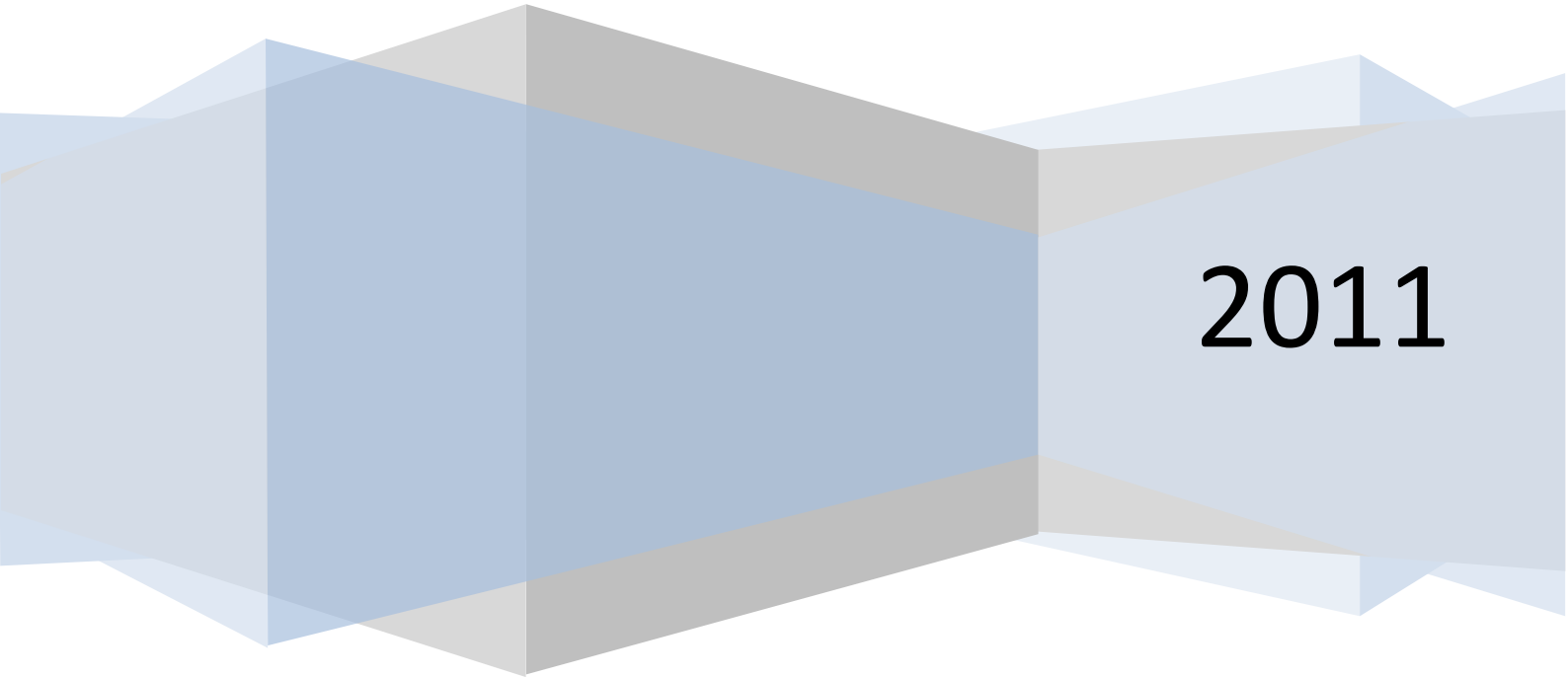


EFREI

Projet Information Numérique

LEPOT Florian – FABRE Maxime



2011

Introduction

Nous devons pour ce premier projet d'électronique mettre en œuvre la réalisation d'un calculateur numérique. Le travail était reparté en plusieurs séances distinctes que nous indiquerons.

Ce premier projet consistait également à concevoir de petits circuits les plus simples possibles et les plus efficaces, dans l'optique de réduire les éventuels coûts de fabrications.

Table des matières

Introduction.....	1
Sommaire	Erreur ! Signet non défini.
Séance n°1 : Réalisation d'additionneurs.....	3
Réalisation d'un additionneur à propagation de retenue.....	3
2.1 Demi additionneur	3
2.2 Additionneur complet	4
2.3 Additionneur à propagation de retenue	5
Séance n°2 : Réalisation d'additionneur soustracteur	6
1.3 Réalisation d'un additionneur 8 bits	6
Le 74LS83N est un additionneur 4 bits complet rapide, le circuit aurait pu être remplacé par celui deux 74LS83N.....	7
2.1 Réalisation d'un demi soustracteur de deux éléments binaires	7
2.2 Réalisation d'un soustracteur complet	8
2.3 Réalisation d'un additionneur-soustracteur complet	9
2.4 Réalisation d'un additionneur-soustracteur de deux mots binaires de 4 bits.	9
Séance n°3 : Réalisation de multiplieurs	11
Multiplieur de mots de deux bits par des mots de deux bits.....	11
1.1. Réalisation de la brique élémentaire M	11
1.2* Multiplieur d'un nombre de 2 bits par un autre de 2 bits.....	11
Multiplieur de mots de deux bits par des mots de n bits	12
2.1. Réalisation d'un multiplieur d'un nombre de deux bits par un nombre sur n bits	12
Conclusion	15

