Arithmétique des Résidus

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

Opérateurs détectant les fautes

codes avec résidu modulo P

$\Rightarrow$ valeurs de P pratiquement utilisées $\in \{3, 7, 15, 31\}$

Opérateurs corrigeant les fautes

codes avec bi-résidu

$\Rightarrow$ pas d'applications pratiques (?)

Opérateurs rapides

processeurs de traitement de signal utilisant le code à résidu (RNS)

$\Rightarrow$ valeurs de P pratiquement utilisées telles que $\lceil \log_2 P \rceil \leq 12$

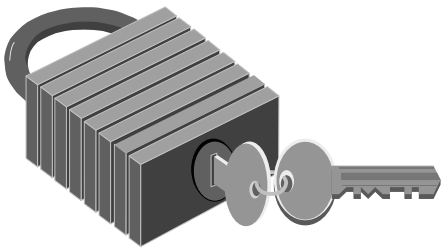
Opérateurs de Cryptographie

multiplication et exponentiation modulo P très grand

$\Rightarrow$ valeurs de P pratiquement utilisées: centaines à milliers de bits

Remarque:

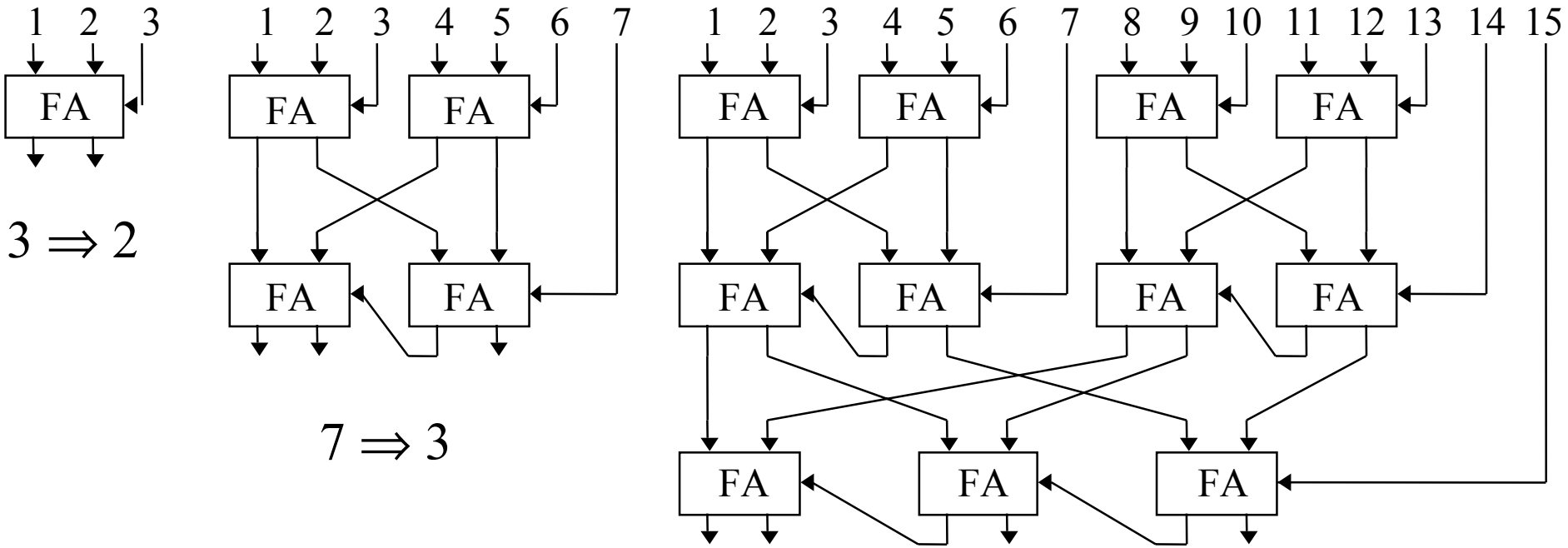
Dans cette partie nous nous limiterons à P petit



Arbre de Wallace

Compter les bits à 1 dans une chaîne

Tout assemblage cohérent de “FA” conserve la propriété:
La somme pondérée de ce qui sort est égale à la somme pondérée de ce qui entre.



