Aix*Marseille université

Année 2012-13

Domaine Sciences et Technologies
LICENCE D'INFORMATIQUE

Automates et circuits : TD 9
Code UE : ENSIN2U2

Circuits combinatoires et séquentiels

Exercice 1 On souhaite réaliser un circuit permettant de calculer la partie entière de la racine carrée d'un entier codé sur 4 bits. Ex : la racine de $(12)_{10} = (1100)_2$ est $(3)_{10} = (11)_2$.

Question 1.1 *Combien le circuit doit-il avoir de sorties ?*

Solution : $|\sqrt{15}| = (3)_{10} = (11)_2$ *donc deux bits de sorties suffisent.*

Question 1.2 *Ecrivez la table de vérité du circuit. Déduisez-en les fonctions booléennes sous forme canonique disjonctive représentant chaque sortie du circuit.*

Solution :

e_3	e_2	e_1	e_0	s_1	s_0
0	0	0	0	0	0
0	0	0	1	0	1
0	0	1	0	0	1
0	0	1	1	0	1
0	1	0	0	1	0
0	1	0	1	1	0
0	1	1	0	1	0
0	1	1	1	1	0
1	0	0	0	1	0
1	0	0	1	1	1
1	0	1	0	1	1
1	0	1	1	1	1
1	1	0	0	1	1
1	1	0	1	1	1
1	1	1	0	1	1
1	1	1	1	1	1

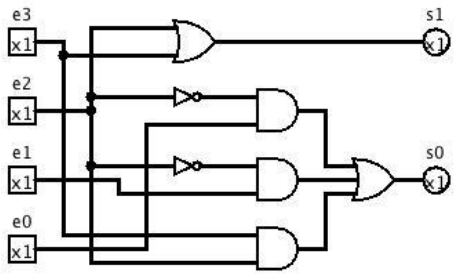
$s_1 = \bar{e_3}e_2\bar{e_1}\bar{e_0} \vee ... \vee e_3e_2e_1e_0$ *(somme booléennes de 12 produits consécutifs).* $s_0 = \bar{e_3}\bar{e_2}\bar{e_1}e_0 \vee ...$ *(somme booléennes de 10 produits).*

Question 1.3 *Déterminez les formules polynomiales minimales de ces fonctions booléennes à l'aide de la méthode des consensus.*

Solution : $s_1 = e_2 \vee e_3$ *et* $s_0 = \bar{e_2}e_0 \vee \bar{e_2}e_1 \vee e_3e_2$.

Question 1.4 *Dessinez le circuit correspondant aux formules précédentes uniquement à l'aide de portes OU, ET et NON.*

Solution :



Exercice 2 On veut réaliser la soustraction de deux entiers codés chacun sur n bits. Pour cela, nous disposons des portes logiques de base : NON, OU, ET, X-OU, etc. Nous disposons aussi

d'additionneurs n bits, c'est-à-dire de circuits logiques ayant en entrée $2n$ bits (n bits par entier dont on fait la somme) et en sortie n bits (pour le résultat de la somme). Ce circuit ne prend pas en compte la possibilité d'une retenue finale : si le résultat tient sur $n+1$ bits alors le bit de gauche est perdu.

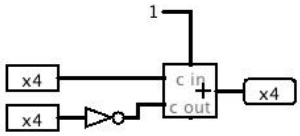
Pour réaliser un soustracteur n bits, nous allons utiliser le fait que $a - b = a + (-b)$. Nous allons donc créer un circuit qui fait la somme entre a et l'opposé de b. L'opposé de b s'obtient en faisant un complément à 2 de b : une inversion des bits de b puis une incrémentation (par un) de l'entier obtenu.

