INSTITUT DE LA PROMOTION SUPERIEURE DU TRAVAIL

ARCHITECTURE DES MACHINES
&
SYSTEMES INFORMATIQUES

LOGIQUE COMBINATOIRE
FONCTIONS LOGIQUES DE BASE

A. M'ZOUGHI

LOGIQUE COMBINATOIRE
FONCTIONS LOGIQUES DE BASE

I. ETUDE DE L'ADDITIONNEUR BINAIRE

L'addition est l'opération la plus fréquemment réalisée dans un calculateur : il faut l'optimiser au maximum. Faire un circuit à 2 étages (ET-OU) d'une addition par exemple, de 2 nombres de 32 bits est probablement possible, mais conduit à un volume de logique important. Heureusement, une autre approche, plus naturelle est possible, calquée sur notre façon de procéder :

	rn-1…	r2	r1	
	an-1…	a2	a1	a0
	bn-1…	b2	b1	b0
	_____	_____	_____	_____
Sn = r_n	Sn-1	S2	S1	S0

il suffit de réaliser n fois la logique suivante (figure 1) :

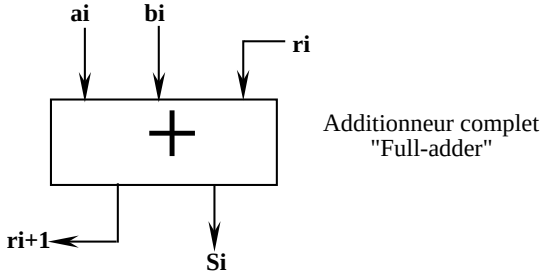


Figure 1 : représentation symbolique d'un additionneur complet

I. 1. Association de demi-additionneurs

Considérons l'addition de 2 bits : a (plus) b, donnant pour résultat : un bit de somme S et un bit de report R.

a	b	a plus b	R	S
0	0	0	0	0
0	1	1	0	1
1	0	1	0	1
1	1	10	1	0

De cette table de vérité, on obtient les équations logiques : $R = a.b$

$S = a\.b + a.b\ = a \oplus b$

Le schéma logique correspondant est représenté sur la figure 2 :

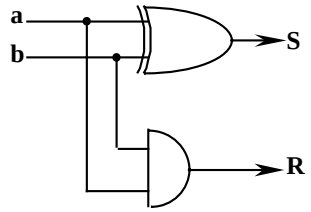


Figure 2 : demi-additionneur

Considérons le cas le plus général où on doit effectuer l'addition de 3 bits a plus b plus r.
r étant le report issu de l'addition des bits du rang précédent (r est dit "report entrant" "Carry In). Le résultat sera : 1 bit de somme S
1 bit de report R (R est dit "report sortant" "Carry Out").

r	**a**	**b**	**r plus a plus b**	**R**	**S**
0	0	0	0	0	0
0	0	1	1	0	1
0	1	0	1	0	1
0	1	1	10	1	0
1	0	0	1	0	1
1	0	1	10	1	0
1	1	0	10	1	0
1	1	1	11	1	1

De cette table de vérité, on obtient les fonctions R et S

R

ab r	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$R = a.b + a\backslash.b.r + a.b\backslash.r = a.b + r.(a \oplus b)$$

S

ab r	00	01	11	10
0	0	1	0	1
1	1	0	1	0

$$S = a\backslash.b\backslash.r + a\backslash.b.r\backslash + a.b\backslash.r\backslash + a.b.r = r\backslash.(a\backslash.b + a.b\backslash) + r.(a\backslash.b\backslash + a.b) = a \oplus b \oplus r$$

Les fonctions R et S peuvent être réalisées en utilisant deux demi-additionneurs et un opérateur OU logique. La figure 3 représente un additionneur complet réalisé à partir de deux demi-additionneurs.

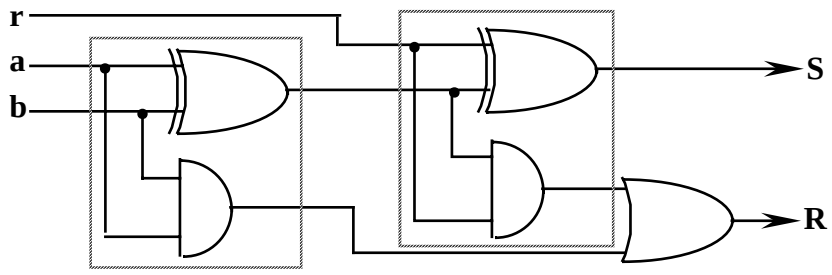


Figure 3 : schéma logique d'un additionneur complet à partir de deux demi-additionneurs

I. 2. Additionneur complet intégré

En réécrivant l'équation du report R à partir de la table de vérité de l'additionneur complet : (a plus b plus r).

R

ab / r	00	01	11	10
0	0	0	1	0
1	0	1	1	1

$$R = a.b + a.r + b.r = a.b + r.(a + b)$$

On a S = a\.b\.r + a\.b.r\ + a.b\.r\ + a.b.r = r\.(a\.b + a.b\) + r.(a\.b\ + a.b) = a $\oplus$ b $\oplus$ r. La figure 4a montre le schéma de l'additionneur complet intégré.

