

Projet d'information numérique

ADDITIONNEUR

ARTHUR GAUTIER ET KAREN HAYOUN

Table des matières

Introduction	2
I. L'additionneur à propagation de retenue	2
II. L'additionneur à anticipation de retenue.....	8
Conclusion	11

Introduction

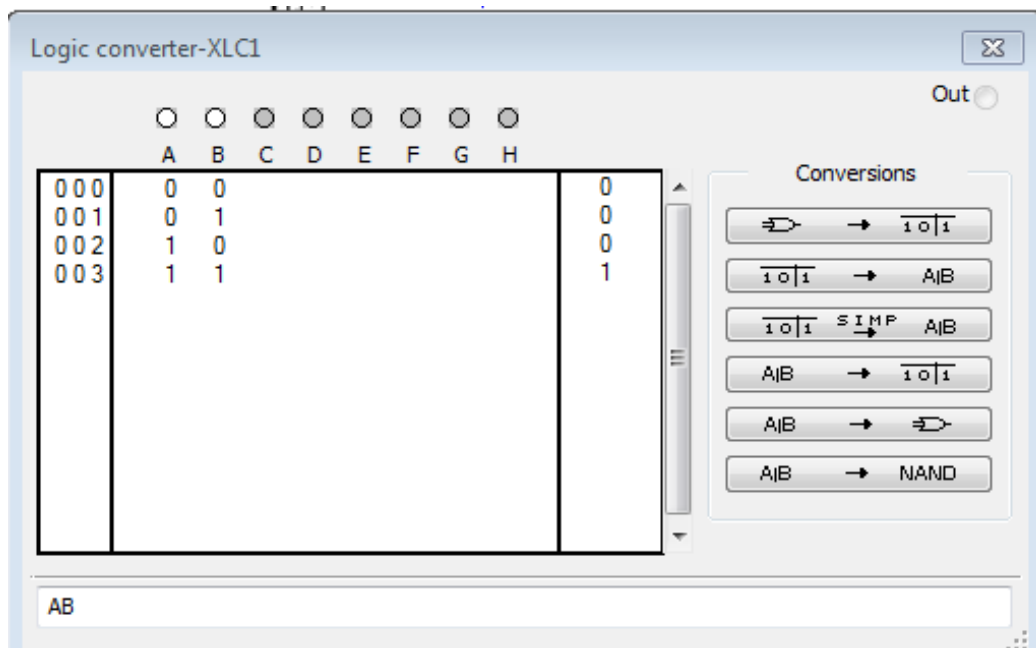
Dès la découverte des nombres, l'homme a essayé de les manipuler pour simplifier sa vie quotidienne. Depuis l'arrivée des machines, ces manipulations ont été largement simplifiées. Toutefois, nous sommes en droit de nous demander : « De quelle façon les machines réalisent-elles ces opérations ? ».

Ici, nous étudierons le fonctionnement d'une addition dans l'ordinateur. On peut se rendre compte que la machine a deux façons de réaliser ces additions : la propagation de retenue et l'anticipation de retenue.

I. L'additionneur à propagation de retenue

Dans une addition, il arrive que la somme des nombres dépasse la valeur unitaire de la base dans laquelle on fait l'addition. Dans ce cas, nous nous retrouverons en présence d'une retenue qui affectera la somme suivante, qui est la somme immédiatement à gauche de la somme juste effectuée.

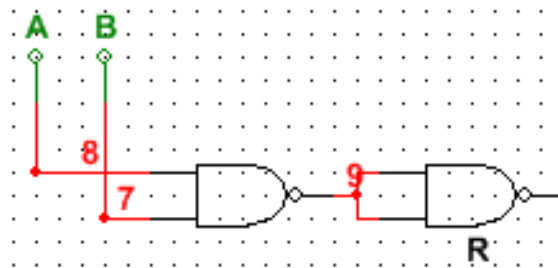
Dans le cas d'une addition en base 2, on peut se demander à quel moment une retenue apparaît.



