

COMPTE RENDU SUR :

Les TP
d'architecture des
Ordinateurs.

G-I

Effectué par :

. GHANOUCHE IsSaM

. SAFRAOUI Majdouline

Encadré par : M. FIHRI

20/05/2014

Sommaire

A-Introduction général : -----	0
B-Feux tricolores : -----	3
1. Feux tricolores scandinaves ou du Royaume-Uni :-----	3
2-Feux tricolores français : -----	4
3-Feux tricolores : (jour et nuit) -----	6
C- Additionneur complet (4 bits) :-----	8
D-Distributeur de café : -----	11
E-Distribution de Boissons : -----	14
F-Conclusion : -----	18

***A-**Introduction général :*

Dans ce rapport, nous avons tenté de présenter de manière simple mais rigoureuse les principaux circuits Logiques qu'on a réalisé dans les séances des TP sur Electronics Workbench.

La présentation de quelques montages simples est une illustration des fonctions.

Notons qu'il existe deux types de circuits logiques : les circuits combinatoires et les circuits séquentiels.

Les circuits logiques exécutent des opérations sur des variables logiques, transportent et traitent des signaux logiques. Les circuits combinatoires qui sont des circuits idéalisés où le temps de propagation des signaux n'est pas pris en considération. Les signaux de sortie ne dépendent que des signaux d'entrée, appliqués à l'instant considéré. Les circuits séquentiels qui sont des circuits où il faut tenir compte du temps de propagation des signaux et de la mémoire du circuit. Les signaux de sortie dépendent des signaux d'entrée appliqués antérieurement.

B- Feux tricolores :

1. Feux tricolores scandinaves ou du Royaume-Uni :

→ Cycle : vert, orange, rouge, orange, vert, ...

- Contrairement aux feux type français, un état ne correspond pas à une couleur allumée.

C'est un compteur à 4 états (0 à 3), donc il faut 2 bascules pour le circuit

Graphe de transition :

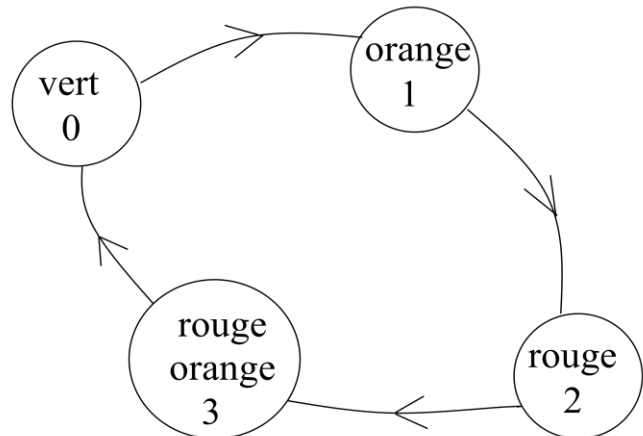
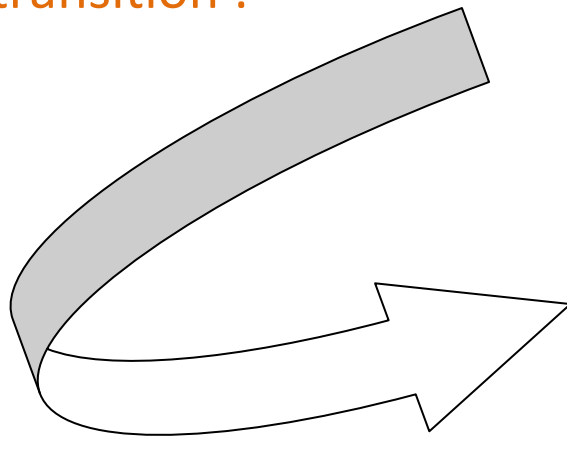
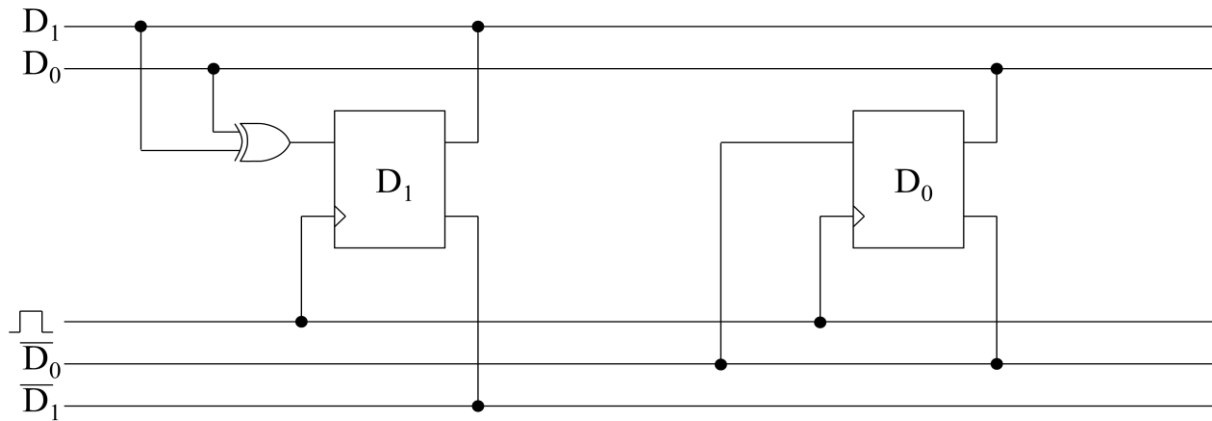


Table de transition :