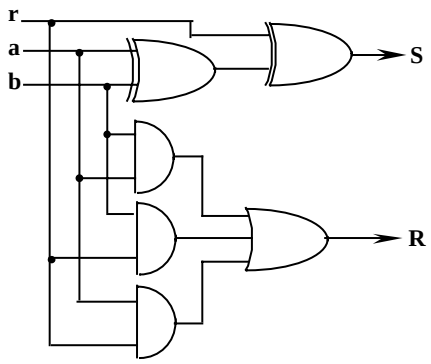


Figure 4a : additinneur complet intégré

En exprimant les fonctions S et R avec des opérateurs NON-ET logique, on obtient :

S = (a/a/b/b/r)/(a/a/b/r/r)/(a/b/b/r/r)/(a/b/r) R = (a/b)/(a/r)/(b/r)

Le schéma logique est donné sur la figure 4b :

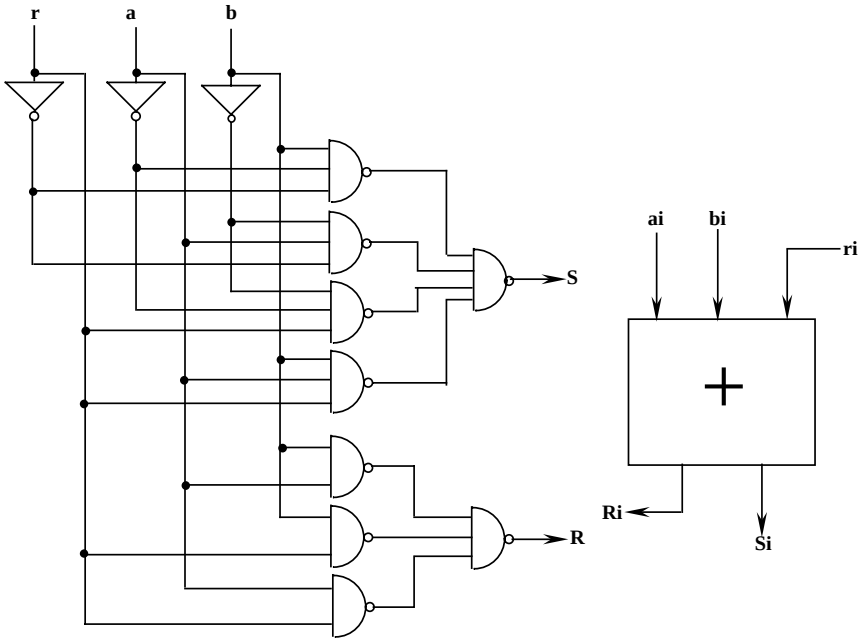


Figure 4 : additinneur complet intégré réalisé avec des portes logiques NON-ET

Comparaisons des deux solutions :

L'intérêt de la solution avec deux demi-additionneurs est qu'elle nécessite moins de portes logiques. Mais on a obtenu de la simplicité au détriment de la performance. Le facteur bloquant est le temps de propagation de la report :

- dans la 1ère solution : 3 étages de logique par bit
- dans la 2ème solution : 2 étages de logique par bit

Pour additionner 32 bits, le résultat n'est donc obtenu qu'au bout d'un temps correspondant à la traversée de 65 couches d'opérateurs pour la première solution contre 64 couches pour la seconde. Ces additionneurs sont dits des additionneurs a report en cascade (Figure 5).

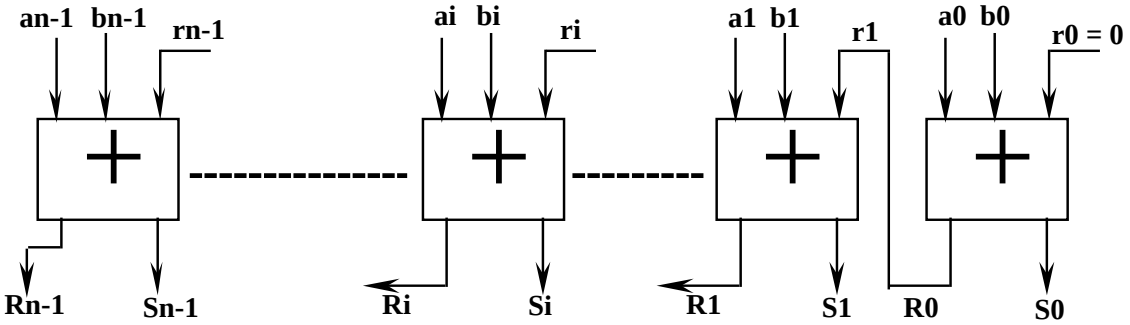


Figure 5 : additionneur à report en cascade

I. 3. Addition rapide à report anticipé (Carry look-ahead).

Dans un additionneur à report anticipé on fait pour chaque rang, le calcul de Ri directement en fonction des bits ai, bi, tous les rangs inférieurs et de r0. Il faut donc déterminer les fonctions logiques qui permettent d'effectuer ce calcul de R pour un rang quelconque.
Nous avons vu que le report d'un additionneur complet est donné par :

$$R = a.b + a.r + b.r = a.b + r.(a + b) = a.b + r.(a\.b + a.b\ + ab) = a.b.(1 + r) + r.\ (a\.b + a.b\)$$