	A	B	C	D	E	F	G	H	Out
000	0	0							0
001	0	1							0
002	1	0							0
003	1	1							1

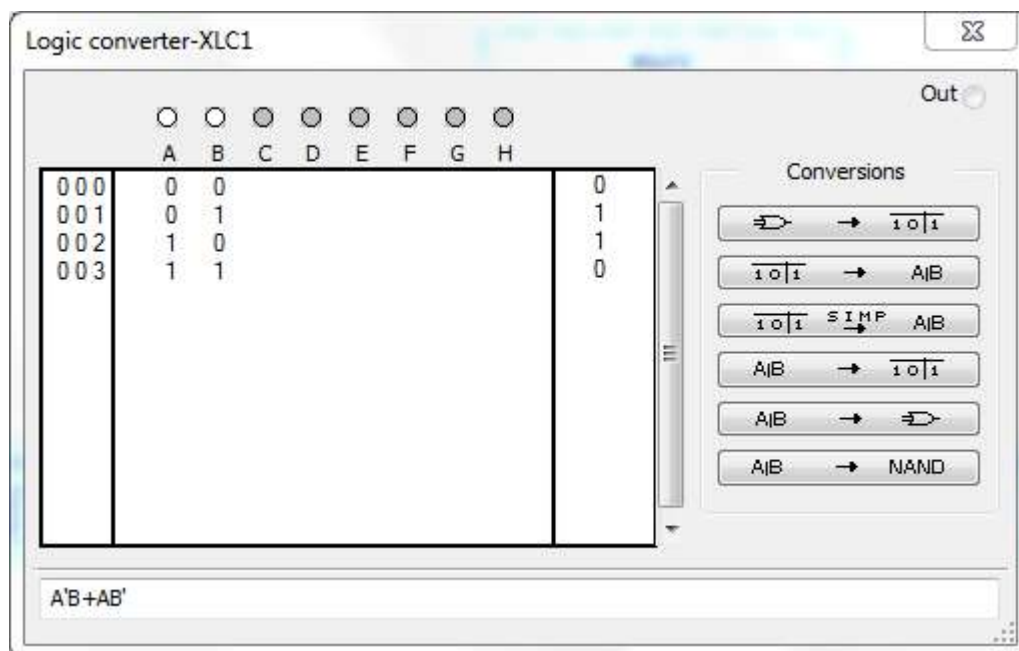
On remarque alors qu'il y a apparition de retenue lorsque les deux opérandes valent 1. Cela correspond à la table de vérité d'une porte ET.

Le circuit correspondant est le suivant :

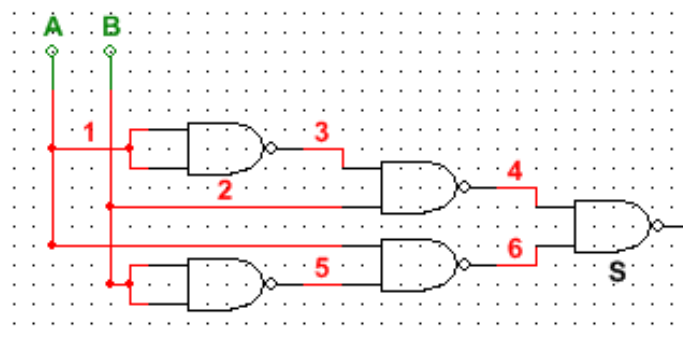


A la fin de ce circuit, nous obtiendrons la retenue R du demi-additionneur.

Pour la somme, on peut se poser la même question.

