

Projet 1 : L'information numérique

Séance 1&2&3

Réalisations d'additionneurs

1.1) Demi-additionneur

1,1,1) Table de vérité d'un demi-additionneur

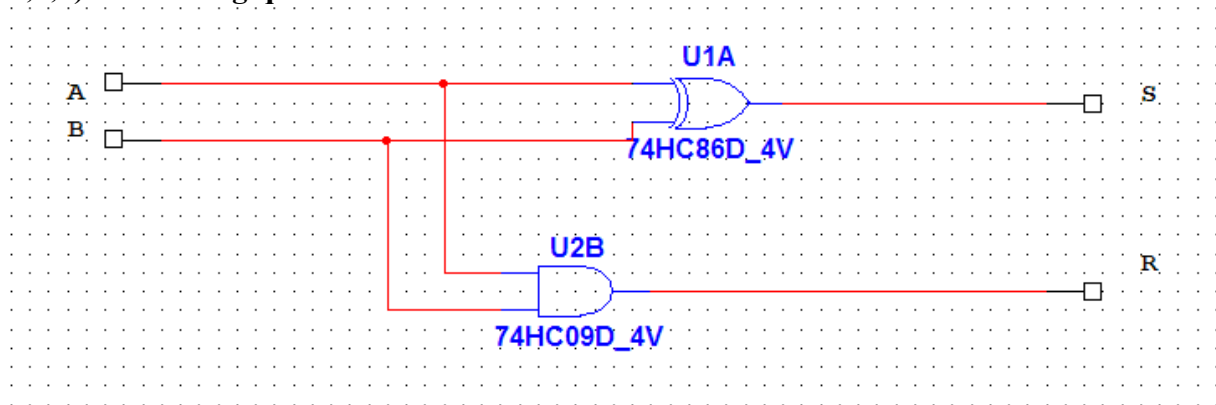
<i>A</i>	<i>B</i>	<i>R</i>	<i>S</i>
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

1,1,2) Equation logique des sorties

$$S = A \text{ XOR } B$$

$$R = A \cdot B$$

1,1,3) Schéma logique d'un demi-additionneur





1.2) Additionneur complet

1,2,1) Table de vérité d'un additionneur complet

A_i	B_i	R_{i-1}	R_i	S_i
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

1,2,2) Equations logiques des sorties

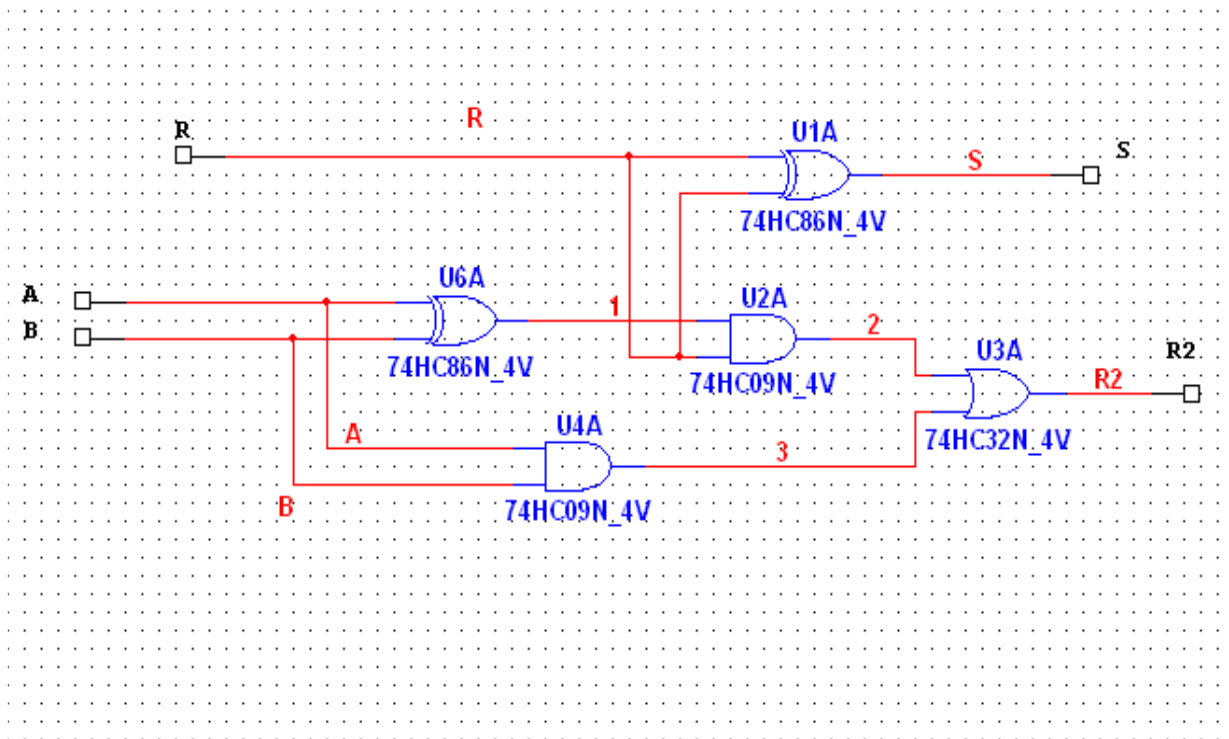
$$S_i = A_i B_i R_{i-1} + A_i B_i R'_{i-1} + A_i B'_i R_{i-1} + A_i B_i R'_{i-1}$$

$$R_i = A_i B_i R_{i-1} + A_i B'_i R_{i-1} + A_i B_i R'_{i-1} + A_i B'_i R'_{i-1}$$

1,2,3)