$$\mathbf{R = a.b + r.(a \oplus b)}$$

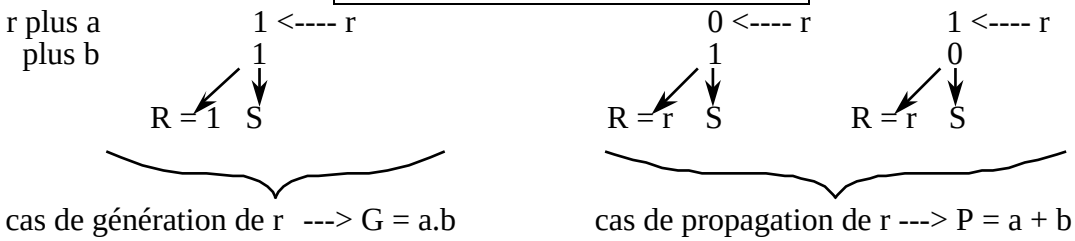
Cette équation signifie que R = 1 si et seulement si une des deux conditions suivantes est vérifiée :

- Si a = b = 1 quelque soit r ; il y a génération systématique d'un report. On définit alors **un terme de génération.**

$$\mathbf{G = a.b}$$

- Si r = 1 et si l'un des deux bits a ou b est égal 1 ; il y a alors propagation du report r. On définit alors **un terme de propagation**.

$$\mathbf{P = a \oplus b}$$



L'équation de R peut alors se mettre sous la forme : R = G + P.r. Tandis que S = a $\oplus$ b $\oplus$ r.
Supposons maintenant qu'on possède des circuits qui délivrent Gi, et Pi (figure 6).

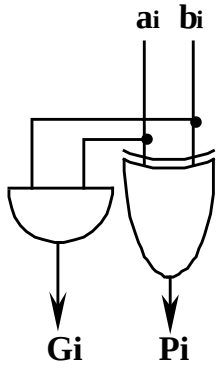


Figure 6 : calcul de des termes de génération et de propagation

Avec de tels circuits on se propose d'effectuer l'addition à report anticipé de 2 nombres A et B de 4 bits chacun : A → a3 a2 a1 a0
B → b3 b2 b1 b0

Le circuit de rang 0 nous donne les termes G0 et P0,
Le circuit de rang 1 nous donne les termes G1 et P1,
Le circuit de rang 2 nous donne les termes G2 et P2,
Le circuit de rang 3 nous donne les termes G3 et P3.

D'après la structure du circuit il nous faut calculer :
R0 = G0 + P0.r0 (bien que dans ce cas r0 = 0)
R1 = G1 + P1.r1 avec r1 = R0
R2 = G2 + P2.r2 avec r2 = R1
R3 = G3 + P3.r3 avec r3 = R2
Soit en remplaçant les termes r par leurs valeurs :

R0 = G0 + P0.r0
R1 = G1 + P1.(G0 + P0.r0)
R1 = G1 + P1.G0 + P1.P0.r0
R2 = G2 + P2.(G1 + P1.G0 + P1.P0.r0)
R2 = G2 + P2.G1 + P2.P1.G0 + P2.P1.P0.r0
R3 = G3 + P3.(G2 + P2.G1 + P2.P1.G0 + P2.P1.P0.r0)
R3 = G3 + P3.G2 + P3.P2.G1 + P3.P2.P1.G0 + P3.P2.P1.P0.r0

Ces équations nous conduisent à la réalisation de la figure 7 sur laquelle on constate que le calcul du report R de chaque rang (y compris le dernier R3) ne se fait qu'à travers 3 couches d'opérateurs seulement au lieu de 9 couches pour un additionneur à report en cascade.
En contre partie le nombre d'opérateurs nécessaires reste très important et c'est pourquoi ceci ne présente un réel intérêt que dans l'optique des techniques d'intégration à moyenne et grande échelle (MSI, LSI et VLSI). Le nombre d'opérateurs nécessaires prend alors une importance tout à fait secondaire en regard des autres avantages du procédé.