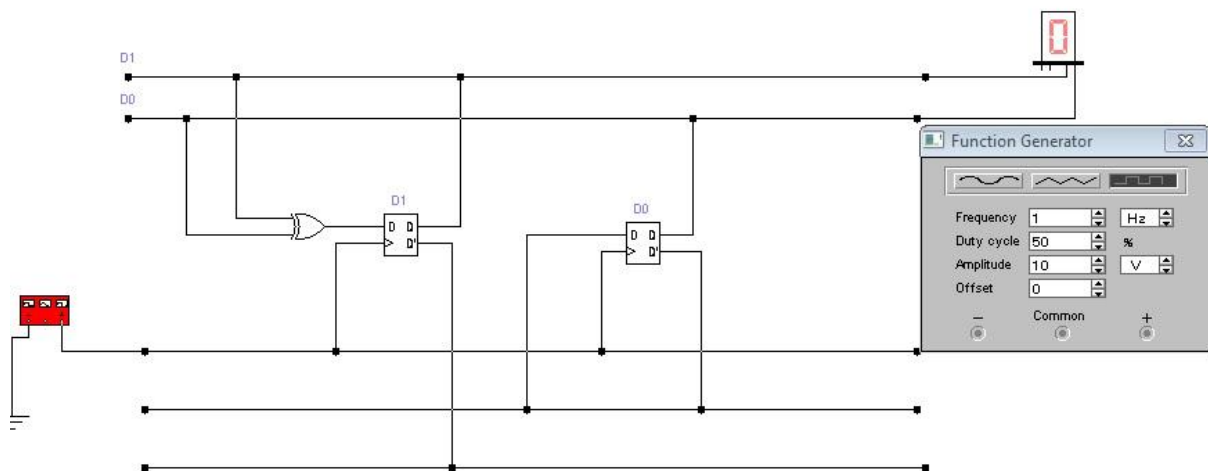


t		t+1	
D ₁	D ₀	D ₁	D ₀
0	0	0	1
0	1	1	0
1	0	1	1
1	1	0	0

$$\begin{cases} D_1^{t+1} = D_1 \oplus D_0 \\ D_0^{t+1} = \overline{D_0} \end{cases}$$



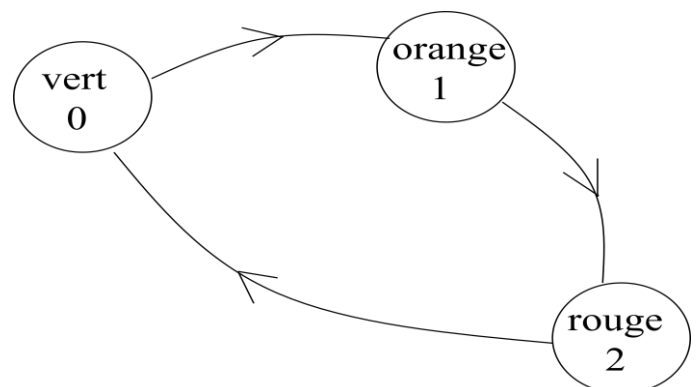
Le circuit sur Electronics Workbench :



2-Feux tricolores français :

Un feu tricolore est un compteur.

**Le graphe de transitions
est le suivant :**



Le graphe a 3 états numérotés de 0 à 2.

Le nombre 2 se code 10 en binaire (2 bits) donc il y aura 2 bascules dans ce circuit.

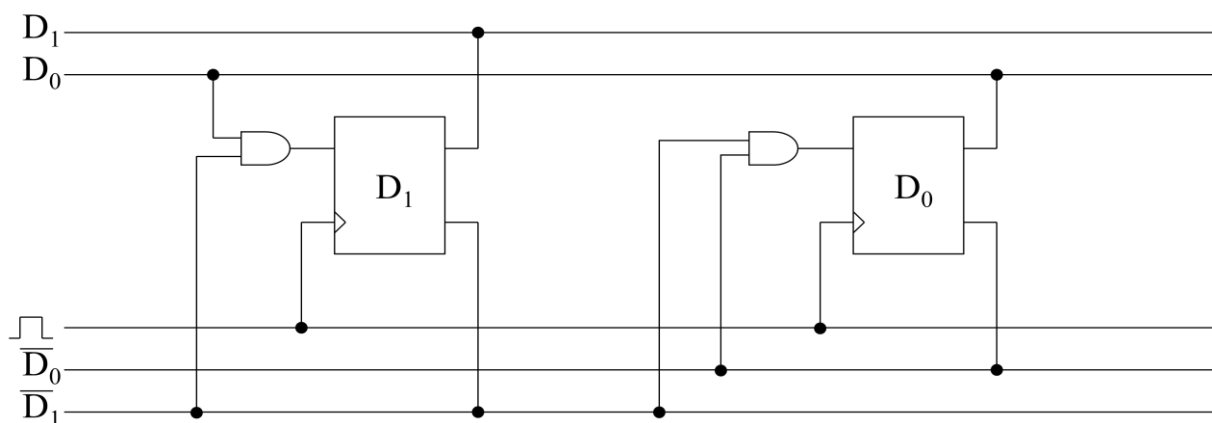
Table de transition :

t		t+1	
D ₁	D ₀	D ₁	D ₀
0	0	0	1
0	1	1	0
1	0	0	0
1	1	0	0

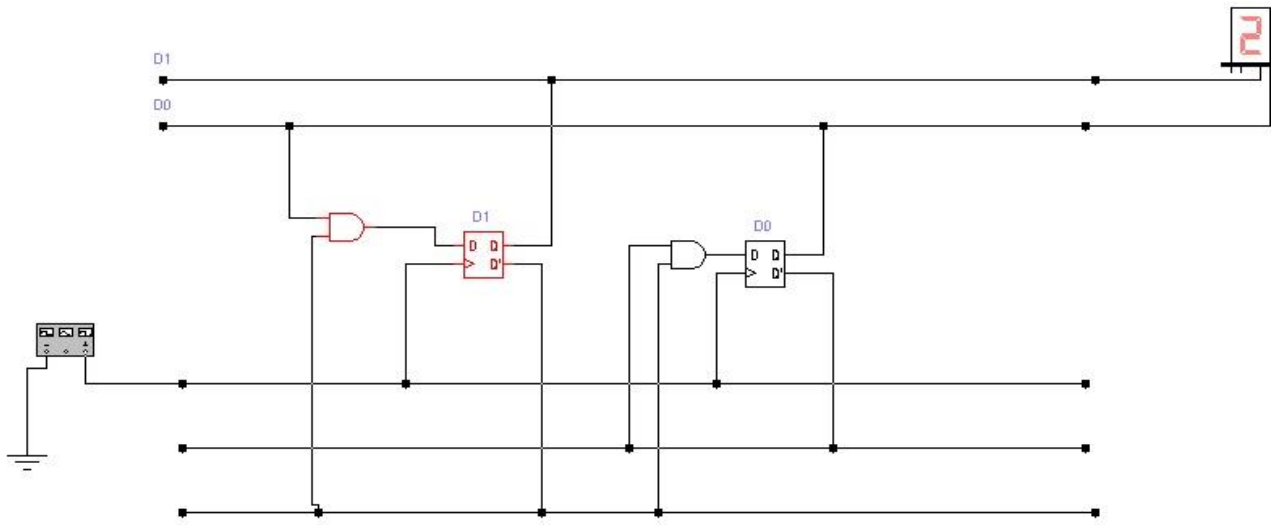


$$\begin{cases} D_1^{t+1} = \overline{D_1}.D_0 \\ D_0^{t+1} = \overline{D_1}.\overline{D_0} \end{cases}$$

À partir des équations issues de la table de transitions, on dessine le circuit :



Le circuit sur Electronics Workbench :



3-Feux tricolores : (jour et nuit)

Un feu tricolore peut avoir 2 types de fonctionnement différents :

de jour : vert, orange, rouge, vert, ... ;

de nuit : orange, éteint, orange, éteint, ... ;

- 2 cycles différents donc 1 bit pour les différencier.