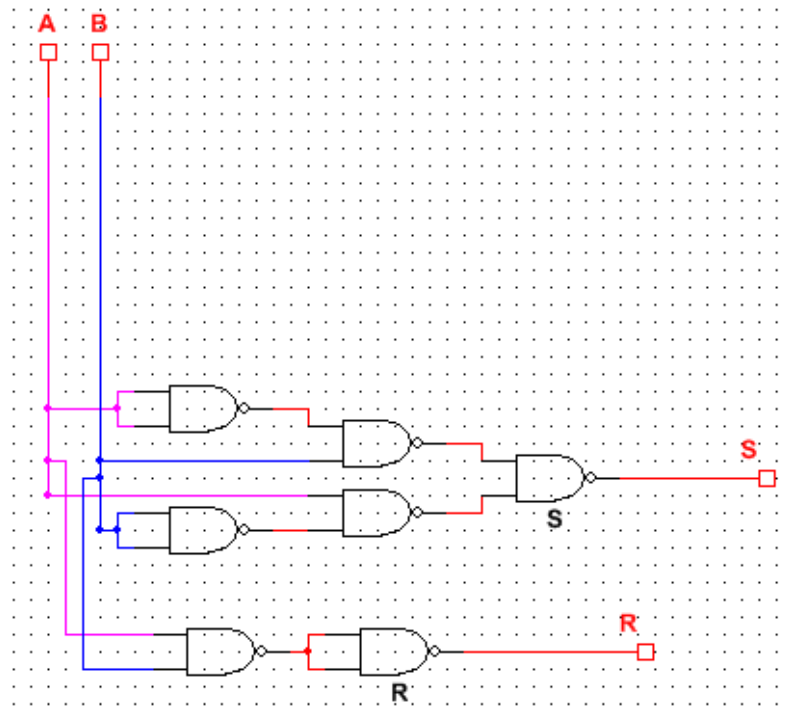


La somme vaut 1 lorsque les deux opérandes n'ont pas les mêmes valeurs. C'est la table de vérité d'une porte XOR (également appelé OU exclusif). On obtient alors le circuit suivant :

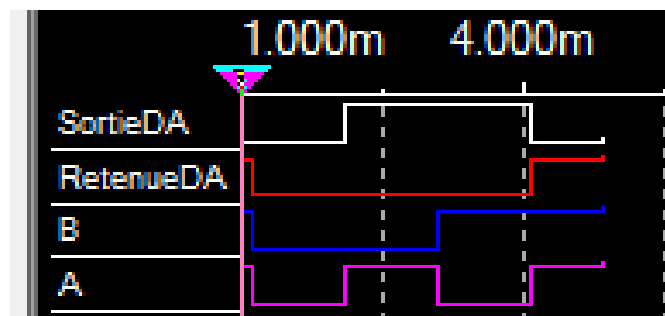


Ce circuit permet d'obtenir la somme du demi-additionneur.

Ces deux circuits réunis forment un demi-additionneur :



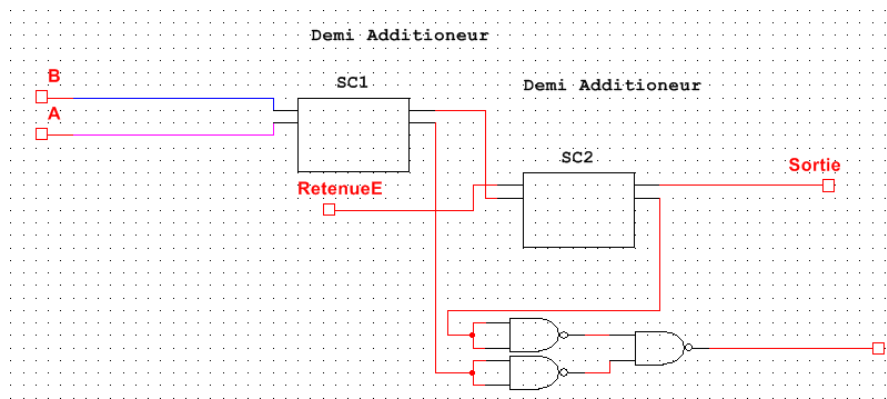
Ce circuit permet d'obtenir les courbes suivantes.



On voit ainsi que lorsque les deux entrées valent 0, il n'y a pas de retenue et la somme vaut 0. Si l'un des entrées vaut 1, il n'y a toujours pas de retenue mais la somme vaut alors 1. Enfin, si les deux entrées valent 1, on voit l'apparition d'une retenue mais la somme vaut alors 0.

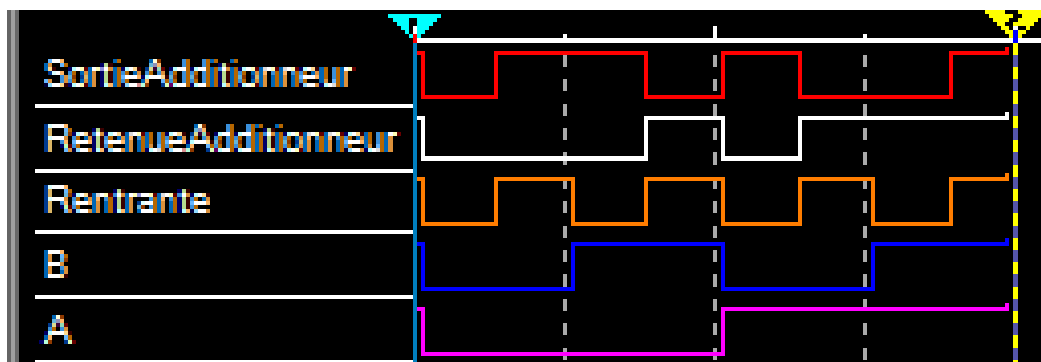
Pour construire un additionneur complet, il est nécessaire de réunir deux demis-additionneurs à l'aide d'une porte OU.