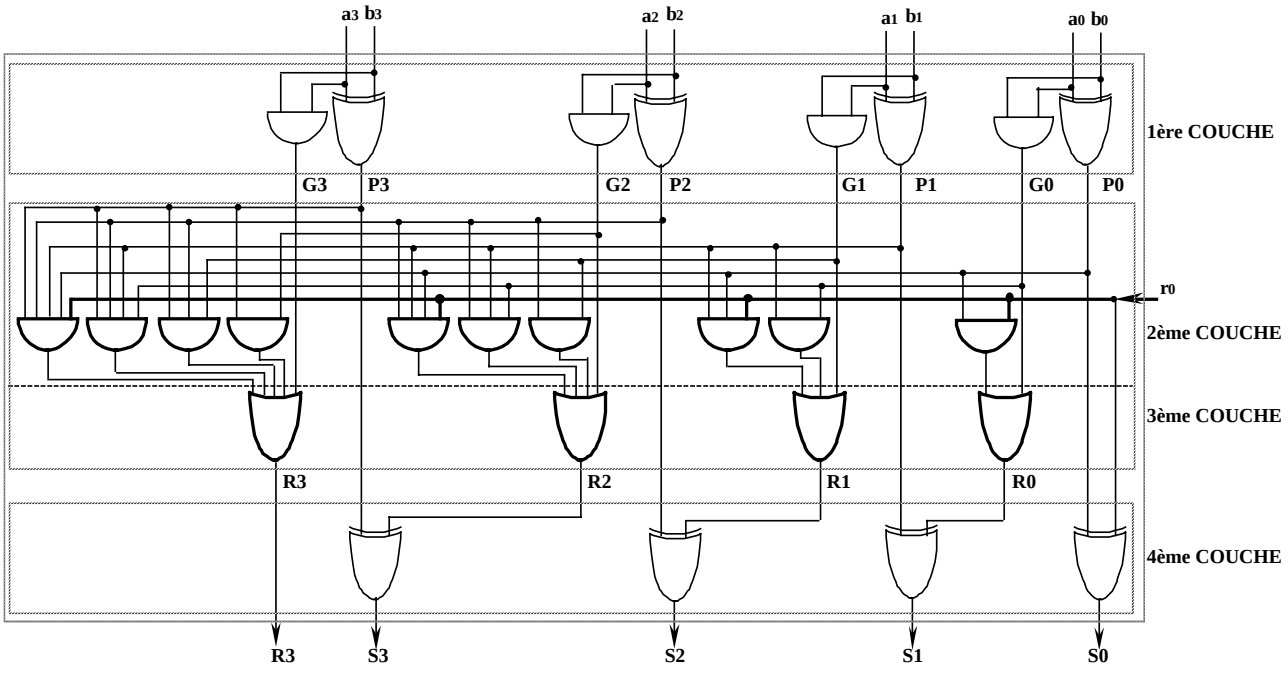


Figure 7 : additionneur 4 bits à report anticipé

Les circuits intégrés proposés actuellement effectuent l'addition rapide avec anticipation du report de 2 nombres de 4 bits chacun. Pour additionner des nombres comportant plus de 4 bits, il y a deux possibilités :

I. 3. 1. Addition à report anticipé à un seul niveau

soit à additionner 2 nombres X et Y de 32 bits, par exemple, on peut monter des additionneurs 4 bits en cascade, comme le montre la figure 8.

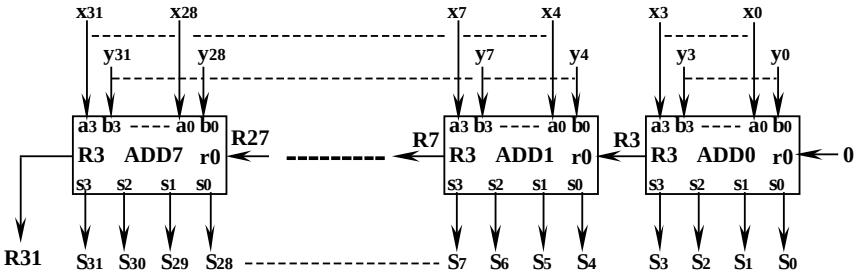


Figure 8 : additionneur 32 bits à un seul niveau

Avec ce montage le report R3 par exemple doit se propager à travers 2 couches d'opérateurs dans chacun des additionneurs ADD1 à ADD7 soit 14 couches au total. Le report R0 issu de l'addition des bits de plus faible poids se propage à travers 14 + 3 = 17 couches d'opérateurs.

I. 3. 2. Addition à report anticipé à plusieurs niveaux

On peut également avec certains circuits, dits générateurs de report anticipé, monter des additionneurs rapide sur plusieurs niveaux et anticiper le calcul du report entre les additionneurs.

Reprenons la fonction R3.

R3 = G3 + P3.G2 + P3.P2.G1 + P3.P2.P1.G0 + P3.P2.P1.P0.r0

On remarque que R3 dépend :
- d'un terme de génération : **G** = G3 + P3.G2 + P3.P2.G1 + P3.P2.P1.G0
- d'un terme de propagation : **P** = P3.P2.P1.P0

De ce fait R3 peut se mettre sous la forme : R3 = **G** + **P**.r0

Considérons maintenant des circuits additionneurs 4 bits qui fournissent ces fonctions G et P en sortie au lieu du report sortant. La figure 9a donne le schéma du circuit modifié.