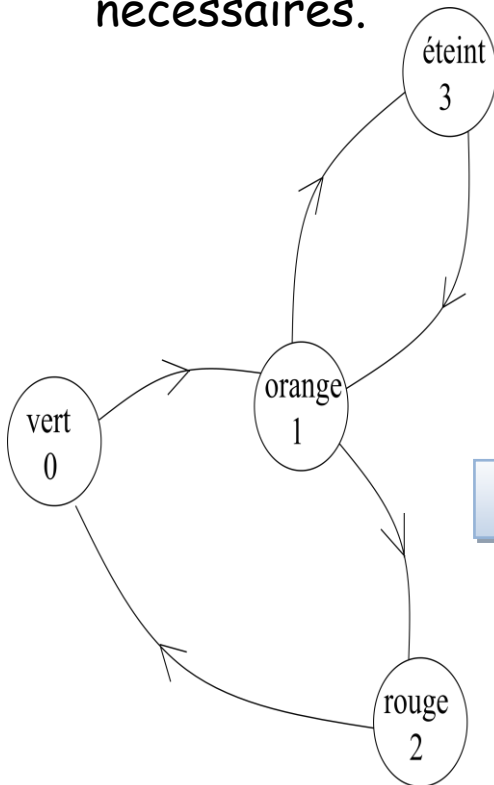
On le note c et on choisit $c=0$ de jour et $c=1$ de nuit.

On a 4 états possibles :

- état 0 : vert allumé (le reste éteint) ;
- état 1 : orange allumé (le reste éteint) ;
- état 2 : rouge allumé (le reste éteint) ;

- état 3 : tout éteint.

=> 4 états donc 2 bascules nécessaires.



c	t		t+1	
	D ₁	D ₀	D ₁	D ₀
0	0	0	0	1
0	0	1	1	0
0	1	0	0	0
0	1	1	0	0
1	0	0	0	1
1	0	1	1	1
1	1	0	0	1
1	1	1	0	1

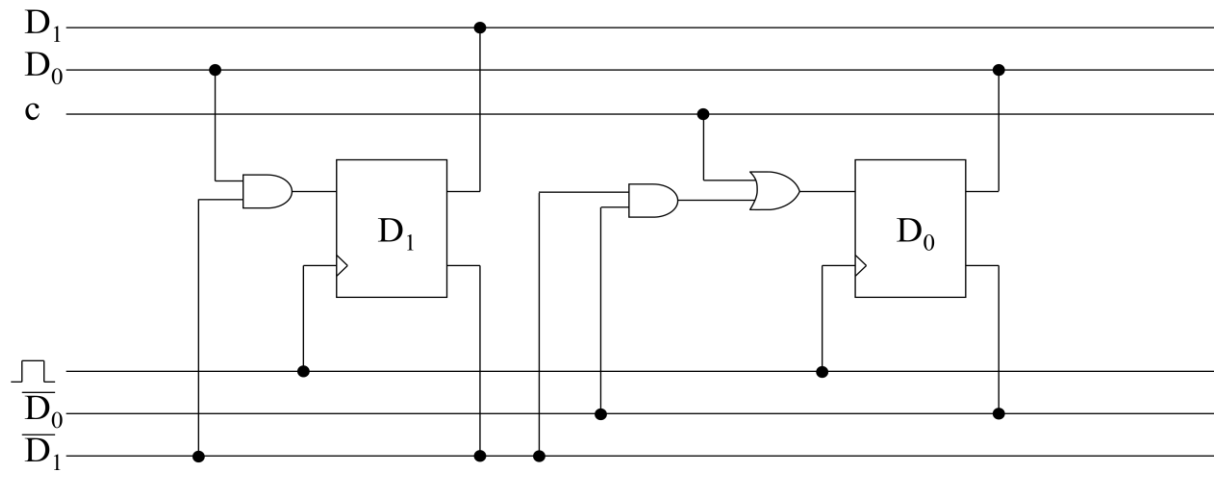
c\D ₁ D ₀	00	01	11	10
0	0	1	0	0
1	0	1	0	0

= Les Tableaux de Karnaugh :

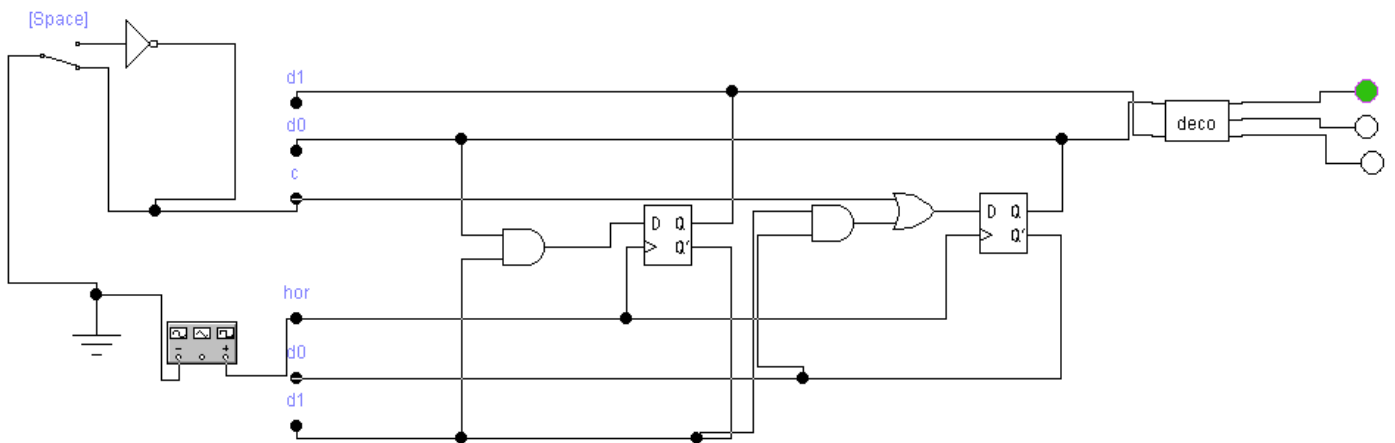
c\D ₁ D ₀	00	01	11	10
0	1	0	0	0
1	1	1	1	1

$$\begin{cases} D_1^{t+1} = \overline{D_1} \cdot D_0 \\ D_0^{t+1} = \overline{D_1} \cdot \overline{D_0} + c \end{cases}$$

RAPPORT (Architecture des Ordinateurs)



Le circuit sur Electronics Workbench :



C- Additionneur complet (4 bits) :

- **idée** : combinaison de plusieurs circuits travaillant sur 1 bit
- **avantages** : plus facile à réaliser, travail avec une taille de n bits (n quelconque), conception plus claire, meilleur méthodologie, schéma logique plus clair, réalisation matérielle du circuit plus simple.
- **Circuit primitif** : additionneur 1 bit complet

Prise en compte de la retenue précédente !