Ce qui compte, c’est ce qu’on peut compter
Harpagon

Construction d'un arbre de Wallace

2^4	2^3	2^2	2^1	2^0	*poids des bits*
				22	*22 bits à réduire*
				7 FA	
			7	8	*15 bits restant à la 1ère étape*
			2 FA	2 FA	
		2	5	4	*11 bits restant à la 2eme étape*
			1 FA	1 FA	
		3	4	2	*9 bits restant à la 3eme étape*
		1 FA	1 FA	1 HA	
	1	2	3	1	*7 bits restant à la 4eme étape*
			1 FA		
	1	3	1	1	*6 bits restant à la 5eme étape*
		1 FA			
	2	1	1	1	*5 bits restant à la 6eme étape*
	1 HA				
1	1	1	1	1	*résultat de 5 bits* (22 $\Rightarrow$ 5)

Codes arithmétiques décimaux

Ces opérations sont elles correctes ?

391 * 972 = 390052

365481 + 44154 = 409535

365481 – 44154 = 331327

Soit $A = \sum_{i=0}^{n-1} a_i * 2^{10}$

A est multiple de 9 si ($a_{n-1} + a_{n-2} + \ldots + a_1 + a_0$) est multiple de 9.

A est multiple de 11
si $(-1)^{n-1}a_{n-1} + (-1)^{n-2}a_{n-2} + \ldots - a_1 + a_0$) est multiple de 11.

891 * 969 = 854379

Codes arithmétiques

On note $|A|_P$ le reste de la division entière de A par P.

$0 \leq |A|_P \leq P-1.$

Le nombre codé est $(A, |A|_P)$

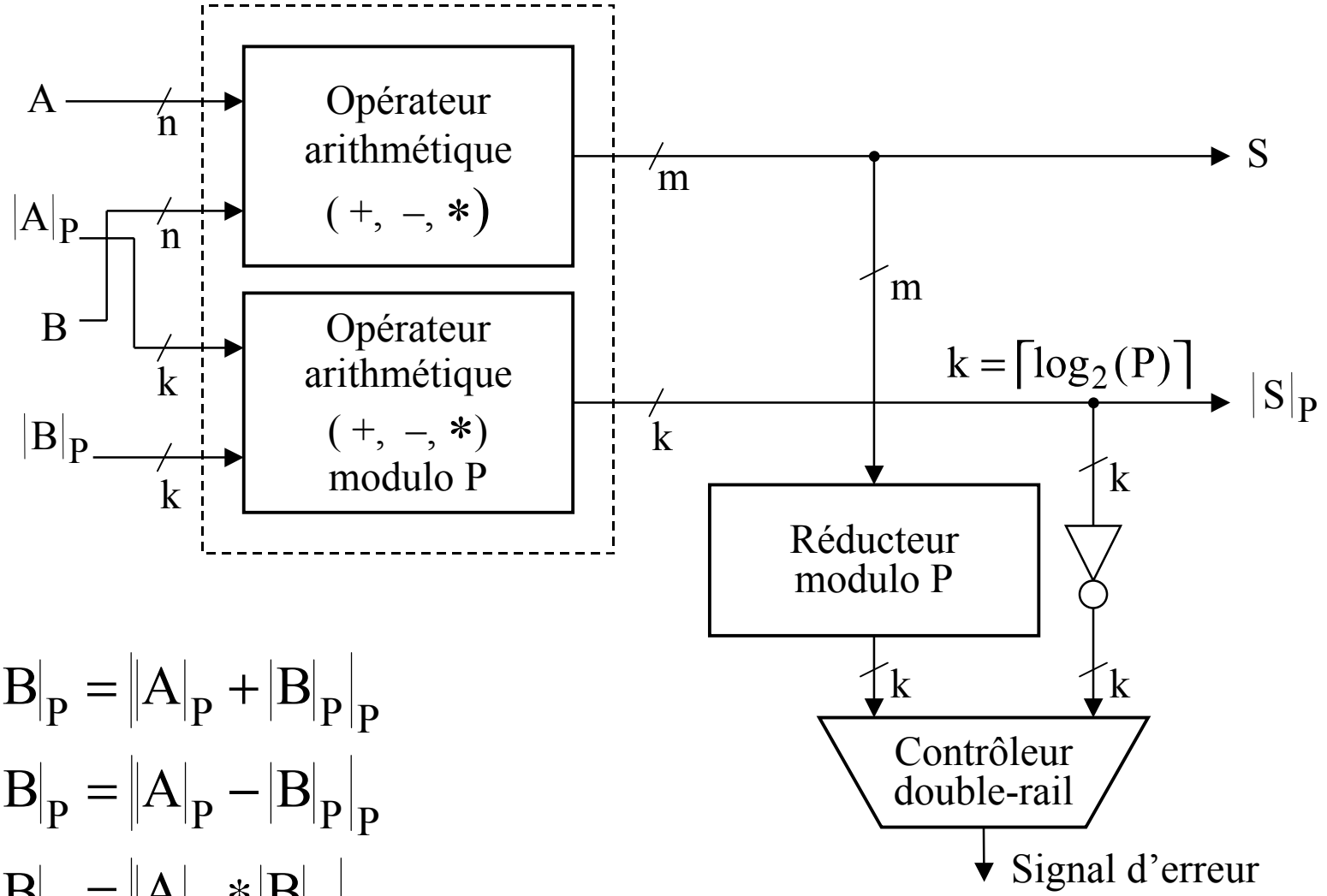
Pour les opérations d'addition, soustraction, multiplication, on a:

$$|A+B|_P = \left||A|_P + |B|_P\right|_P$$

$$|A-B|_P = \left||A|_P - |B|_P\right|_P$$

$$|A*B|_P = \left||A|_P * |B|_P\right|_P$$

Opérateurs arithmétiques auto-testables



$$|A+B|_P = ||A|_P + |B|_P|_P$$

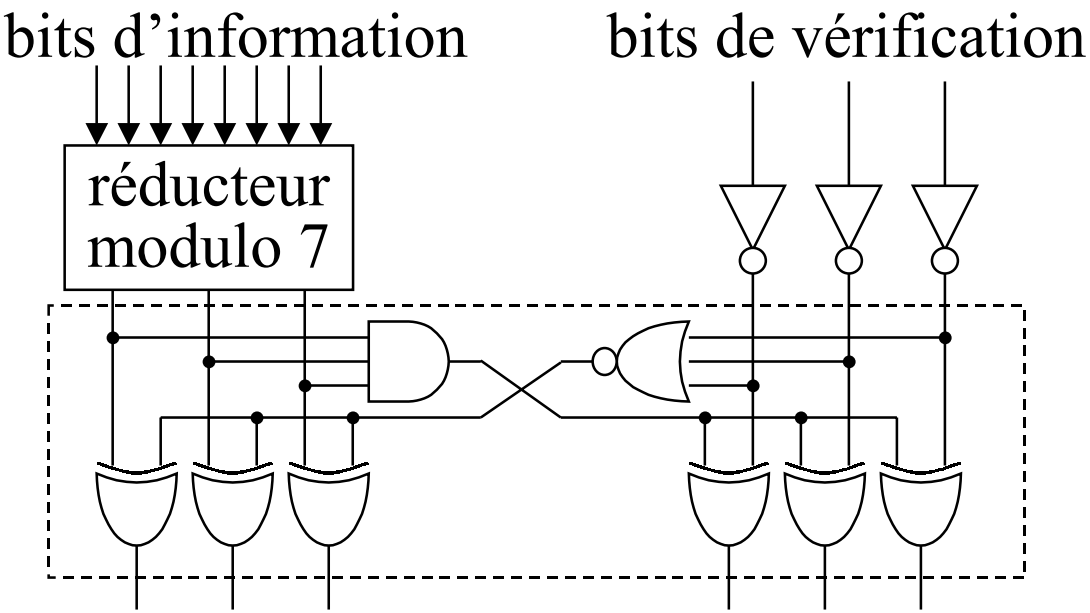
$$|A-B|_P = ||A|_P - |B|_P|_P$$

$$|A*B|_P = ||A|_P * |B|_P|_P$$

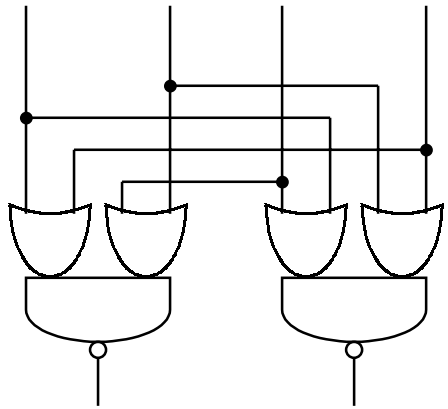
Contrôleur auto-testable

Un contrôleur auto-testable a au moins 2 sorties
En pratique on utilise le code "double rail" :

