Master ACSI

Architectures pour le Traitement Numérique

Cours Habib MEHREZ

Laboratoire d’Informatique de Paris 6, Université Pierre et Marie Curie

4, place Jussieu, 75252 Paris cedex 5

MULTIPLICATION

Signe et grandeur

$$A = (-1)^{a_0} \sum_{i=-m}^{i=-1} a_i 2^i$$

$$B = (-1)^{a_0} \sum_{i=-m}^{i=1} b_i 2^i$$

$$A * B = (-1)^{a_0 + b_0} \sum_{i=-m}^{i=-1} \left(\sum_{j=-m}^{j=-1} a_i b_j 2^j \right) 2^i$$

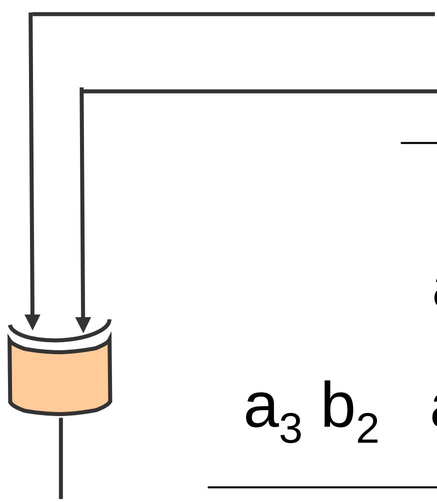
Complément à 2

$$A = (-a_0) + \sum_{i=-m}^{i=-1} a_i 2^i$$

$$B = (-b_0) + \sum_{i=-m}^{i=-1} b_i 2^i$$

$$A * B = a_0 b_0 - a_0 \sum b_j 2^j - b_0 \sum a_i 2^i + \sum \left(\sum a_i b_j 2^j \right) 2^i$$

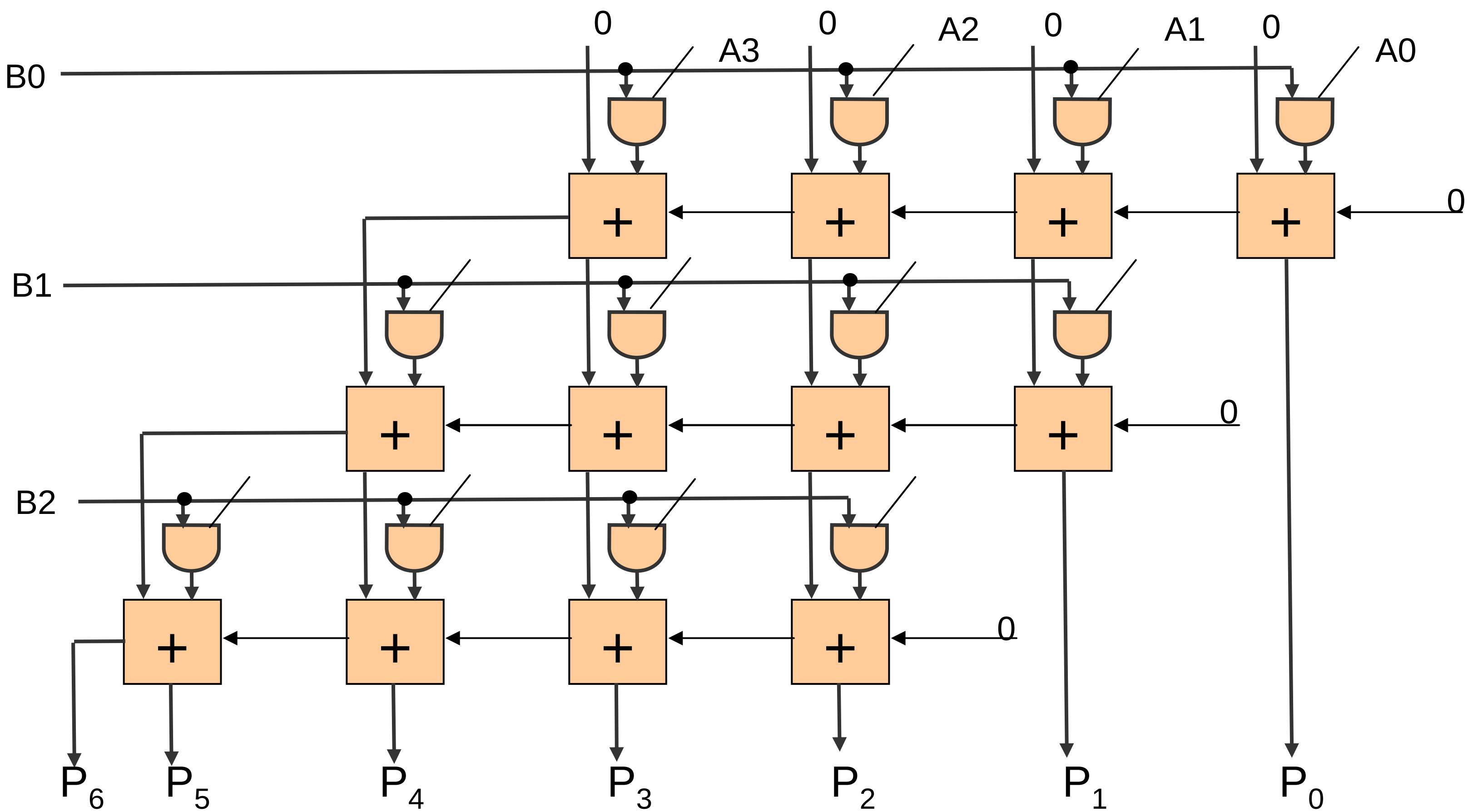
Multiplication naive (Signe et grandeur)