Question 2.1 *Indiquez comment réaliser un circuit permettant la soustraction de deux entiers de n bits à l'aide d'un additionneur n bits et de n portes NON.*

Solution : *Inverser chaque bits de b à l'aide de portes NON. On obtient $-b-1$, qu'on additionne avec a à l'aide d'un additionneur n bits dont on met la retenue entrante à 1.*

Question 2.2 *Dessinez ce circuit pour $n = 4$.*

Solution :



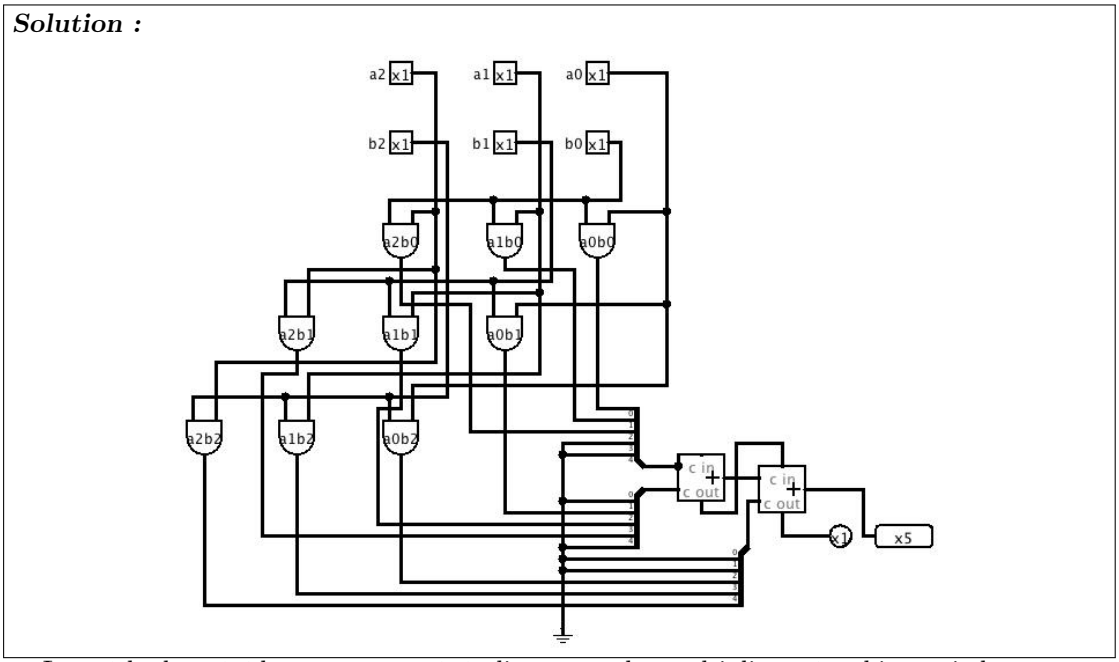
Exercice 3 Multiplieur 3 bits

On veut réaliser un circuit qui effectue le produit de deux entiers codés chacun sur 3 bits. Pour ce faire, on va s'inspirer de la multiplication "à la russe" :

		a_2	a_1	a_0
	$\times$	b_2	b_1	b_0
		a_2b_0	a_1b_0	a_0b_0
	a_2b_1	a_1b_1	a_0b_1	
a_2b_2	a_1b_2	a_0b_2		
c_4	c_3	c_2	c_1	c_0

Question 3.1 *Réalisez le circuit effectuant la multiplication en prenant deux entiers codés par les variables a_2, a_1, a_0, b_2, b_1 et b_0 et dont le résultat est codé par les variables c_4, c_3, c_2, c_1 et c_0.* Pour ceci, on a à notre disposition toutes les portes logiques de base ainsi que deux additionneurs 5 bits (sans bit de retenue finale en sortie).

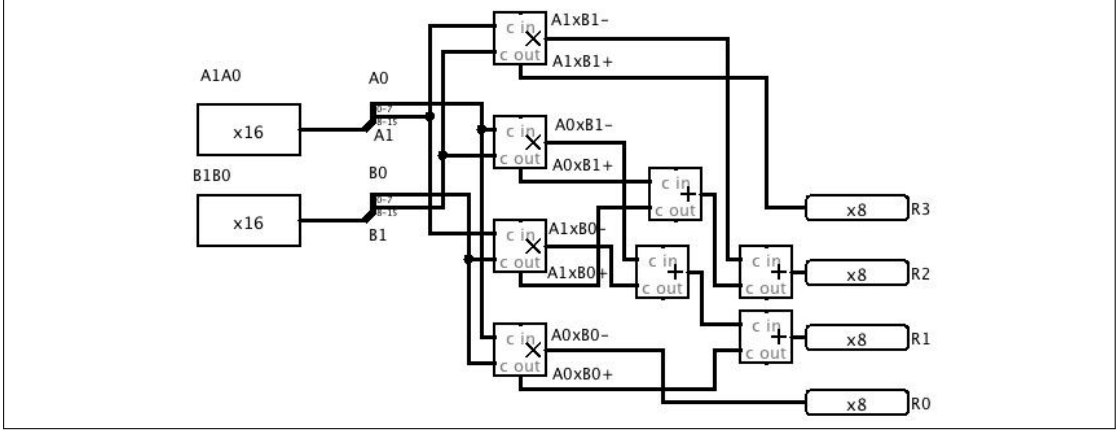
Solution :