- 01 10 : opération correcte
- 00 11 : erreur



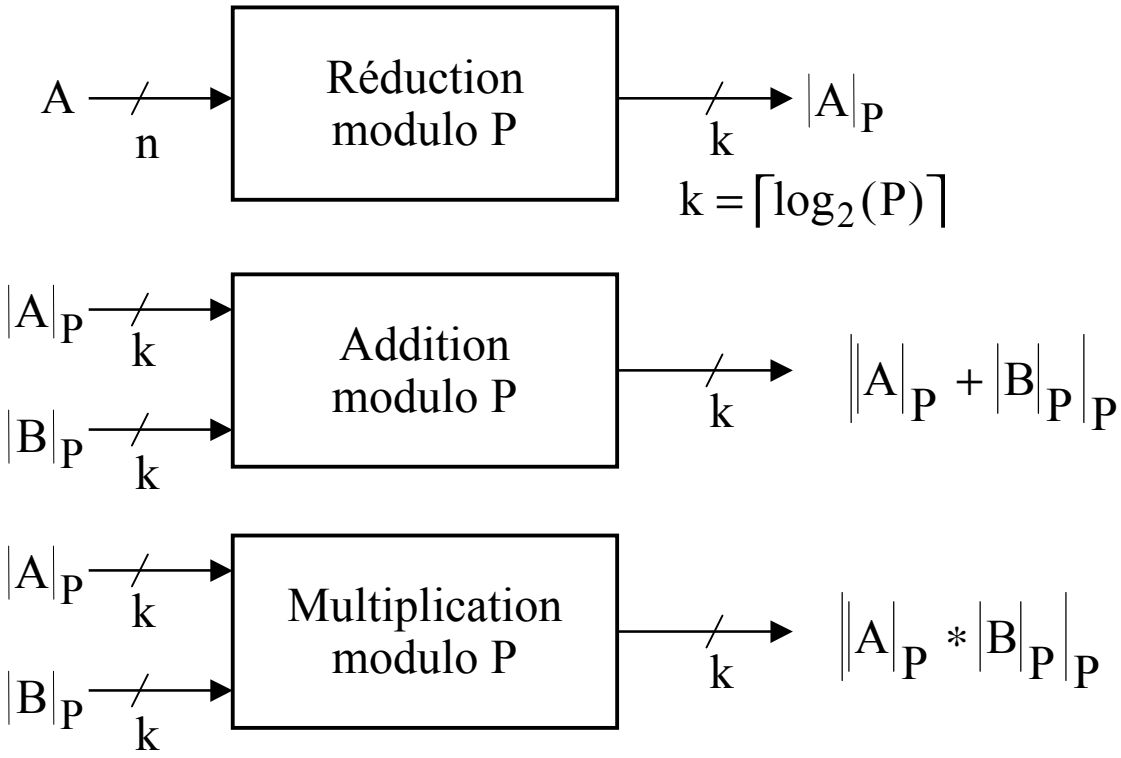
Convertisseur en code "double rail"



Porte "double rail"

