Multiplication Matérielle

Alain GUYOT

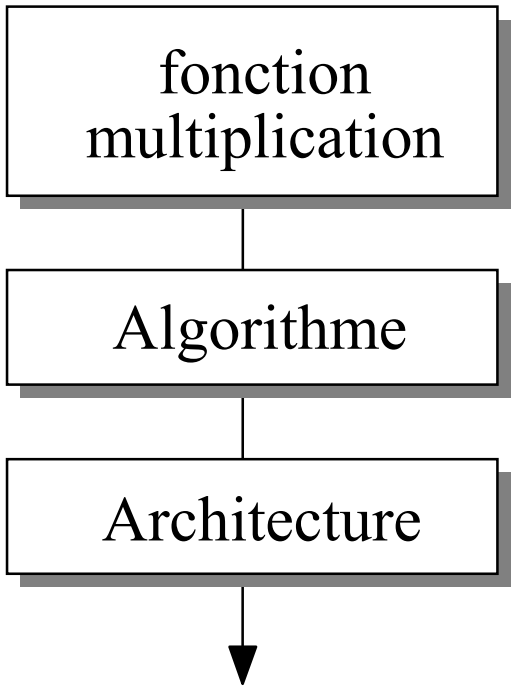
Concurrent Integrated Systems
TIMA

(33) 04 76 57 46 16
Alain.Guyot@imag.fr
http://tima-cmp.imag.fr/Homepages/guyot
Techniques de l'Informatique et de la Microélectronique pour l'Architecture. Unité associée au C.N.R.S. n° B0706

But
Réaliser des multiplieurs

Optimiser la surface et/ou la vitesse

Problèmes
- Propagation de la retenue



Plan

- Multiplieur "naïf" d'entiers ≥ 0 — 1,5 n
- Multiplieur "carry save" d'entiers ≥ 0 — n
- Multiplieur "naïf" d'entiers relatifs — 1,5 n
- Multiplieur "carry save" d'entiers relatifs — n
- Multiplieur performant — $\log_{3/2} n$, $\log_2 n$

Multiplication d'entiers

Soient $A = \sum_{i=0}^{n-1} a_i * 2^i$; $B = \sum_{j=0}^{n-1} b_j * 2^j$ avec $a_i, b_i \in \{0,1\}$.

Le produit $P = A * B$ s'écrit $\left(\sum_{i=0}^{n-1} a_i * 2^i\right) * \left(\sum_{j=0}^{n-1} b_j * 2^j\right)$

ou encore en appliquant la distributivité somme / produit

$$P = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \left(a_i * b_j * 2^{i+j} \right).$$

Or $A < 2^n$ et $B < 2^n$ impliquent que $P < 2^{2n}$,

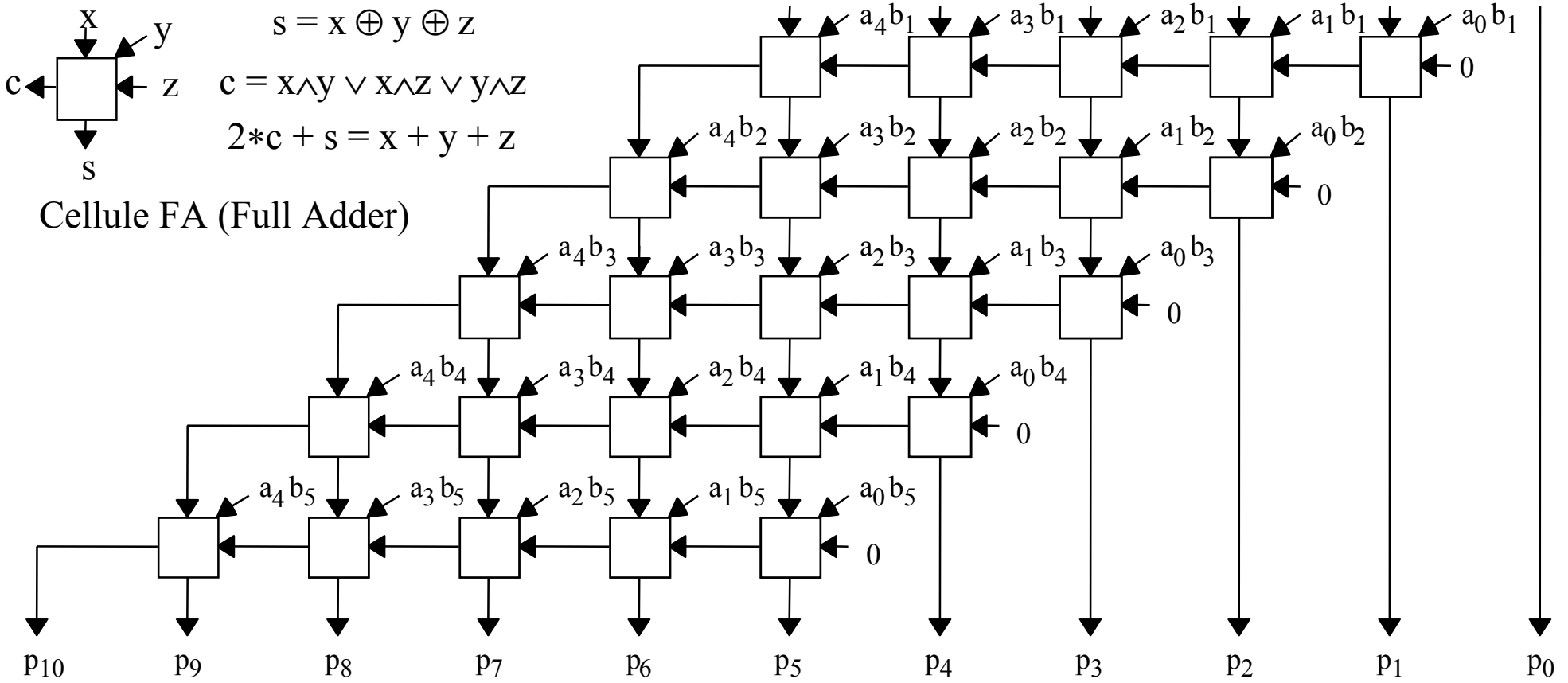
Par conséquent P peut être écrit avec 2n bits.

Le problème est alors de réduire la somme pondérée de n^2 bits en la somme pondérée de 2n bits.

Cette réduction peut être faite en temps $O(n)$, $O(\sqrt[2]{n})$, $O(\log_{3/2} n)$, $O(\log_2 n)$

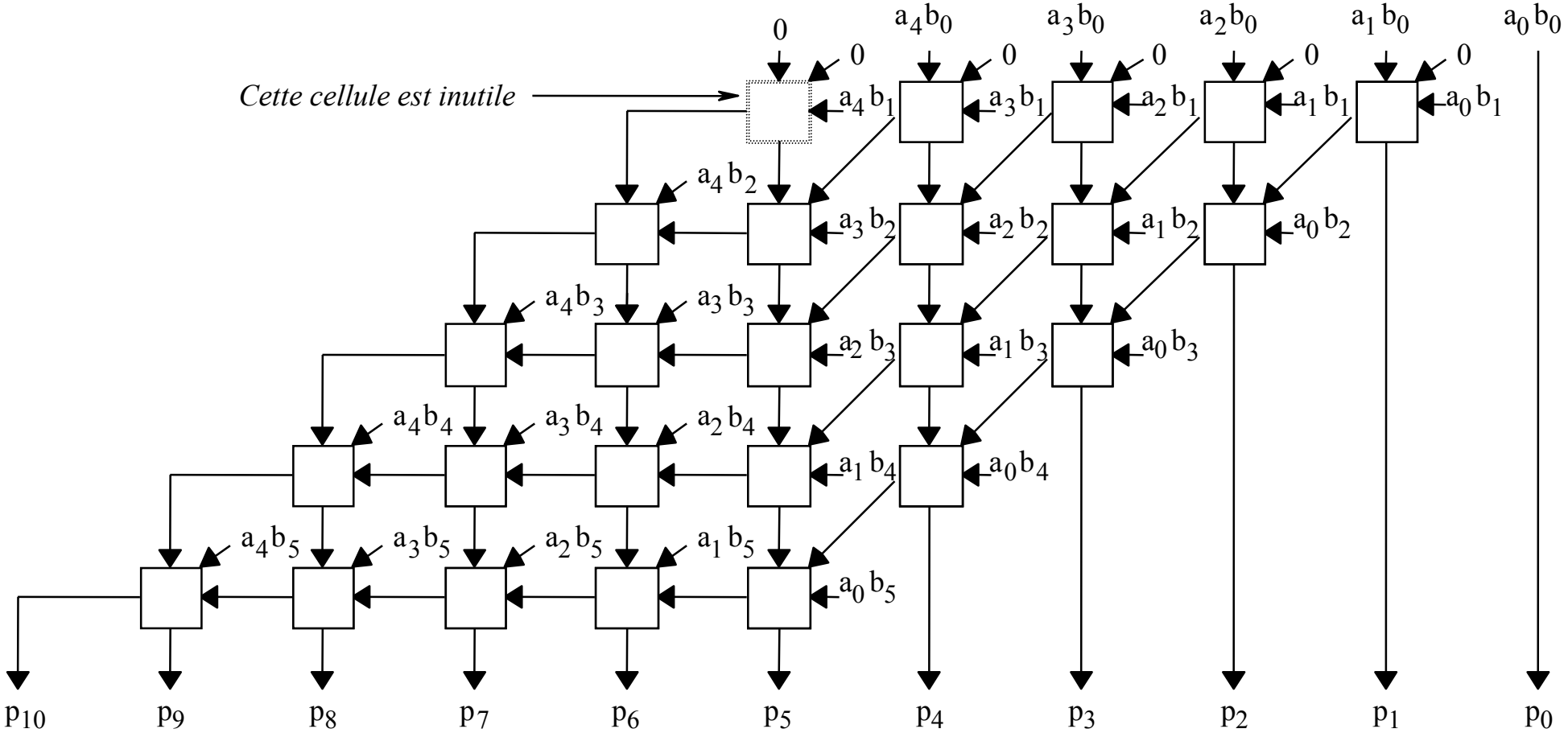
Multiplieur naïf

$$A = \sum_{i=0}^{4} a_i 2^i \qquad B = \sum_{i=0}^{5} b_i 2^i \qquad P = A*B = \sum_{i=0}^{10} p_i 2^i$$

