 : Logique combinatoire

TD – 6 h

Travail demandé

1. a) Pour les expressions booléennes $Q_1 = a + a.b$ et $Q_2 = a + \bar{a}.b$ établir le ***logigramme*** puis le ***schéma électrique équivalent*** avec des contacts a et b reliés à la phase (L) en entrée et une lampe Q reliée au neutre (N) en sortie. En déduire une ***expression plus simple*** respectivement pour Q_1 et Q_2.

   b) ***En extension*** (i.e. par une table de vérité) puis ***en intention*** (i.e. par calcul formel algébrique), ***démontrer*** les propriétés remarquables d'absorption $a + a.b = a$ et $a + \bar{a}.b = a + b$ (cf. cours p. 2).

2. a) ***Simplifier*** les expressions booléennes suivantes : $Q_1 = (a + b).(a + \bar{b})$ $\quad Q_2 = \bar{a}.b.\bar{c} + \bar{a}.b.c + a.b.\bar{c} + a.b.c$

   b) ***Démontrer*** algébriquement les propriétés remarquables suivantes :

   i. $\overline{a \oplus b} = \bar{a} \oplus b = a \oplus \bar{b} = a \odot b$ ii. $a \oplus b = \bar{a} \oplus \bar{b}$ iii. $a . (b \oplus c) = (a . b) \oplus (a . c)$

   iv. $a \oplus (b . c) \neq (a \oplus b) . (a \oplus c)$ v. $(a \oplus b) \oplus c = a \oplus (b \oplus c)$

   Conclure quant aux ***propriétés de l'opérateur*** $\oplus$.

   vi. $a \uparrow \overline{(b \uparrow c)} = \overline{(a \uparrow b)} \uparrow c$ vii. $a \uparrow (b \uparrow c) \neq (a \uparrow b) \uparrow c$

   viii. $a \uparrow (b . c) \neq (a \uparrow b) . (a \uparrow c)$ ix. $a \uparrow (b + c) \neq (a \uparrow b) + (a \uparrow c)$

   Conclure quant aux ***propriétés de l'opérateur*** $\uparrow$.

3. Soit la ***fonction logique Q*** définie par la table de vérité ci-contre.

a	b	c	Q
0	0	0	**0**
0	0	1	**0**
0	1	0	**0**
0	1	1	**1**
1	0	0	**0**
1	0	1	**1**
1	1	0	**1**
1	1	1	**1**

   a) À partir de ses valeurs **1**, déterminer l'expression de Q comme une ***somme de produits*** d'atomes (ou de compléments d'atome) a, b, c ; la simplifier autant que possible.

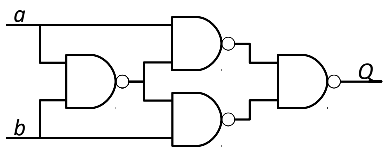
   b) À partir des valeurs **0** de son complément, déterminer l'expression de Q comme un ***produit de sommes*** d'atomes (ou de compléments d'atome) a, b, c ; la simplifier pour retrouver celle obtenue à la question a).

   c) De quelle fonction remarquable s'agit-il ? Tracer son ***logigramme***.

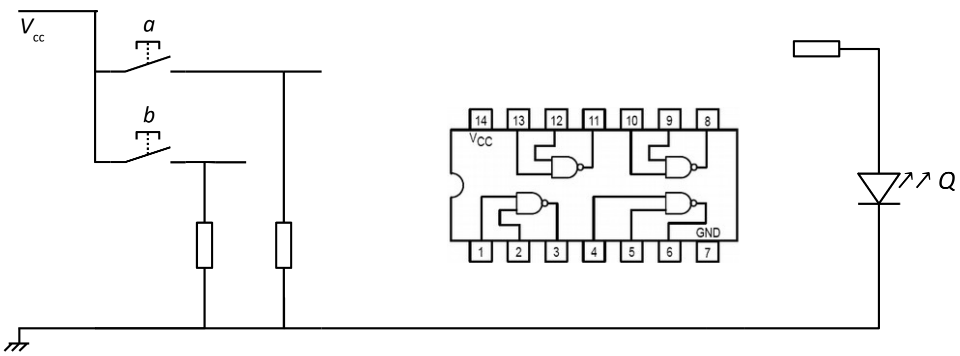
   d) Écrire l'***expression*** de Q avec la syntaxe du ***langage C***.

4. a) En employant uniquement des portes ***nand***, composer un ***logigramme*** pour recréer chacune des portes logiques ***non***, ***et***, ***ou*** ; écrire l'***expression booléenne*** de chaque logigramme et la simplifier.

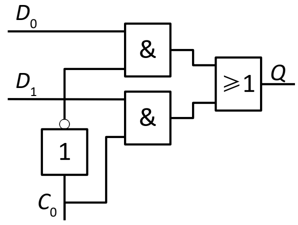
   b) Sur le logigramme ci-contre, en partant des entrées a et b, indiquer l'***expression booléenne simplifiée*** (avec ***et***, ***ou***, ***non***) de chaque branche jusqu'à Q. Quelle est la ***porte logique*** implémentée par cette fonction ?



   c) Sur le schéma du ***circuit imprimé double face*** ci-dessous (avec un circuit *74HC00*, deux boutons-poussoirs a et b, une led Q, et leurs résistances associées), tracer un ***câblage*** de la fonction définie par le logigramme ci-dessus, en employant une ***couleur distincte par face*** pour les pistes, sans croisements !