		(a_4)	a_3	a_2	a_1	a_0
			(b_3)	b_2	b_1	b_0
			$a_3\,b_0$	$a_2\,b_0$	$a_1\,b_0$	$a_0\,b_0$
		$a_3\,b_1$	$a_2\,b_1$	$a_1\,b_1$	$a_0\,b_1$	
	$a_3\,b_2$	$a_2\,b_2$	$a_1\,b_2$	$a_0\,b_2$		
P_6	P_5	P_4	P_3	P_2	P_1	P_0

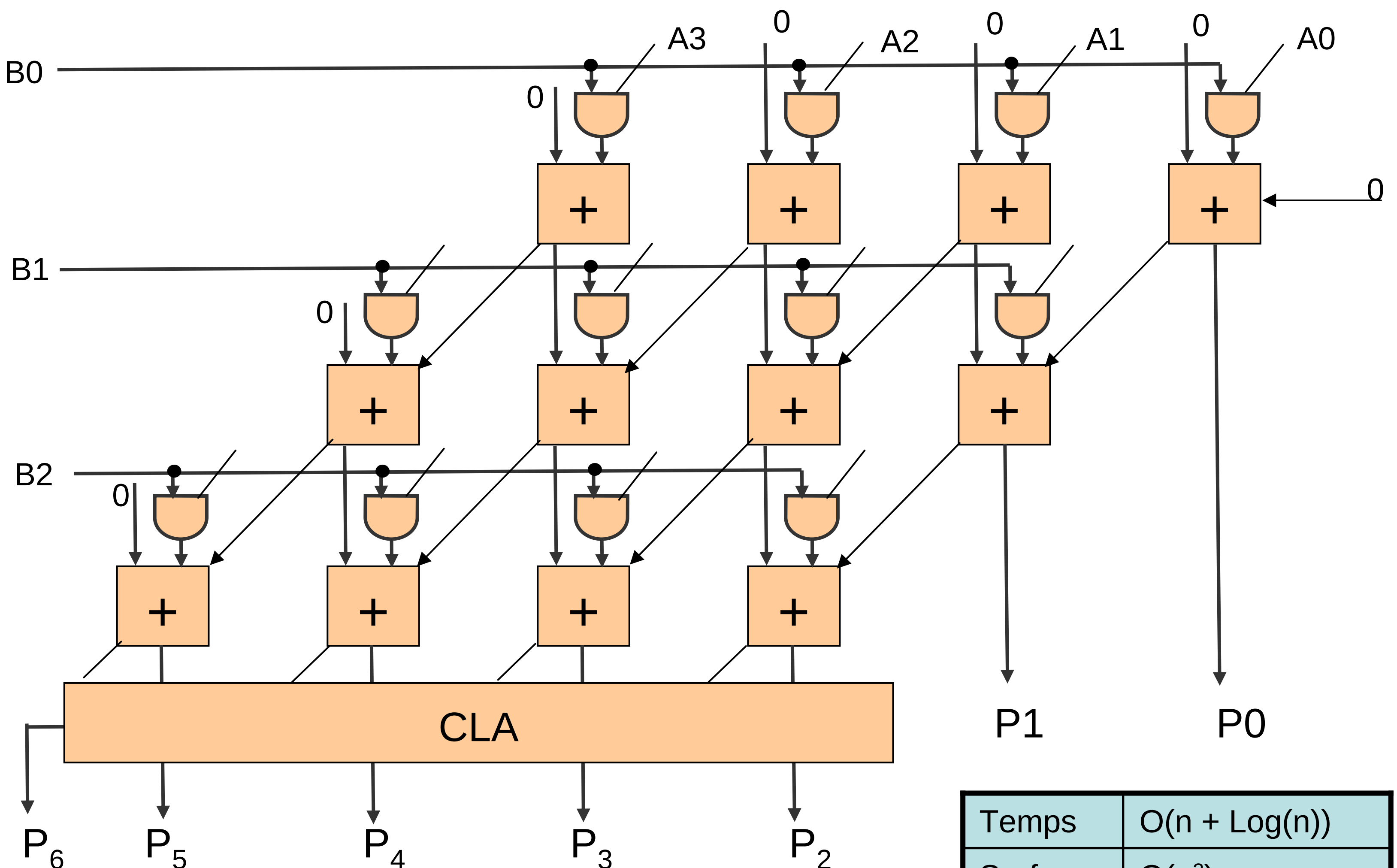
➔

Multiplication naive: Implantation



Temps	$O(n^2)$
Surface	$O(n^2)$

Multiplication BRAUN



Temps	O(n + Log(n))
Surface	O(n^2)

Multiplication BOOTH

Soit un entier B en complément à 2:

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

Avec b_{n-1} bit de signe de B

En considérant que $b_{-1} = 0$ (le 1er bit LSB ignoré, à droite de b_0),

Booth montre que B peut s'écrire:

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

Ainsi le produit B*A peut s'écrire:

$$B * A = \left(\sum_{i=0}^{i=n-1} (b_i - b_{i-1}) * 2^i\right) * A$$

On examine alors les transitions bi et on multiplie par A

Multiplication BOOTH

$$B * A = \left(\sum_{i=0}^{i=n-1} (b_i - b_{i-1}) * 2^i \right) * A$$

$b_i - b_{i-1}$ = 0 ou 1 ou -1

b_i	b_{i-1}	OPERATION
0	0	Décalage + 0*A (Annulation)
0	1	Décalage + A (addition)
1	0	Décalage – A (soustraction)
1	1	Décalage + 0 (Annulation)