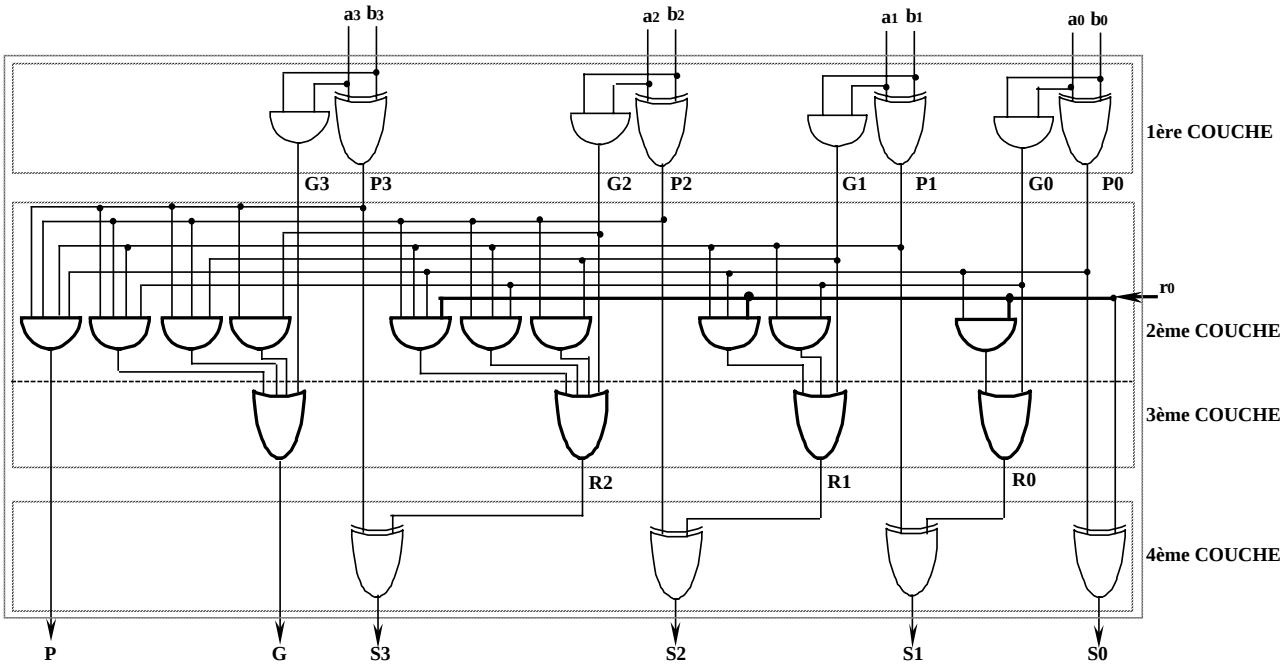


Figure 9a : additionneur 4 bits à report anticipé modifié

Considérons maintenant 4 de ces additionneurs 4 bits numérotés m, n, p et q et leurs sorties G, P qui, pour l'instant sont les seuls à nous intéresser (figure 9).

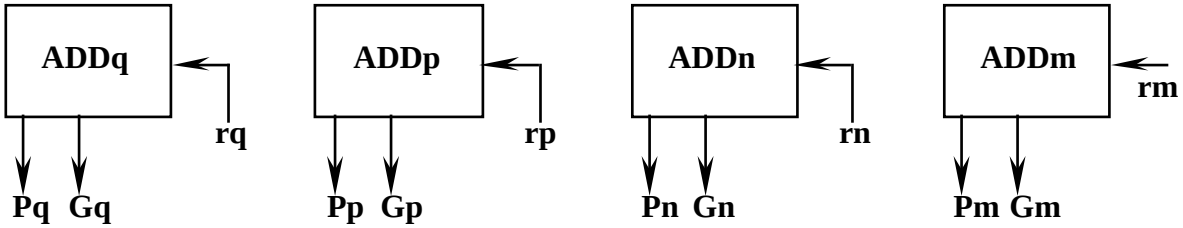


Figure 9b : additionneurs à report anticipé munis de sorties P et G

D'après l'équation établie plus haut (R3) on peut écrire :
Rm = Gm + Pm.rm
Rn = Gn + Pn.rn avec rn = Rm
Rp = Gp + Pp.rp avec rp = Rn
Rq = Gq + Pq.rq avec rq = Rp

d'où en développant :

Rm = Gm + Pm.rm
Rn = Gn + Pn.(Gm + Pm.rm)
Rn = Gn + Pn.Gm + Pn.Pm.rm
Rp = Gp + Pp.(Gn + Pn.Gm + Pn.Pm.rm)
Rp = Gp + Pp.Gn + Pp.Pn.Gm + Pp.Pn.Pm.rm
Rq = Gq + Pq.(Gp + Pp.Gn + Pp.Pn.Gm + Pp.Pn.Pm.rm)

Rq = Gq + Pq.Gp + Pq.Pp.Gn + Pq.Pp.Pn.Gm + Pq.Pp.Pn.Pm.rm

On peut encore mettre Rq sous la forme : Rq = G + P.rm
avec : G = Gq + Pq.Gp + Pq.Pp.Gn + Pq.Pp.Pn.Gm
et P = Pq.Pp.Pn.Pm
Il existe des circuits intégrés générateurs de report anticipé (type 74 182) qui à partir de Gm, Pm, Gn, Pn, Gp, Pp, Gq, Pq et rm fournissent en sortie les fonctions Rm, Rn, Rp, G et P (figure 10).

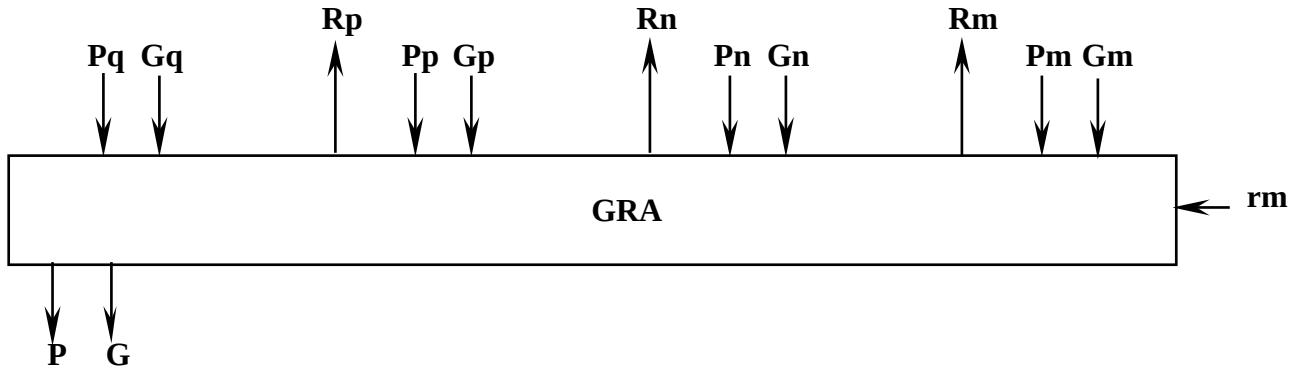


Figure 10 : représentation symbolique d'un générateur de report anticipé

La figure 11 donne le schéma logique de principe d'un circuit générateur de report anticipé, sur lequel on constate que le calcul de toutes les fonctions se fait à travers 2 couches d'opérateurs.