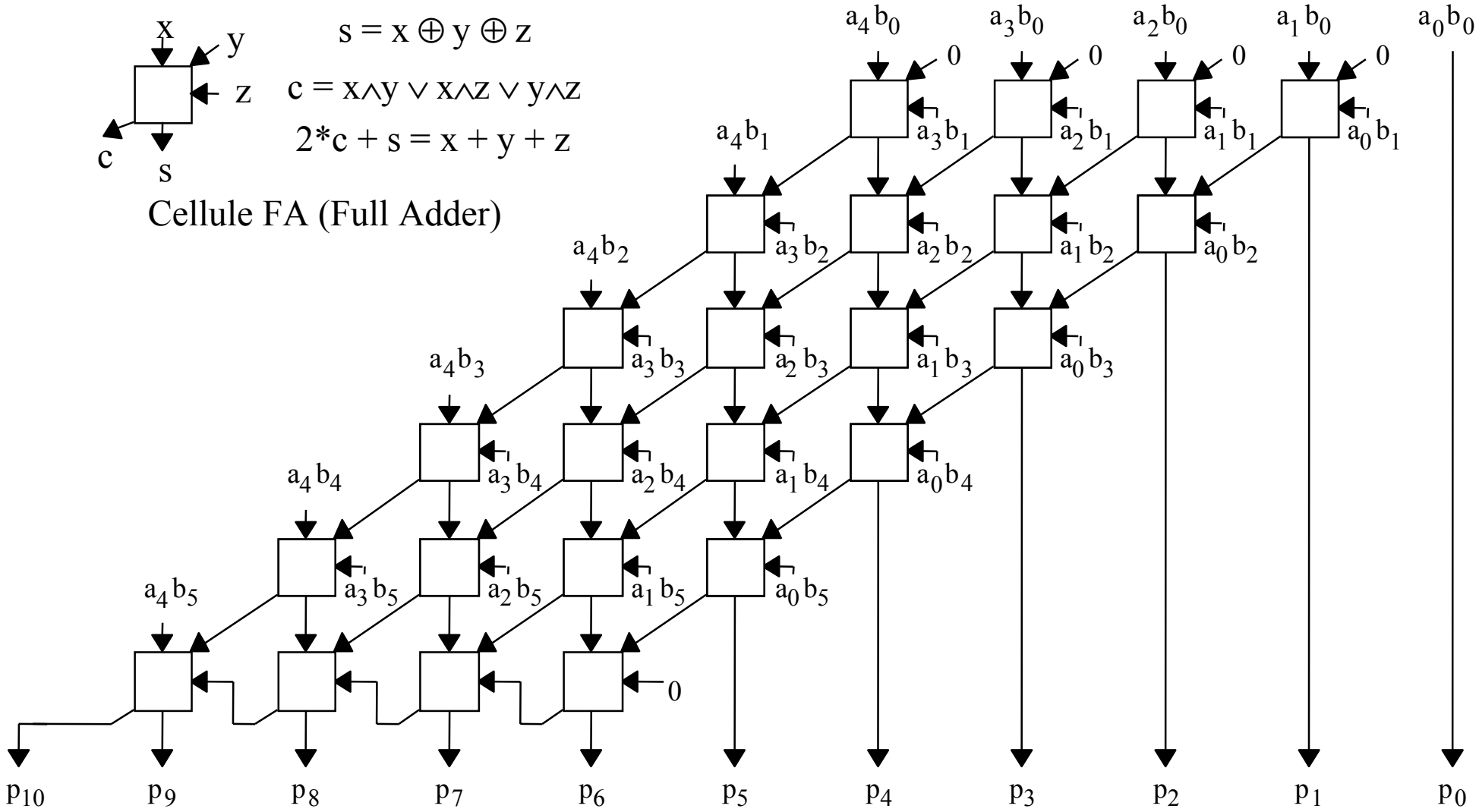


Multiplieur amélioré

Tous les chiffres d'une même colonne sont de même poids.
En jouant sur l'associativité de l'addition, on peut réduire le chemin critique de 13 à 9.



Multiplieur de Braun



Multiplieur signé

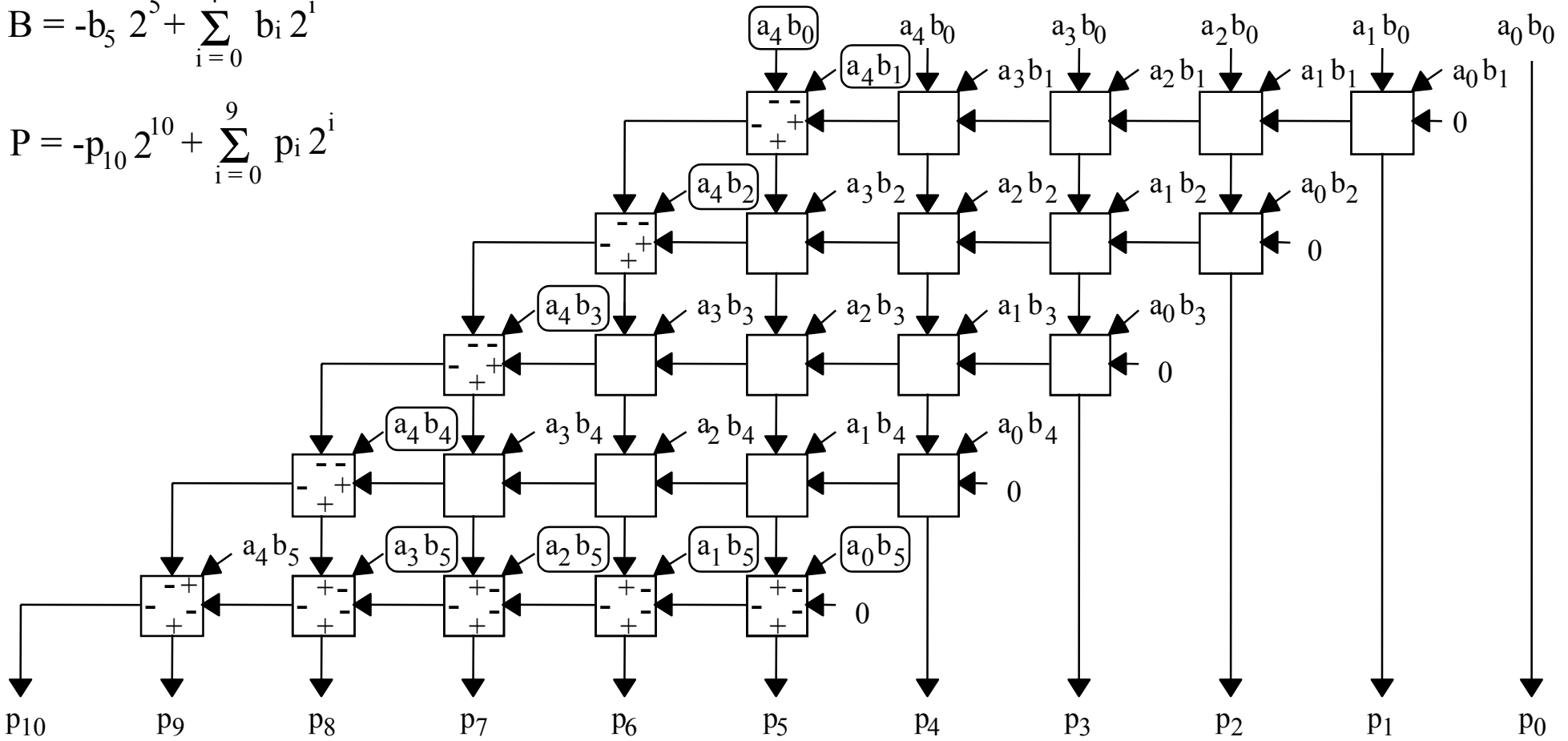
$$A = -a_4\,2^4 + \sum_{i=0}^{3} a_i\,2^i$$

$$B = -b_5\,2^5 + \sum_{i=0}^{4} b_i\,2^i$$

$$P = -p_{10}\,2^{10} + \sum_{i=0}^{9} p_i\,2^i$$

Multiplieur de Pezaris

Pezaris, S.D. "A 40 ns 17 bit by 17 bit Array Multiplier"
IEEE Transactions on Computers (1971) pp 442-447.