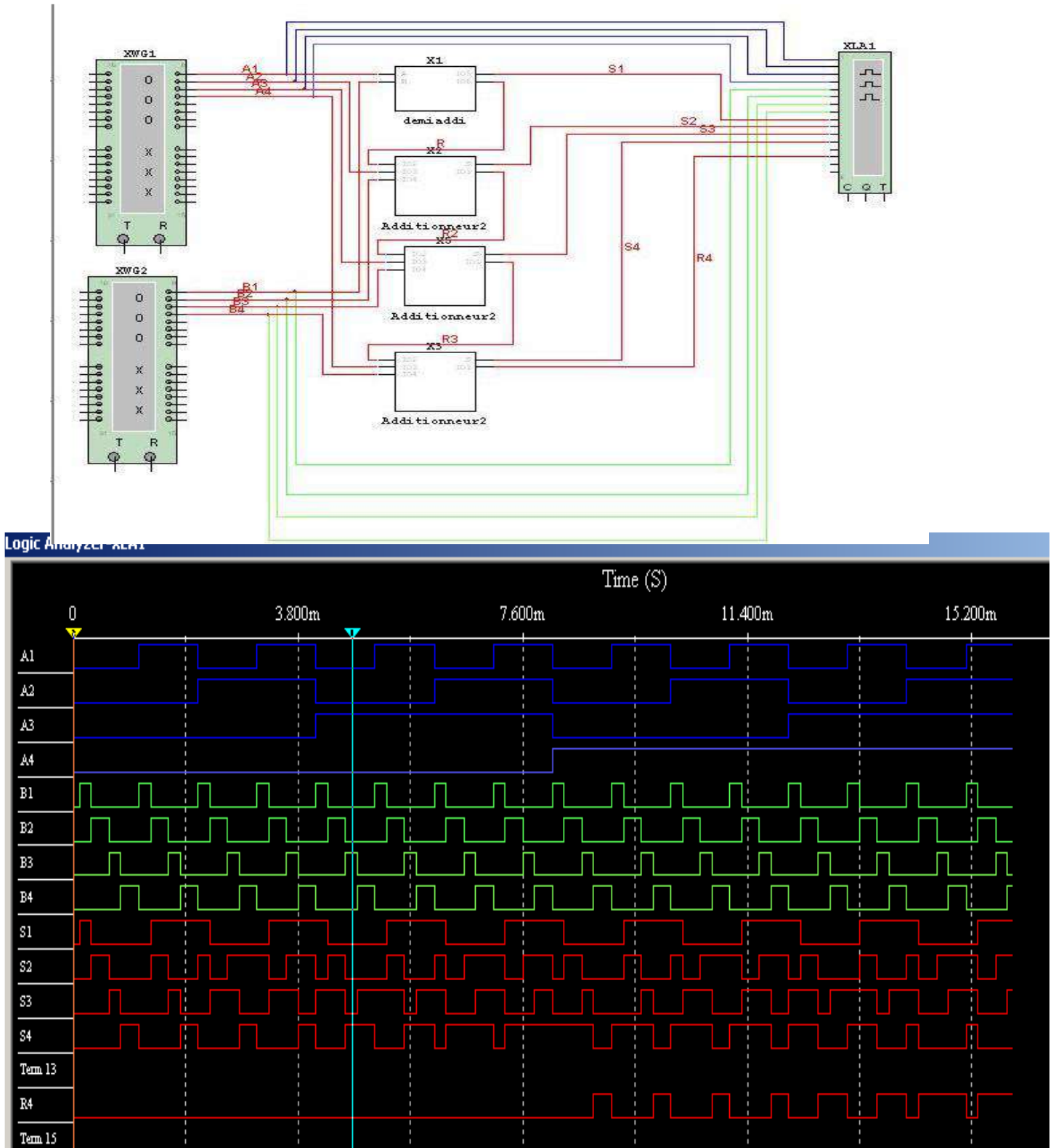
L'additionneur complet fait la somme de la retenue précédente et du résultat de l'addition deux mot binaire donc il peut être construit à partir de deux demi-additionneur et d'une porte OU (pour la prochaine opération)

1.2.4&1.2.5) Schéma logique du circuit réalisé sur Multisim



1.3) Additionneur à propagation de retenue

Voici le schéma obtenu à l'aide de Multisim



1.4) Additionneur à retenue anticipée

On a : $G_i = A_i \cdot B_i$ $P_i = A_i + B_i$

Calcul de R_{i+1}

$$R_{i+1} = R_i \cdot A_i + A_i \cdot B_i + R_i \cdot B_i$$

$$R_{i+1} = R_i (A_i + B_i) + A_i \cdot B_i$$

$$R_{i+1} = R_i (P_i) + G_i$$

L'expression de R_1, R_2, R_3

$$R_0 = 0$$

$$R_1 = G_0 + R_0 \cdot P_0$$

$$R_2 = G_1 + (G_0 + R_0 \cdot P_0) \cdot P_1 = G_1 + G_0 \cdot P_1 = G_1 + R_1 \cdot P_1$$

$$R_3 = G_2 + (G_1 + (G_0 + R_0 \cdot P_0) \cdot P_1) \cdot P_2 = G_2 + (G_1 + G_0 \cdot P_1) \cdot P_2 = G_2 + R_2 \cdot P_2$$

Expressions des sorties S_0, S_1, S_2 et de R_2 en fonction des G_i et P_i .