Ainsi, le circuit d'un additionneur est le circuit suivant :



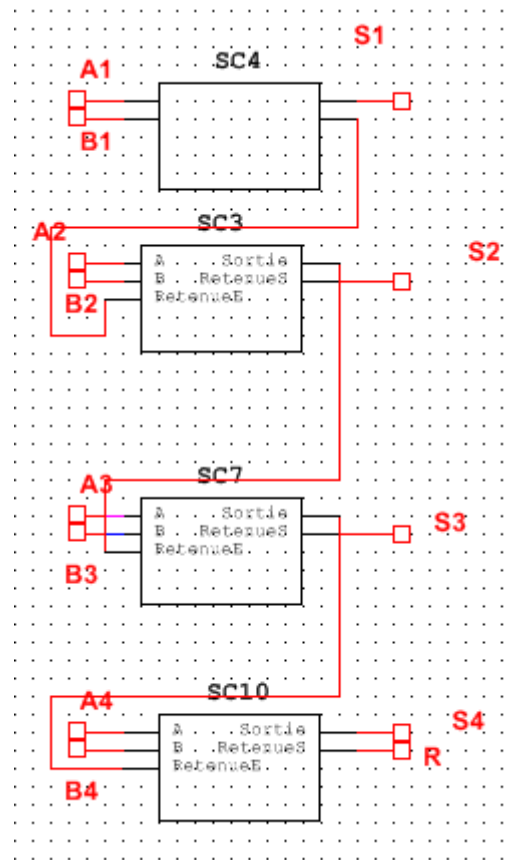
Le circuit du premier demi-additionneur a déjà été présenté. Dans le second, on remplace l'entrée des mots binaires A et B par la somme précédent et une retenue entrante (la retenue provenant de la somme précédente). La somme de ce second demi-additionneur correspond à la somme finale. La retenue finale est donnée par la porte OU située en bas du circuit. Ainsi, si jamais au moins l'une des retenues vaut 1, la retenue finale vaudra 1.

Ce circuit nous donne les courbes suivantes :



On peut ainsi voir que la retenue vaut 1 si deux entrées ou plus valent 1 et que la sortie vaut 1 quand un nombre impair d'entrées est à 1.