5. Un ***multiplexeur « n vers 1 »*** est un circuit qui canalise en sortie sur **1** seule voie (***output Q***) des données binaires en entrée sur ***n*** voies (***data*** $D_0, D_1 \dots D_{n-1}$) via une fonction de sélection du ***numéro de voie***, un entier naturel ***C*** codé sur ***k*** bits (***channel*** $C_0, C_1 \dots C_{k-1}$). Ce type de circuit est notamment employé pour réduire le nombre de fils de câblage entre un microcontrôleur et des afficheurs 7 segments ou des matrices de points. On donne ci-dessous le logigramme d'un multiplexeur ***« 2 vers 1 »*** (2 voies sélectionnées par un bit C_0).

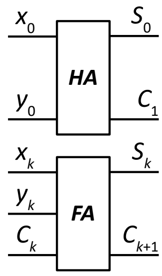


a) Établir la ***table de vérité*** de ***Q*** pour toutes les combinaisons des entrées C_0, D_1, D_0 codées <u>dans cet ordre</u> en binaire pur, i.e. avec C_0 comme MSB et D_0 comme LSB.

b) Déterminer l'***expression booléenne*** de ***Q*** en fonction de C_0, D_1 et D_0. Montrer algébriquement et vérifier avec la table que $Q = D_k$ si $C_0 = k$ pour $k = 0$ et 1.

c) Sur le même principe, en utilisant des portes multi-entrées, concevoir le ***logigramme*** d'un ***multiplexeur « 4 vers 1 »*** (4 bits de donnée D_0 à D_3, 2 bits de voie C_0 et C_1 codant $2^2 = 4$ numéros de voies 0, 1, 2, 3).

6. Un ***additionneur binaire naturel*** à ***n bits*** est un composant qui additionne deux ***entiers naturels x*** et ***y*** codés en binaire sur *n* bits ; il est intégré dans les unités arithmétiques et logiques des microprocesseurs. Il se compose de ***n additionneurs monobits***, dont les entrées-sorties de retenues sont câblées en série :



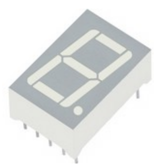
- le 1er est un circuit à deux entrées x_0 et y_0 (bits de rang 0 de ***x*** et ***y***) et deux sortie binaires S_0 (en anglais ***sum***) et C_1 (retenue pour le calcul de S_1, en anglais ***carry-out***), tels que, <u>au sens arithmétique</u>, $S_0 = (x_0 + y_0) \bmod 2$[1] et $C_1 = (x_0 + y_0) \div 2$ (division entière); ce circuit est appelé ***demi-additionneur*** (en anglais ***half adder***, abrégé ***HA***) car sans retenue en entrée ;
- les suivants sont des ***additionneurs complets*** (en anglais ***full adder***, abrégé ***FA***) à trois entrées x_k, y_k, C_k et deux sorties $S_k = (x_k + y_k + C_k) \bmod 2$ et $C_{k+1} = (x_k + y_k + C_k) \div 2$.

a) Établir la ***table de vérité***, le ***logigramme*** et l'***expression logique*** de S_0 et C_1 du circuit ***HA***.

b) Procéder de même pour le circuit ***FA***. Montrer que $S_k = C_k \oplus (x_k \oplus y_k)$ et $C_{k+1} = \mathrm{maj}(x_k, y_k, C_k)$.

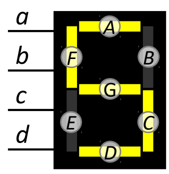
7. Un ***afficheur 7 segments*** – cf. fig. 1 – est un composant permettant d'afficher (en général, avec une led) ***un chiffre*** ou éventuellement quelques lettres ou symboles ; chaque ***segment*** étant ***activé ou non*** selon le symbole à afficher (e.g. A).

Pour un affichage limité aux chiffres, on emploie un ***circuit décodeur*** – cf. fig. 2 – qui pilote l'activation des segments (ici repérés de ***A*** à ***G*** – cf. fig. 3) en fonction du chiffre à afficher, qui est codé sur ***4 bits***, ici repérés ***a*** (LSB) à ***d*** (MSB). On étudie ici les fonctions logiques d'un tel décodeur.



1

2



3

a) Compléter les ***lignes 2 à 5*** de la table de vérité ci-contre donnant la valeur **0** ou **1** des segments ***A*** à ***G*** en fonction des valeurs des bits ***a*** à ***d*** du chiffre codé en entrée.

b) En déduire l'***expression booléenne*** des segments ***A*** et ***E*** en fonction de ***a***, ***b***, ***c*** et ***d***. Puis :

- simplifier l'expression de ***A*** pour minimiser le nombre d'opérateurs employés, sans limite de type d'opérateur ni de nombre d'entrées par opérateur (e.g. on peut utiliser un opérateur ***et*** à 3 entrées) ; on rappelle que $\overline{a \oplus b} = a \odot b$;
- simplifier l'expression de ***E*** pour n'utiliser que des fonctions ***non*** et ***nor***.

c) Représenter les ***logigrammes*** des expressions simplifiées des segments ***A*** et ***E***.

chiffres					segments						
	d	*c*	*b*	*a*	*G*	*F*	*E*	*D*	*C*	*B*	*A*
0	0	0	0	0	0	1	1	1	1	1	1
1	0	0	0	1	0	0	0	0	1	1	0
2											
3											
4											
5											
6	0	1	1	0	1	1	1	1	1	0	1
7	0	1	1	1	0	0	0	0	1	1	1
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	0	1	1	1	1

1. I.e. le signe + symbolise ici une addition cyclique sur 1 bit et non pas l'opérateur *ou* ; en termes logiques, on peut remplacer ce + par ⊕.