$$S_0 = P_0$$

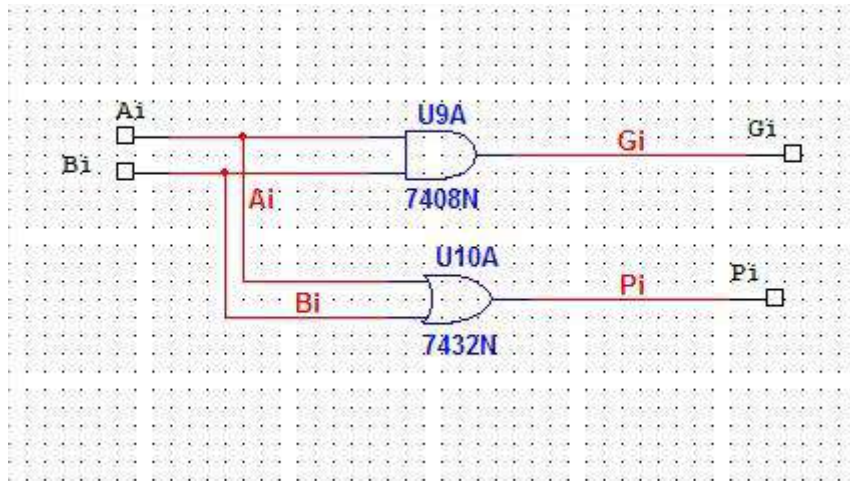
$$S_1 = (R_0 \oplus a_1 \oplus b_1)$$

$$S_2 = (R_1 \oplus a_2 \oplus b_2)$$

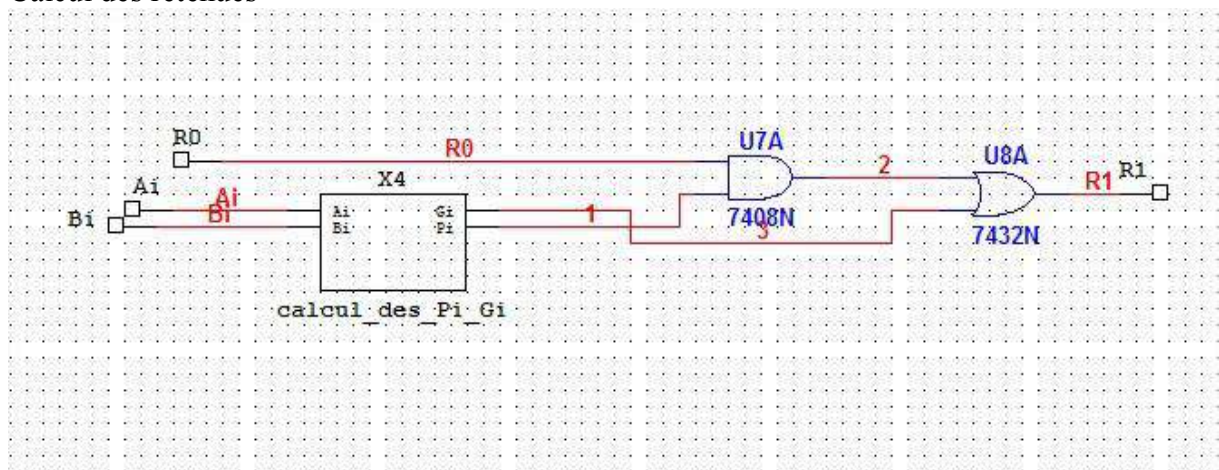
$$S_3 = (R_2 \oplus a_3 \oplus b_3)$$

Réalisation du circuit sous Multisim

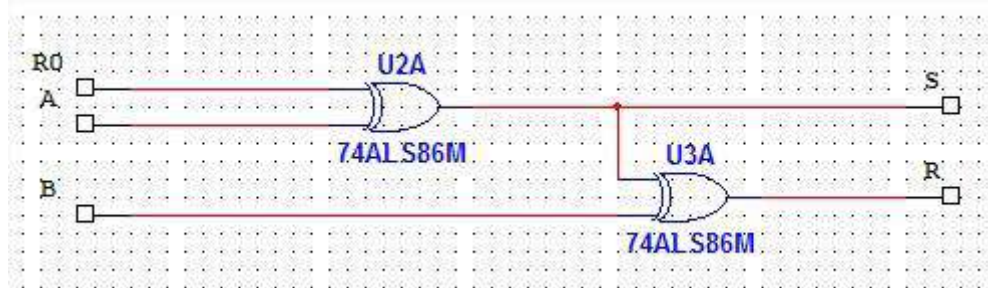
Calcul des Gi et Pi :



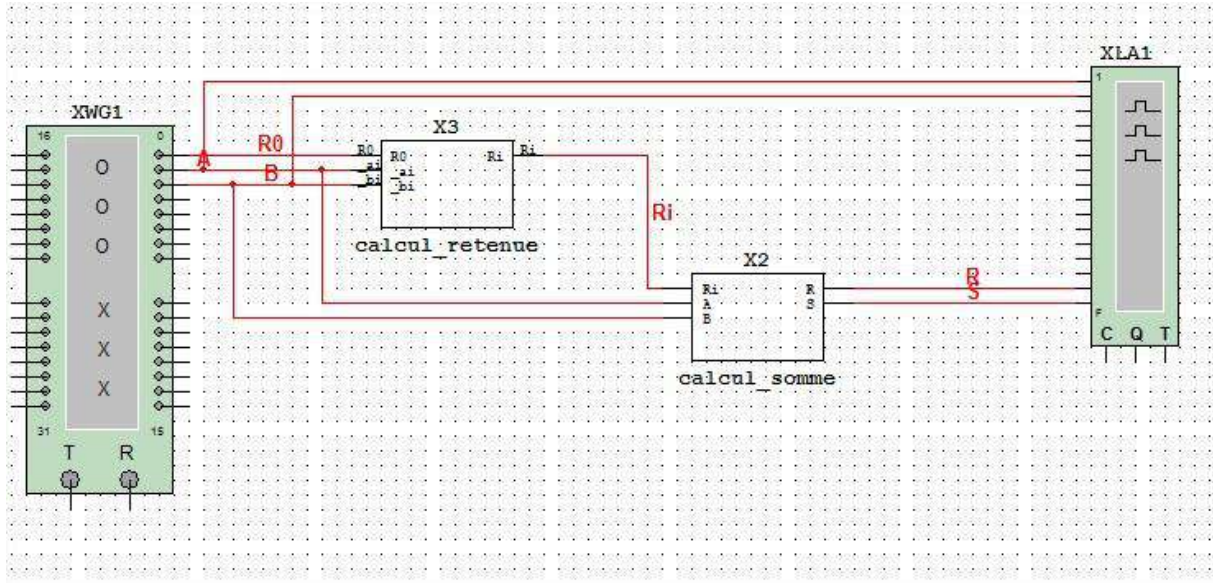
Calcul des retenues



Calcul des sommes



Circuit entier :



1.5) Comparaison des deux méthodes

1) Additionneur classique

Demi-Additionneur

Dans un demi additionneur nous n'avons qu'un passage de porte pour chaque sortie donc $T = 10\text{ns}$.

Additionneur complet

Dans un additionneur complet le nombre maximum de porte pour une sortie est de 3 donc $T = 3 * 10 = 30\text{ns}$

Additionneur de mot de 4bits

Un additionneur de mot de 4bits est composé de 3 additionneurs complet et d'un demi-additionneur donc $T = 3 * 30 + 10 = 100\text{ns}$

Additionneur de mot de 8bits

Un additionneur de mot de 8bits est composé de 7 additionneurs complet et d'un demi-additionneur donc $T = 7 * 30 + 10 = 220\text{ns}$

2) Additionneur à anticipation de retenue

Calcul de G_i , P_i , R_i et S_i

$G_i = A_i \bullet B_i$ $P_i = A_i + B_i$ Le temps de propagation pour G_i et pour P_i n'est donc que de 10ns (une porte)

Additionneur de mot de 4bits avec anticipation

Il y a ici 4 portes ET en parallèles puis une porte OU, nous avons donc $T = 20\text{ns}$ (2portes)