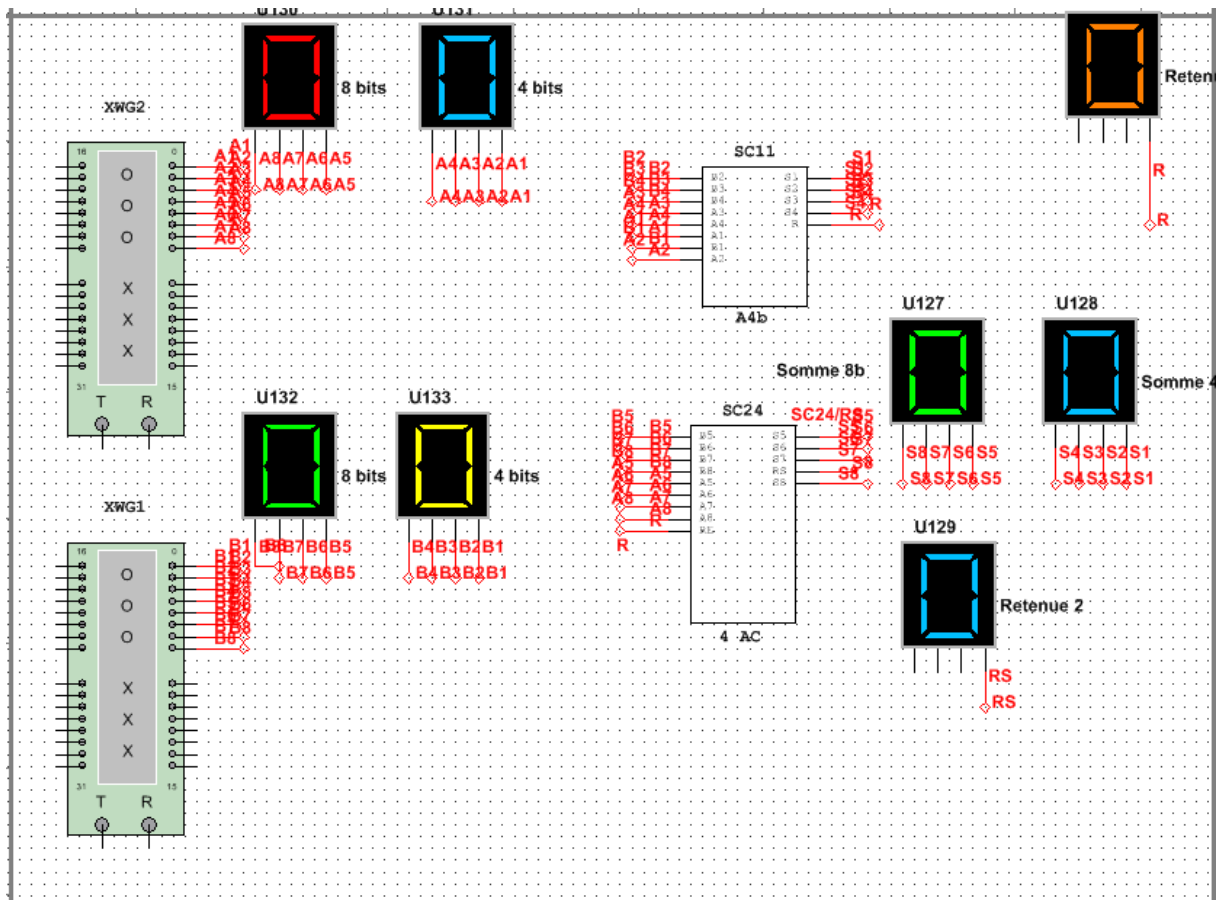
A l'aide de 3 de ces additionneurs et d'un demi-additionneur, on peut réaliser un additionneur 4 bits avec propagation de retenue. Le circuit de cet additionneur est le suivant :



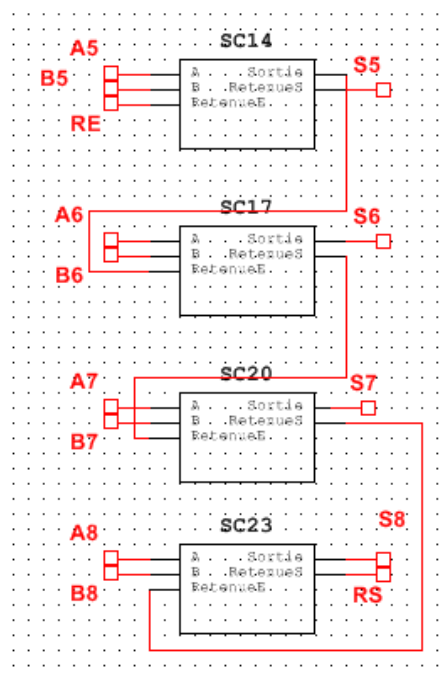
Les $A_i B_i$ représentent les entrées, les S_i représentent les sommes et R représente la dernière retenue. Ainsi, pour ce circuit, on obtiendra un résultat de la forme : R $S_4 S_3 S_2 S_1$.

A l'aide de cet additionneur complet de 4 bits et de 4 additionneurs complets, on peut réaliser un additionneur qui pourra réaliser l'addition de deux mots de 8 bits.

Le circuit sera le suivant :



Le premier sous-circuit (le A4b) a été détaillé avant. Voilà le circuit du second :



Ce circuit est composé de 4 additionneurs complets en série.