RAPPORT (Architecture des Ordinateurs)

Table de vérité :

a	b	c	r	s
0	0	0	0	0
0	0	1	0	1
0	1	0	0	1
0	1	1	1	0
1	0	0	0	1
1	0	1	1	0
1	1	0	1	0
1	1	1	1	1

Tableau de Karnaugh pour r :

a\bc	00	01	11	10
0	0	0	1	0
1	0	1	1	1

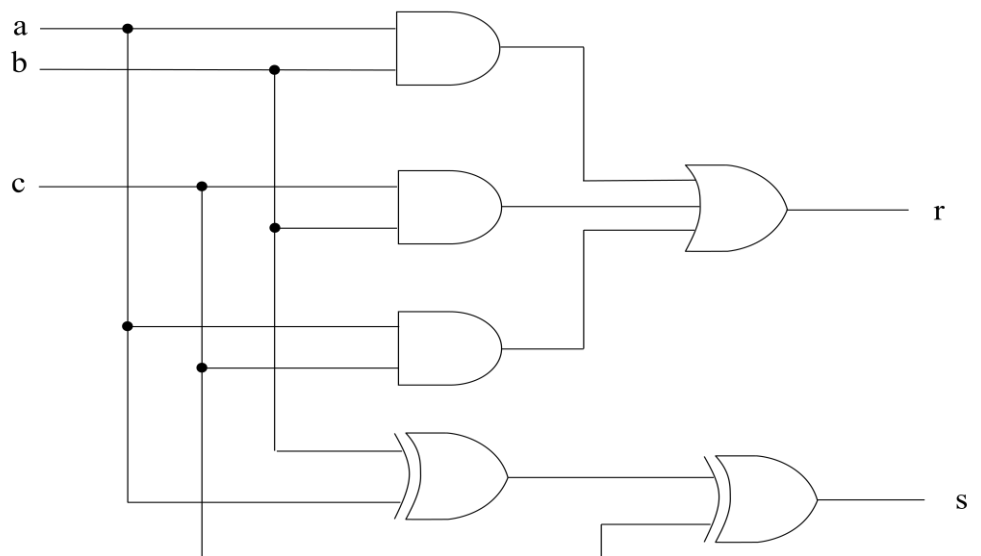
Tableau de Karnaugh pour s :

a\bc	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$r = ac + bc + ab$$

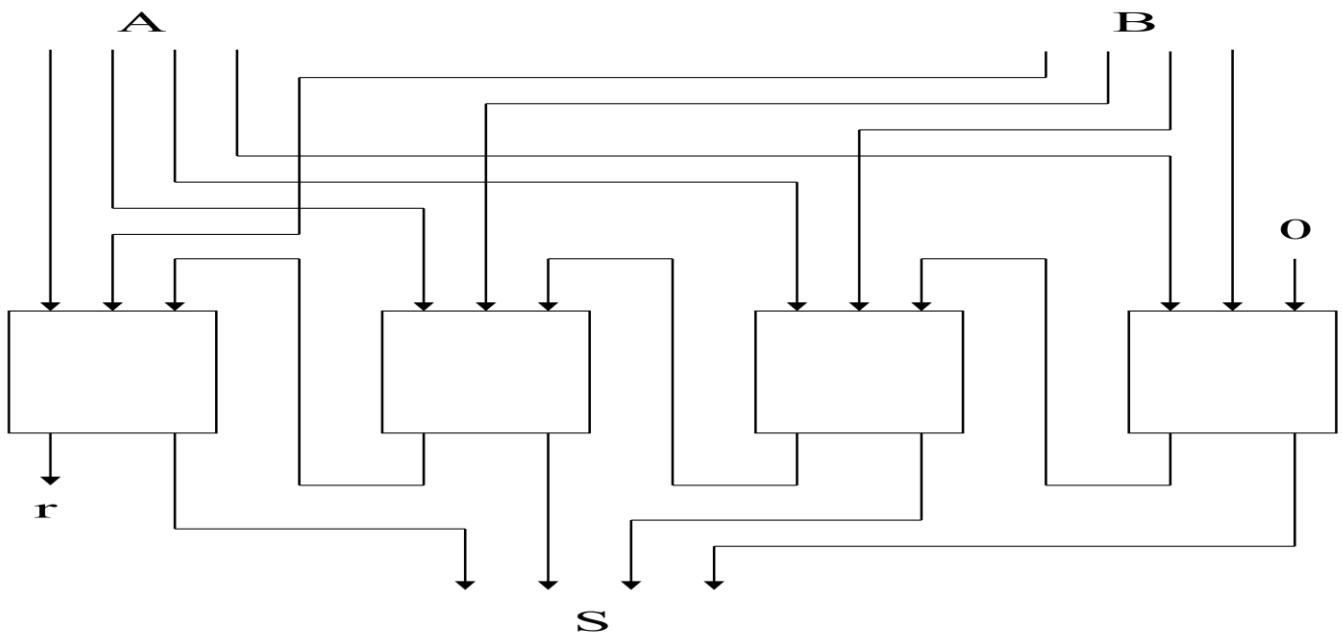
$$s = a \oplus b \oplus c$$

Schéma
logique :

