

TD d'architecture des ordinateurs II

Arithmétique binaire

Exercice 1 : Addition binaire de nombres non signés

1. Expliquer le fonctionnement de l'additionneur 4 bits de la figure 1.

Réponse _____

L'additionneur est composé de :

- 2 registres à décalage A (5 bits) et B (4 bits) qui gardent respectivement les opérandes (un signal *new* sur le registre permet de charger une valeur dans le bit 4) ;
- un additionneur ;
- un registre permettant de conserver la retenue (un signal *raz* permet de remettre à 0) ;
- un compteur 2 bits (de 00 à 11) (un signal *raz* permet de remettre à 0, *clock* incrémentation du compteur ; *fin* obtenu quand le compteur passe à 11).

Initialisation :

- entrée des opérandes dans a(3..0) à b (3..0) ;
- mise à zéro de C ;
- mise à zéro du compteur.

Un cycle de fonctionnement :

- une addition des bits de poids faible (positionnement de la retenue et du résultat à la sortie de l'additionneur) ;
- un signal *new* charge le résultat dans le bit a(4) du registre à décalage ;
- décalage des bits ;
- incrémentation du compteur.

_____ ♣

2. Pourquoi le registre à décalage A possède 5 bits ?

Réponse _____

Pour pouvoir stocker le résultat de l'addition sans perdre d'information sur a.

_____ ♣

3. Dérouler le fonctionnement pour les additions 6+3 et 6+11.

Réponse _____

compteur	a(4..0)	b(3..0)	retenue C	état
00	X0110	0011	0	init
00	X0110	0011	0	add
00	10110	0011	0	new
00	X1011	X001	0	dec
01	X1011	X001	1	add
01	01011	X001	1	new
01	X0101	XX00	1	dec
10	X0101	XX00	1	add
10	00101	XX00	1	new
10	X0010	XXX0	1	dec
11	X0010	XXX0	0	add
11	10010	XXX0	0	new
11	X1001	XXXX	0	dec

compteur	a(4..0)	b(3..0)	retenue C	état
00	X0110	1011	0	init
00	X0110	1011	0	add
00	10110	1011	0	new
00	X1011	X101	0	dec
01	X1011	X101	1	add
01	01011	X101	1	new
01	X0101	XX10	1	dec
10	X0101	XX10	1	add
10	00101	XX10	1	new
10	X0010	XXX1	1	dec
11	X0010	XXX1	1	add
11	00010	XXX1	0	new
11	X0001	XXXX	0	dec

4. Comment peut détecter l'overflow ?

Réponse _____

La retenue C est à 1.

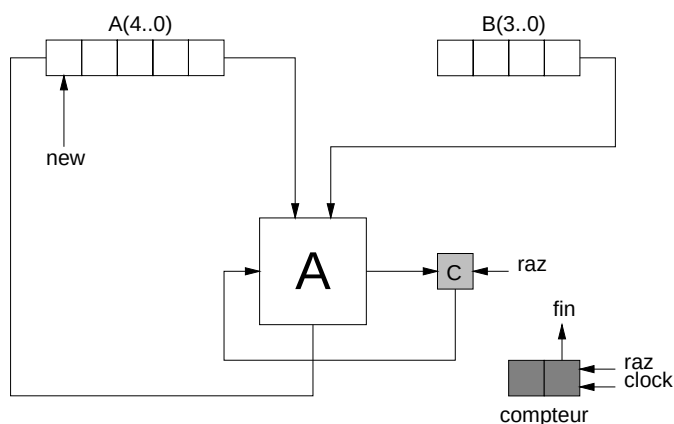


FIG. 1 – Additionneur binaire sériel

Exercice 2 : Addition binaire parallèle en complément à 2

1. Expliquer le fonctionnement de l'additionneur 4 bits de la figure 2.

Réponse _____

On positionne les opérandes dans les registres et on attend stabilisation du système (c'est-à-dire propagation des retenues).

2. Calculer le résultat et l'overflow des additions suivantes : $2+4$, $4+5$, $(-2)+(-4)$, $(-4)+(-5)$, $4+(-2)$ et $(-4)+2$. En déduire la condition d'overflow.