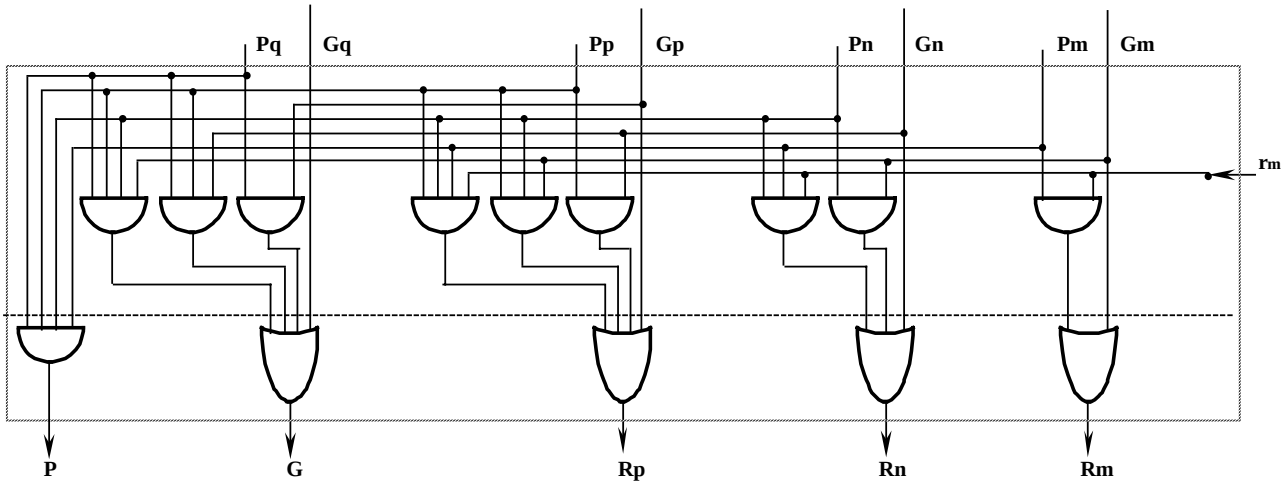
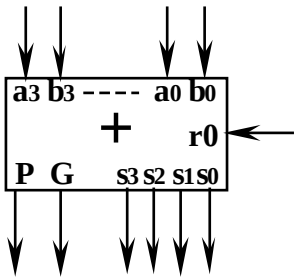


Figure 11 : schéma logique d'un générateur de report anticipé

A l'aide de tels circuits on peut réaliser des montages additionneurs à report anticipé :

- **à deux niveaux** : les additionneurs utilisés ADD0 à ADD7 sont tous du type ci-dessous.



Sur le schéma de la figure 12, on constate que le report issu de l'addition de (x0, y0), avant d'arriver en R31 va traverser :

- 3 couches d'opérateurs dans ADD0
- 2 couches d'opérateurs dans GRAa
- 2 couches d'opérateurs dans ADD3
- 2 couches d'opérateurs dans GRAb
- 2 couches d'opérateurs dans ADD7

Soit au total 11 couches d'opérateurs au lieu de 17 couches avec des additionneurs à report anticipé monté en cascade.

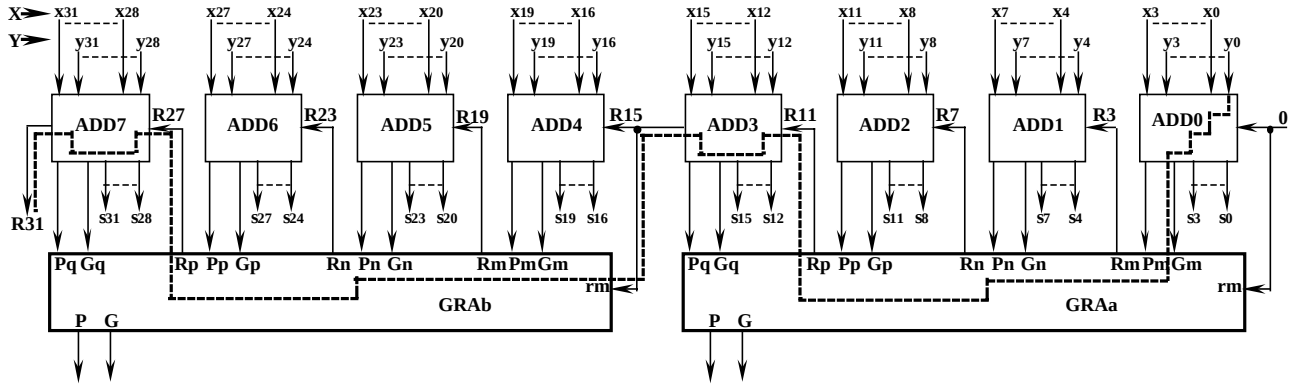


Figure 12 : additionneur 32 bits à 2 niveaux de calcul de report anticipé

- **à trois niveaux ou plus** : Sur le schéma de la figure 13, on constate cette fois que le report R0 issu de l'addition de (x0, y0), avant d'arriver en R31 va traverser :
  - 3 couches d'opérateurs dans ADD0
  - 2 couches d'opérateurs dans GRAa
  - 2 couches d'opérateurs dans GRAc

Soit au total 7 couches d'opérateurs au lieu de 11 couches avec des additionneurs à report anticipé monté en deux niveaux.