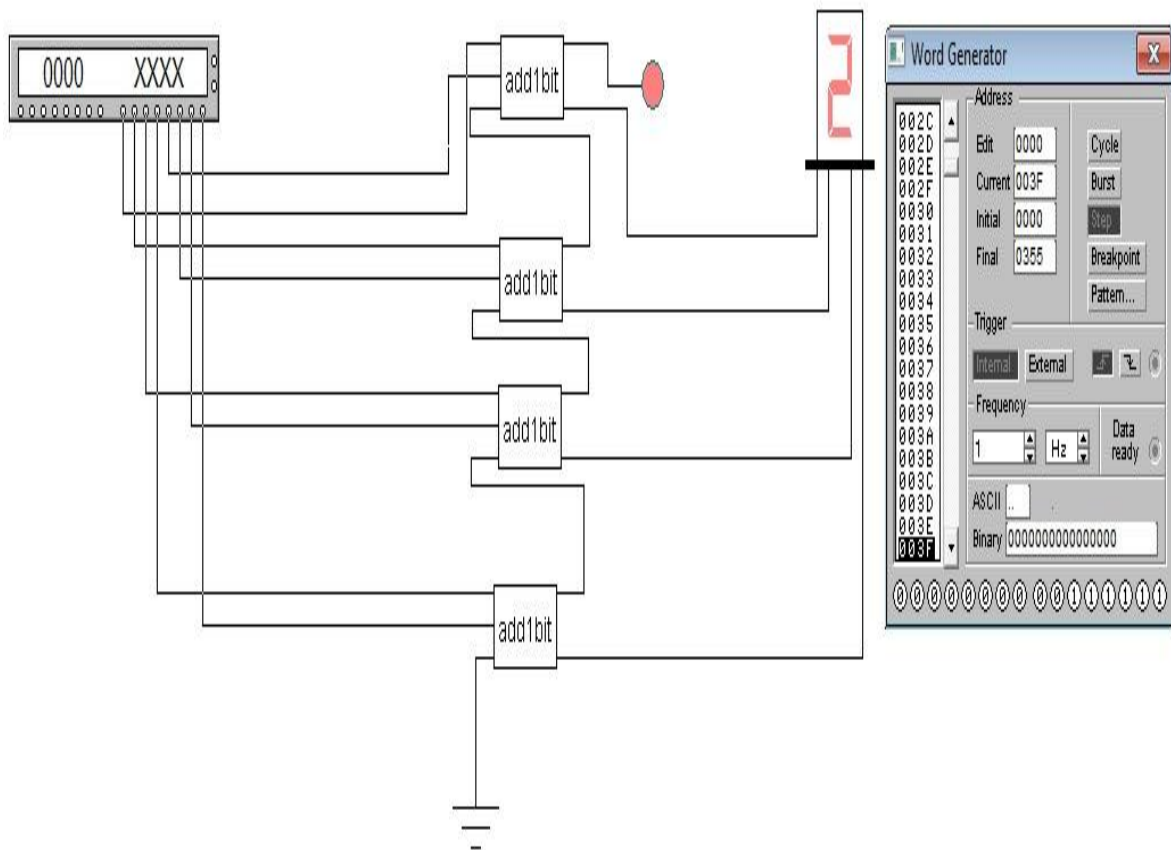


RAPPORT (Architecture des Ordinateurs)

Schéma logique d'un additionneur 4 bits constitué d'additionneurs complets :



Le circuit sur Electronics Workbench :



D-Distributeur de café :

On veut réaliser un automate contrôlant un distributeur de café.

On suppose qu'il distribue du café, avec ou sans sucre, avec ou sans lait.

Il y a donc 4 fonctionnements possibles :

1. fournir du café noir ;
2. fournir du café sucré ;
3. fournir du café au lait ;
4. fournir du café au lait sucré.

Ces 4 cas correspondent à 4 cycles différents du distributeur.

Les cycles sont numérotés de 0 à 3 (2 bits de codage) :

00 : gobelet, café lyophilisé, eau ;

01 : gobelet, café lyophilisé, sucre, mélangeur, eau ;

10 : gobelet, café lyophilisé, lait lyophilisé, eau ;

11 : gobelet, café lyophilisé, sucre, mélangeur, lait lyophilisé, eau.

On définit les états (certaines actions peuvent être regroupées) :

état 0 : repos (attente d'une commande de boisson)

état 1 : commande de boisson déclenchée

un gobelet

du café lyophilisé

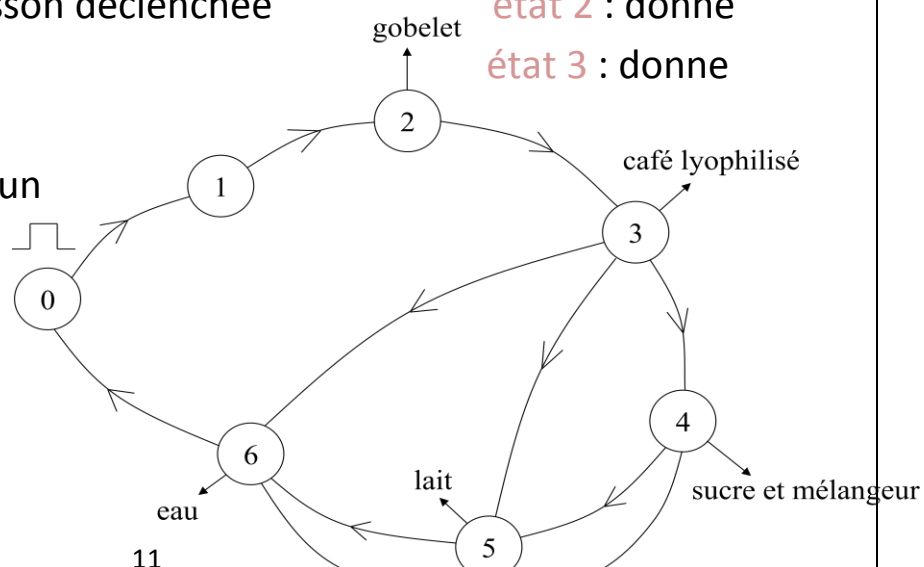
état 4 : donne du sucre et un
mélangeur

état 5 : donne du lait
lyophilisé

état 6 : donne de l'eau

état 2 : donne

état 3 : donne



Graphe de transitions :

RAPPORT (Architecture des Ordinateurs)

Il y a 7 états (0 à 6) donc 3 bascules seront nécessaires.

Table de Transition :

c ₁ c ₀	t D ₂ D ₁ D ₀	t+1 D ₂ D ₁ D ₀	c ₁ c ₀	t D ₂ D ₁ D ₀	t+1 D ₂ D ₁ D ₀
00	000	001	10	000	001
00	001	010	10	001	010
00	010	011	10	010	011
00	011	110	10	011	101
00	100	xxx	10	100	xxx
00	101	xxx	10	101	110
00	110	000	10	110	000
00	111	xxx	10	111	xxx
01	000	001	11	000	001
01	001	010	11	001	010
01	010	011	11	010	011
01	011	100	11	011	100
01	100	110	11	100	101
01	101	xxx	11	101	110
01	110	000	11	110	000
01	111	xxx	11	111	xxx

- On remplace les X par 0 qui permet de **revenir à l'état 0 lorsqu'on démarre dans un état non géré par l'automate.** Même qu'il est possible de démarrer dans n'importe quel état valide de l'automate !