Séance n°1 : Réalisation d'additionneurs

Réalisation d'un additionneur à propagation de retenue

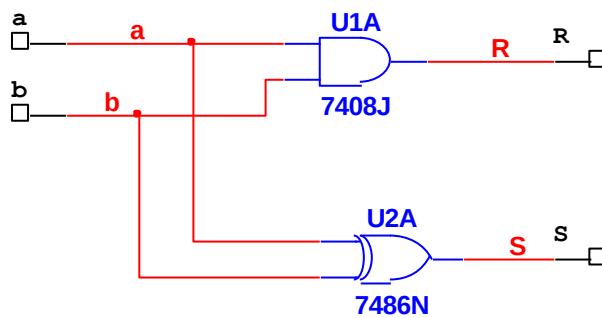
2.1 Demi additionneur

a. Table de vérité

a	b	R	S
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Equation logique : $S = a \oplus b$ et $R = a.b$ (Le logic converter confirme les équations)

b. Schéma d'un demi-additionneur



a : bit de poids fort
b : bit de poids faible
S : Résultat
R : Retenue

2.2 Additionneur complet

a. Table de vérité d'un additionneur complet

Ai	Bi	Ri	Ri+1	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

On détermine maintenant les équations de R et S.

Equation simplifiée de R

$$R = \overline{A_i}B_iR_i + A_i\overline{B_i}R_i + A_iB_i\overline{R_i} + A_iB_iR_i$$

$$R = A_iB_i(R_i + \overline{R_i}) + R_i(\overline{A_i}B_i + A_i\overline{B_i})$$

$$R = R_i.A \oplus B_i + A_iB_i$$

Equation simplifiée de S

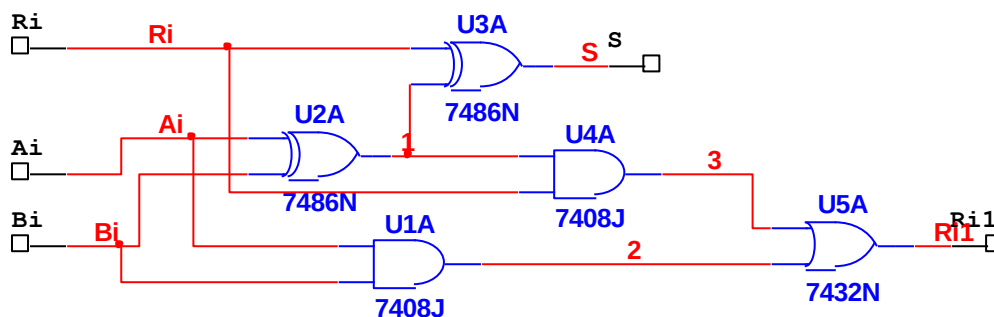
AiBi\R	0	1
00	0	1
01	1	0
11	0	1
10	1	0

Avec le tableau de Karnaugh, on obtient un damier, on a donc une fonction XOR.

$$S = R_i \oplus A_i \oplus B_i$$

b. Un additionneur est composé de 2 demis additionneurs et d'une porte ET, comme le montre le schéma ci-dessous.

c. Schéma d'un additionneur complet



On place ce circuit dans un sous-circuit nommé SubAdd.

d. Circuit d'un additionneur complet

Voir annexe Séance 1 a).