Opérateurs modulo P

Réaliser les opérateurs suivants
en minimisant coût et délai



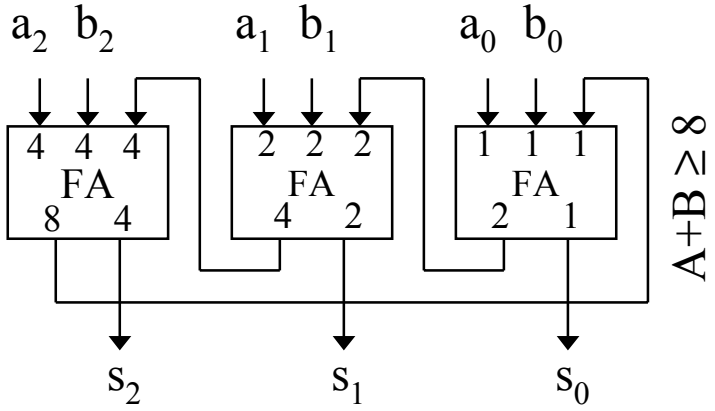
Poids des bits modulo P

$$\text{Soit } A = \sum_{i=0}^{11} a_i * 2^i$$

$$\text{Alors } |A|_P = \left| \sum_{i=0}^{11} a_i * |2^i|_P \right|_P$$

Poids des a_i	a_{11}	a_{10}	a_9	a_8	a_7	a_6	a_5	a_4	a_3	a_2	a_1	a_0
	2048	1024	512	256	128	64	32	16	8	4	2	1
$\|2^i\|_8$	0	0	0	0	0	0	0	0	0	4	2	1
$\|2^i\|_7$	4	2	1	4	2	1	4	2	1	4	2	1
$\|2^i\|_9$	-4	-2	-1	4	2	1	-4	-2	-1	4	2	1

Carry End around Adder



$$S = |A + B|_7$$

La valeur 0 de S a deux représentations: 000 ou 111

$$S = |A + B + c_0|_8$$

$$c_0 = c_3 = (A + B + c_0) \geq 8$$

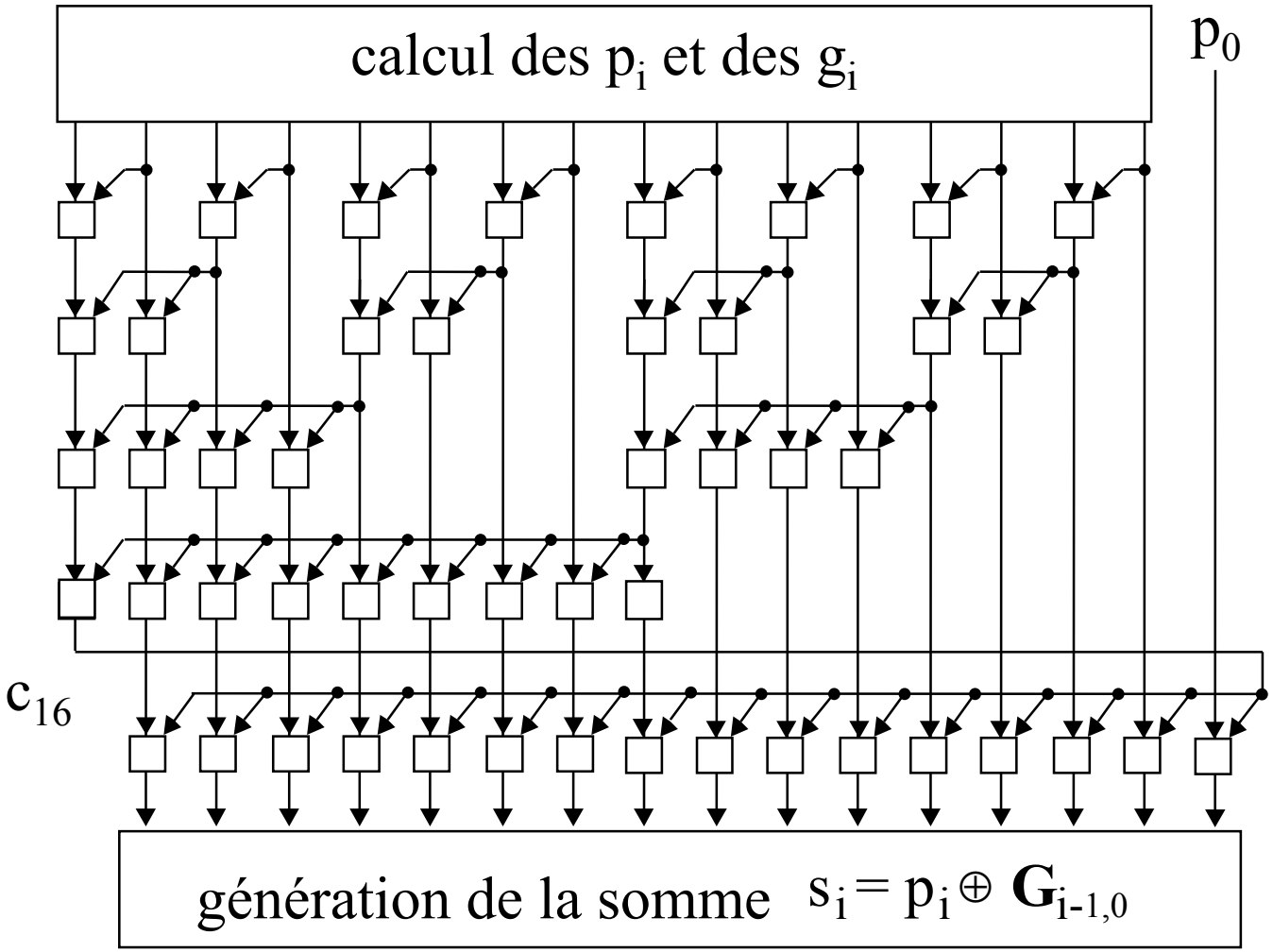
A+B	$\lvert A+B \rvert_7$
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	0 ou 7
8	1
9	2
10	3
11	4
12	5
13	6
14	7