Additionneur de mot de 8bits avec anticipation

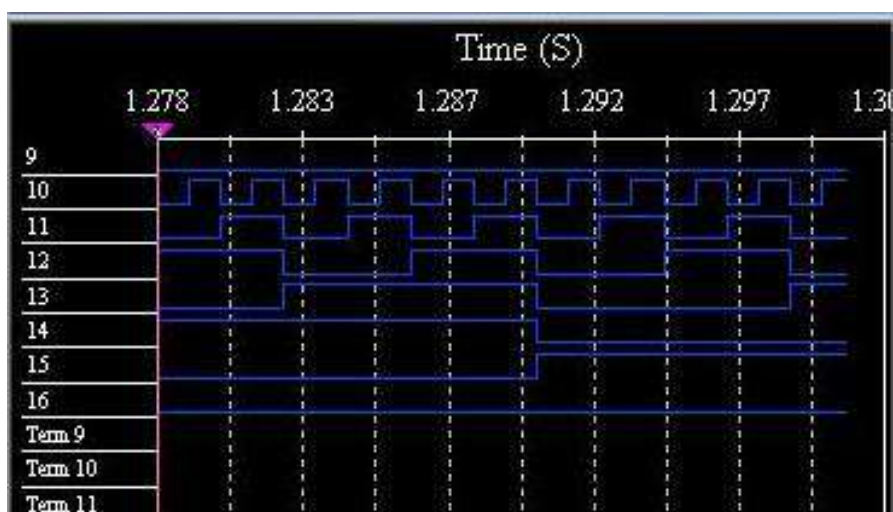
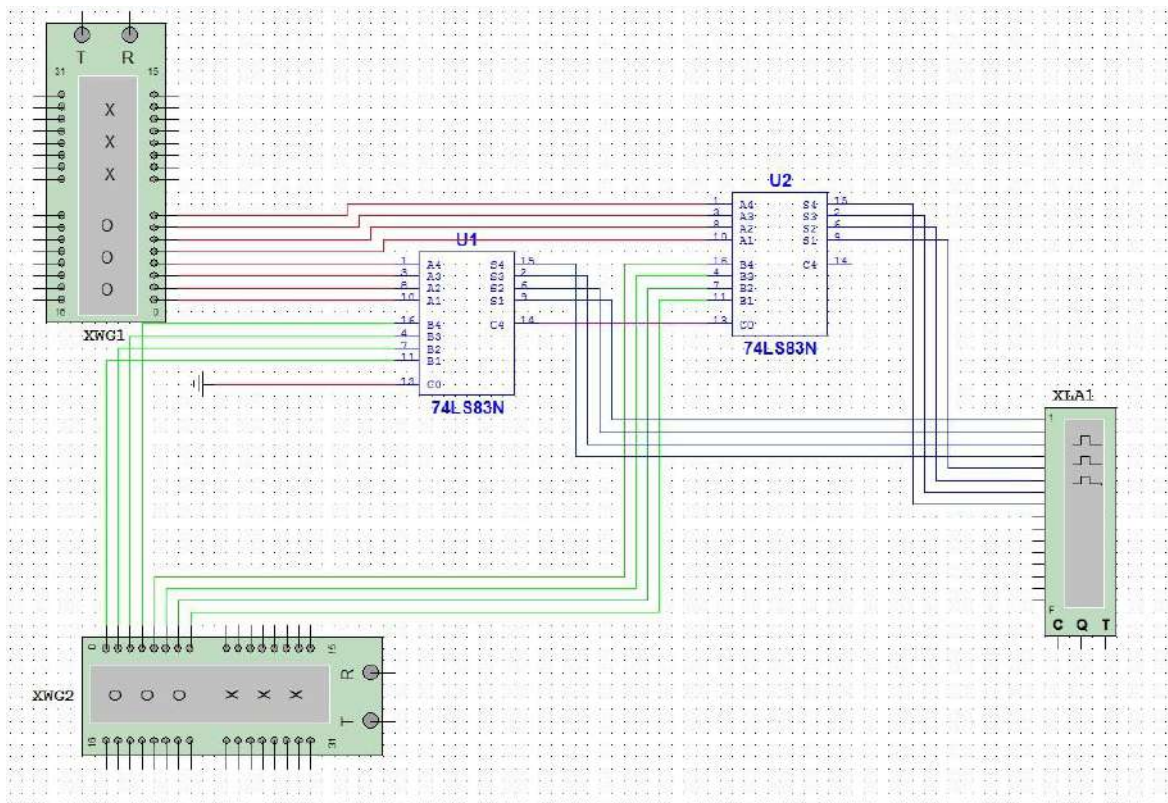
Pour obtenir un additionneur de mot de 8bits nous avons 2 additionneurs de mots de 4bits travaillant en parallèle puis une porte ET. Nous avons donc $T = 30\text{ns}$ (3portes)

3) Tableau comparatif

Additionneur à retenue anticipée		Additionneur à propagation de retenue	
Avantage	Défaut	Avantage	Défaut
- Temps de propagation rapide.	- Schéma complexe - Besoin de place	- Structures répétitives donc possibilités de mettre des sous-circuits (schéma simple) - Taille réduite	- Temps de propagation lent (il dépend des mots d'entrées)

1.6) Réalisation d'un additionneur de 8 bits

Le 74LS83N est additionneur complet d'un mot de 2 mots de 4bits, il prend donc en entrée les 2 mots (de façon décroissante, le bit de poids le plus fort est en haut) et éventuellement une retenue, fais la somme de ceci et retourne ce mot « somme » associé à sa retenue éventuelle.



2) Réalisation d'additionneur soustracteur

2.1 Réalisation d'un demi soustracteur de deux éléments binaires

Si=B-A

A	B	R	S
0	0	0	0
0	1	0	1
1	0	1	1
1	1	0	0

Après lecture sur les 1 et simplification nous obtenons les équations de R et S suivantes :

$$R = A \cdot \bar{B} \quad \text{et} \quad S = \bar{A} \cdot B + A \cdot \bar{B} = A \oplus B$$