Le temps de calcul pour l'additionneur à propagation de retenue vaut :

$$T = (\text{nombre de bits du mot}) * (\text{temps de calcul / étage}).$$

On travaille avec des mots de 4 bits. Et on a, pour chaque étage, 3 passages de portes. Or, on nous dit que les portes travaillent en 10 ns. On a donc un temps de calcul / étage qui vaut : $3 * 10 = 30$ ns

Le temps total pour un additionneur de 4 bits à propagation de retenue est donc :

$$T_{4\text{bits}} = 4 * 30 = 120 \text{ ns.}$$

Pour un additionneur 8 bits, on passe sur des mots de 8 bits, on a ainsi :

$$T_{8\text{bits}} = 8 * 30 = 240 \text{ ns.}$$

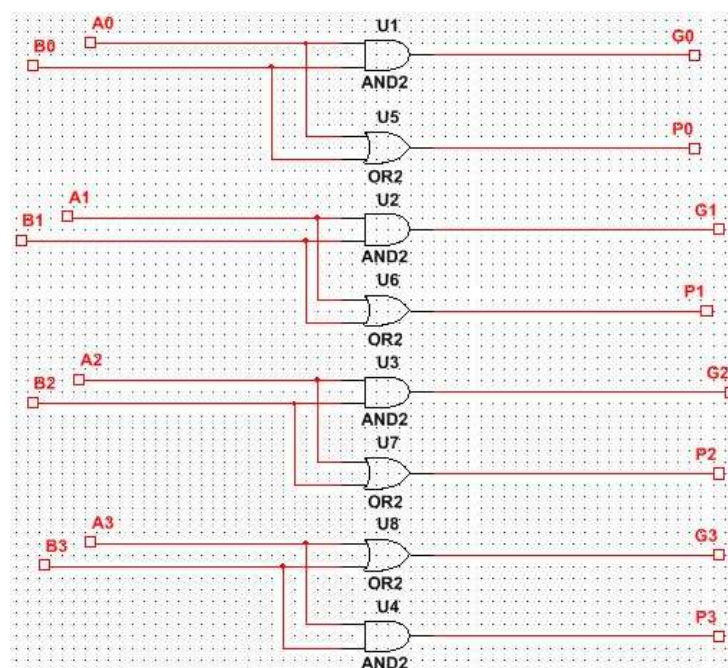
Désormais, passons au second type d'additionneur, l'additionneur à anticipation de retenues.

II. L'additionneur à anticipation de retenue

Ce second type d'additionneur fonctionne d'une façon différente. En effet, là où dans le premier additionneur, les sommes partielles dépendaient des retenues précédentes, ici, les sommes partielles ne dépendent que de la première retenue (très généralement nulle) et de la retenue associée à la somme.

Cet additionneur calcule en trois temps. Tout d'abord, il réalise le calcul des termes de propagation et de génération respectivement appelés P_i et G_i . Le terme de propagation vaut $P_i = A_i \text{ OU } B_i$ et le terme de génération vaut $G_i = A_i \text{ ET } B_i$.

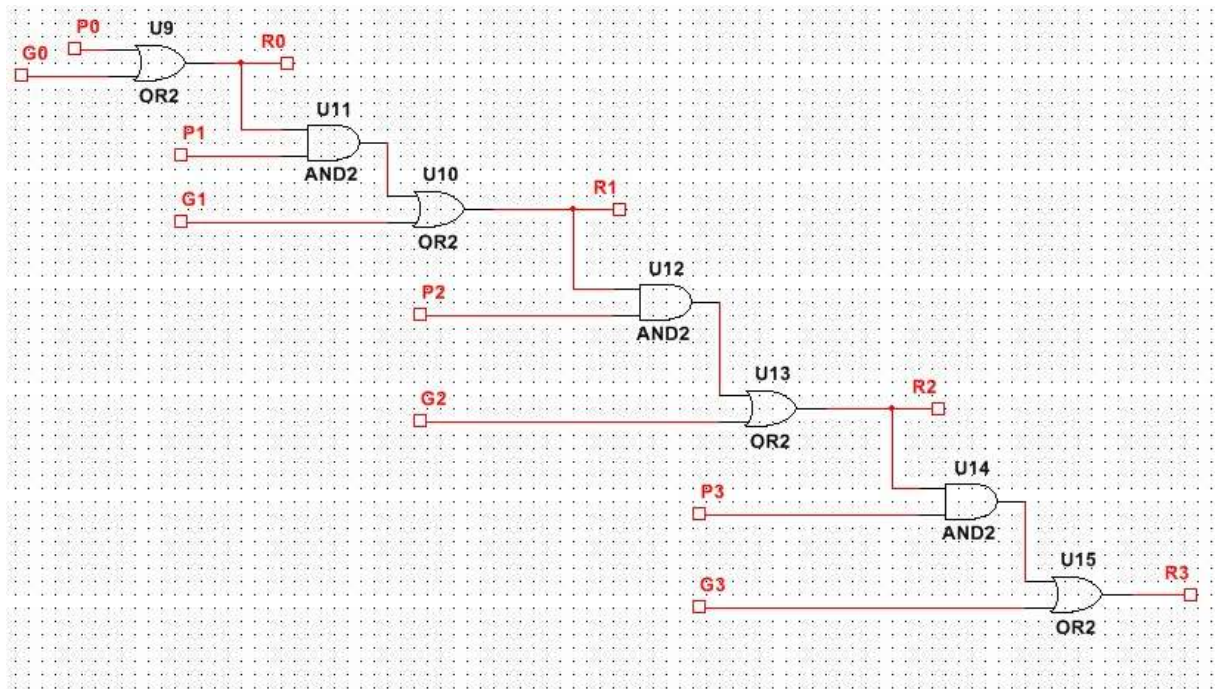
Leur calcul se réalise à l'aide du circuit suivant :