Transition 1: 0→1 donc soustraction

Transition 2: 1→1 donc annulation

Transition 3: 1→0 donc addition

4 (A)

*****

3 (B)

Exemple: 3 * 4=

+4 = 0 1 0 0

3 transitions

+ 3= * 0 1 1 0

1 1 1 1 0 0 Soustraction -4 => 1100 (-A)

0 0 0 0 0 Décalage simple (+0)

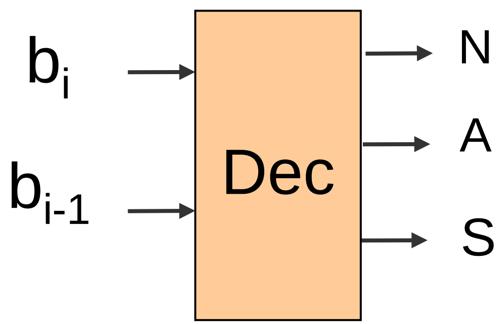
+ 0 1 0 0 Addition (+A)

0 0 1 1 0 0 = 12

: Extension de signe

Multiplication BOOTH: implantation

Dec: Décodeur de Booth



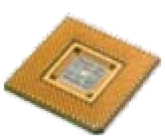
$$N = b_i b_{i-1} + b'_i b'_{i-1}$$

$$A = b'_i b_{i-1}$$

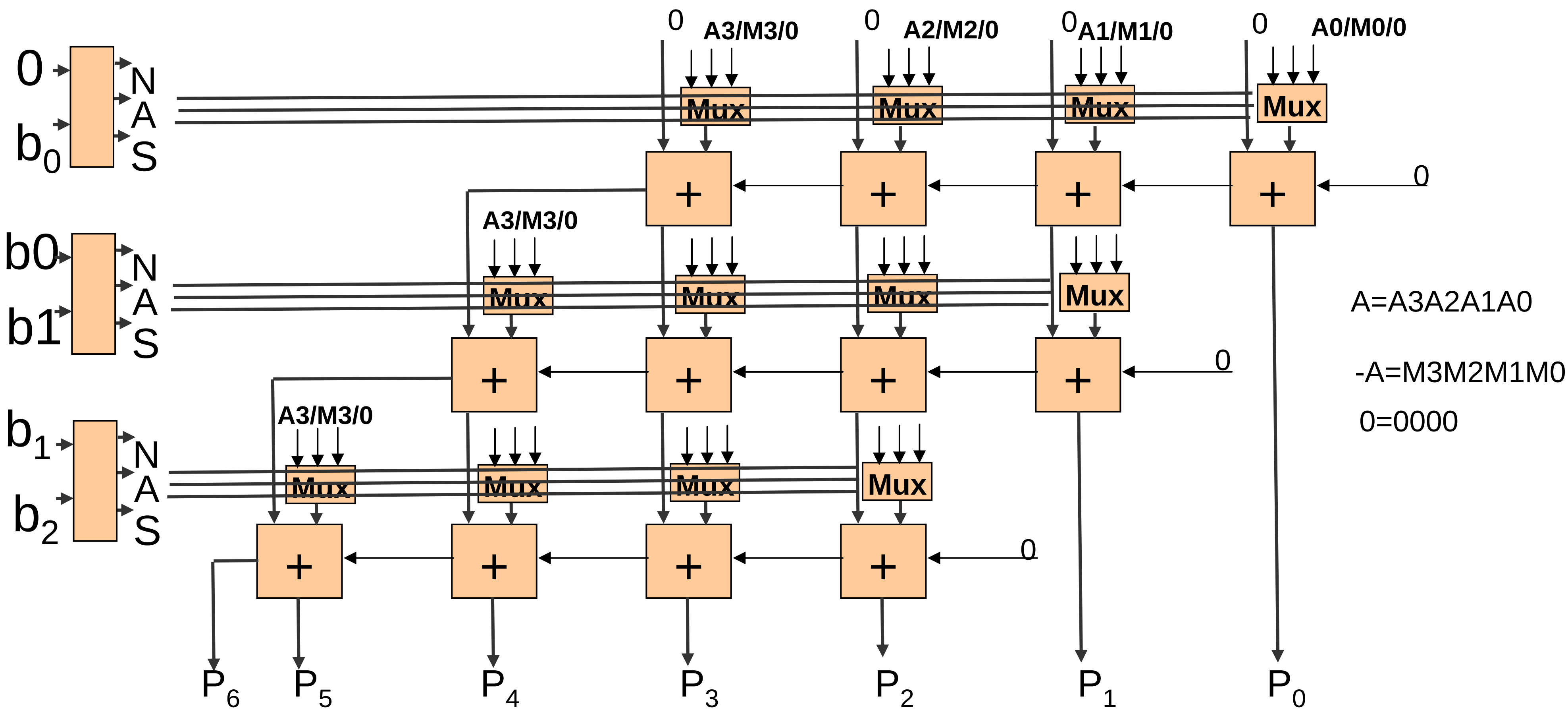
$$S = b_i b'_{i-1}$$

Notation: b'=NOT(b)

Temps	O(n + Log(n))
Surface	O(n^2)