Lorsque A + B = 7, S = 000 ou S = 111 suivant la valeur initiale de c_0

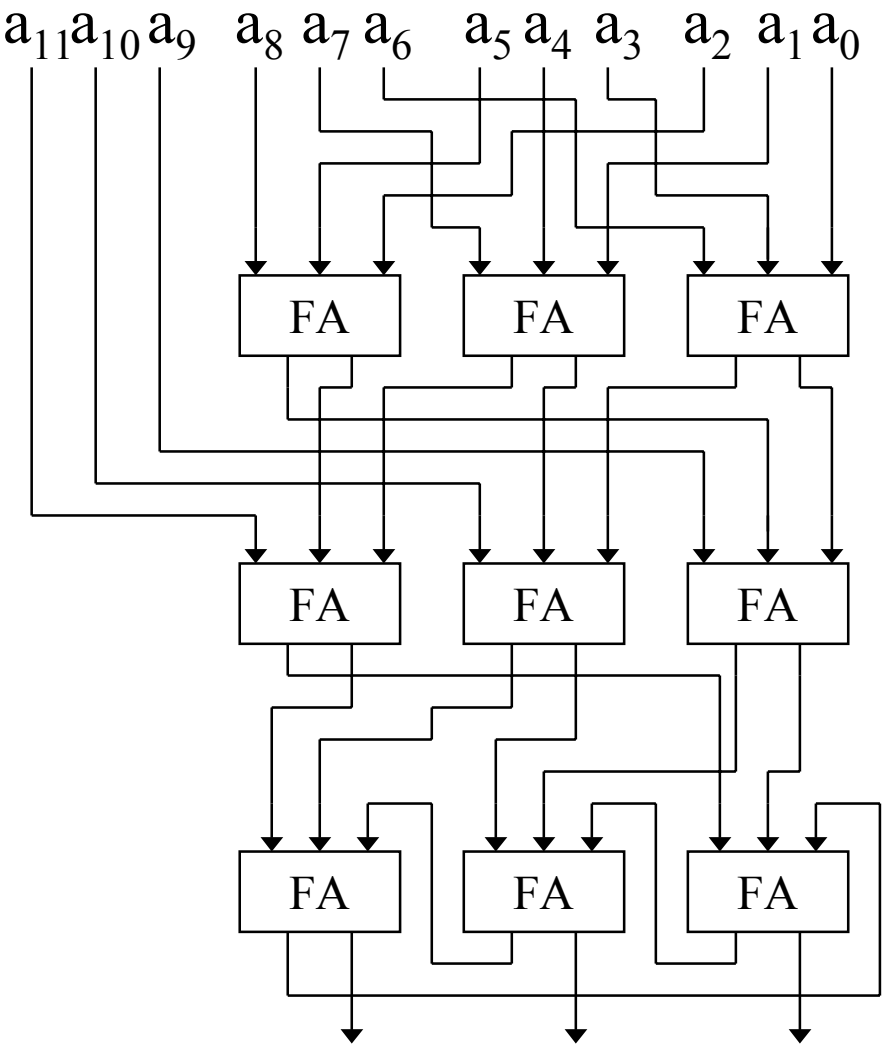
⇒ On admet une double représentation de la valeur 0,

⇒ il faut en tenir compte lors de la comparaison des résidus.