Réponse _____

0	0	0	0			0	1	0	0			1	1	0	0		
	0	0	1	0	(2)		0	1	0	0	(4)		1	1	1	0	(-2)
	0	1	0	0	(4)		0	1	0	1	(5)		1	1	0	0	(-4)
	0	1	1	0	(6)		1	0	0	1	(-7)		1	0	1	0	(-6)
1	0	0	0			1	1	0	0			0	0	0	0		
	1	1	0	0	(-4)		0	1	0	0	(4)		1	1	0	0	(-4)
	1	0	1	1	(-5)		1	1	1	0	(-2)		0	0	1	0	(2)
	0	1	1	1	(7)		0	0	1	0	(2)		1	1	1	0	(7)

$$\text{overflow} = c_n \oplus c_{n-1}$$

3. Si le temps de stabilisation d'un additionneur est τ , donner le temps nécessaire pour obtenir le résultat de l'addition.

Réponse _____

Exemple : addition de 1111 (1) et 0001 (1) temps de propagation des retenues pour obtenir le résultat $n\tau$.

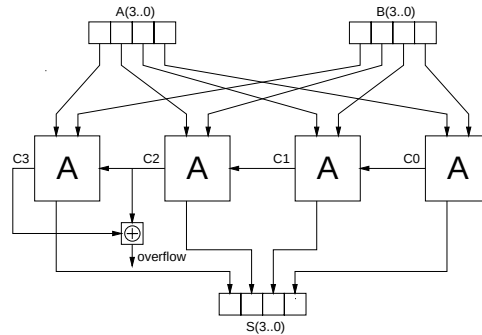


FIG. 2 – Additionneur binaire parallèle en complément à 2

Exercice 3 : Anticipation de retenues

On définit les fonctions suivantes (dans le cas de l'addition de deux bits a_i et b_i) :

- la génération de retenue : $g_i = a_i \cdot b_i$;

Réponse _____

Quand on additionne on génère une retenue de toute façon. Si une retenue arrive de l'étage précédent, elle est absorbée.

- la propagation de retenue : $p_i = a_i \oplus b_i$;

Réponse _____

Quand on additionne 1 et 0, on génère une retenue si une retenue arrive de l'étage précédent : on propage la retenue précédente.

- la retenue : $c_i = g_i + (p_i \cdot c_{i-1})$.

Réponse _____

La retenue de l'étage i est positionnée si on génère une retenue ou si on propage une retenue de l'étage précédent.

- la somme : $s_i = p_i \oplus c_{i-1}$;

On veut réaliser un additionneur 3bits utilisant la méthode d'anticipation de retenue.

1. Donner les formules de s_0 , c_0 , s_1 , c_1 , s_2 et c_2 en fonction de g_0 , g_1 , g_2 , p_0 , p_1 , p_2 et c_e (c_e est la retenue provenant de l'étage précédent).

Réponse _____

$$\begin{cases} c_0 = g_0 + p_0 \cdot c_e \\ s_0 = p_0 \oplus c_e \end{cases} \quad \begin{cases} c_1 = g_1 + p_1 \cdot c_0 \\ = g_1 + p_1 \cdot g_0 + p_1 \cdot p_0 \cdot c_e \\ s_1 = p_1 \oplus c_0 \end{cases} \quad \begin{cases} c_2 = g_2 + p_2 \cdot c_1 \\ = g_2 + p_2 \cdot g_1 + p_2 \cdot p_1 \cdot g_0 + \\ p_2 \cdot p_1 \cdot p_0 \cdot c_e \\ s_2 = p_2 \oplus c_1 \end{cases}$$



2. Compléter le schéma du circuit de la figure 3 et justifier le gain par rapport à l'additionneur parallèle de la question 2.