Multiplieur de Baugh-Wooley (1)

$$P = A * B = \left(-a_4 2^4 + \sum_{i=0}^{3} a_i 2^i \right) * \left(-b_4 2^4 + \sum_{j=0}^{3} b_j 2^j \right) = a_4.b_4 2^8 + \sum_{i=0}^{3}\sum_{j=0}^{3} a_i.b_j 2^{i+j} - a_4 2^4 \sum_{j=0}^{3} b_j 2^j - b_4 2^4 \sum_{i=0}^{3} a_i 2^i$$

$$= a_4.b_4 2^8 + \sum_{i=0}^{3}\sum_{j=0}^{3} a_i.b_j 2^{i+j} + a_4 2^4 \left(-2^4 + \sum_{j=0}^{3} \overline{b_j} 2^j + 1 \right) + b_4 2^4 \left(-2^4 + \sum_{i=0}^{3} \overline{a_i} 2^i + 1 \right)$$

$$= a_4.b_4 2^8 + \sum_{i=0}^{3}\sum_{j=0}^{3} a_i.b_j 2^{i+j} + \left(\overline{a_4} - 1 + \overline{b_4} - 1 \right) 2^8 + (a_4 + b_4) 2^4 + \sum_{i=0}^{3} \left(a_4.\overline{b_i} + \overline{a_i}.b_4 \right) 2^{i+4}$$

2^9	2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
					$a_4.\overline{b_0}$	$a_3.b_0$	$a_2.b_0$	$a_1.b_0$	$a_0.b_0$
				$a_4.\overline{b_1}$	$a_3.b_1$	$a_2.b_1$	$a_1.b_1$	$a_0.b_1$	
			$a_4.\overline{b_2}$	$a_3.b_2$	$a_2.b_2$	$a_1.b_2$	$a_0.b_2$		
		$a_4.\overline{b_3}$	$a_3.b_3$	$a_2.b_3$	$a_1.b_3$	$a_0.b_3$			
	$a_4.b_4$	$\overline{a_3}.b_4$	$\overline{a_2}.b_4$	$\overline{a_1}.b_4$	$\overline{a_0}.b_4$				
	$\overline{a_4}$				a_4				
1	$\overline{b_4}$				b_4				

2^9	2^8
	$a_4.b_4$
	$\overline{a_4}$
1	$\overline{b_4}$
2^9	2^8
$a_4 \vee b_4$	$a_4 \vee b_4$

Amélioration de Blankenship

Multiplieur de Baugh-Wooley (2)

$$A = -a_4 2^4 + \sum_{i=0}^{3} a_i 2^i$$

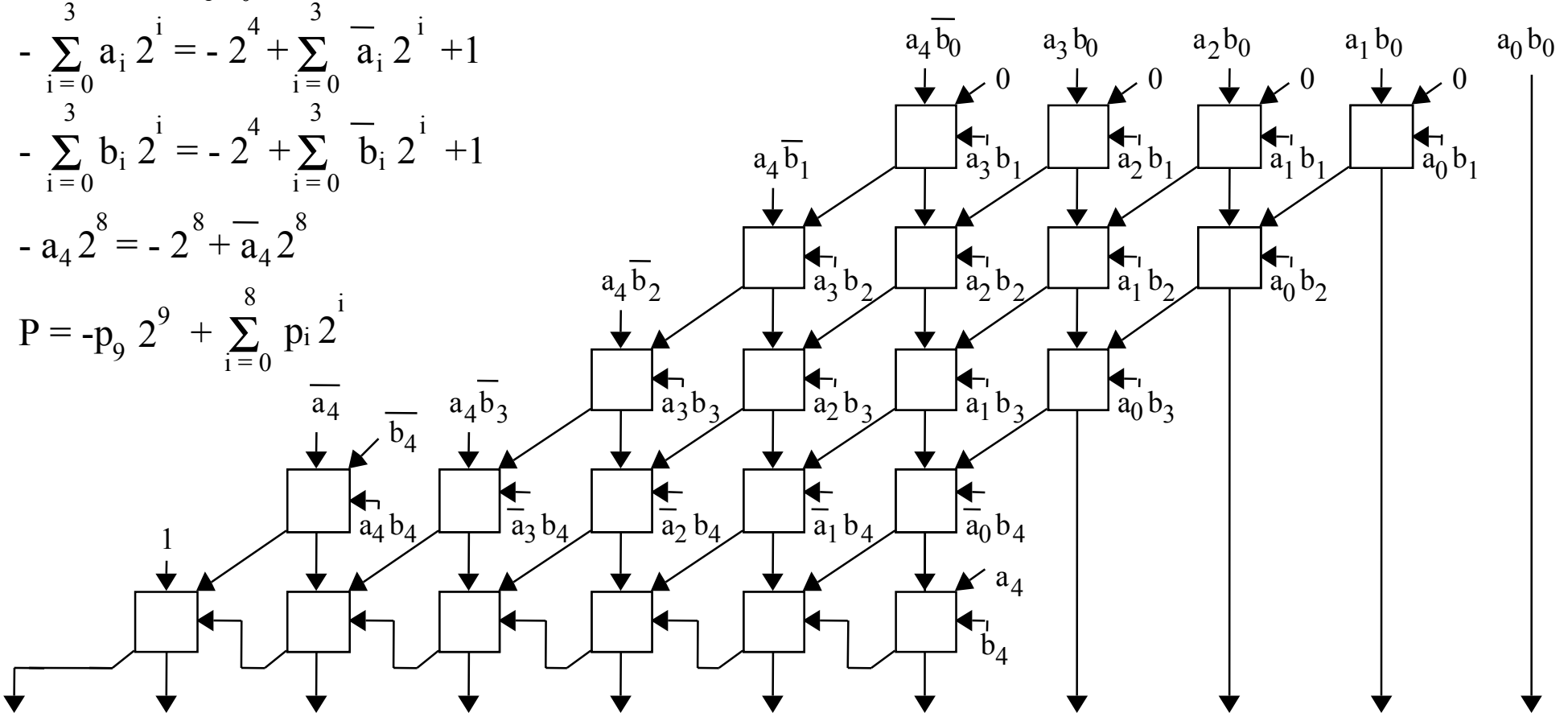
$$B = -b_4 2^4 + \sum_{i=0}^{3} b_i 2^i$$

$$-\sum_{i=0}^{3} a_i 2^i = -2^4 + \sum_{i=0}^{3} \overline{a}_i 2^i + 1$$

$$-\sum_{i=0}^{3} b_i 2^i = -2^4 + \sum_{i=0}^{3} \overline{b}_i 2^i + 1$$

$$-a_4 2^8 = -2^8 + \overline{a}_4 2^8$$

$$P = -p_9 2^9 + \sum_{i=0}^{8} p_i 2^i$$



Multiplieur rapide

Stratégie

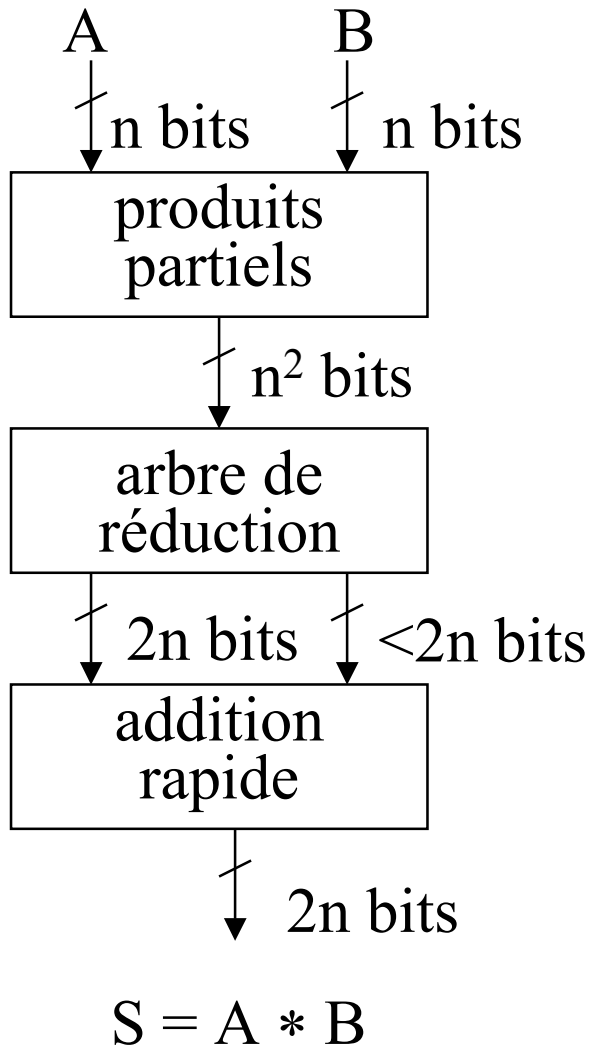
1- Au début calculer simultanément tous les produits partiels $a_i * b_j$

(il y en a n^2 pour le produit naïf, $n^2 + 5$ pour Baugh-Wooley et $n/2(n+4)$ pour Booth modifié)

2- Puis réduire ces produits partiels à 1 ou 2 bit(s) pour chaque poids sans propager de retenue

3- Enfin additionner les deux nombres de l'étape précédente le plus vite possible (ça, on sait faire)

Pour l'étape 2, on sait déjà combien de cellule "FA" et "HA" il y aura dans chaque colonne, cependant on va perdre la régularité des connections.