Multiplication Booth: Implantation



Temps	$O(n^2)$
Surface	$O(n^2)$

Multiplication BOOTH Modifié

Soit un entier B en complément à 2:

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

Avec b_{n-1} bit de signe de B

En considérant que $b_{-1} = 0$ (le 1er bit LSB ignoré, à droite de b_0),

Booth montre que B peut s'écrire:

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

Avec i := i+2

Ainsi le produit B*A peut s'écrire:

$$B * A = \sum_{i=0}^{i=n-2} (-2b_{i+1} + b_i + b_{i-1}) * 2^i * A$$

On examine alors les transitions bi sur trois bits et on multiplie par A

Multiplication BOOTH modifié (base 4)

B_{i+1}	b_i	B_{i-1}	Code	Décalage	Annulation	Complément
0	0	0	0	0	1	0
0	0	1	+1(+A)	0	0	0
0	1	0	+1 (+A)	0	0	0
0	1	1	+2 (+2A)	1	0	0
1	0	0	-2(-2A)	1	0	1
1	0	1	-1(-A)	0	0	1
1	1	0	-1(-A)	0	0	1
1	1	1	0	0	1	0

Transitions 1: 000 donc annulation (+0)

Transition 2: 100 donc soustraction + décalage (-2A)

Transition 3: 001 donc addition (+A)

12

*

8

Exemple: 8 * 12

0 1 1 0 0

0 0 1 0 0 0 0 Rajout obligatoire d'un «0»

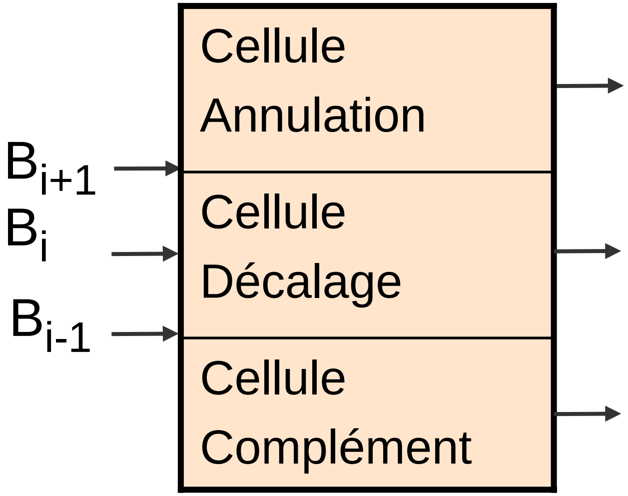
0 0 0 0 0 0 0 0 0 0 Annulation

1 1 1 0 1 0 0 Complémentation + décalage

0 0 1 1 0 0 addition

0 0 0 1 1 0 0 0 0 0 = 96

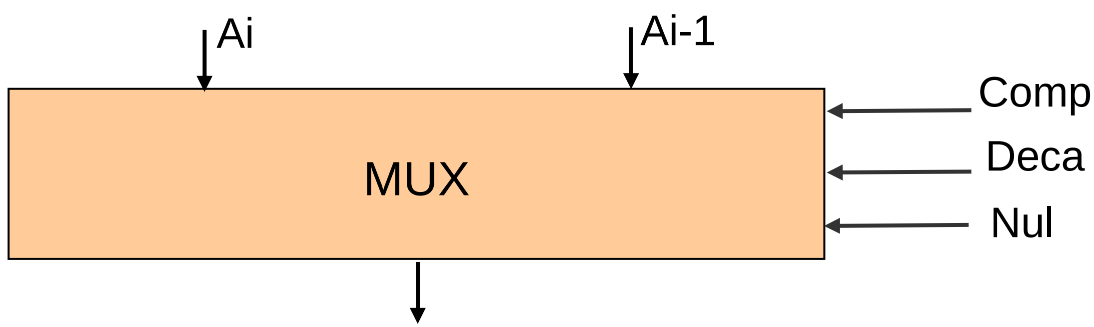
Multiplication BOOTH modifié: cellules de codage