La méthode précédente peut se généraliser pour des multiplieurs à n bits mais lorsque n est une puissance de 2, on peut utiliser une autre méthode qui permet de construire un multiplieur $2n$ bits à partir de quatre multiplieurs n bits. Si A et A' sont des entiers codés sur n bits, on note A_A' l'entier codé sur $2n$ bits dont la partie de poids fort est A et la partie de poids faible est A'. On a A_A' = A $\times 2^n$+ A' donc A_A' $\times$ B_B' = (A.2^n + A') $\times$ (B.2^n + B') = (A $\times$ B).2^{2n} + (A $\times$ B' + A' $\times$ B).2^n + A' $\times$ B'.

Question 3.2 *Déduisez-en comment on peut réaliser un multiplieurs $2n$ bits uniquement à partir de 4 multiplieurs n bits (ayant deux entiers de n bits chacun en entrée et un résultat en sortie sur $2n$ bits) et de 8 additionneurs n bits.*

Solution :



Exercice 4 Le but final de l'exercice est de réaliser un circuit avec mémoire qui indique le nombre d'objets contenus dans une boîte au cours du temps. Cette boîte peut contenir de 0 à 3 objets. Un fournisseur peut de temps à autre décider d'ajouter un objet dans la boîte. Un consommateur peut de temps à autre décider de retirer un objet de la boîte.

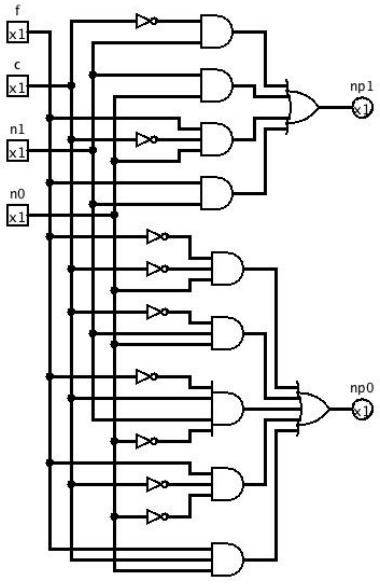
Question 4.1 Dans un premier temps, nous allons construire un circuit C avec 4 bits d'entrée n_1, n_0, f et c et 2 bits de sortie n'_1 et n'_0. Les entrées n_1 et n_0 codent en représentation binaire

le nombre d'objets contenus actuellement dans la boîte ($n_1 = 1$ et $n_0 = 0$ signifie qu'il y a deux objets). $f = 1$ signifie que le fournisseur veut ajouter un objet tandis que $f = 0$ signifie qu'il ne fait rien. $c = 1$ signifie que le consommateur veut retirer un objet et $c = 0$ signifie qu'il ne fait rien. Les sorties n'_1 et n'_0 codent le nombre d'objets que contiendra la boîte après que le fournisseur et/ou le consommateur aient modifié le nombre d'objets. Remarque : si $f = c$, le nombre d'objets ne change pas ; si $f = 1$ et $c = 0$, il y aura un objet de plus (s'il n'y en avait pas trois) ; si $f = 0$ et $c = 1$, il y aura un objet de moins (s'il y en avait au moins un).

Ecrivez la table de vérité des fonctions logiques qui indique les valeurs des sorties n'_1 et n'_0. Puis, déterminez les fonctions booléennes correspondantes (dans leur forme polynômiale minimale), puis dessinez le circuit correspondant.