Exercice

On veut multiplier 2 nombres entiers A et B de n bits chacun.

1- Combien y a t'il de produits partiels ($a_j * b_k$) au total ?

2- Combien y a t'il de produits partiels ($a_j * b_k$) de poids 2^i ($0 \leq i \leq 2n-2$) ?

3- Combien faut-il de Cellules FA pour réduire tous ces produits partiels à (2n - 1) bits ?

4- Combien faut-il de Cellules HA ?

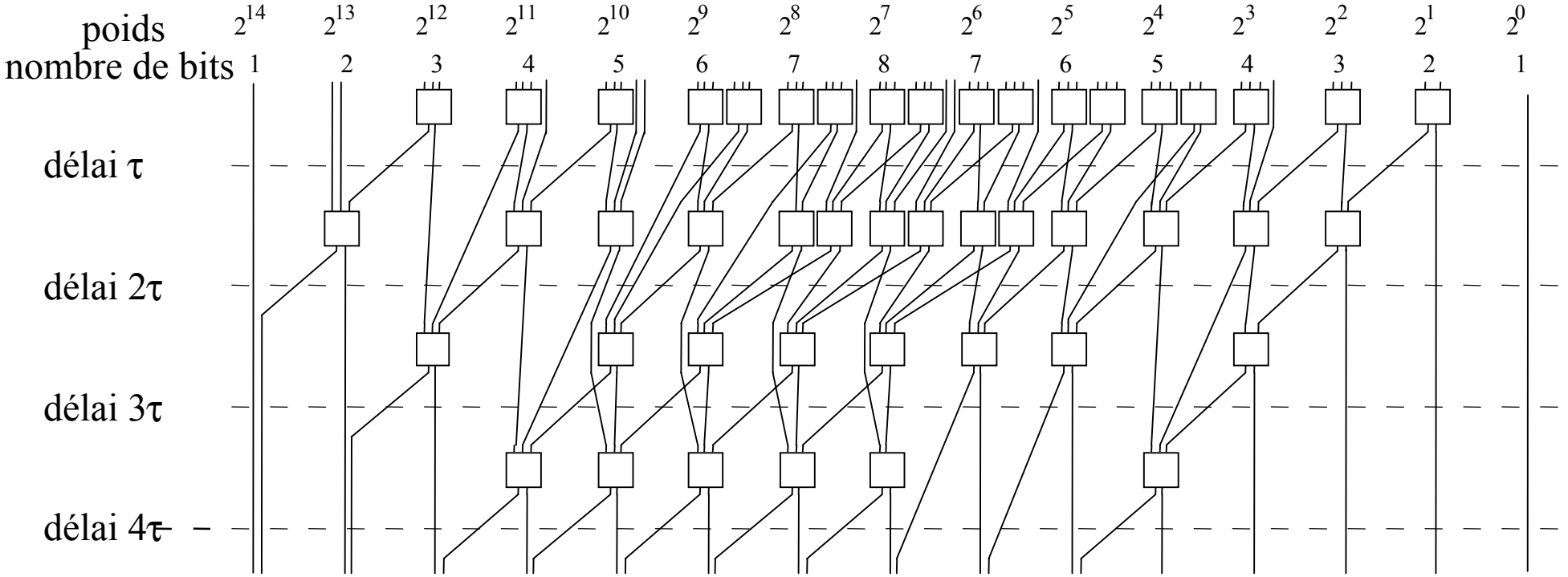
5- En comptant les retenues, combien y a t'il de bits de poids 2^i à réduire ?

6- Combien de cellules FA faut il traverser au minimum pour réduire k bits de poids 2^i en

un bit de poids 2^i et $1+\left\lceil \frac{k-3}{2} \right\rceil$ bits de poids 2^{i+1} ?

Arbre de Wallace

Calcul des n^2 produits partiels $a_i * b_j$
(exemple de 64 produits bit à bit)

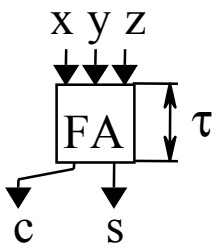


Vers addition rapide (10 bits)

La somme pondérée des bits qui entrent est égale à la somme pondérée des bits qui sortent !

Algorithme de Dadda

Equations logiques et modèle de délai

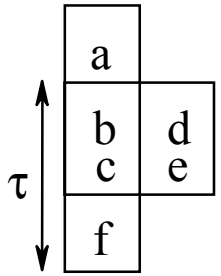


$s = x \oplus y \oplus z$

$c = x \wedge y \vee x \wedge z \vee y \wedge z$

$2*c + s = x + y + z$

a, f : # bits
b, d : # FA
c, e : # HA



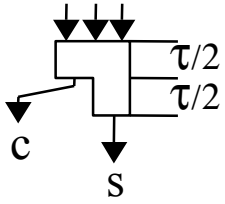
$3*b + 2*c \leq a$

$f = a + d + e - 2*b - c$

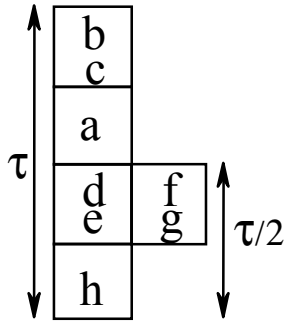
1	2	3	4	5	6	7	8	7	6	5	4	3	2	1
		FA	FA	FA	FA FA	FA FA	FA FA	FA FA	FA FA	FA HA	FA	FA	HA	
1	3	2	3	5	4	5	6	5	4	3	3	2	1	1
	FA		FA	FA	FA	FA HA	FA FA	FA HA	FA	FA	FA	HA		
2	1	3	2	4	4	4	4	3	3	2	2	1	1	1
		FA		FA	FA	FA	FA	FA	FA		HA			
2	2	1	3	3	3	3	3	2	1	3	1	1	1	1
			FA	FA	FA	FA	HA			FA				
2	2	2	2	2	2	2	2	2	2	1	1	1	1	1

Algorithme de Dadda

Modèle de délai plus précis



La retenue c se calcule 2 fois plus vite que la somme s



a, h : # bits
b, d, f : # FA
c, e, g : # HA

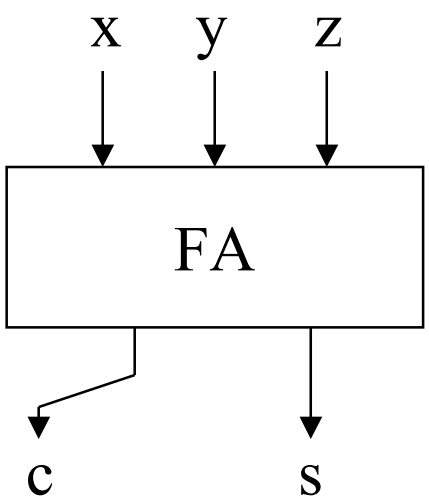
$3*d + 2*e \leq a$

$h = a + b + c + f + g - 3*d - 2*e$

1	2	3	4	5	6	7	8	7	6	5	4	3	2	1
		FA	FA	FA	FA FA	FA FA	FA FA	FA FA	FA FA	FA	FA	FA	HA	
1	3	1	2	4	2	3	4	3	1	3	2	1	0	1
	FA			FA		FA	FA	FA		FA				
1	0	2	4	2	5	3	4	2	4	1	3	2	1	1
			FA		FA	FA	FA	HA	FA		FA	HA		
2	1	3	1	4	3	2	3	2	1	3	1	0	1	1
		FA		FA	FA	HA	FA			FA				
2	2	0	3	2	2	2	1	3	3	0	2	1	1	1
								FA	FA		HA			
2	2	1	3	3	3	3	3	1	0	2	0	1	1	1
			FA	FA	FA	FA	FA			HA				
2	2	2	1	1	1	1	0	2	2	0	1	1	1	1
2	2	2	2	2	2	2	1	2	2	1	1	1	1	1

τ/2

Modèles de délai de « FA »



$x \leq y \leq z$

(x arrive avant ou en même temps que y)

(y arrive avant ou en même temps que z)

$s = \max.\ (\ y + \delta_1,\ \ z + \delta_2)$

$c = z + \delta_3$

Valeurs courantes :