2.3 Additionneur à propagation de retenue

b. $R_i = (A_i B_i) + (R_{i-1})(A_i + B_i)$

Or $G_i = A_i B_i$ et $P_i = A_i + B_i$

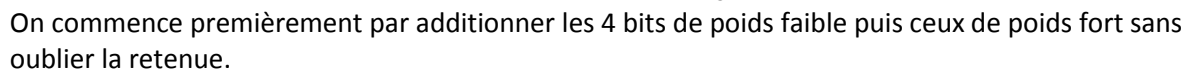
On a donc $R_i = G_i + (R_{i-1})P_i$

c. $R_1 = G_1 + R_0 P_1 = G_1 + G_0 P_1$

$$R_2 = G_2 + R_1 P_2 = G_2 + (G_1 + G_0 P_1) P_2$$

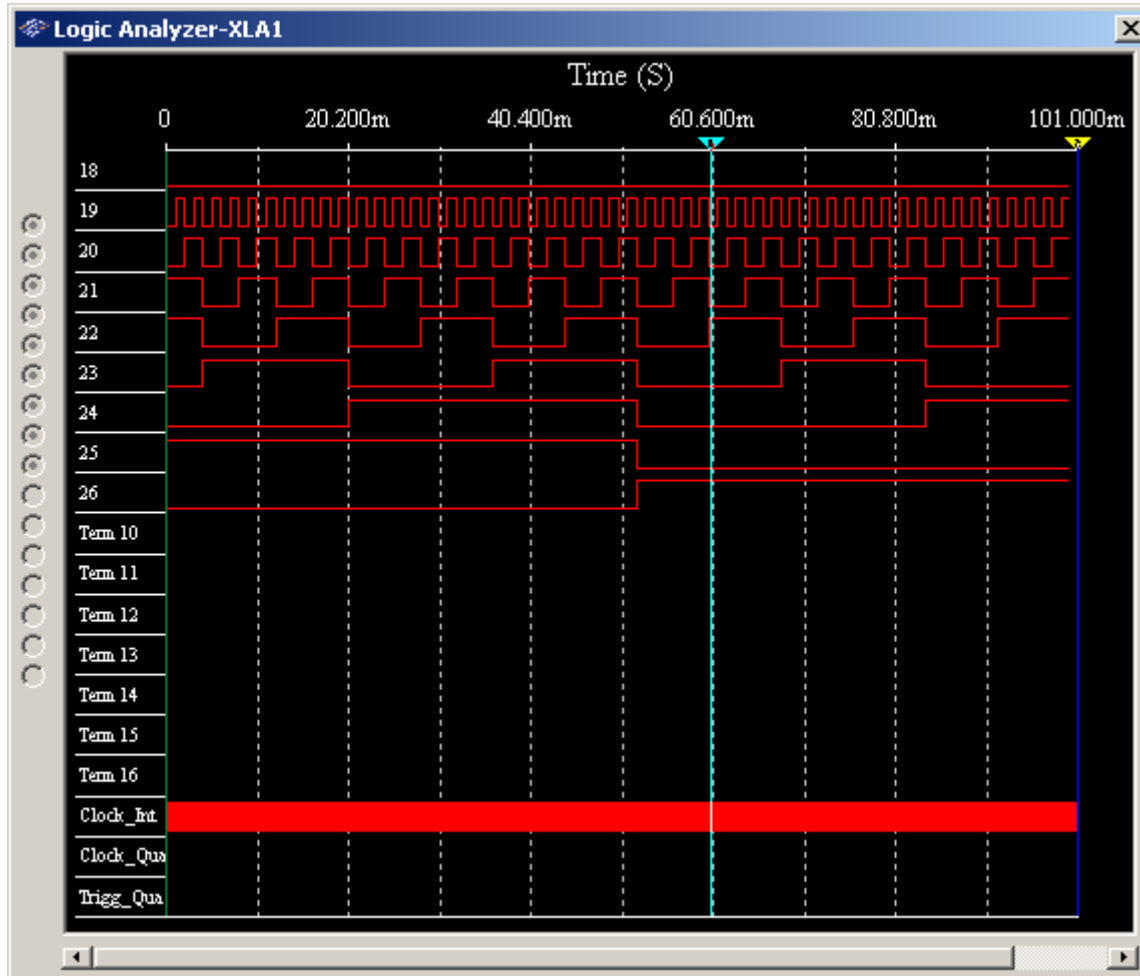
$$R_3 = G_3 + R_2 P_3 = G_3 + (G_2 + (G_1 + G_0 P_1) P_2) P_3$$

1.3 Réalisation d'un additionneur 8 bits



Pour les entrées on a d'abord tous les bits d'un mot (en commençant par le plus faible) puis on enchaîne sur tous les bits du second mot.

Sorties



Les sorties nous affiches le mot en commençant par le bit le plus faible et en terminant par la retenue.

Le 74LS83N est un additionneur 4 bits complet rapide, le circuit aurait pu être remplacé par celui deux 74LS83N

2.1 Réalisation d'un demi soustracteur de deux éléments binaires

$$Si = B - A$$

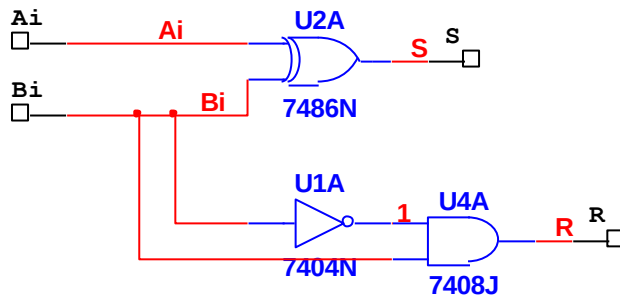
Table de vérité

A	B	Si	R
0	0	0	0
0	1	1	0
1	0	1	1
1	1	0	0

$$S = A \oplus B$$

$$R = A \cdot \bar{B}$$

On obtient le schéma suivant :



2.2 Réalisation d'un soustracteur complet

$$S_i = B_i - (A_i + R_{i-1})$$

On obtient la table de vérité suivante :

Ai	Bi	Ri	S	Ri+1
0	0	0	0	0
0	0	1	1	1
0	1	0	1	0
0	1	1	0	0
1	0	0	1	1
1	0	1	0	1
1	1	0	0	0
1	1	1	1	1