Solution :

n_1	n_0	f	c	n'_1	n'_0
0	0	0	0	0	0
0	0	0	1	0	0
0	0	1	0	0	1
0	0	1	1	0	0
0	1	0	0	0	1
0	1	0	1	0	0
0	1	1	0	1	0
0	1	1	1	0	1
1	0	0	0	1	0
1	0	0	1	0	1
1	0	1	0	1	1
1	0	1	1	1	0
1	1	0	0	1	1
1	1	0	1	1	0
1	1	1	0	1	1
1	1	1	1	1	1

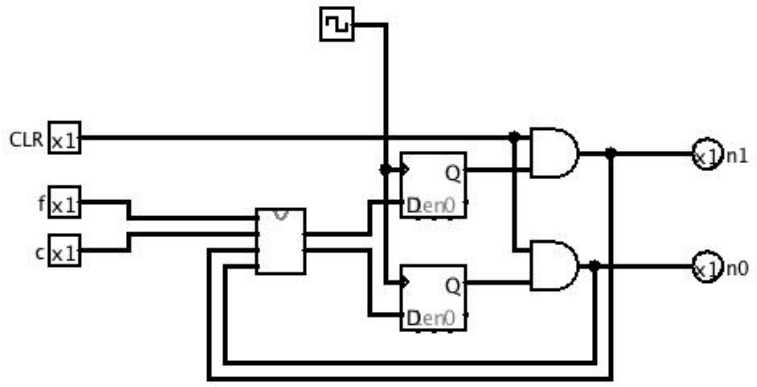


Question 4.2 Maintenant, on veut réaliser le circuit avec mémoire qui indique le nombre d'objets contenus dans la boîte au cours du temps. Ce circuit possède une entrée d'horloge nommée CK, trois entrées f, c et CLR et deux sorties n_1 et n_0. Les sorties n_1 et n_0 indiquent le nombre actuel de jetons dans la boîte. Les entrées f et c ont le même sens qu'auparavant. Lorsque l'entrée CLR est à 0, la boîte est vidée, c'est-à-dire, les sorties n_1 et n_0 sont à 0. Lorsque CLR est à 1, à chaque

impulsion d'horloge les sorties sont modifiées en fonction de f, de c et des précédentes valeurs de n_1 et n_0.

Dessinez le circuit décrit ci-dessus. Pour ceci, on n'utilisera que le circuit C, des bascules D et des portes logiques NON, OU ou ET.

Solution :



Exercice 5 Le but final de l'exercice est de réaliser un circuit avec mémoire qui sert de générateur de nombres pseudo-aléatoires. En informatique, une façon simple de simuler la génération de nombres aléatoires est de définir une suite du type $u_{n+1} = (u_n \times a + b)$ modulo k. (On rappelle que n modulo k est égal au reste de la division entière de n par k). En fixant la valeur de la graine u_0, nous pouvons obtenir tous les autres éléments de la suite. Lorsque a et b sont bien choisis, on peut obtenir une suite périodique d'entiers de période k, contenant tous les entiers compris entre 0 et $k-1$, dont l'ordre semble aléatoire.

Dans le cadre de cet exercice, nous fixons $k = 16$, $a = 5$ et $b = 1$ et, à partir de la graine $u_0 = 1$, nous obtenons la suite de période 16 : $u_0 = 1$, $u_1 = 6$, $u_2 = 15$, $u_3 = 12$, $u_4 = 13$, etc.

Dans un premier temps, nous allons construire un circuit C avec 4 bits d'entrée e_3, e_2, e_1 et e_0 et 4 bits de sortie s_3, s_2, s_1 et s_0. Les 4 bits d'entrées codent un entier binaire, e_3 étant le bit de poids fort (le plus à gauche). Ex : l'entier binaire 0100 est codé par $e_3 = 0$, $e_2 = 1$, e_1=0 et $e_0 = 0$. Les 4 bits de sortie codent aussi un entier binaire avec les mêmes conventions.

Si un entier x ($0 \leq x \leq 15$) est fourni en entrée du circuit C alors celui-ci fournit la valeur $(x \times 5 + 1)$ modulo 16 en sortie.