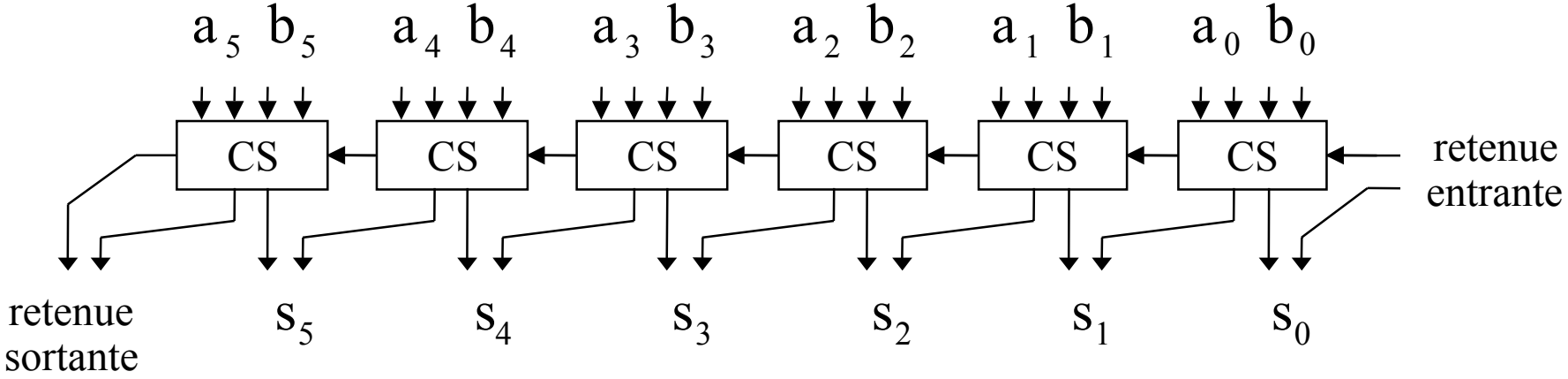
$\delta_1 = 1\ ,\ \delta_2 = 1\ ,\ \delta_3 = 1$

$\delta_1 = 1\ ,\ \delta_2 = 1\ ,\ \delta_3 = 1/2$

$\delta_1 = 1\ ,\ \delta_2 = 1/2\ ,\ \delta_3 = 1/2$

Rappel sur l'additionneur CS (ou réducteur 4 donne 2)

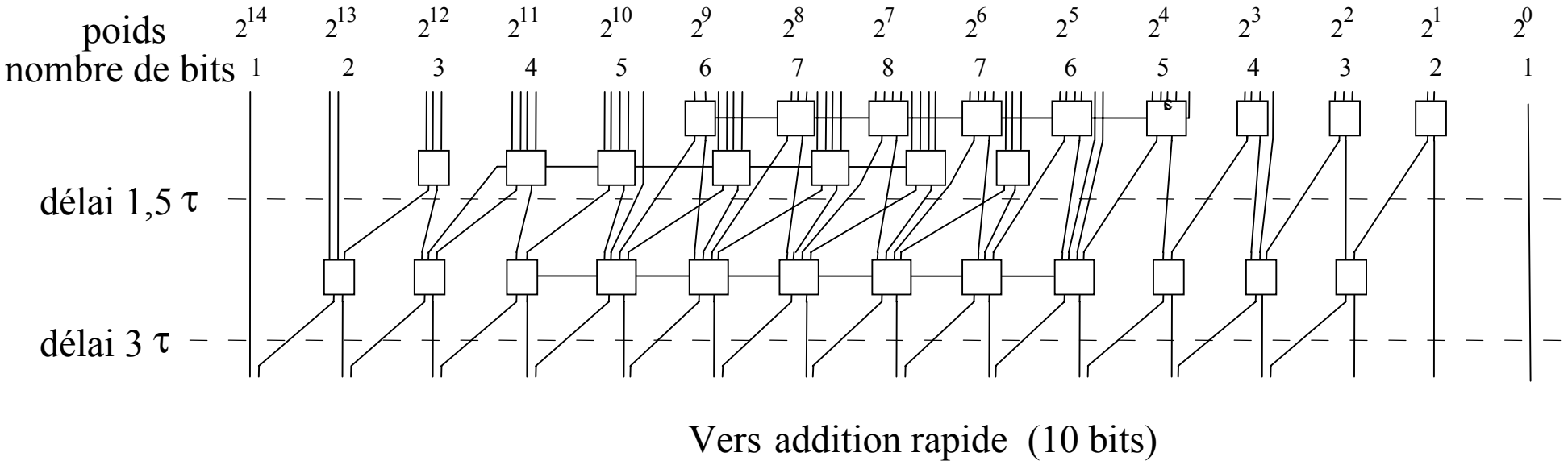
$$A = \sum_{i=0}^{n-1} a_i\, 2^i \qquad B = \sum_{i=0}^{n-1} b_i\, 2^i \qquad S = \sum_{i=0}^{n} s_i\, 2^i \qquad a_i, b_i, s_i \in \{0,1,2\}$$



La somme pondérée des bits qui entrent est égale à la somme pondérée des bits qui sortent !

Multiplieur arborescent

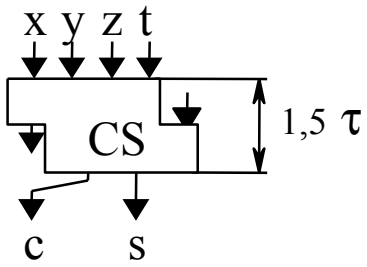
Calcul des n^2 produits partiels $a_i * b_j$
(exemple de 64 bits)



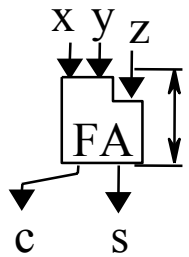
La somme pondérée des bits qui entrent est égale à la somme pondérée des bits qui sortent !

Multiplieur arborescent

1	2	3	4	5	6	7	8	7	6	5	4	3	2	1	# bits à réduire
0	0	1	1	1	2	2	1	1	1	0	0	0	0	0	# retenue horizontale
		FA	CS	CS	FA CS	CS CS	CS CS	CS FA	CS	CS	FA	FA	HA		1,5 τ
1	3	3	2	4	4	4	4	3	4	2	3	2	1	1	
0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	
	FA	FA	FA	CS	CS	CS	CS	CS	CS	HA	FA	HA			
2	2	2	2	2	2	2	2	2	2	2	2	1	1	1	



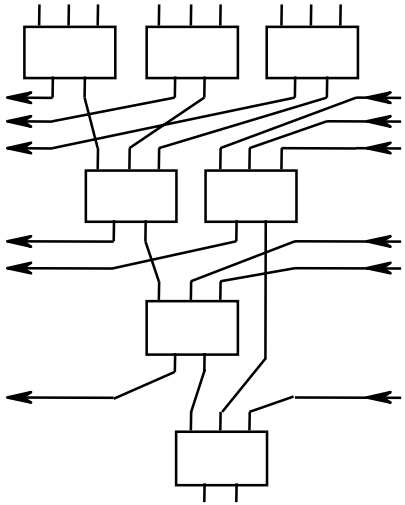
	-4		-5
+1	-1	+1	0
	CS	ou	CS
+1	+1	+1	+1



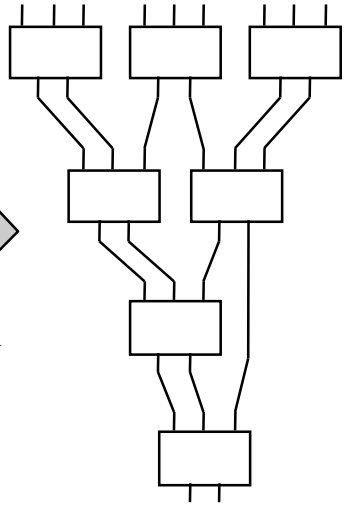
	-2		-3
0	-1	0	0
	FA	ou	FA
+1	+1	+1	+1

Modèle de réducteur

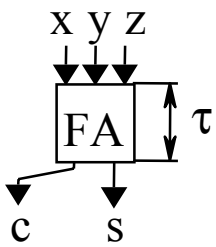
Le nombre et la position des retenues qui sortent sont les mêmes pour les retenues qui entrent