On simplifie les équations S et Rn+1

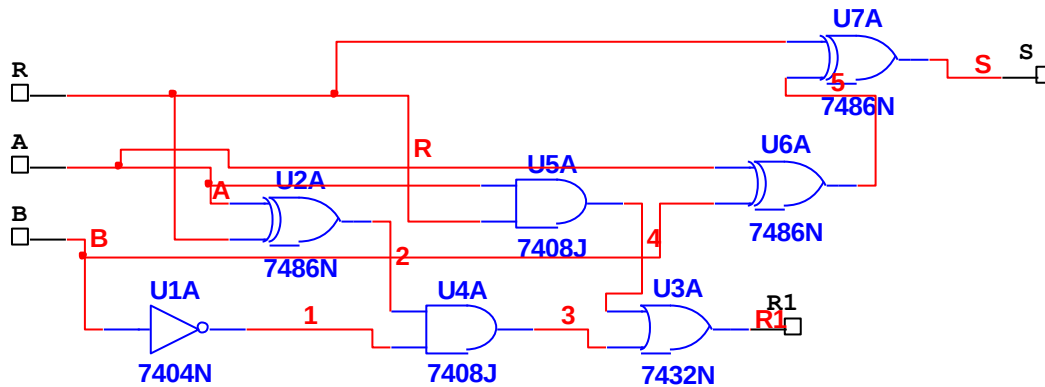
AB\R	0	1
00	0	1
01	1	0
11	0	1
10	1	0

$$\begin{aligned}
 S &= \bar{A}\bar{B}R + \bar{A}B\bar{R} + A\bar{B}\bar{R} + ABR \\
 S &= R(\bar{A}\bar{B} + AB) + \bar{R}(\bar{A}B + A\bar{B}) \\
 S &= R(\bar{A} \oplus \bar{B}) + \bar{R}(A \oplus B) \\
 S &= R \oplus A \oplus B
 \end{aligned}$$

On simplifie R.

$$\begin{aligned}
 R &= \bar{A}\bar{B}R + \bar{A}B\bar{R} + A\bar{B}R + ABR \\
 R &= AR + \bar{B}(A \oplus R)
 \end{aligned}$$

On obtient le circuit suivant :



2.3 Réalisation d'un additionneur-soustracteur complet

Dans un circuit soustracteur et additionneur l'équation de sortie S est la même $R \oplus A \oplus B$.

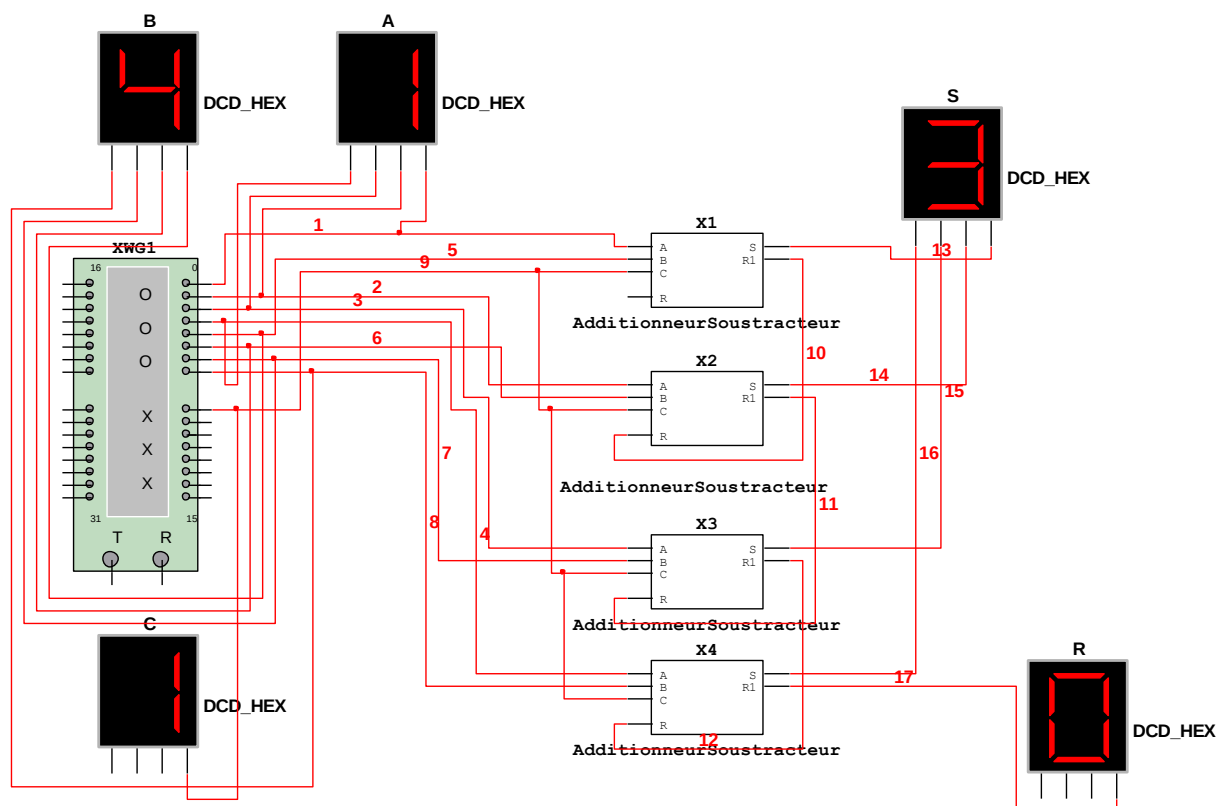
L'équation du reste lors d'une addition est : $AR + B$ ($A \oplus R$)