Question 5.1 *Ecrire la table de vérité de la fonction logique qui indique les valeurs des sorties s_3, s_2, s_1 et s_0 en fonction des valeurs d'entrée e_3, e_2, e_1 et e_0.*

Solution :

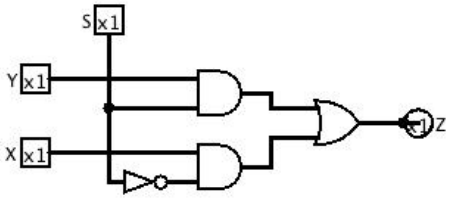
e_3	e_2	e_1	e_0	s_3	s_2	s_1	s_0
0	*0*	*0*	*0*	*0*	*0*	*0*	*1*
0	*0*	*0*	*1*	*0*	*1*	*1*	*0*
0	*0*	*1*	*0*	*1*	*0*	*1*	*1*
0	*0*	*1*	*1*	*0*	*0*	*0*	*0*
0	*1*	*0*	*0*	*0*	*1*	*0*	*1*
0	*1*	*0*	*1*	*1*	*0*	*1*	*0*
0	*1*	*1*	*0*	*1*	*1*	*1*	*1*
0	*1*	*1*	*1*	*0*	*1*	*0*	*0*
1	*0*	*0*	*0*	*1*	*0*	*0*	*1*
1	*0*	*0*	*1*	*1*	*1*	*1*	*0*
1	*0*	*1*	*0*	*0*	*0*	*1*	*1*
1	*0*	*1*	*1*	*1*	*0*	*0*	*0*
1	*1*	*0*	*0*	*1*	*1*	*0*	*1*
1	*1*	*0*	*1*	*0*	*0*	*1*	*0*
1	*1*	*1*	*0*	*0*	*1*	*1*	*1*
1	*1*	*1*	*1*	*1*	*1*	*0*	*0*

Question 5.2 *En utilisant la méthode des consensus, écrivez les fonctions booléennes sous forme polynômiale minimale correspondant aux valeurs de sorties de s_3, s_2, s_1 et s_0.*

Solution : $s_3 = \bar{e_3}e_1\bar{e_0} \vee \bar{e_3}e_2\bar{e_1}e_0 \vee e_3\bar{e_2}\bar{e_1} \vee e_3\bar{e_1}\bar{e_0} \vee e_3e_1e_0$, $s_2 = \bar{e_2}\bar{e_1}e_0 \vee e_2\bar{e_0} \vee e_2e_1$, $s_1 = \bar{e_1}e_0 \vee e_1\bar{e_0}$ *et* $s_0 = \bar{e_0}$.

Question 5.3 *Dessinez un multiplexeur à 3 entrées X, Y, S et une sortie Z, sachant que lorsque S=0, on a Z=X, et lorsque S=1, on a Z=Y.* Pour ceci, on n'a le droit d'utiliser que la porte logique unaire NON et les portes logiques OU et ET. (Vous pouvez retrouver la composition du multiplexeur à partir de sa table de vérité.)

Solution :



Question 5.4 Maintenant, on veut réaliser le circuit à mémoire simulant une suite d'entiers tirés aléatoirement. Ce circuit possède une entrée d'horloge nommée CK, cinq entrées g_3, g_2, g_1, g_0 et CLR et quatre sorties n_3, n_2, n_1 et n_0. Les 4 entrées g_i codent la graine de la suite et les sorties n_i codent un élément de la suite. Lorsque CLR est à 0, on a la graine en sortie du circuit. Lorsque CK=1, à chaque cycle d'horloge, on passe en sortie à l'élément suivant de la suite (le suivant de celui qui était en sortie au cycle précedent).

Dessiner le circuit décrit ci-dessus. Pour ceci, on n'utilisera que le circuit C (défini en début d'exercice), des multiplexeurs à 3 entrées (cf question **5.3**), des bascules D et des portes logiques NON, OU ou ET. Dans cette question, le circuit C et les multiplexeurs sont considérés comme des boîtes noires, on les dessinera comme des rectangles (avec leurs entrées et leurs sorties) et on ne redessinera donc pas les portes logiques qui les composent.

Solution :