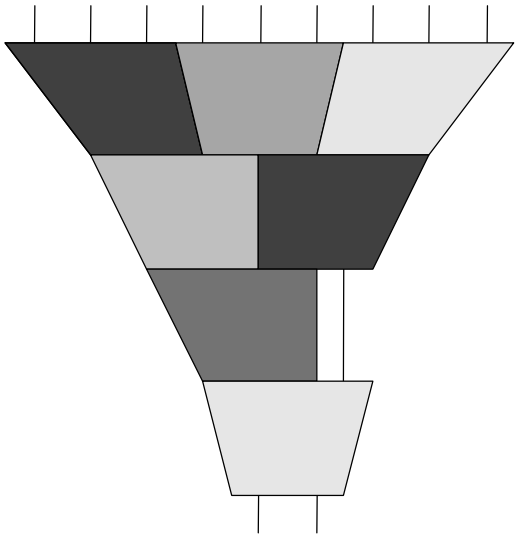


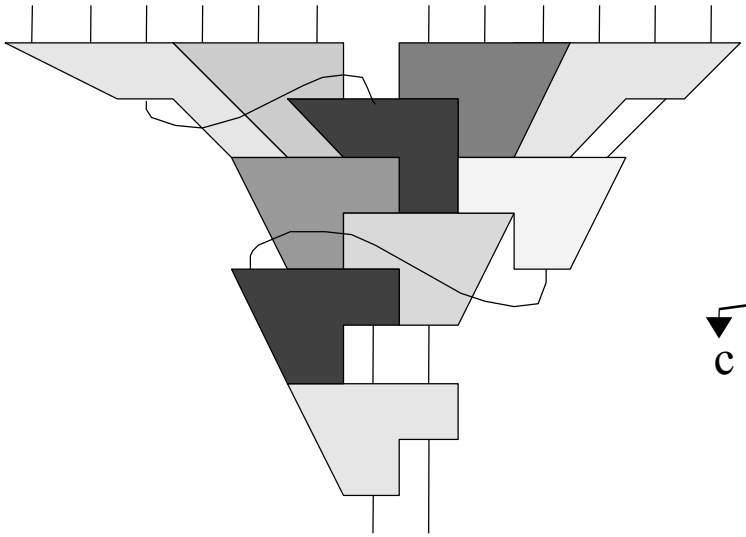
même délai

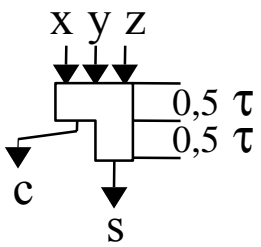


Ce modèle de réducteur est lui même réducteur: les retenues ne sont pas uniformément réparties

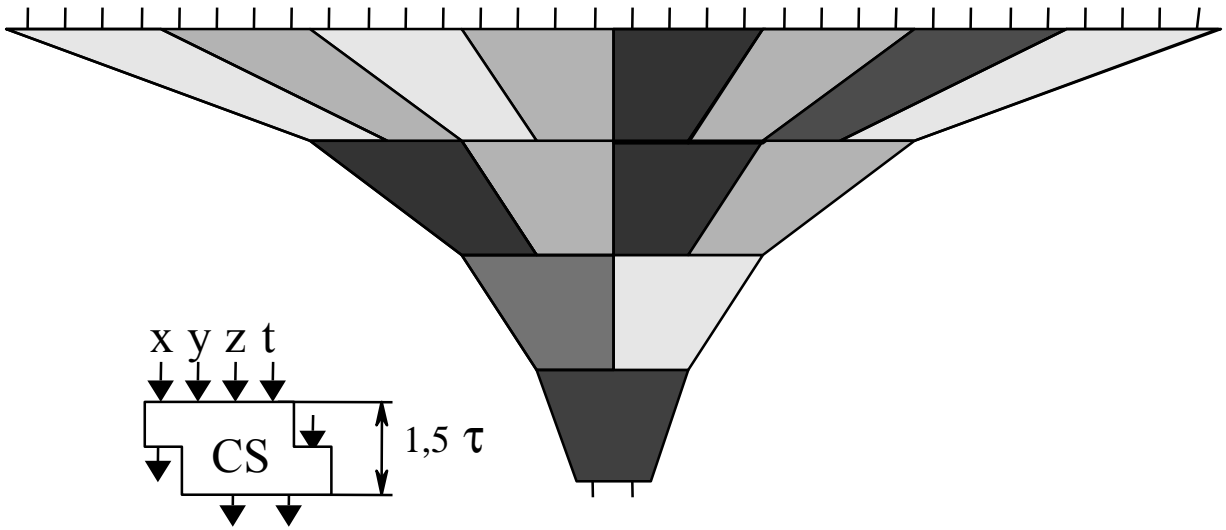




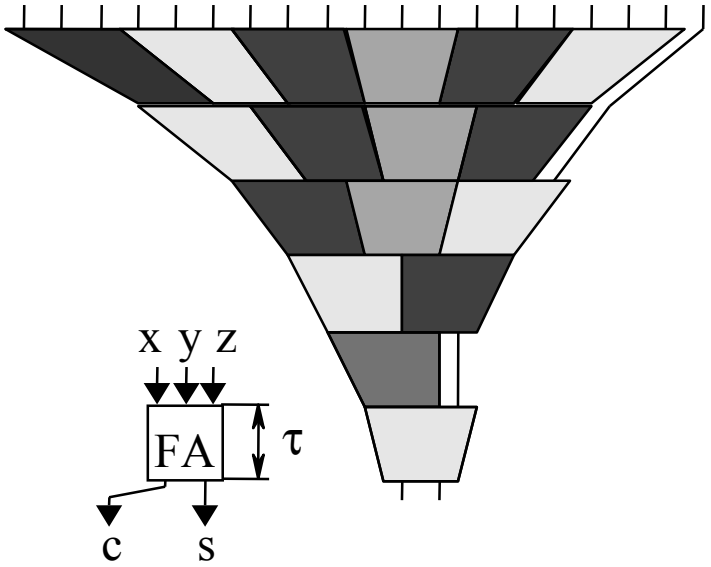




Comparaison de réducteur

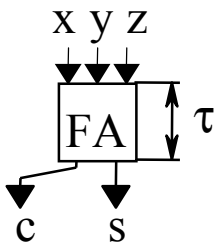


La retenue horizontale est cachée

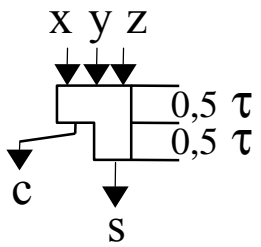


Exemple de multiplication 8 bits par 8 bits:

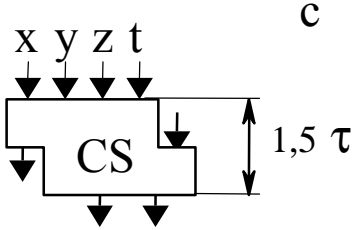
Arbre de Wallace, modèle 1 délai = 4 τ



Arbre de Wallace, modèle 2 délai = 3,5 τ



Réducteur 4 donne 2 (CS) délai 3 τ



Code de Booth

code de B	$B = \sum_{i=0}^{n-1} b_i * 2^i$	$B = \sum_{i=0}^{(n-2)/2} c_i * 4^i$
produit A * B	$A*B = \sum_{i=0}^{n-1} A*b_i*2^i$	$A*B = \sum_{i=0}^{(n-2)/2} A*c_i*4^i$
valeurs des chiffres	$b_i \in \{0,1\}$	$c_i \in \{-2, -1, 0, 1, 2\}$
nombre d'opérations	n-1	$(n-2)/2$
type d'opération	addition	addition/ soustraction

Conversion conventionnel en code de Booth

$$B = \sum_{i=0}^{n-1} b_i * 2^i \quad \Rightarrow \quad B = \sum_{i=0}^{\frac{n-2}{2}} c_i * 4^i$$

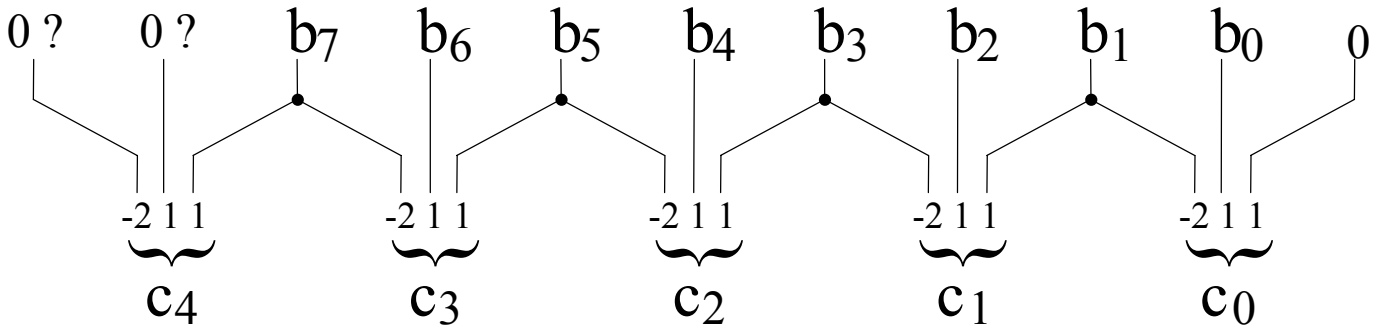
Soient $B'' = \sum_{i=0}^{\frac{n-2}{2}} b_{2i} * 2^{2i}$ termes de rang pair