$Nul = B_{i+1}(B'_i + B'_{i-1})$

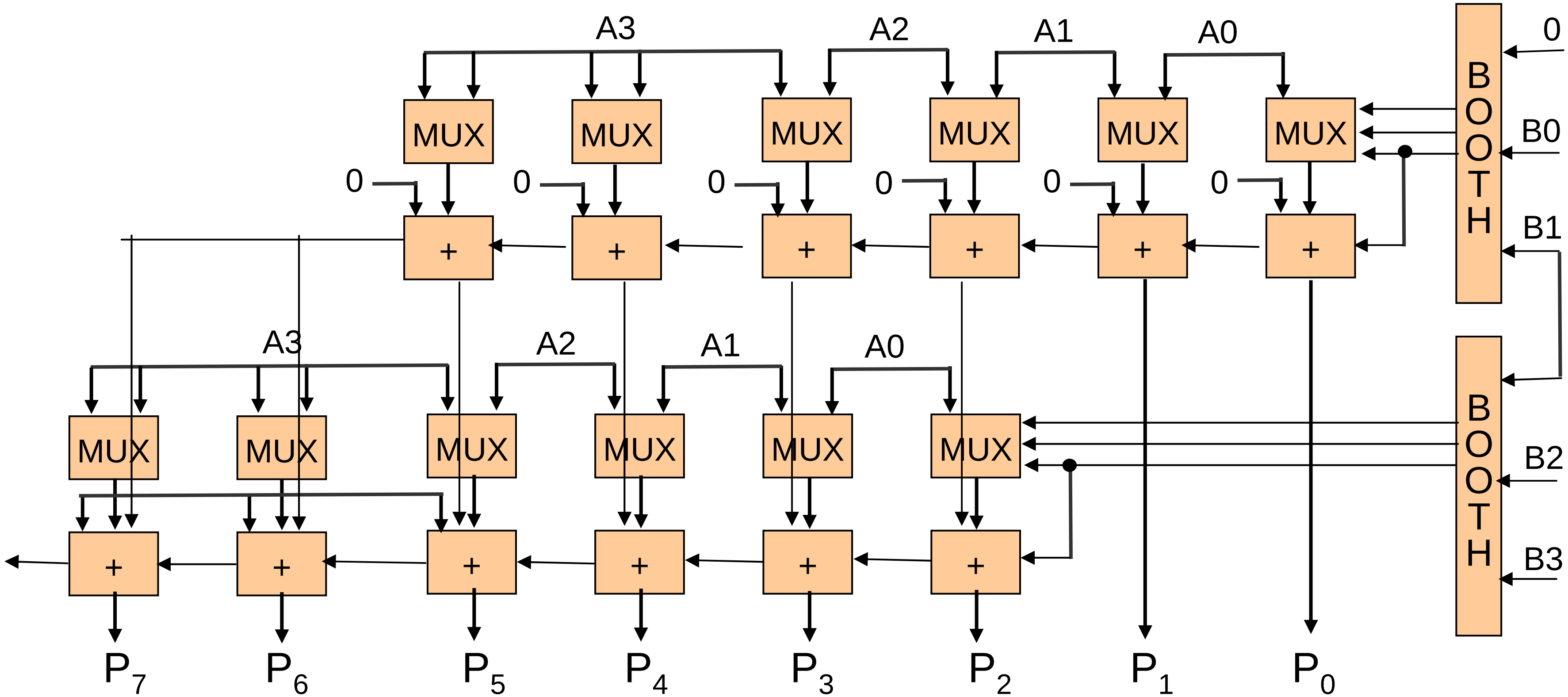
$Comp = B_{i+1}(B_i + B_{i-1})$

$Deca = B_{i+1}B_i B_{i-1} + B_{i+1}B'_i B'_{i-1}$



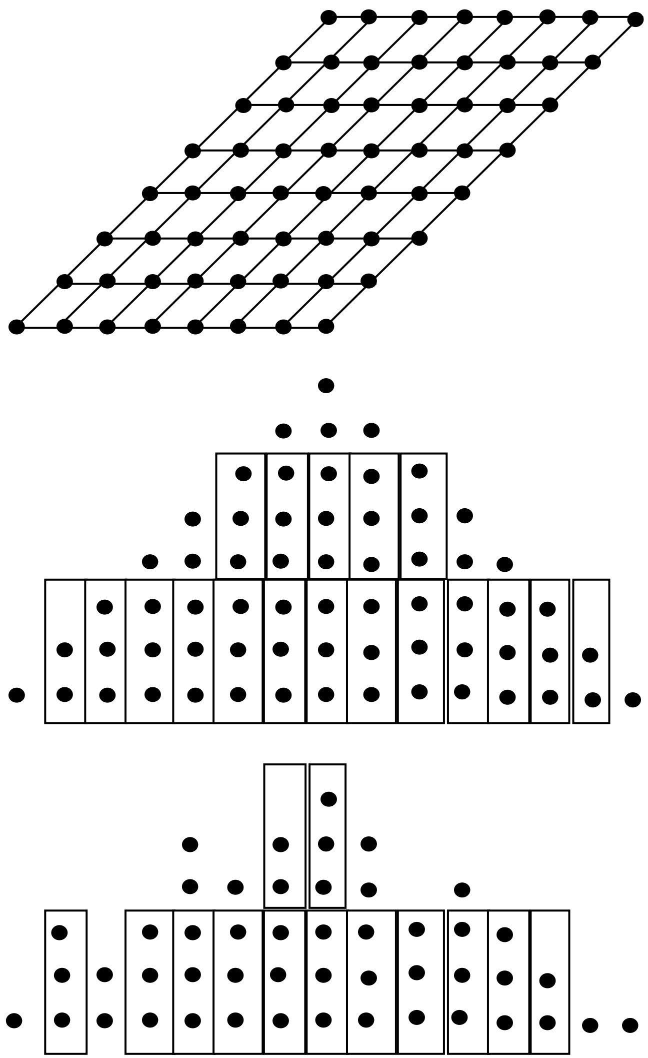
$M = \overline{Nul} \,.\, (Comp \oplus (Ai.\, \overline{Deca} + Ai\text{-}1 \,.\, Deca))$

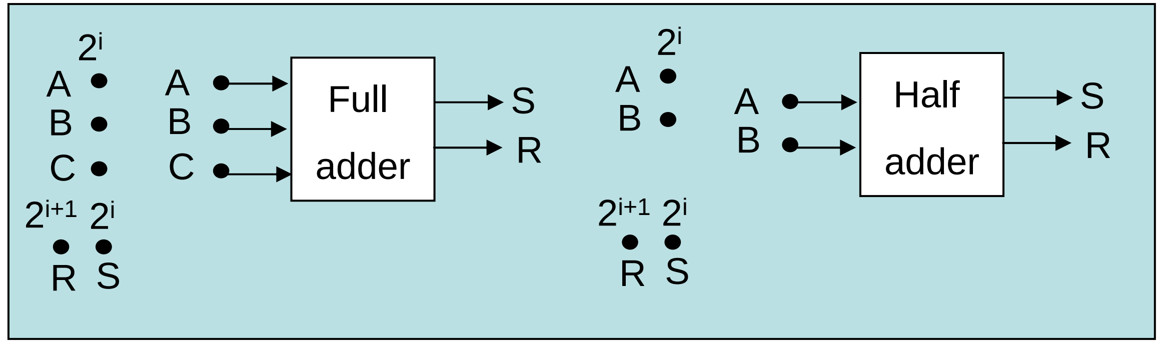
Multiplication BOOTH modifié: implémentation