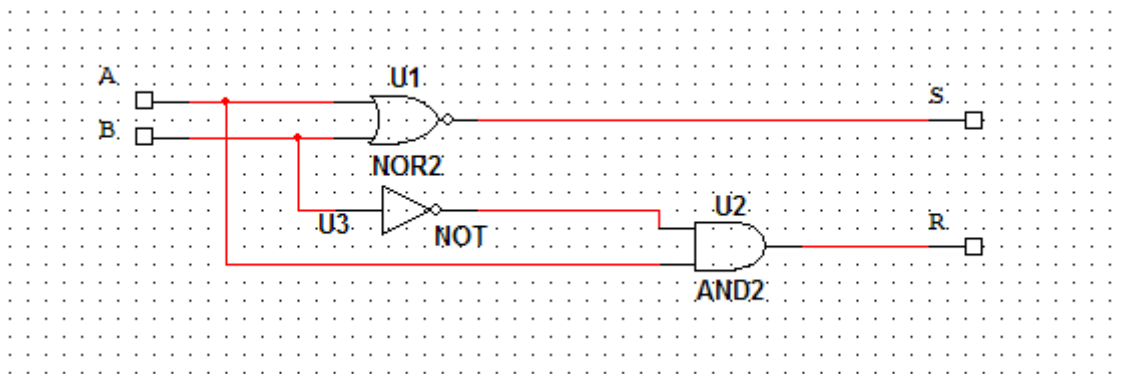


Schéma logique du demi-soustracteur

2.2 Réalisation d'un soustracteur complet

Table de vérité

A	B	R _{n-1}	R	S
0	0	0	0	0
0	0	1	1	1
0	1	0	0	1
0	1	1	0	0
1	0	0	1	1
1	0	1	1	0
1	1	0	0	0
1	1	1	1	1

Méthode de Karnaugh

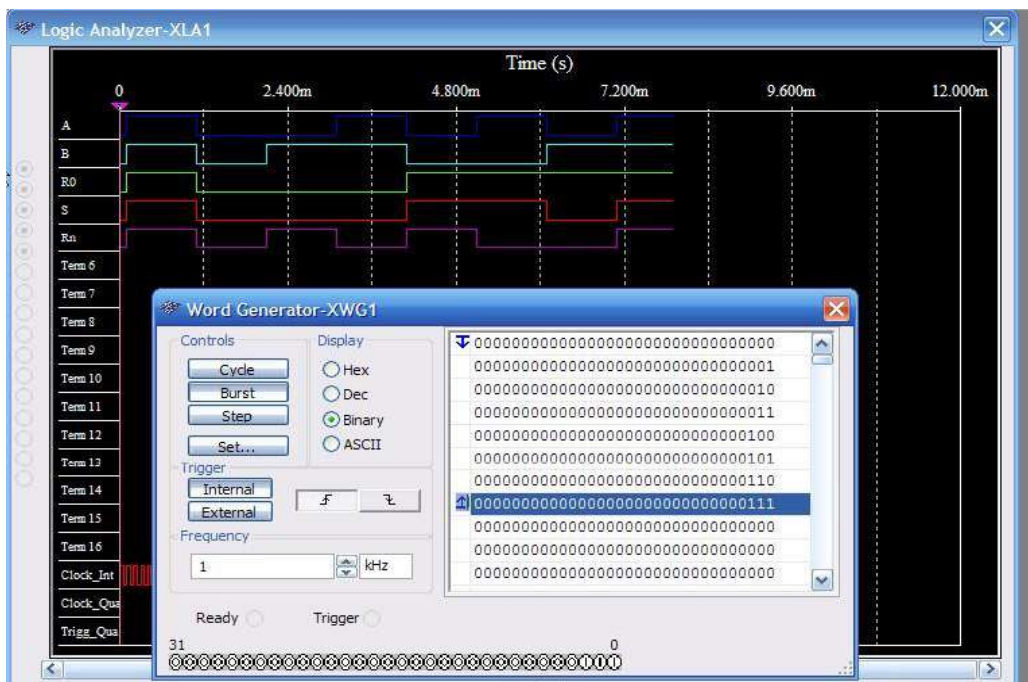
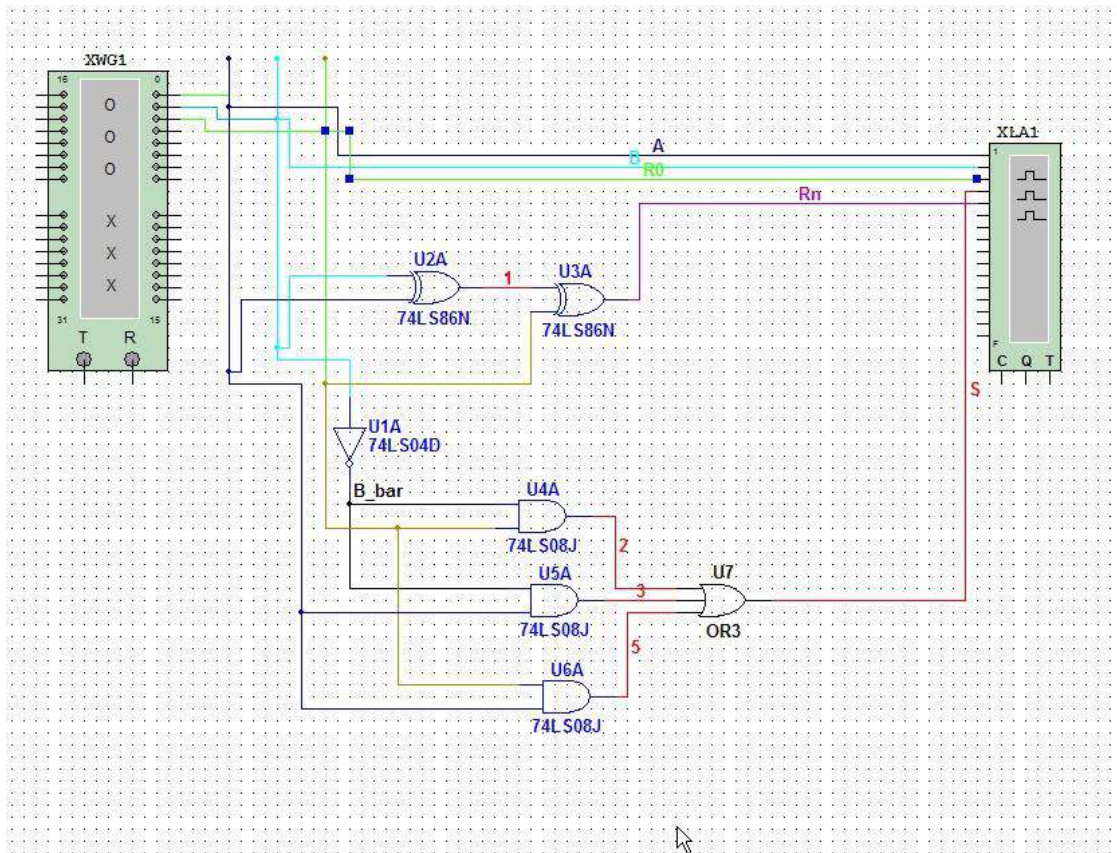
Equation de S

AB / R _{n-1}	0	1
00	0	1
01	0	0
11	0	1
10	1	1

$$S = A \cdot \bar{B} + \bar{B} \cdot R_{n-1} + \bar{A} \cdot B \cdot \bar{R}_{n-1}$$