L'équation du reste lors d'une soustraction est : $AR + \bar{B}$ ($A \oplus R$).

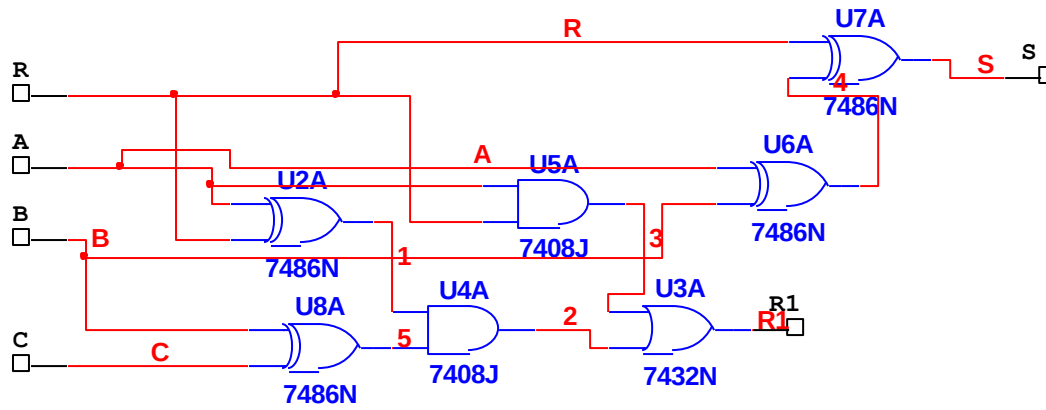
Seul le B change. Il faut donc trouver un moyen de changer la valeur de B selon la valeur de C afin de réaliser un circuit qui fait une addition quand $C = 0$, et une soustraction quand $C = 1$.

Le XOR est la porte la plus adapté.

2.4 Réalisation d'un additionneur-soustracteur de deux mots binaires de 4 bits.



Le circuit est constitué d'un sous-circuit (le soustracteur additionneur complet) réalisée à la question précédente.



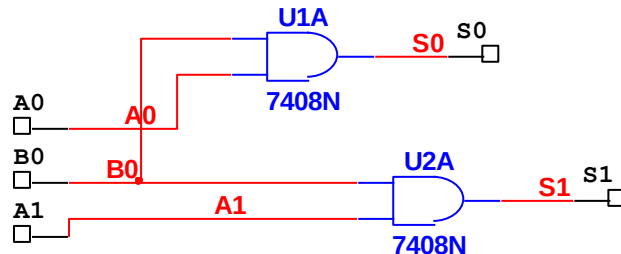
Circuit du soustracteur additionneur complet. Seules la porte XOR à la place de la porte NON et l'ajout de C changent du soustracteur complet.

Séance n°3 : Réalisation de multiplieurs

Multiplieur de mots de deux bits par des mots de deux bits

1.1. Réalisation de la brique élémentaire M

Pour l'opération « Multiplier », on utilise une porte ET :



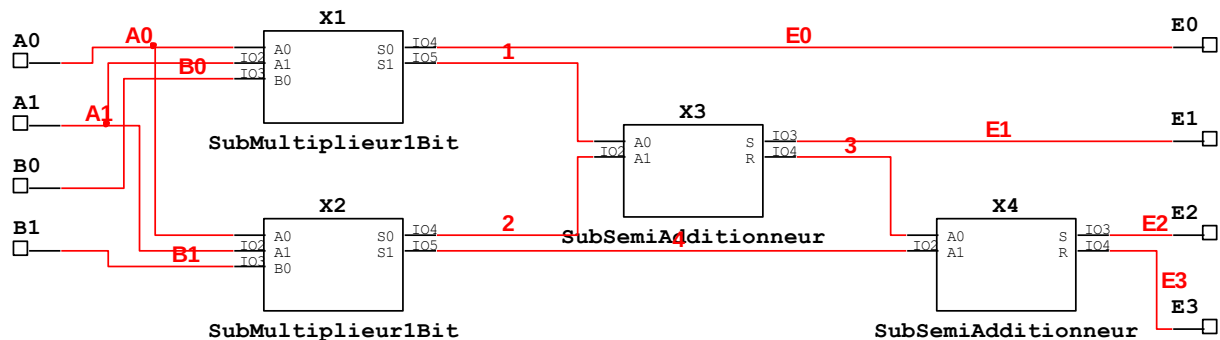
Si $A0 = 0$, $A1 = 1$ et $B0 = 1$, $S0 = 0$ et $S1 = 1$.