Temps	$O(n^2)$
Surface	$O(n^2)$

Multiplicateur de WALLACE



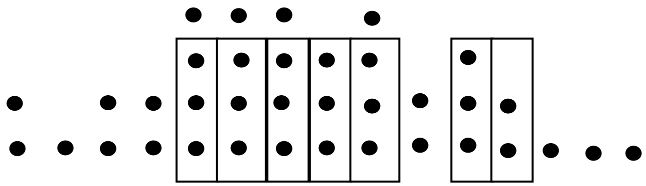


a) Représentation des produits (8*8)

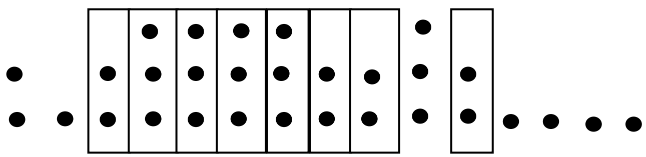
b) Réarrangement et sommation 1ère couche: chaque groupe de 3 bits de poids i additionnés par Full adder donne une somme S de poids i et une retenue de poids i+1. Le Half adder n'additionne que deux bits et donne aussi une somme S de poids i et une retenue de poids i+1

c) Sommation 2^e couche

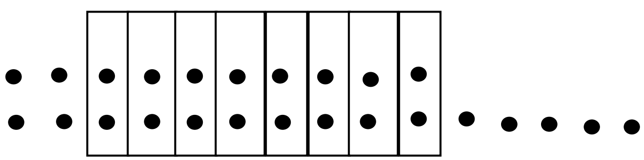
Multiplicateur de WALLACE



c) Sommation 3e couche



d) Sommation 4e couche

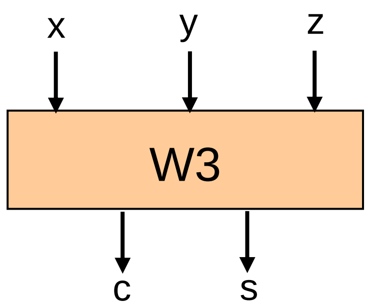


e) Sommation finale

Voir un autre exemple p 21

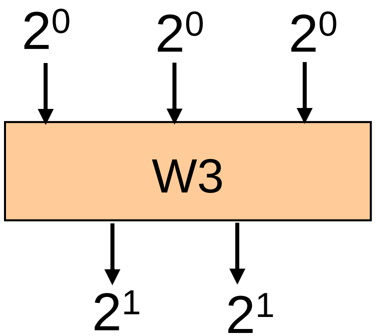
Arbre de WALLACE

- Arbre 3 entrées => 2 sorties

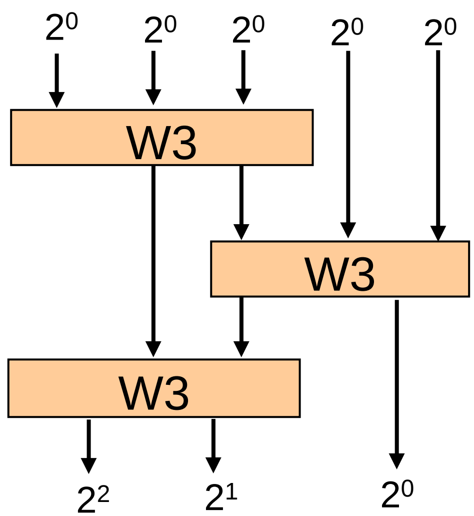


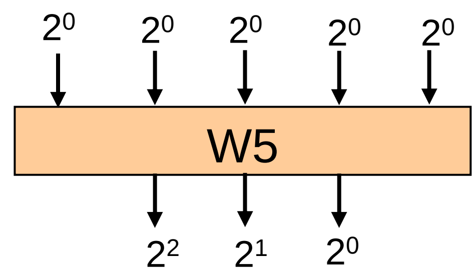
$S = x \oplus y \oplus z$

$C = Maj(x,y,z)$



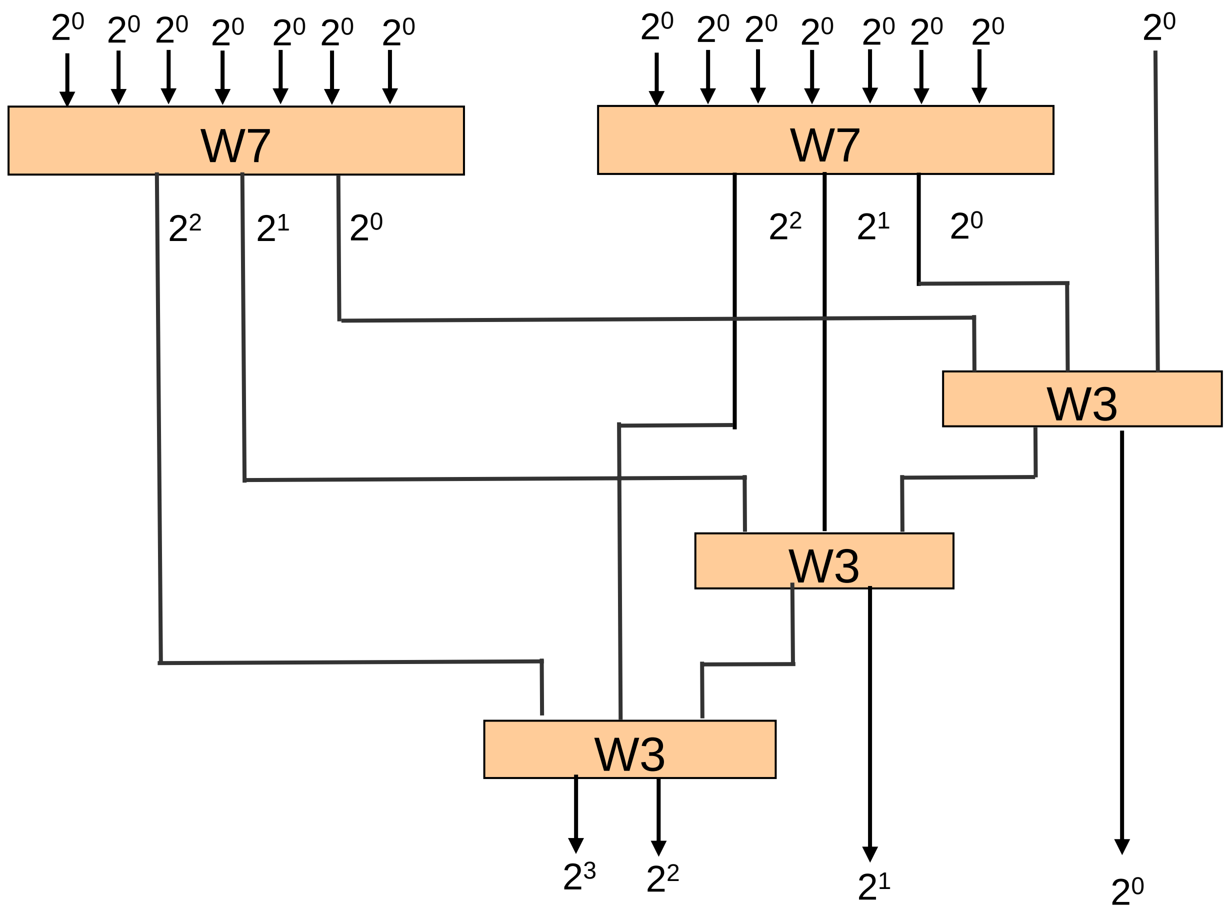