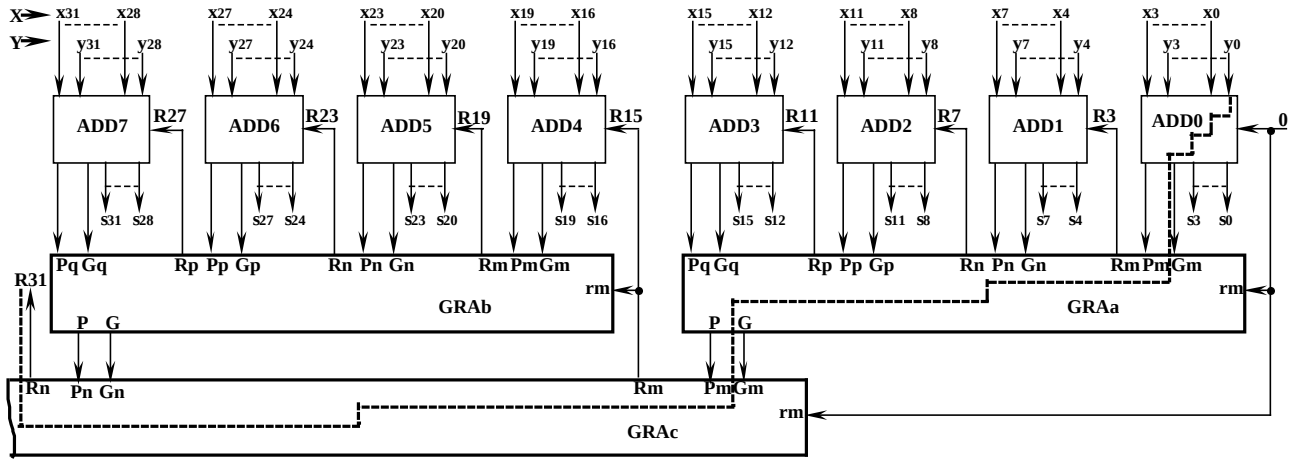


Figure 13 : additionneur 32 bits à 3 niveaux de calcul de report anticipé

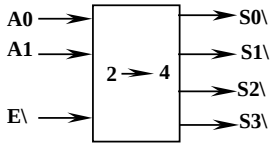
I. 4. Tableau récapitulatif de la comparaison portant sur le report R0

Addition de 2 nombres	Nombres de couches d'opérateurs			
	8 bits	16 bits	32 bits	64 bits
additionneurs complet avec report en cascade	17(16)[1]	33(32)	65(64)	129(128)
additionneurs à report anticipé (1 niveau)	5	9	17	33
additionneurs à report anticipé (2 niveaux)	-	7	11	19
additionneurs à report anticipé (3 niveaux)	-	-	7	9

[1] Les valeurs entre parenthèses représentent le nombre de couches dans le cas d'un additionneur complet intégré.

II. ETUDE DU DECODEUR

Un décodeur est un circuit possédant n entrées et N sorties tel que ($N = 2^n$) et une entrée de validation (Enable) ou de commande. La représentation symbolique et la table de vérité d'un décodeur "2 vers 4" sont les suivants :



E	A	B	S0	S1	S2	S3	Remarque
1	X	X	1	1	1	1	inactif
0	0	0	0	1	1	1	actif
0	0	1	1	0	1	1	
0	1	0	1	1	0	1	
0	1	1	1	1	1	0	

S0\ = E + A + B $\Rightarrow$ S0\\\ = (E + A + B)\\ = (E\.A\.B\)\
S1\ = E + A + B\ $\Rightarrow$ S1\\\ = (E + A + B\ )\\ = (E\.A\.B)\
S2\ = E + A\ + B $\Rightarrow$ S2\\\ = (E + A\ + B)\\ = (E\.A.B\)\
S3\ = E + A\ + B\ $\Rightarrow$ S3\\\ = (E + A\ + B\)\\ = (E\.A.B)\

Un décodeur est identificateur de minterm. On remarquera qu'il fonctionne en logique positive sur les entrées et en logique négative sur les sorties et l'entrée de validation. La réalisation d'un décodeur de "2 vers 4" avec portes logiques NON-ET est donnée sur la figure 14.

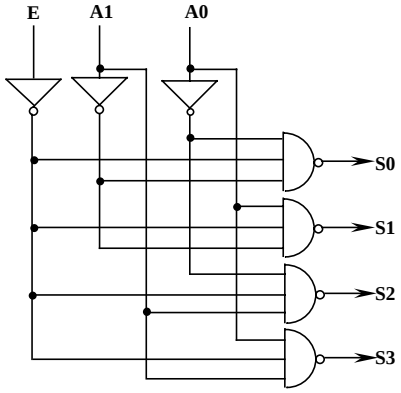


Figure 15 : schéma logique d'un décodeur de 2 vers 4

II. 1. Association de décodeurs

Le signal de validation permet d'associer deux décodeurs "2 vers 4" pour obtenir un décodeur de 3 vers 8 (3 entrées, 8 sorties) (figure 16).