$$S = \bar{B}(A + R_{n-1}) + B(\bar{A} \cdot \bar{R}_{n-1})$$

Réalisation sous Multisim



Equation de Rn

AB	/	Rn-1	0	1
00			0	1
01			1	0
11			0	1
10			1	0

$$R = A \cdot \bar{B} + A \cdot R_{n-1} + \bar{A} \cdot \bar{B} \cdot R_{n-1}$$

$$R = \bar{B}(A + \bar{A} + R_{n-1}) + A \cdot R_{n-1}$$

$$R = \bar{B} \cdot R_{n-1} + A \cdot R_{n-1}$$

$$R = R_{n-1}(A + \bar{B})$$

2.3 Réalisation d'un additionneur-soustracteur complet

Table de vérité

Ci	Ai	Bi	Si	Ci+1
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

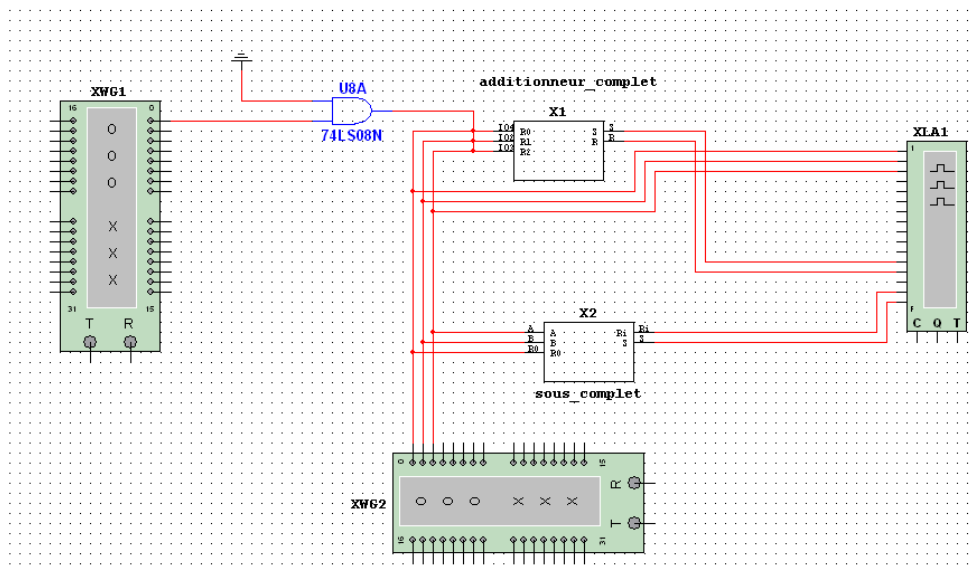
Ai et Bi sont les deux bits d'entrée, Ci est l'élément binaire qui détermine si le calcul est une addition (C égal à 0) ou si c'est une soustraction (C égal à 1). Ci est la retenue initiale et Ci+1 est la retenue générée par le calcul. Si est le bit de sortie.

$$S_i = C_i \oplus A_i \oplus B_i$$

$$C_{i+1} = A_i B_i + C_i (A_i \oplus B_i)$$

Réalisation sous Multisim

Nous remplaçons le soustracteur complet précédent par le sous-circuit « sous_complet » et nous réutilisons le sous-circuit « additionneur_complet » réalisé lors de la séance 1.



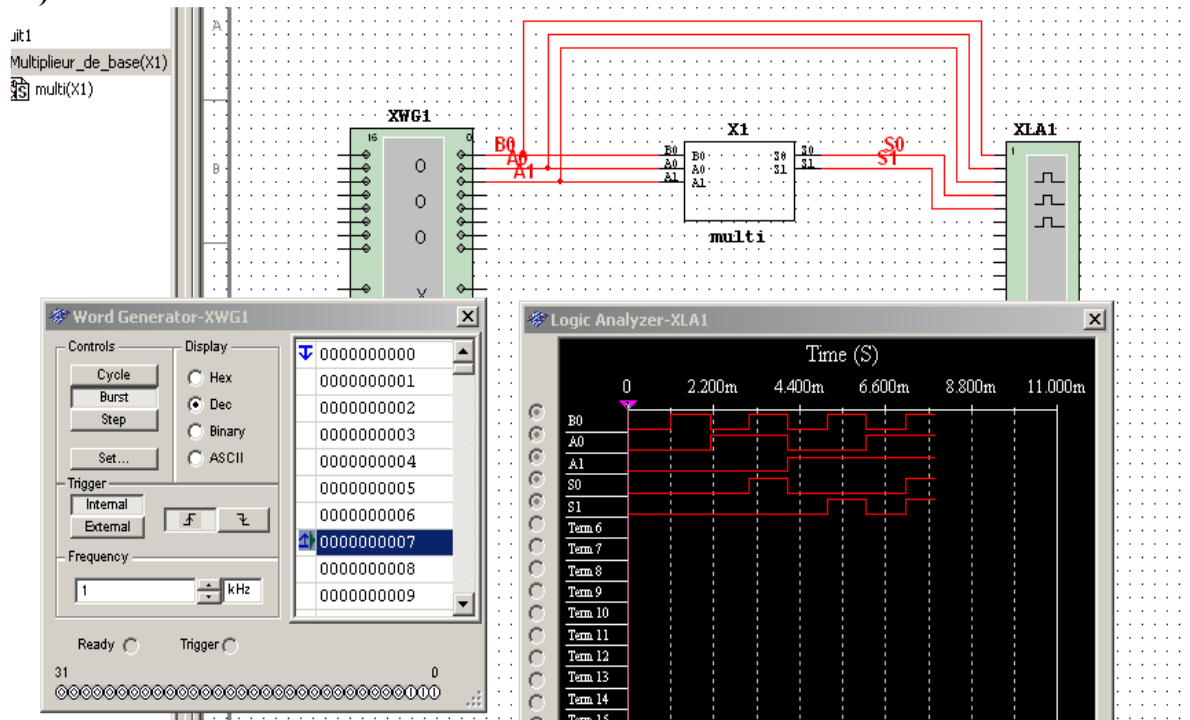
2.4 Réalisation d'un additionneur-soustracteur de deux mots binaires de 4 bits.

Nous réutilisons l'additionneur de 4 bits vu précédemment qui remplace le sous-circuit « additionneur_complet » et nous remplaçons le soustracteur complet par un soustracteur de 2 mots de 4bits.

Réalisation de multiplieurs

1) Multiplieur de mots de deux bits par des mots de deux bits

1.1)