On remarque que ces termes sont calculés en parallèle et non en série.

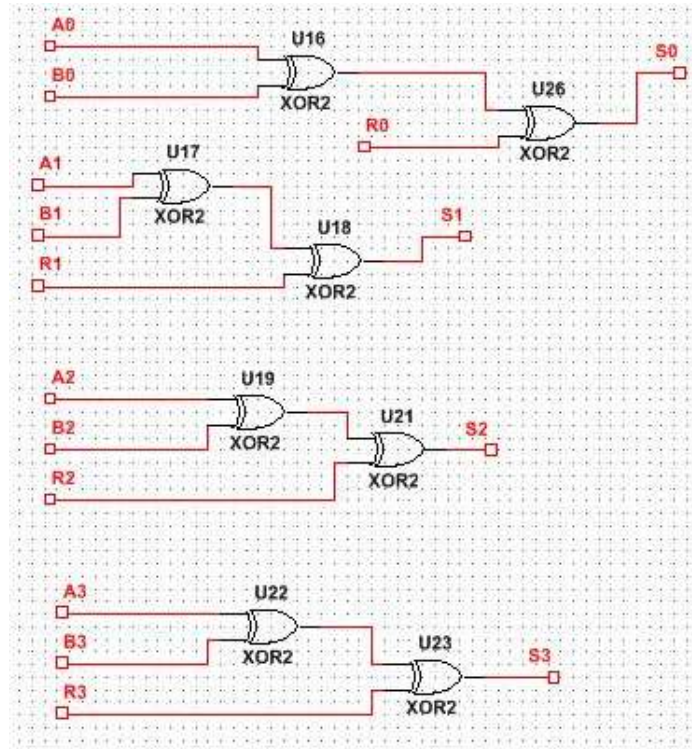
Ensuite, on passe au calcul des retenues R_i . Ces retenues à l'aide des retenues qui les précèdent ainsi que des termes de générations et propagation qui leur sont associés.

On calcule ainsi les retenues à l'aide du circuit suivant :