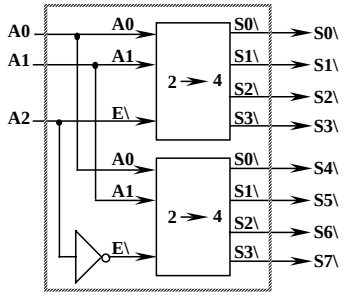


Figure 16 : décodeur de 3 vers 8 par association de 2 décodeurs de 2 vers 4

Ce décodeur de 3 vers 8 ne possède pas de signal de validation. Il est possible de le rajouter si nécessaire en conditionnant par deux portes logiques OU les deux signaux /E des deux décodeurs par ce signal global de validation (figure 17).

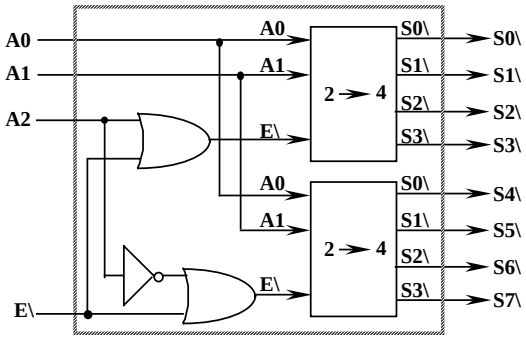
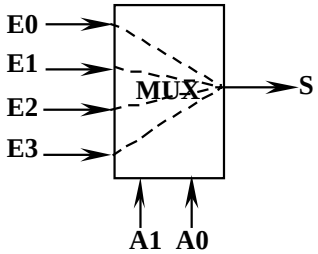


Figure 17 : décodeur de 3 vers 8 avec un signal de validation à partir de 2 décodeurs de 2 vers 4

En procédant, de façon récursive on voit qu'il est possible d'obtenir un décodeur de taille quelconque.

III. ETUDE DU MULTIPLEXEUR

Un multiplexeur permet de réaliser un "aiguillage" d'informations, il possède N entrées de données tel que ($N = 2^n$) qui représentent les informations à aiguiller, une sortie de donnée et n entrées de contrôle permettant de donner le numéro de l'entrée à aiguiller sur la sortie.
La représentation symbolique et la table de vérité d'un multiplexeur de "4 vers 1" sont les suivantes :



A1	A0	S
0	0	E0
0	1	E1
1	0	E2
1	1	E3

$S = A\backslash.B\backslash.E0 + A\backslash.B.E1 + A.B\backslash.E2 + A.B.E3$

Le schéma logique du circuit multiplexeur de 4 vers 1 est représenté sur la figure 18 :

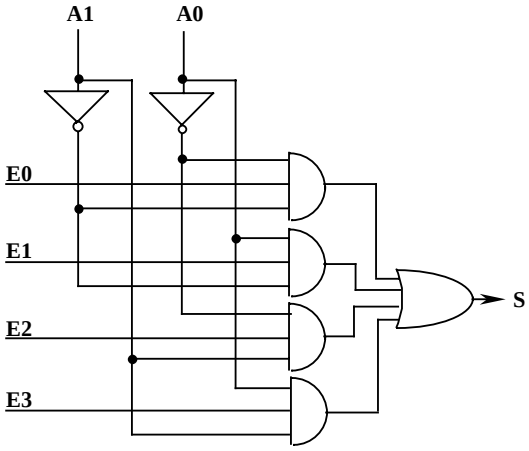


Figure 18 : schéma logique d'un multiplexeur de 4 vers 1

On ne voit pas de signal de validation comme dans le décodeur. Ce signal pourrait être envisagé, mais n'apporterait rien ici : si le multiplexeur est bloqué, cela veut dire qu'il ne peut laisser passer aucune entrée. Mais qu'y aurait-il alors sur sa sortie : il faudrait convenir d'un état 1 ou 0 dans ce cas, résultat qui peut être paramétré en conservant l'une des entrées pour ce cas.

III. 1. Association de multiplexeurs

L'association de multiplexeurs est possible comme le montre la figure suivante : réalisation d'un multiplexeur de 8 vers 1 à partir de 3 multiplexeurs de 4 vers 1 (Figure 19).

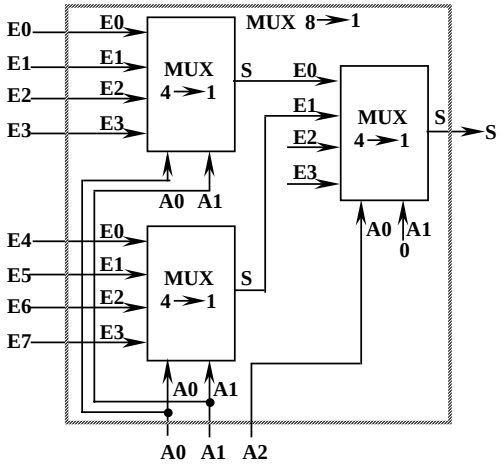


Figure 19 : multiplexeur de 8 vers 1 par association de 3 multiplexeurs de 2 vers 4

Cette association peut être étendue pour obtenir le multiplexeur désiré.