1.2* Multiplieur d'un nombre de 2 bits par un autre de 2 bits

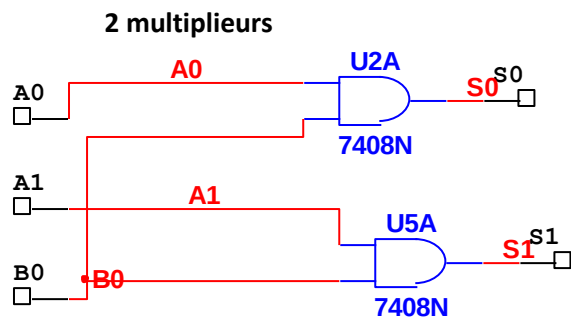
Dans ce schéma, il manque le signe « + » à la 4eme ligne.

Il manque également le bit pour une éventuelle retenue.

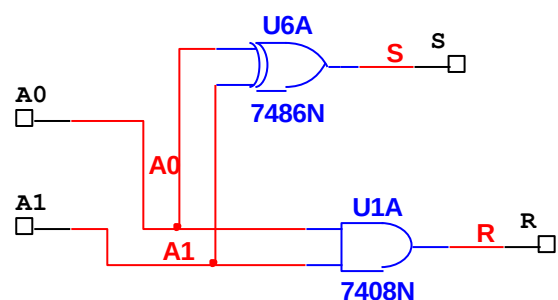
Le résultat est donc sur 4 bits (3 bits apparents sur le schéma, plus celui de la retenue).



Il y'a donc 4 sous circuits utilisés :



2 demi-additionneurs



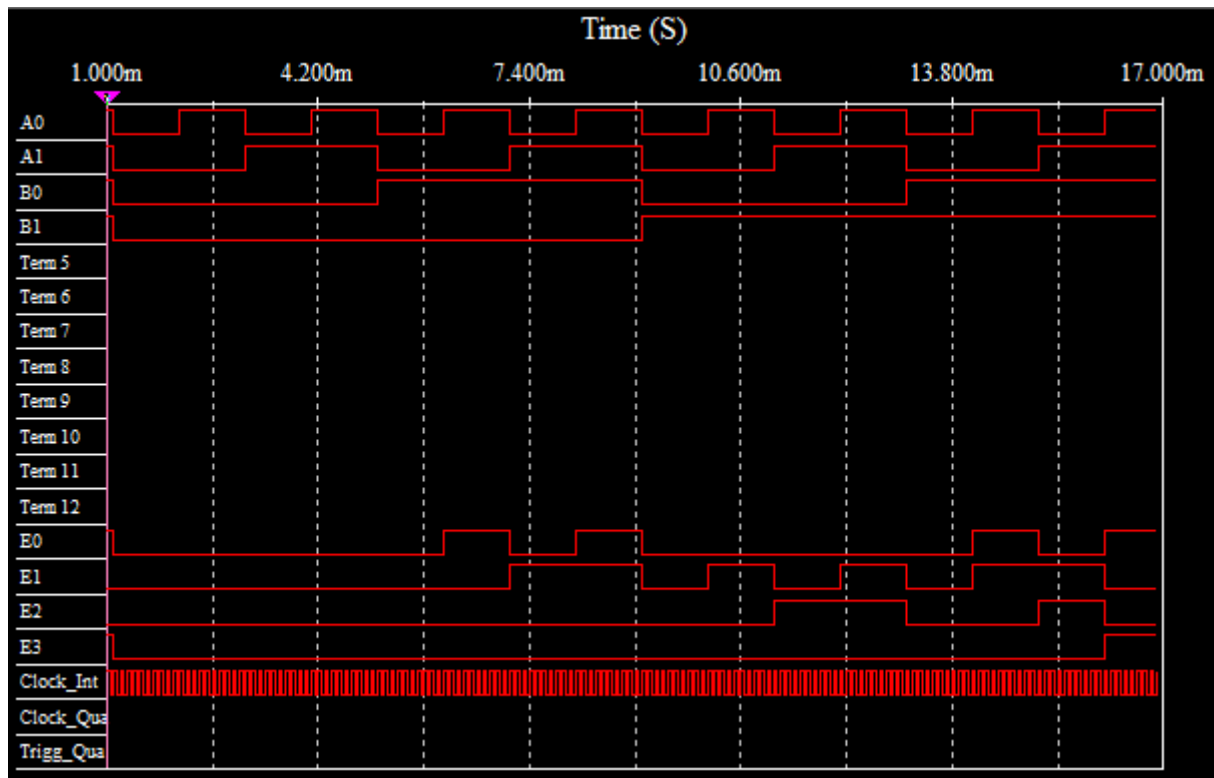
Le fonctionnement est assez simple, on a 2 mots de 2 bits A0A1 et B0B1.