- Les Tableaux de Karnaugh :

Pour D₂ :

c ₁ c ₀ \ D ₂ D ₁ D ₀	000	001	011	010	110	111	101	100
00	0	0	1	0	0	0	0	0
01	0	0	1	0	0	0	0	1
11	0	0	1	0	0	0	1	1
10	0	0	1	0	0	0	1	0

Pour D₁ :

c ₁ c ₀ \ D ₂ D ₁ D ₀	000	001	011	010	110	111	101	100
00	0	1	1	1	0	0	0	0
01	0	1	0	1	0	0	0	1
11	0	1	0	1	0	0	1	0
10	0	1	0	1	0	0	1	0

RAPPORT (Architecture des Ordinateurs)

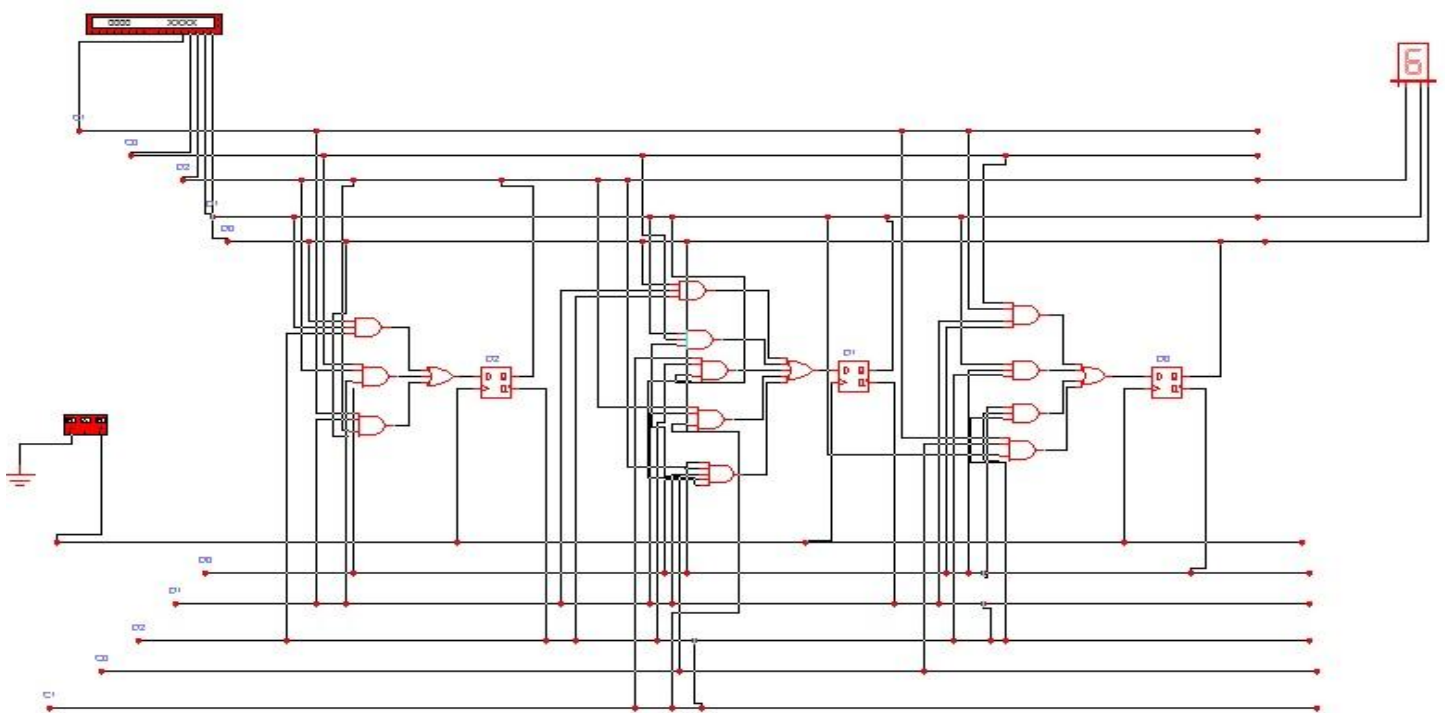
Pour Do :

$c_1c_0 \backslash D_2D_1D_0$	000	001	011	010	110	111	101	100
00	1	0	0	1	0	0	0	0
01	1	0	0	1	0	0	0	0
11	1	0	0	1	0	0	0	1
10	1	0	1	1	0	0	0	0

Les équations sont :