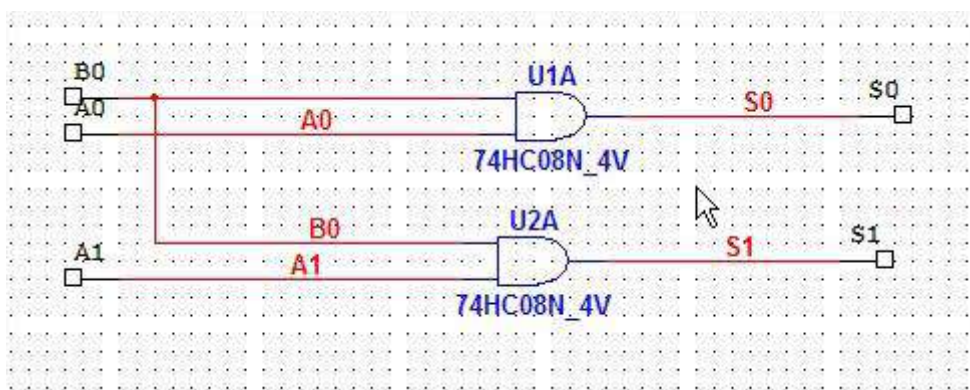


On remarque que

$$S0 = A0.B0$$

$$S1 = A1.B0$$

Voici le schéma du sous-circuit

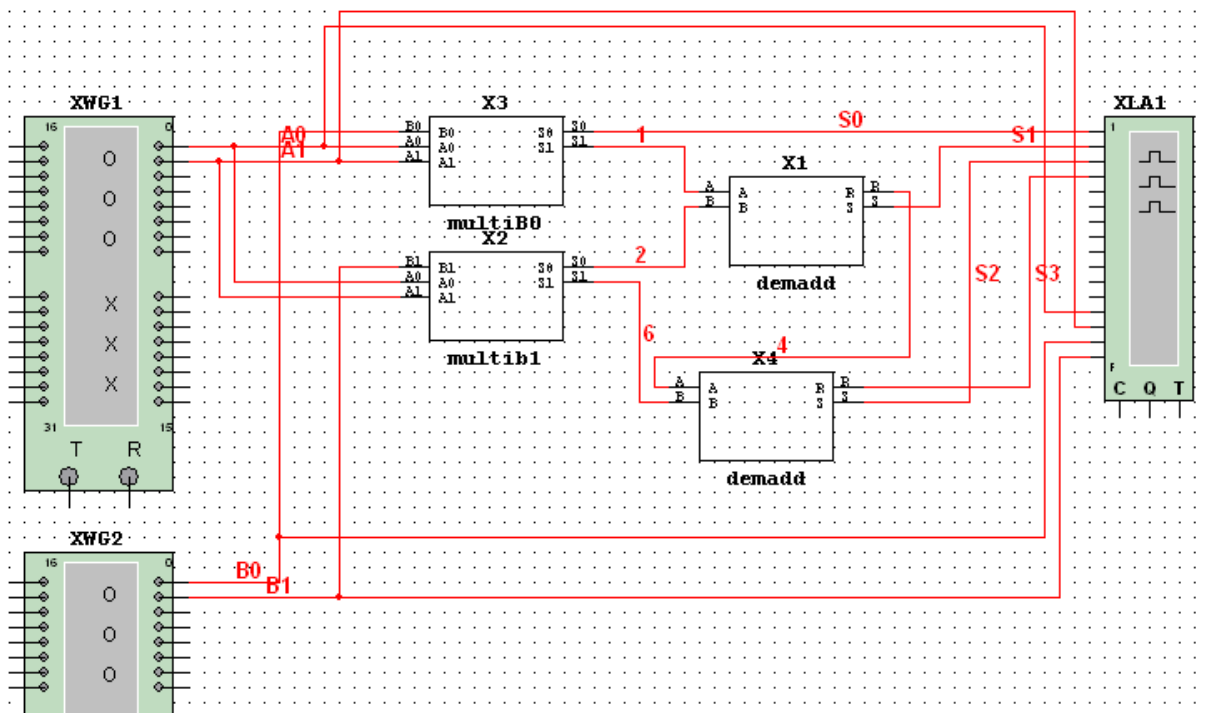


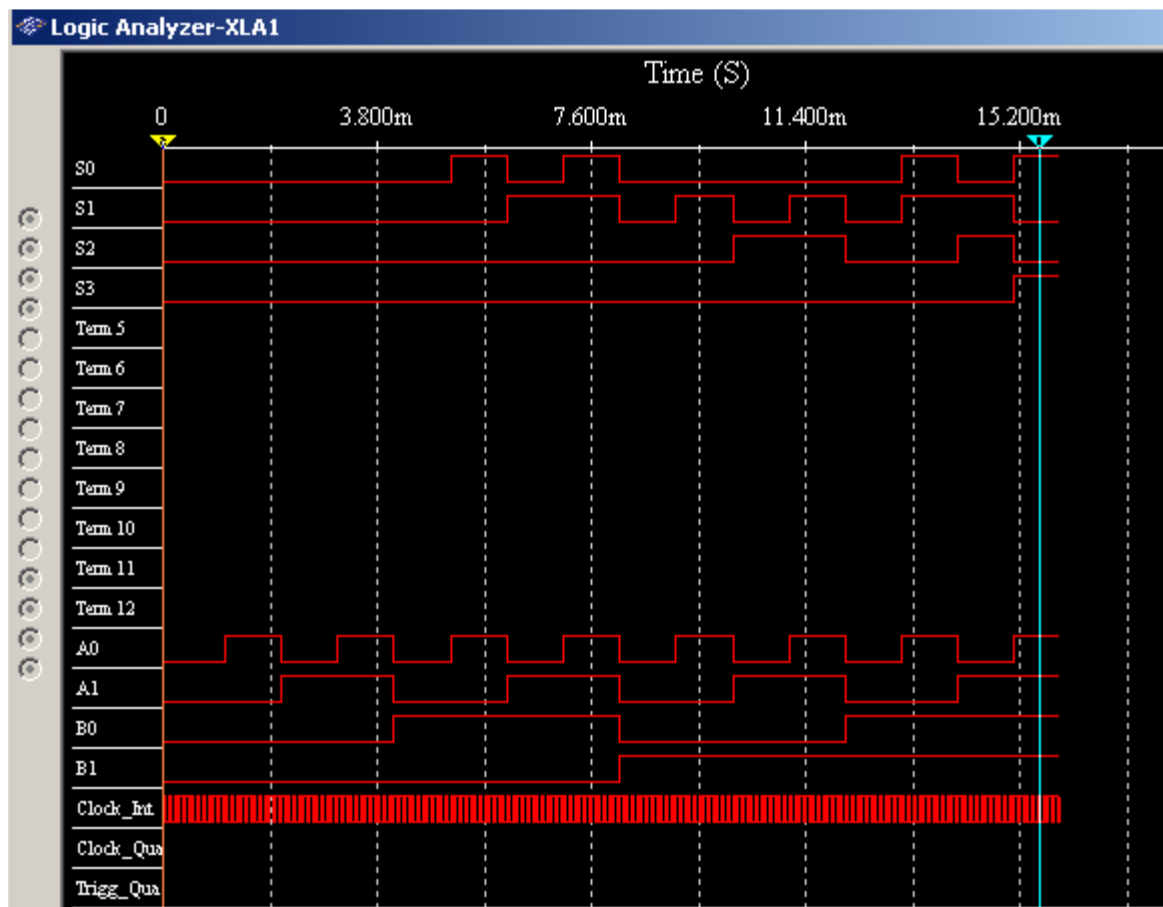
1,2)

Pour une multiplication de 2 mots de 2 bits nous avons :

- Une 1^{ère} multiplication : $B0 * (A1.A0)$
- Une seconde multiplication : $B1 * (A1.A0)$
- L'addition du bit de poids le plus fort de la 1^{ère} multiplication et du bit de poids le plus faible de la 2nde multiplication
- L'addition du bit de poids le plus fort de la seconde multiplication et de la retenue de la 1^{ère} addition