- Arbre 5 entrées => 3 sorties



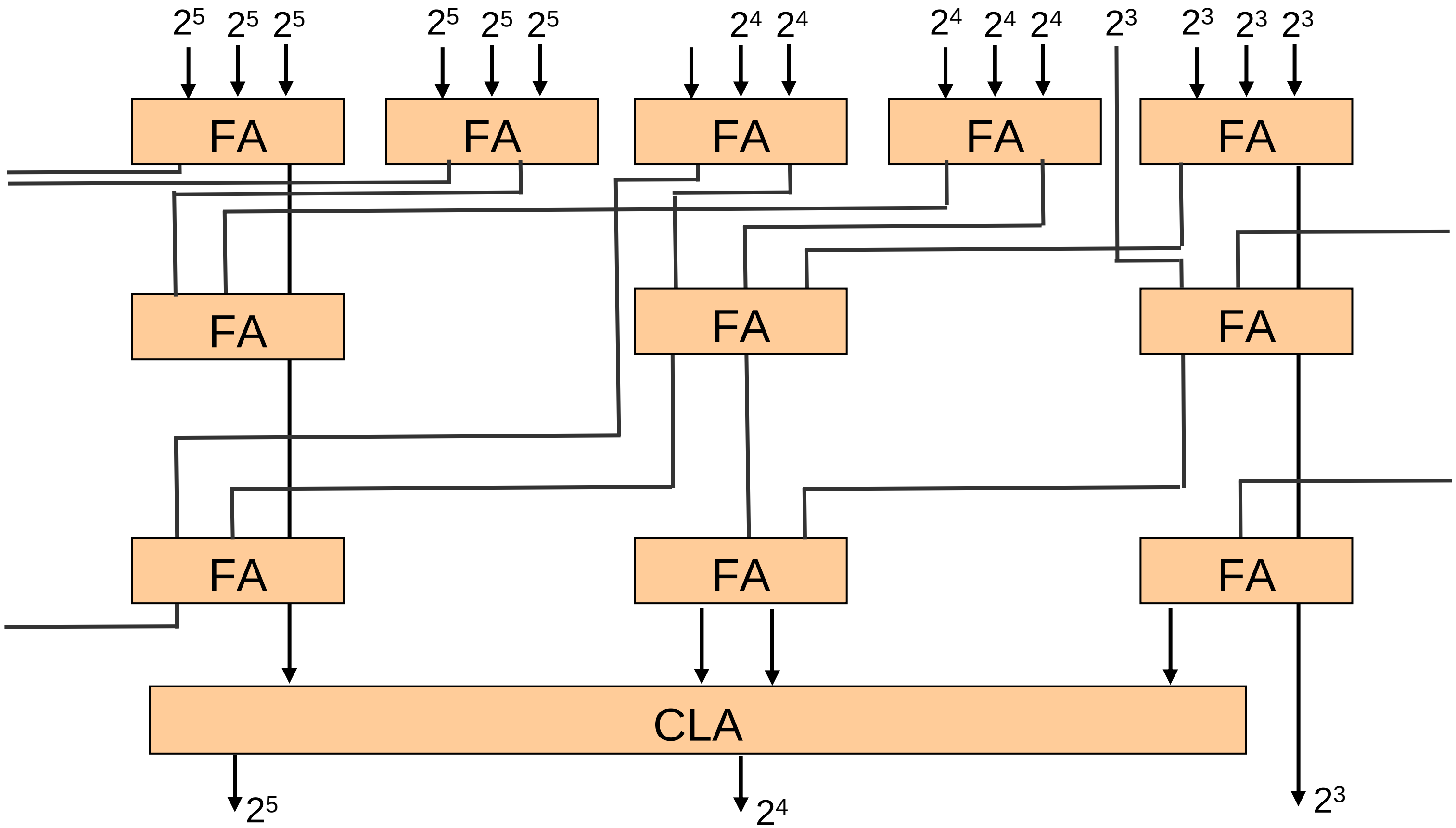


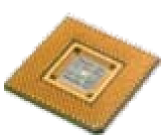
Arbre de WALLACE

- Arbre 15 entrées => 4 sorties



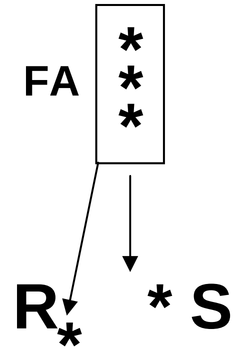
Implémentation de WALLACE

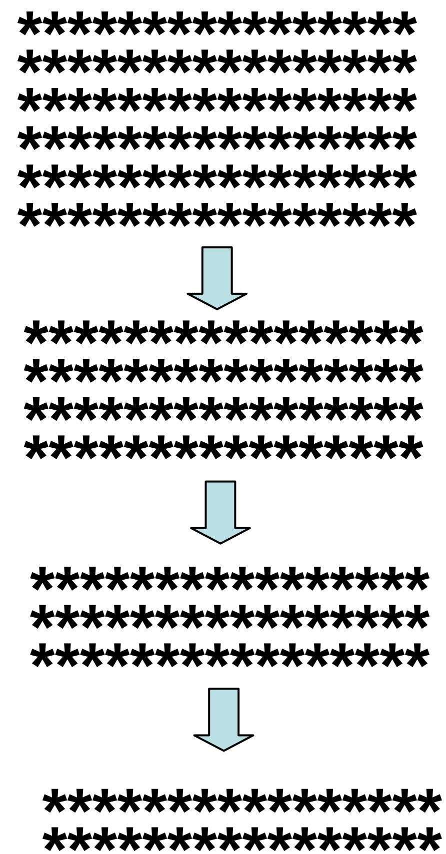




Implémentation de WALLACE

Progression en $\log_{3/2}$





- $u_0 = 2; u_{n+1} = \left\lfloor \frac{3}{2} u_n \right\rfloor$
- Soit la suite :

  $u_0 = 2, u_1 = 3, u_2 = 4, u_3 = 6, u_4 = 9, u_5 = 13, \ldots$

Temps	$O(\log_{3/2}(n))$, avec n nombre d'opérandes
Surface	$< O(n^2)$

Méthode de Fadavi-Ardekani

Sommation à effectuer

```
            1 0 0 1 0
        0 1 0 1 1
    1 0 1 0 0
0 1 1 1 0
_____________________
```

Solution avec extention de signe :

```
1 1 1 1 1 1 1 0 0 1 0
0 0 0 0 0 1 0 1 1
1 1 1 0 1 0 0
0 1 1 1 0
_____________________
0 1 0 1 1 0 1 1 1 1 0
```

Solution avec la méthode de Fadavi–Ardekani

Calcul de la constante de signe :

```
1 1 1 1 1 1 1 X X X X
1 1 1 1 1 X X X X