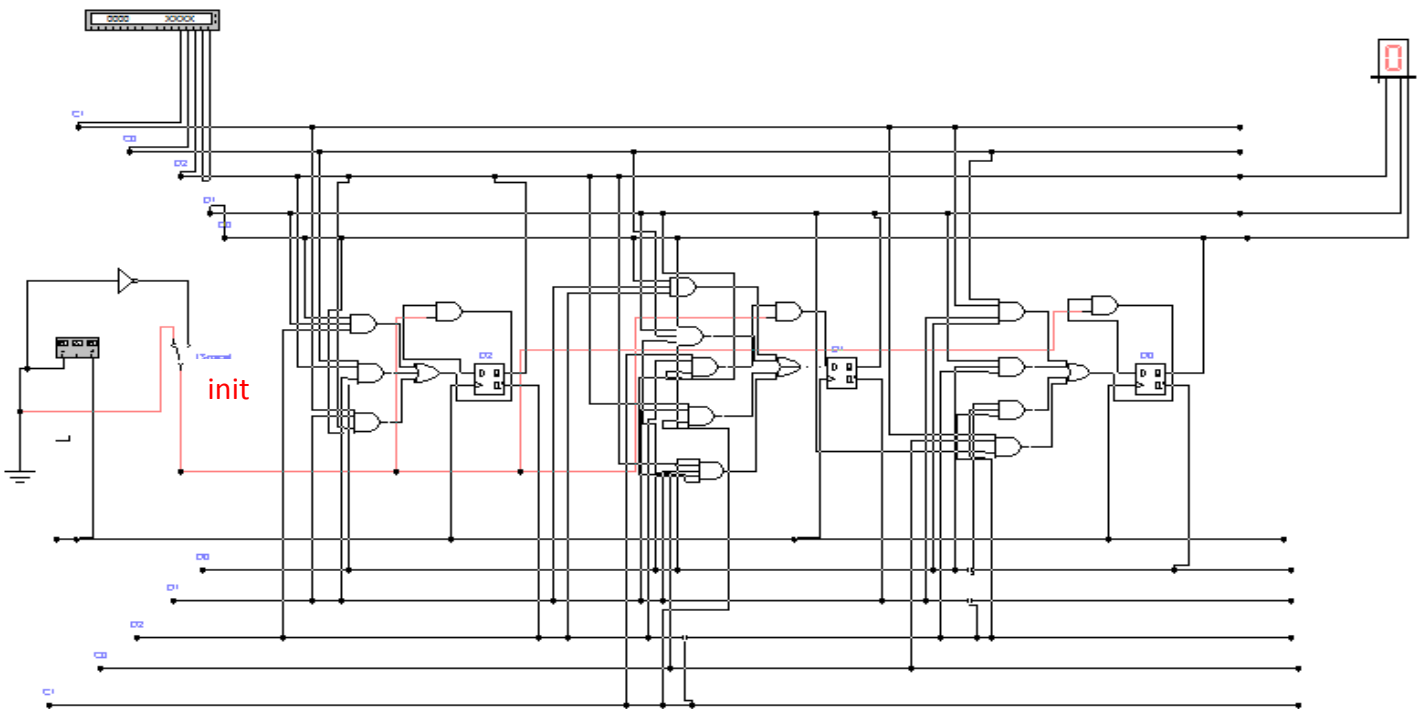
$$\begin{cases} D_2^{t+1} = \overline{D_2} D_1 D_0 + c_1 D_2 \overline{D_1} D_0 \\ D_1^{t+1} = c_1 D_2 \overline{D_1} D_0 + \overline{D_2} D_1 \overline{D_0} + \overline{c_1} D_2 \overline{D_1} + \overline{c_1} \overline{c_0} D_2 D_1 + \overline{D_2} \overline{D_1} D_0 + \overline{c_1} c_0 D_2 \overline{D_1} D_0 \\ D_0^{t+1} = \overline{D_2} D_1 \overline{D_0} + c_0 c_1 \overline{D_1} D_0 + c_1 \overline{c_0} D_2 D_1 + \overline{D_2} D_1 \overline{D_0} \end{cases}$$

Le circuit sur Electronics Workbench :



Il est impératif d'utiliser un signal init parce que :

le signal init permet de **toujours démarrer à l'état 0** (état de repos) .



E-Distribution de Boissons :

On veut réaliser un automate contrôlant un distributeur de boisson.

On suppose qu'il distribue un boisson , soit dans une bouteille ,
canette ou d'un mixe.

Il y a donc 3 fonctionnements possibles :

1. fournir d'une bouteille de boisson ;
2. fournir d'une canette ;
3. fournir d'un Mix.

Ces 3 cas correspondent à 3 cycles différents du distributeur.

Les cycles sont numérotés de 0 à 2 (2 bits de codage) :

00 : Lavage, Température, remplissage, Fermeture, Stock.

RAPPORT (Architecture des Ordinateurs)

01: Remplissage, fermeture, soudage, stock.

10 : Remplissage, fermeture, stock

On définit les états :

État 0 : repos

état 1 : attente d'une commande

état 2 : Lavage

état 3 : Température

état 4 : Remplissage

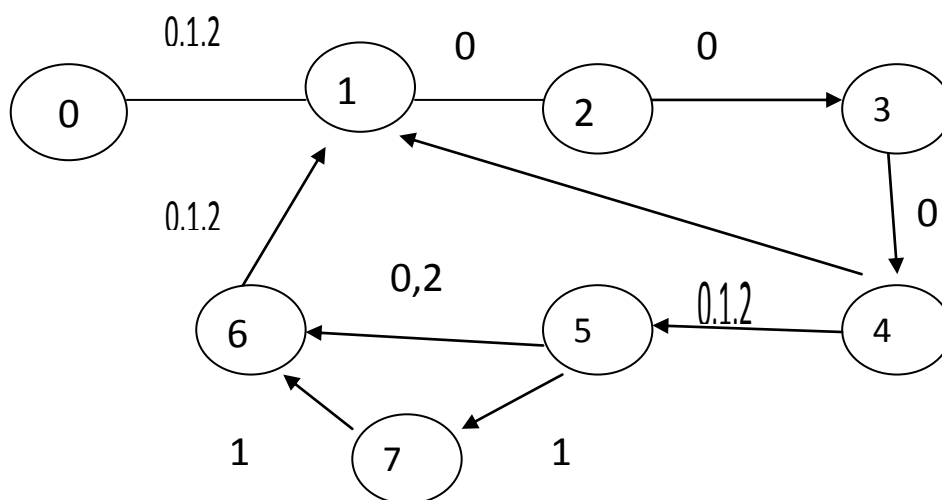
état 5 : Fermeture

état 6 : Stock

État 7 : Soudage

Il y a 8 états (0 à 7) donc 3 bascules seront nécessaires.

Graphe de transitions :



RAPPORT (Architecture des Ordinateurs)

Table de Transition :

c_1c_0	t $D_2D_1D_0$	t+1 $D_2D_1D_0$	c_1c_0	t $D_2D_1D_0$	t+1 $D_2D_1D_0$
00	000	001	10	000	001
00	001	010	10	001	100
00	010	011	10	010	xxx
00	011	100	10	011	xxx
00	100	101	10	100	101
00	101	110	10	101	110
00	110	001	10	110	001
00	111	xxx	10	111	xxx
01	000	001	11	000	xxx
01	001	100	11	001	xxx
01	010	xxx	11	010	xxx
01	011	xxx	11	011	xxx
01	100	101	11	100	xxx
01	101	110	11	101	xxx
01	110	001	11	110	xxx
01	111	xxx	11	111	xxx

Remplaçant les X par 0.