$B' = \sum_{i=0}^{\frac{n-2}{2}} b_{2i+1} * 2^{2i+1}$ termes de rang impair

$B = B'' + B' = B'' + (2*B') - 2*(1/2*B')$

$(2*B')$: décalé à gauche

$(1/2*B')$: décalé à droite



la valeur d'un chiffre c_i est la somme pondérée de ses bits

$$c_i = -2\, b_{2i+1} + b_{2i} + b_{2i-1} \qquad c_i \in \{-2, -1, 0, 1, 2\}$$

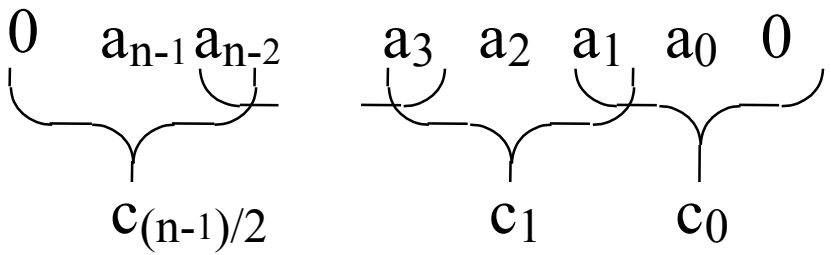
La somme pondérée des chiffres qui entrent est égale à la somme pondérée des chiffres qui sortent

Remarques sur le code de Booth

Entiers positifs

$$A = \sum_{i=0}^{n-1} a_i \, 2^i$$

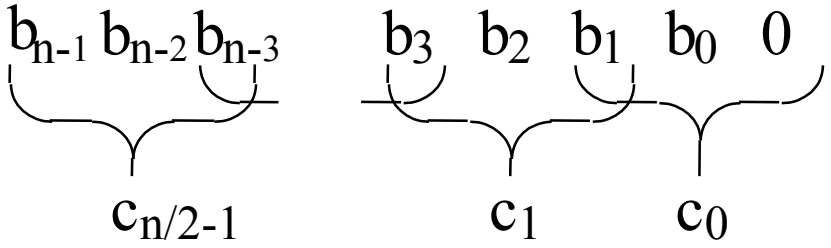
$\Leftrightarrow$



Entiers signés

$$B = -b_{n-1} 2^{n-1} + \sum_{i=0}^{n-2} b_i \, 2^i$$

$\Leftrightarrow$

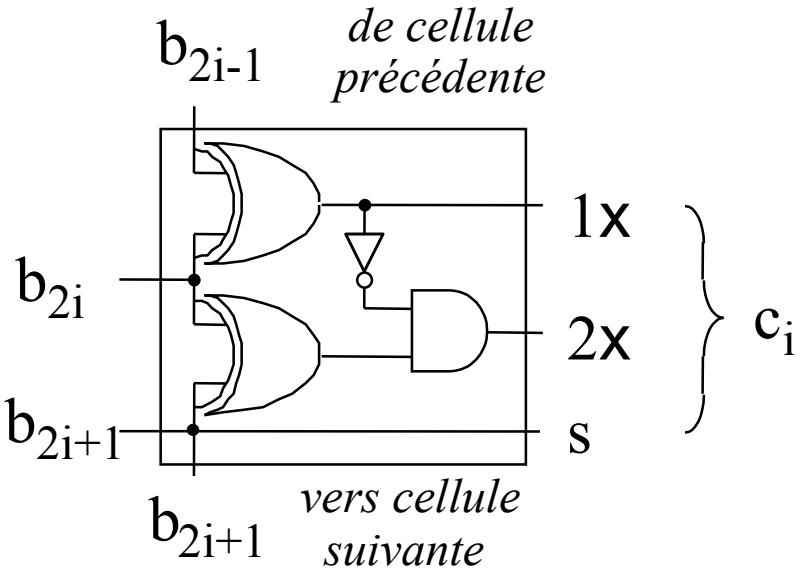


La notation de Booth est redondante: $(0\ 2)_4 = (1\ \overline{2})_4$

(c'est la notation symétrique de redondance minimale en base 4)

Réécriture des chiffres en signe/valeur absolue

b_{2i+1}	b_{2i}	b_{2i-1}	c_i	1x	2x	s
0	0	0	0	0	0	0
0	0	1	1	1	0	0
0	1	0	1	1	0	0
0	1	1	2	0	1	0
1	0	0	-2	0	1	1
1	0	1	-1	1	0	1
1	1	0	-1	1	0	1
1	1	1	-0	0	0	1



$$c_i * A = \left(s \oplus \left(A \wedge 1x \vee \left(2 * A\right) \wedge 2x\right)\right) + s$$

La multiplication d'un nombre par un chiffre est plus aisée mais la valeur d'un chiffre c_i n'est plus la somme pondérée de ses bits (code signe + valeur absolue).

Réduction de chiffres signés

1- Multiples du Multiplicande A

$2*A = -a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0\ 0$

$1*A = -a_6\ a_6\ a_5\ a_4\ a_3\ a_2\ a_1\ a_0$

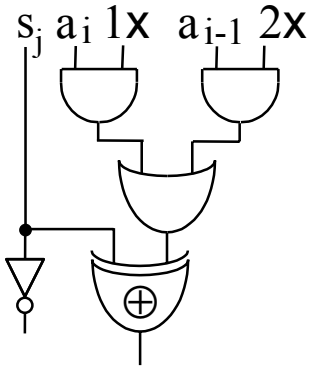
$0*A = 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0$

$-0*A = -1\ 1\ 1\ 1\ 1\ 1\ 1\ 1\ +1$

$-1*A = \overline{-a_6}\ \overline{a_6}\ \overline{a_5}\ \overline{a_4}\ \overline{a_3}\ \overline{a_2}\ \overline{a_1}\ \overline{a_0}\ +1$

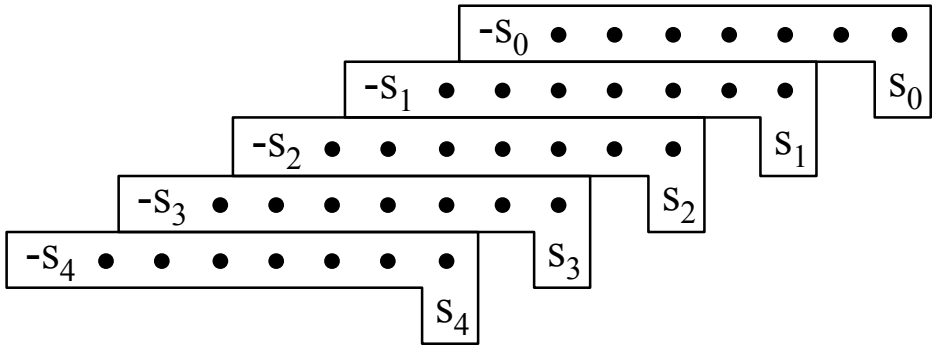
$-2*A = \overline{-a_6}\ \overline{a_5}\ \overline{a_4}\ \overline{a_3}\ \overline{a_2}\ \overline{a_1}\ \overline{a_0}\ 1\ +1$

$c_i*A = \boxed{-d_7\ d_6\ d_5\ d_4\ d_3\ d_2\ d_1\ d_0 \atop s}$



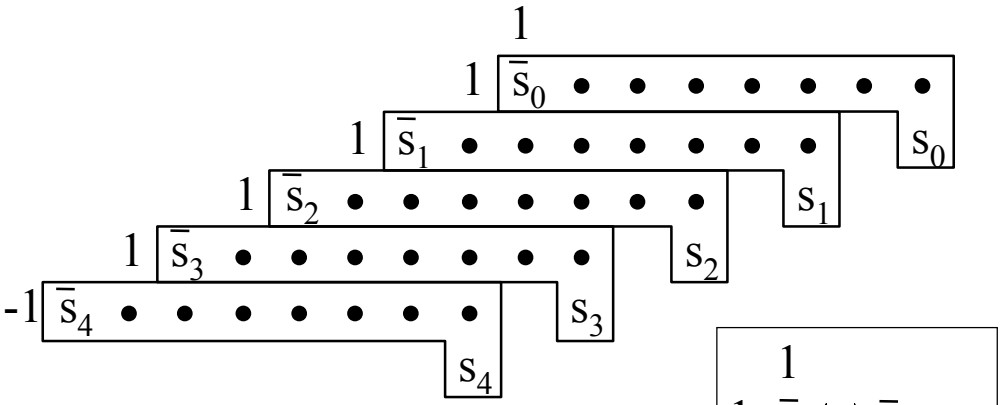
$d_i = s_j \oplus ((a_i \wedge 1x) \vee (a_{i-1} \wedge 2x))$

2- Addition des multiples



3- Propagation des signes pour l'addition

$-s_4\ 0\ -s_3\ 0\ -s_2\ 0\ -s_1\ 0\ -s_0 = -1\ \overline{s}_4\ 1\ \overline{s}_3\ 1\ \overline{s}_2\ 1\ \overline{s}_1\ 1\ \overline{s}_0\ +1$