Carry End around Adder rapide



Réducteur modulo 7



Soit $A = \sum_{i=0}^{11} a_i * 2^i$

2^2	2^1	2^0	
4	4	4	*Bits à réduire*
FA	FA	FA	
3	3	3	*Bits 1ere étape*
FA	FA	FA	
2	2	2	*Bits 2eme étape*
FA ←	FA ←	FA ←	
1	1	1	*Bits 3eme étape*

Addition, soustraction, multiplication modulo 7

$$|A|_7 = \sum_{i=0}^{2} a_i * 2^i \qquad |B|_7 = \sum_{i=0}^{2} b_i * 2^i \qquad S = \sum_{i=0}^{2} s_i * 2^i$$

Bits à réduire

2^2	2^1	2^0
a_2 b_2	a_1 b_1	a_0 b_0

$$S = \left| |A|_7 + |B|_7 \right|_7$$

2^2	2^1	2^0
a_2 $\overline{b_2}$	a_1 $\overline{b_1}$	a_0 $\overline{b_0}$

$$S = \left| |A|_7 - |B|_7 \right|_7$$

$$-B = \overline{B} - 7$$

2^2	2^1	2^0
$a_0\ b_2$ $a_1\ b_1$ $a_2\ b_0$	$a_2\ b_2$ $a_0\ b_1$ $a_1\ b_0$	$a_1\ b_2$ $a_2\ b_1$ $a_0\ b_0$

$$S = \left| |A|_7 * |B|_7 \right|_7$$

Addition, soustraction, multiplication modulo 7

$$|A|_7 = \sum_{i=0}^{2} a_i * 2^i \qquad |B|_7 = \sum_{i=0}^{2} b_i * 2^i \qquad S = \sum_{i=0}^{2} s_i * 2^i$$

Tables de Pithagore

$$S = \left| |A|_7 + |B|_7 \right|_7 \qquad S = \left| |A|_7 - |B|_7 \right|_7 \qquad S = \left| |A|_7 * |B|_7 \right|_7$$

	0	1	2	3	4	5	6	7
0	Z	1	2	3	4	5	6	Z
1	1	2	3	4	5	6	Z	1
2	2	3	4	5	6	Z	1	2
3	3	4	5	6	Z	1	2	3
4	4	5	6	Z	1	2	3	4
5	5	6	Z	1	2	3	4	5
6	6	Z	1	2	3	4	5	6
7	Z	1	2	3	4	5	6	Z

	0	1	2	3	4	5	6	7
0	Z	6	5	4	3	2	1	Z
1	1	Z	6	5	4	3	2	1
2	2	1	Z	6	5	4	3	2
3	3	2	1	Z	6	5	4	3
4	4	3	2	1	Z	6	5	4
5	5	4	3	2	1	Z	6	5
6	6	5	4	3	2	1	Z	6
7	Z	6	5	4	3	2	1	Z

	0	1	2	3	4	5	6	7
0	Z	Z	Z	Z	Z	Z	Z	Z
1	Z	1	2	3	4	5	6	Z
2	Z	2	4	6	1	3	5	Z
3	Z	3	6	2	5	1	4	Z
4	Z	4	1	5	2	6	3	Z
5	Z	5	3	1	6	4	2	Z
6	Z	6	5	4	3	2	1	Z
7	Z	Z	Z	Z	Z	Z	Z	Z

Z = 0 ou 7 (double représentation)

Exercices

1- Combien faut il de de bits pour exprimer le nombre de bits à 1 dans une chaîne de n bits ?