Les Tableaux de Karnaugh :

Pour D_2 :

$c_1c_0 \backslash D_2D_1D_0$	000	001	011	010	110	111	101	100
00	0	0	1	0	0	0	1	1
01	0	1	0	0	0	0	1	1
11	0	0	0	0	0	0	0	0
10	0	1	0	0	0	0	1	1

Pour D_1 :

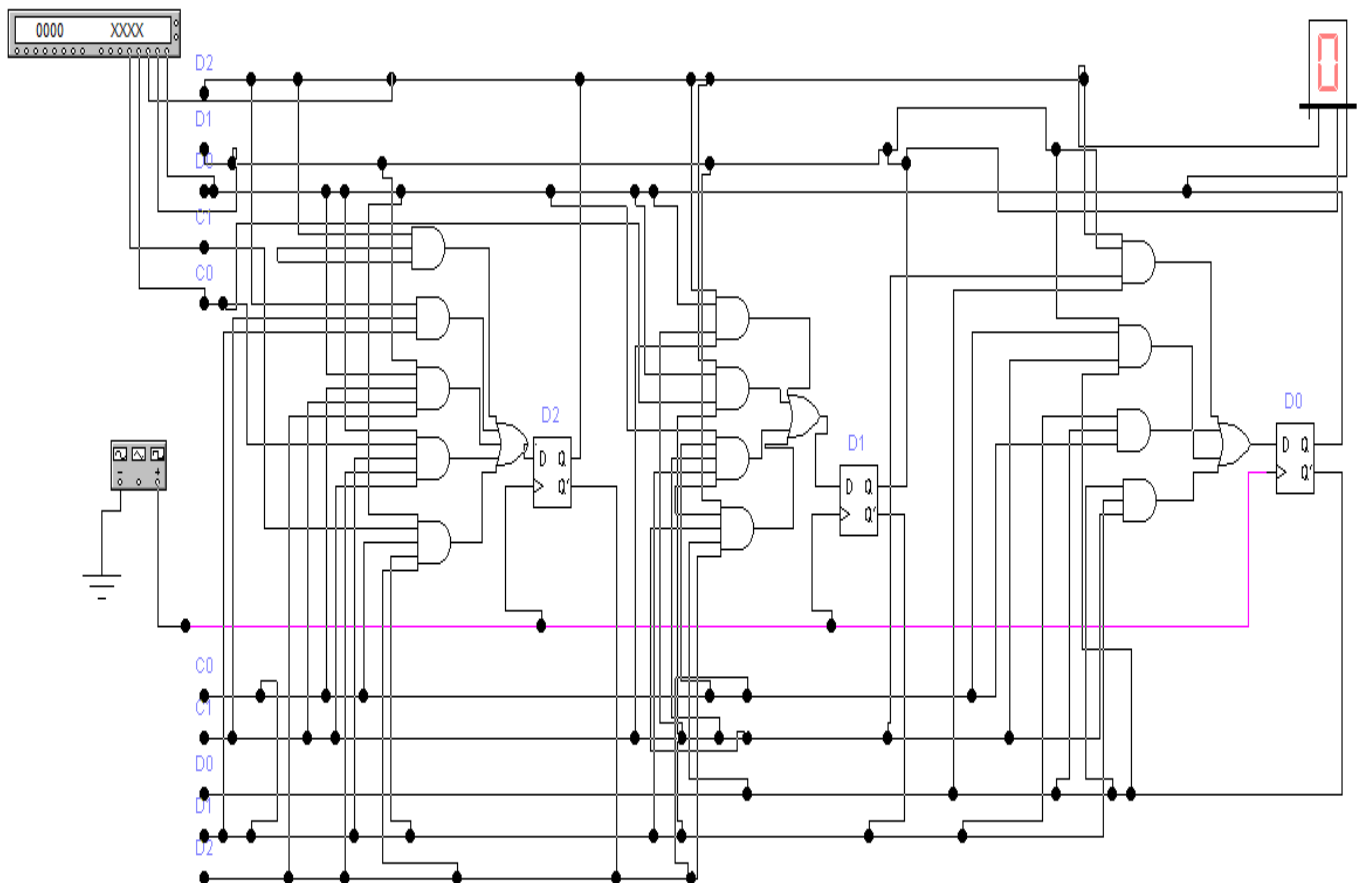
$c_1c_0 \backslash D_2D_1D_0$	000	001	011	010	110	111	101	100
00	0	1	0	1	0	0	1	0
01	0	0	0	0	0	0	1	0
11	0	0	0	0	0	0	1	0
10	0	0	0	0	0	0	0	0

Pour D_0 :

$c_1c_0 \backslash D_2D_1D_0$	000	001	011	010	110	111	101	100
00	1	0	0	1	1	0	0	1
01	1	0	0	0	1	0	0	1
11	0	0	0	0	0	0	0	0
10	1	0	0	0	0	0	0	1

$$\begin{cases} D_2^{t+1} = \overline{c_1} \cdot \overline{D_2} \cdot \overline{D_1} + \overline{c_1} c_0 \overline{D_2} \cdot D_1 \cdot D_0 + c_1 \overline{c_0} \cdot \overline{D_2} \cdot \overline{D_1} \cdot D_0 + c_1 c_0 \overline{D_2} \cdot \overline{D_1} \cdot D_0 + \overline{c_0} \cdot \overline{D_2} \cdot \overline{D_1} \\ D_1^{t+1} = \overline{c_1} \overline{D_2} \cdot \overline{D_1} \cdot D_0 + c_0 \overline{D_2} \cdot \overline{D_1} \cdot D_0 + \overline{c_1} \cdot c_0 \cdot \overline{D_2} \cdot \overline{D_1} \cdot D_0 + \overline{c_1} \cdot c_0 \cdot \overline{D_2} \cdot \overline{D_1} \cdot D_0 \\ D_0^{t+1} = \overline{c_1} \overline{D_2} \cdot \overline{D_1} \cdot D_0 + \overline{c_0} \cdot \overline{D_1} \cdot D_0 + \overline{c_1} \overline{D_2} \cdot \overline{D_1} \cdot D_0 + \overline{c_1} \cdot c_0 \cdot \overline{D_1} \cdot D_0 \end{cases}$$

Le circuit sur Electronics Workbench :



F-Conclusion:

Ce fût un rapport à la hauteur de nos attentes. IL nous a permis de laisser libre recours à notre imagination et à notre savoir faire lors de la réalisation.

La réalisation des circuits logiques combinatoires et séquentiels est un sujet qui nous a passionnés.

Outre l'aspect technique, le travail sur ce thème fût une grande aventure permettant de créer de forts liens amicaux avec tous les collaborateurs d'Electronics Workbench. Ce fût une application directe des méthodes de Négociation, de communication et d'organisation avec les amis(e) de groupe.

Sur le plan personnel, ce projet a été très formateur agissant comme une synthèse de la formation en Informatique à La Faculté des Sciences Techniques de Settat.

REMERCIEMENTS :

Nous tenons à remercier notre enseignant d'Architecture des Ordinateurs, Monsieur FIHRI qui nous a été d'une aide précieuse et qui nous a consacré du temps