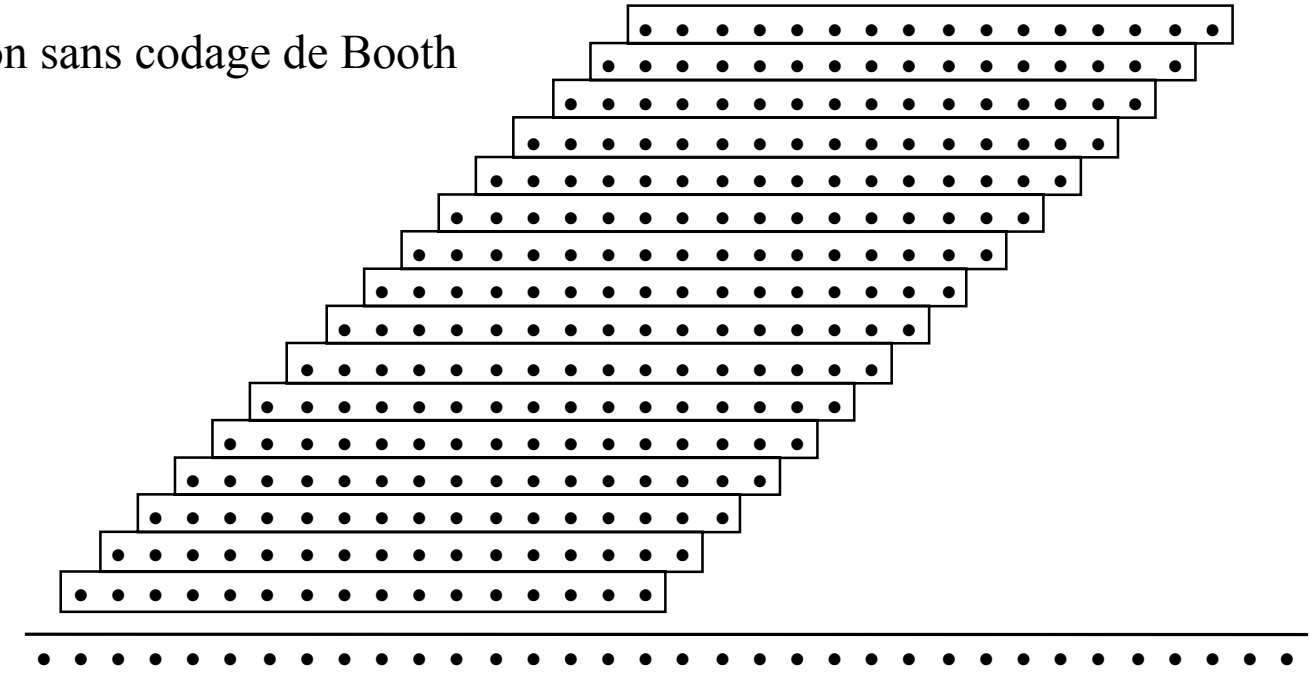


$$\begin{array}{l} 1 \\ 1\ \overline{s}_0 \end{array} \Leftrightarrow \overline{s}_0 s_0 s_0$$

Comparaison Booth-conventionnelle

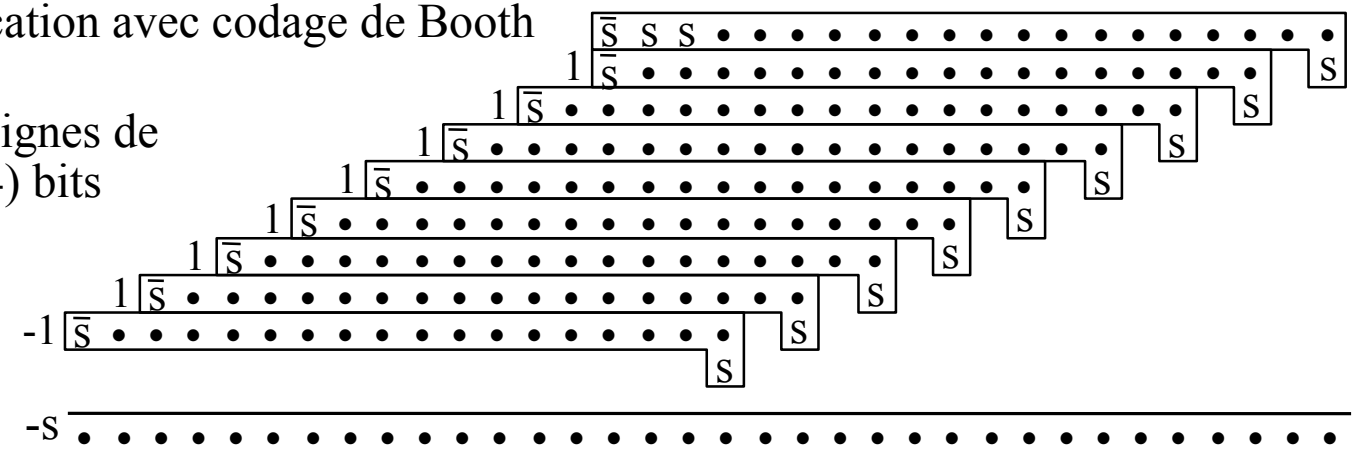
Multiplication sans codage de Booth

n lignes de
n bits



Multiplication avec codage de Booth

n/2 lignes de
(n+4) bits



Multiplieur rapide

multiplicande $A = \sum_{i=0}^{n-1} a_i * 2^i$

multiplieur $B = \sum_{i=0}^{n-1} b_i * 2^i$

Conversion de standard à "code de Booth"

$\sum_{i=0}^{\frac{n-2}{2}} c_i * 4^i$

Arbre binaire d'additionneurs sans propagation

$a_i, b_i, p_i \in \{0,1\}$

$c_i \in \{-2,-1,0,1,2\}$

$s_i \in \{0,1,2\}$

$P = \sum_{i=0}^{2n-1} s_i * 2^i$

conversion de "carry save" à standard

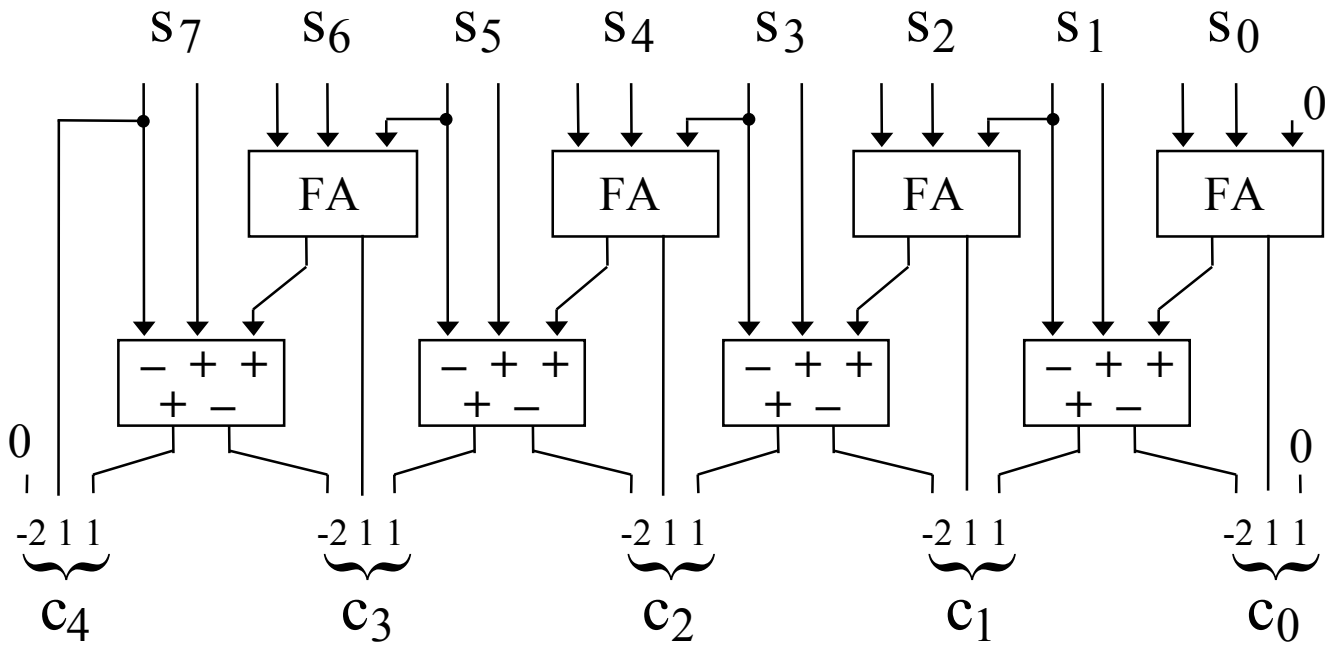
produit $P = A * B = \sum_{i=0}^{2n-1} p_i * 2^i$

Conversion de CS à code de Booth

Si le produit est un addende, un multiplieur ou un dividende, il n'est pas nécessaire de le convertir le « carry save » en notation conventionnelle.

$$P = \sum_{i=0}^{n-1} s_i * 2^i \quad \Rightarrow \quad P = \sum_{i=0}^{\frac{n-2}{2}} c_i * 4^i$$

$$s_i \in \{0,1,2\}$$
$$c_i \in \{-2,-1,0,1,2\}$$



La somme pondérée des chiffres qui entrent est égale à la somme pondérée des chiffres qui sortent