On commence par effectuer une multiplication de A0A1 par B0, on obtient alors une sortie sur 2 bits : S0 et S1. S0 est notre sortie finale E0.

On effectue ensuite la multiplication de A0A1 par B1, on obtient à nouveau une sortie sur 2 bits : S2 et S3. On additionne S1 et S2, on obtient donc une sortie à nouveau sur 2 bits : E1 et R, E1 étant la somme de S1 et S2, et R la retenue. Pour finir, on additionne R et S3, on obtient donc une dernière sortie sur 2 bits : E2 et E3.

On a donc au final, 4 bits de sortie, E0, E1, E2 et E3.

En branchant un Word Generator et un Logic Analyzer, on obtient ceci :



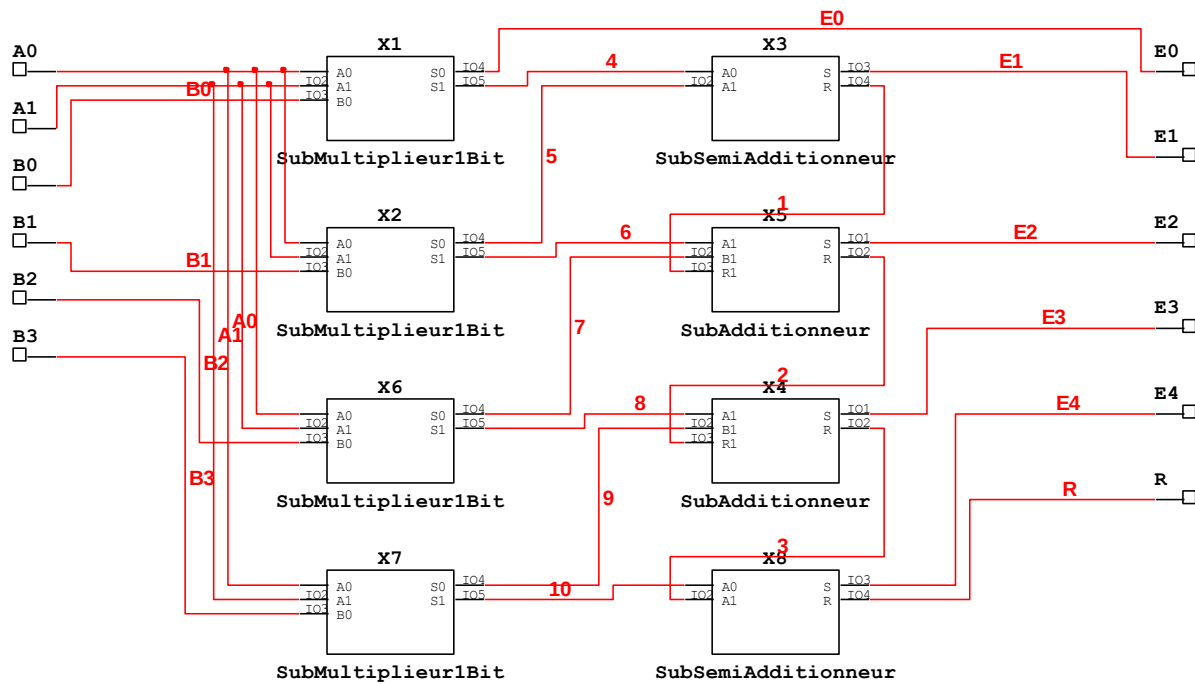
A0, A1, B0 et B1 sont les entrées et E0, E1, E2 et E3 sont les sorties.

Multiplieur de mots de deux bits par des mots de n bits

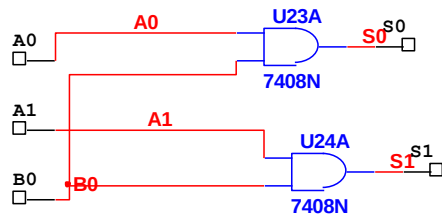
2.1. Réalisation d'un multiplieur d'un nombre de deux bits par un nombre sur n bits

		A1 A0			
X	B3 B2	B1 B0			
+			A1B0	A0B0	
+		A1B1	A0B1	.	
+	A1B2	A0B2	.	.	
+	A1B3	A0B3	.	.	.
=	A1B3	(A1B2 + A0B3)	(A1B1 + A0B2)	(A2B0 + A0B1)	(A0B0)

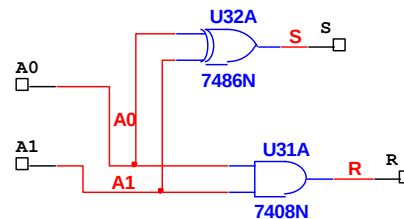
Le résultat est sur n + 2 bits.