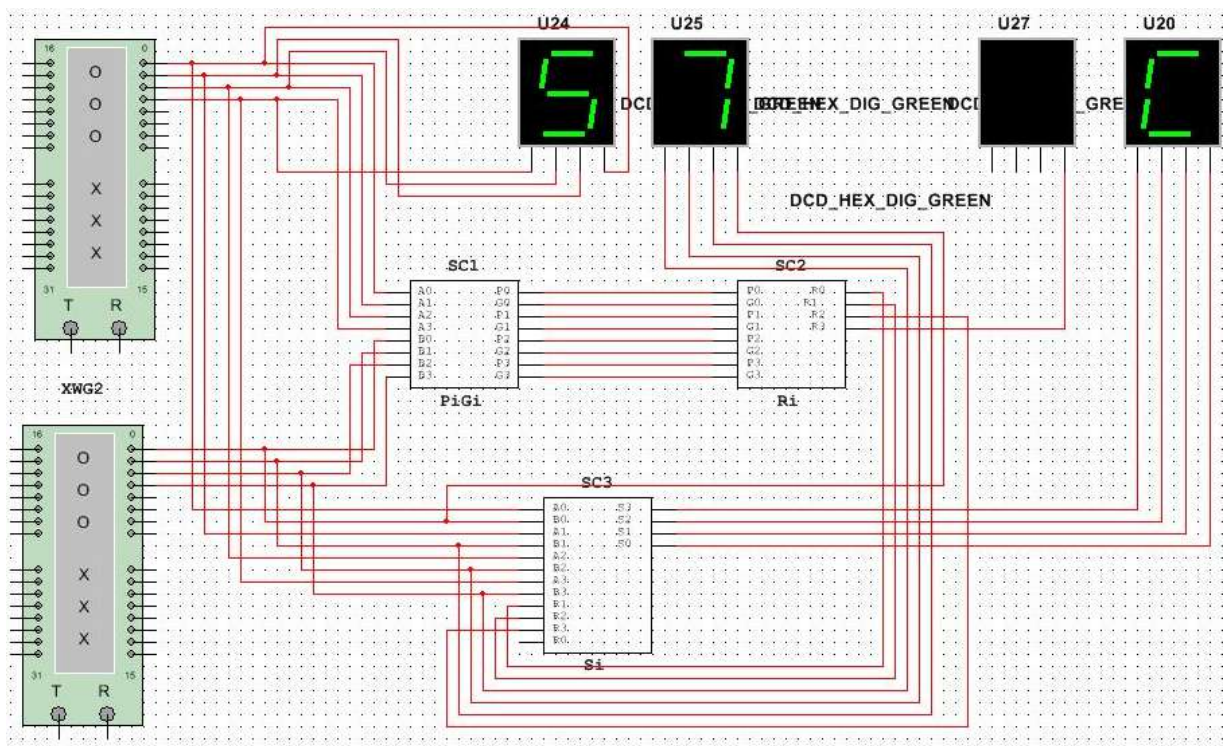


Ce calcul de retenues s'effectue en série mais avant de calculer les sommes partielles.

Passons maintenant à la 3^{ème} étape de l'additionneur, le calcul des sommes partielles S_i . Ces sommes partielles se calculent de la façon suivante : $S_i = A_i \text{ XOR } B_i \text{ XOR } R_i$. Ce calcul correspond au circuit suivant :



En mettant ces trois circuits en série, on obtient un additionneur de deux mots de 4 bits avec anticipation de retenue :



Essayons maintenant de trouver le temps mis par l'additionneur à anticipation de retenue pour un additionneur de 4 bits.

Pour calculer une somme, on passe par 4 portes ET en parallèles puis par une porte ET ayant les 4 sorties précédents comme entrées. On trouve ainsi un temps de travail correspondant au temps de travail de deux portes soit : $T_{4\text{bits}} = 20 \text{ ns}$.

Pour l'additionneur 8 bits (dont le schéma n'est pas ici par manque de temps lors des séances de projet), on utilise deux additionneurs 4 bits en parallèles et dont les sorties passent par une porte ET. Cela fait donc 3 passages de portes soit : $T_{8\text{bits}} = 30 \text{ ns}$.

Nombres de bits	Additionneur à propagation	Additionneur à anticipation
4 bits	120 ns	20 ns
8 bits	240 ns	30 ns

Ainsi, en faisant le rapport anticipation sur propagation on trouve :

(pour l'additionneur 4 bits) : $20 / 120 = 1/6$

(pour l'additionneur 8 bits) : $30 / 240 = 1/8$

Conclusion

On a ainsi mis en évidence que l'additionneur à anticipation de retenue fonctionne bien plus rapidement que l'additionneur à propagation de retenue. En effet, une différence de 10^2 ns sur 4 bits et de plus de $2 \cdot 10^2$ sur 8 bits représente une accélération non négligeable lorsque de nombreux calculs doivent être réalisés. De plus, cette accélération à l'air d'augmenter au fur et à mesure que le nombre de bits augmente. Ainsi, plus on a de bits, plus l'avance de l'additionneur à anticipation sur celui à propagation sera grande.