1 1 1 X X X X
1 X X X X
_____________________
0 1 0 1 0 1 1 0 0 0 0
```

Ajout de la constante de signe
et inversion des bits de signe :

```
            0 0 0 1 0
        1 1 0 1 1
    0 0 1 0 0
1 1 1 1 0
0 1 0 1 0 1 1 0 0 0 0
_____________________
0 1 ~~1~~ ~~1~~ 1 0 1 1 1 1 0
    0 1
```

Correction

Démonstration pour un seul nombre

Soit un nombre quelconque ZXXXX

Avec Z bit de signe en complément à 2

Sur 8 bits:

Si Z=0 on a 0000XXXX

Si Z=1 on a 1111XXXX

La constante de signe est 11110000

Il faut écrire: Z'XXXX avec Z'=NOT(Z)

Donc: 000Z'XXXX

+11110000

Alors si Z=0 donc Z'=1 et la somme est:

0000XXXX

Et si Z=1 donc Z'=0 et la somme est:

1111XXXX

Conclusions

Multiplication	Temps	Surface
Multiplication naive	$O(n^2)$	$O(n^2)$
Multiplication de BRAUN	$O(n)$	$O(n^2)$
Multiplication de BOOTH	$\sim(n+\log(n))$	$O(n^2)$
Multiplication de BOOTH modifiée	$\sim(n/2+\log(n))$	$\sim(n^2/2)$
Multiplication de WALLACE	$O(\log_{3/2}(n))$	$O(n^2)$