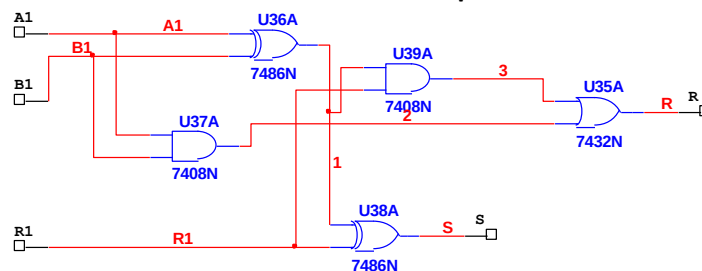
4 multiplieurs 2 bits par 1 bit



2 demi-additionneurs



2 additionneurs complets

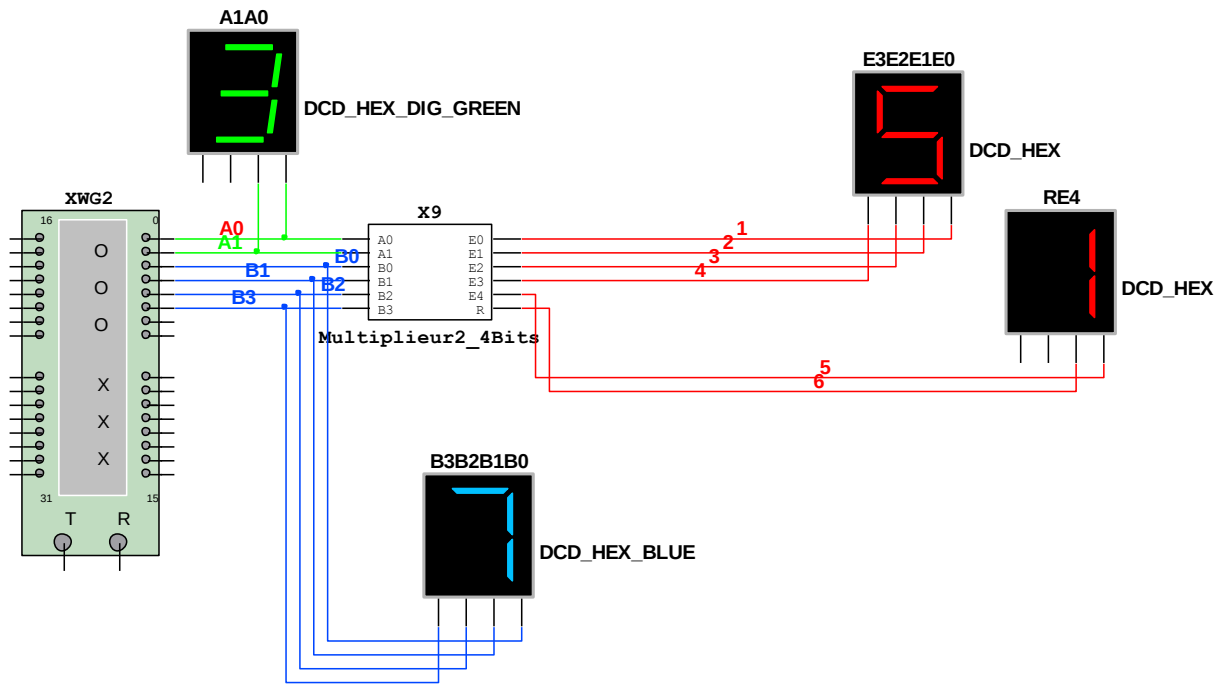


Le fonctionnement est le même que pour le multiplieur 2 bits par 2 bits, on effectue les multiplications de A1A0 par B0, de A1A0 par B1, de A1A0 par B2 et de A1A0 par B3.

Pour chaque multiplication, on obtient une sortie sur 2 bits :

- Pour la multiplication de A1A0 par B0, le premier bit de sortie S0 correspond à E0, le 1^{er} bit du résultat final. Le 2^{eme} bit est quant à lui envoyé dans un demi additionneur.
- Dans ce demi-additionneur, on envoie également le 1^{er} bit de sortie de la 2^{eme} multiplication. On obtient alors la sortie S et la retenue R. La sortie S correspond à E1, le 2^{eme} bit du résultat final, la retenue R est envoyée dans un additionneur complet avec la 2^{eme} sortie de la 2^{eme} addition (de A1A0 par B1 donc) et de la 1^{ere} sortie de la 3^{eme} multiplication.
- On fait de même avec les 2 dernières multiplications, et on obtient au final 6 bits de sortie : E0, E1, E2, E3, E4 et R qui forment le résultat de notre multiplication.

Par exemple, avec des afficheurs hexadécimaux on obtient ceci :



On a ici la multiplication 3 x 7.

La sortie RE4 est la retenue, et correspond à 16 lorsque l'affichage est de 1.

On a donc $3 \times 7 = 16 + 5 = 21$.

Conclusion

En conclusion, ce projet nous a permis de concrétiser les connaissances acquises tout au long du semestre.

Ce projet fut enrichissant, premièrement, d'un point de vue théorique, avec l'étude du fonctionnement d'un calculateur, d'un additionneur, mais également d'un point de vue technique avec la réalisation de petits circuits et l'utilisation de Multisim.