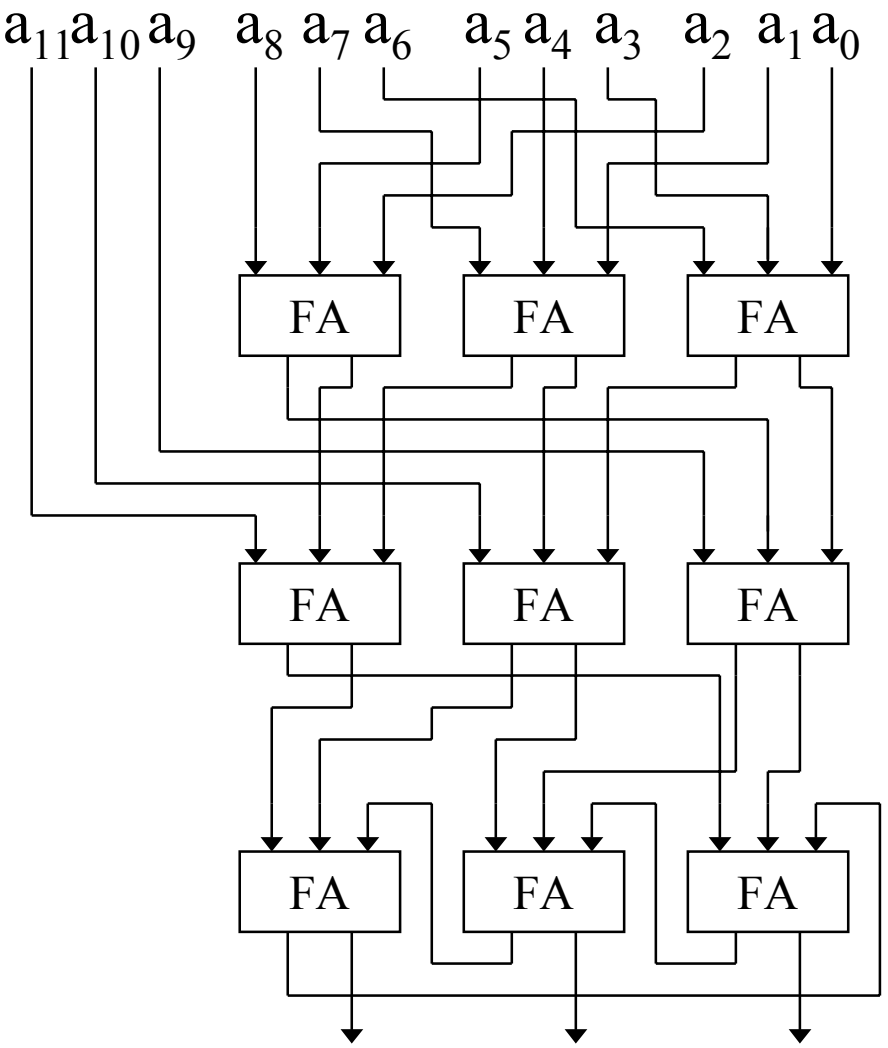
2- Combien faut il de cellule FA (avec toutes les entrées utilisées) pour compter les bits à 1 dans une chaîne de n bits ?

3- Donnez le circuit qui calcule le reste modulo 7 d'un entier signé représenté sur 13 bits:

$$A = -a_{12}\, 2^{12} + \sum_{i=0}^{11} a_i\, 2^i$$

4- Donner le circuit calculant la somme de deux nombres en "signe, valeur absolue"

Réducteur modulo 9



Soit $A = \sum_{i=0}^{11} a_i * 2^i$

2^2	2^1	2^0	
4	4	4	*Bits à réduire*
FA	FA	FA	
3	3	3	*Bits 1ere étape*
FA	FA	FA	
2	2	2	*Bits 2eme étape*
FA ←	FA ←	FA ←	
1	1	1	*Bits 3eme étape*

Périodicité (1)

La réduction modulo 7 est simple car $\left|2^{3k}\right|_7 = 1, \left|2^{3k+1}\right|_7 = 2, \left|2^{3k+2}\right|_7 = 4$

Définition: la période de P est le plus petit entier j tel que $\left|2^{j}\right|_P = 1$

Définition: la demi-période de P est le plus petit entier j tel que $\left|2^{j}\right|_P = -1$

P	3	5	7	9	11	13	15	17	19	21	23	25	27	29	31	33
demi période	1	2	-	3	5	6	-	4	9	-	-	10	9	14	-	5
période	2	4	3	6	10	12	4	8	18	6	11	20	18	28	5	10

P	35	37	39	41	43	45	47	49	51	53	55	57	59	61	63	65
demi période	-	18	-	10	7	-	-	-	-	26	-	9	29	30		6
période	12	36	12	20	14	12	23	21	8	52	20	18	58	60	6	12

Modulo P (P impair quelconque)

La demi-période n'existe pas pour tous les P, mais si elle existe elles vaut la moitié de la période, d'où son nom.

Le calcul du reste modulo P commence par une réduction sur un nombre de bits égal à la demi-période de P ou à défaut la période de P.