1.3) Schéma sous MultiSim



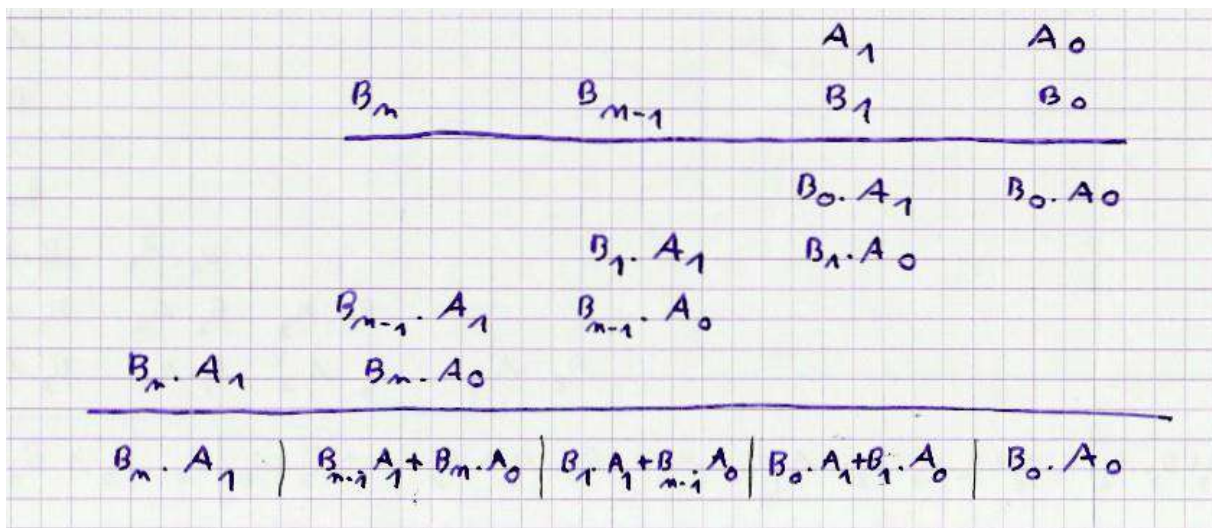


2 Multiplieur de mots de deux bits par des mots de n bits (nombres non signés)

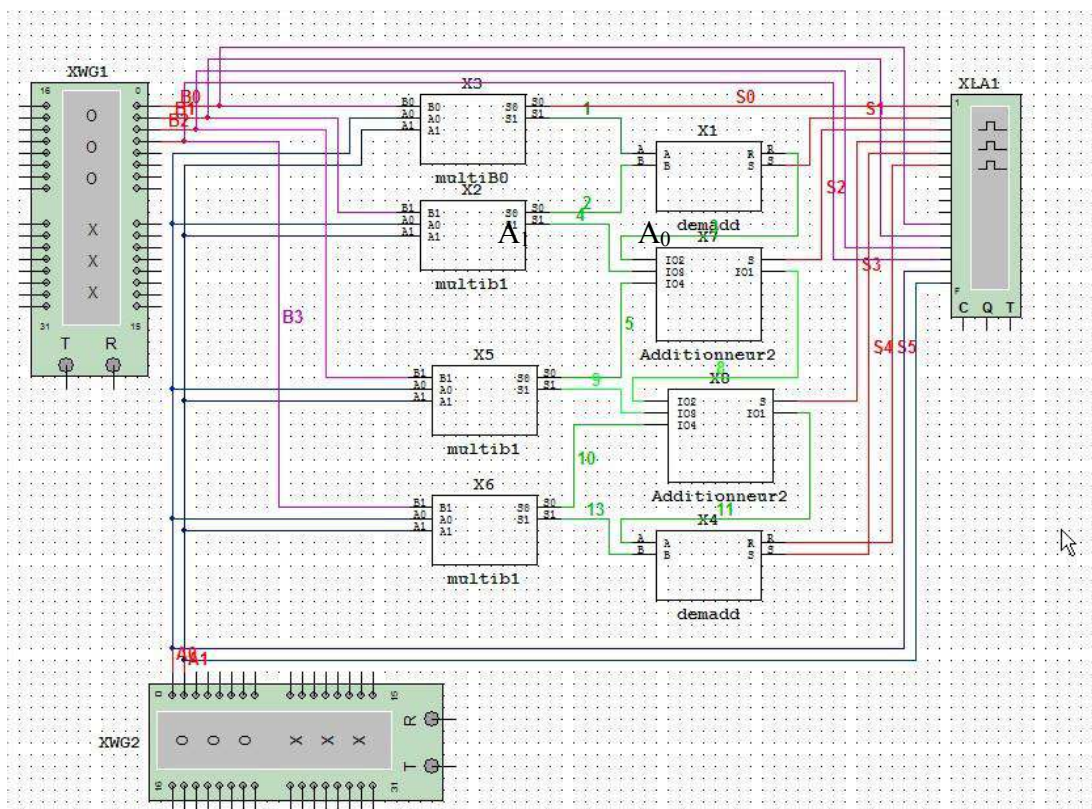
2.1) Pour multiplier un mot de 2 bits et un mot de n bits nous avons simplement à reprendre l'algorithme précédent, il y aura n multiplications du type $B_n \cdot (A_0.A_1)$ et n-1 additions.

- Nous utiliserons le circuit M afin de réaliser les multiplications (du type $B_n \cdot (A_0.A_1)$)
- Nous utiliserons des demi additionneurs afin de faire la somme du bit de poids le plus fort de la multiplication précédente et du bit de poids le plus faible de la nouvelle multiplication

Multiplication d'un mot de 2 bits (A) par un mot de n bits (B).



2.2) Schéma du circuit réalisé sous multisim



Résultat de la simulation :



3) Multiplieur de deux mots de n bits (nombres non signés)

3.1) Multiplication de 2mots de 4bits :

A_3	A_2	A_1	A_0
B_3	B_2	B_1	B_0

$$\begin{array}{r}
 B_0 \cdot A_3 \quad B_0 \cdot A_2 \quad B_0 \cdot A_1 \quad B_0 \cdot A_0 \\
 B_1 \cdot A_3 \quad B_1 \cdot A_2 \quad B_1 \cdot A_1 \quad B_1 \cdot A_0 \\
 B_2 \cdot A_3 \quad B_2 \cdot A_2 \quad B_2 \cdot A_1 \quad B_2 \cdot A_0 \\
 B_3 \cdot A_3 \quad B_3 \cdot A_2 \quad B_3 \cdot A_1 \quad B_3 \cdot A_0 \\
 \hline
 B_2 \cdot A_3 + B_3 \cdot A_2 \quad B_1 \cdot A_3 + B_2 \cdot A_2 + B_3 \cdot A_1 \quad B_0 \cdot A_3 + B_1 \cdot A_2 + B_2 \cdot A_1 + B_3 \cdot A_0 \quad B_0 \cdot A_2 + B_1 \cdot A_1 + B_2 \cdot A_0 \quad B_0 \cdot A_1 + B_1 \cdot A_0 \quad B_0 \cdot A_0
 \end{array}$$