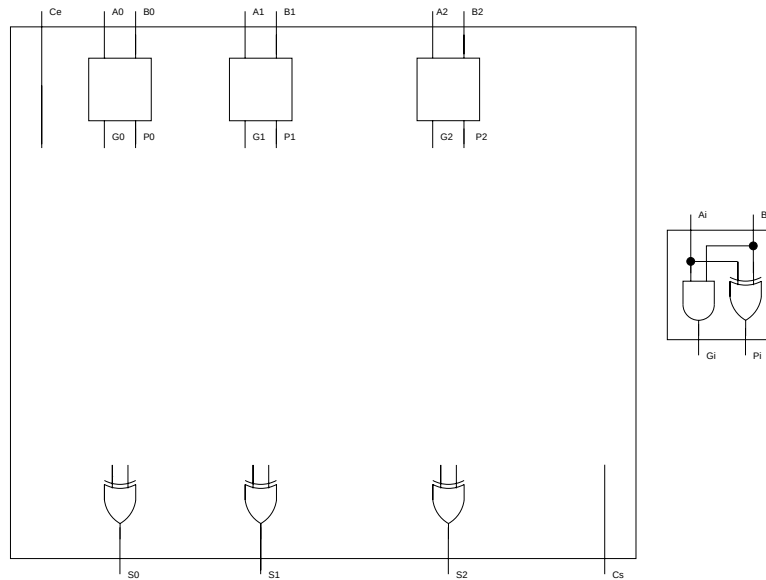


FIG. 3 – Additionneur binaire en complément à 2 avec anticipation de retenue

Réponse _____

Voir Figure-Réponse 1.



Exercice 4 : Préliminaires

1. Effectuer à la main la division binaire suivante $448/17$.
2. Proposer une méthode basée sur des comparaisons, des soustractions et des décalages.

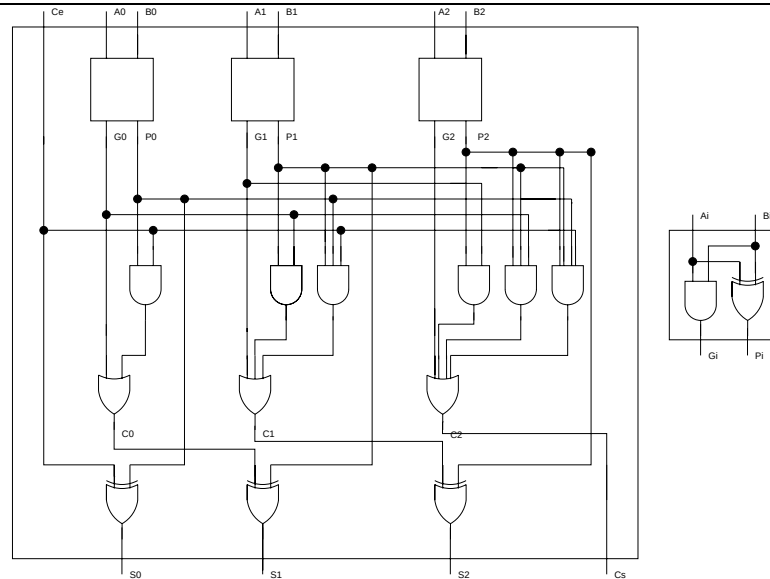
Réponse _____

On considère les premiers bits du dividende. On regarde si ils sont supérieurs au diviseur :

- si oui, on met un bit à 1 dans le quotient ;
- si non, on met un bit à 0 dans le quotient.

On soustrait aux bits du dividende considérés le diviseur. On obtient un reste partiel. On lui ajoute le bit suivant du dividende (décalage de la fenêtre sur les bits du dividende).

FigRep 1 Additionneur binaire en complément à 2 avec anticipation de retenue



On itère le processus.

Dans ce TD, on s'intéresse à la division binaire de nombres non signés codés sur 4 bits.

Exercice 5 : Division binaire avec restauration

1. Que faut-il faire pour effectuer des soustractions sur des nombres non signés.
Réponse _____

Passage en code complément à 2 sur 5 bits (le bit supplémentaire est le bit de signe).

2. Donner une méthode pour comparer deux nombres non signés.

Réponse _____
On fait la soustraction des 2 nombres et on regarde le bit de signe du résultat.

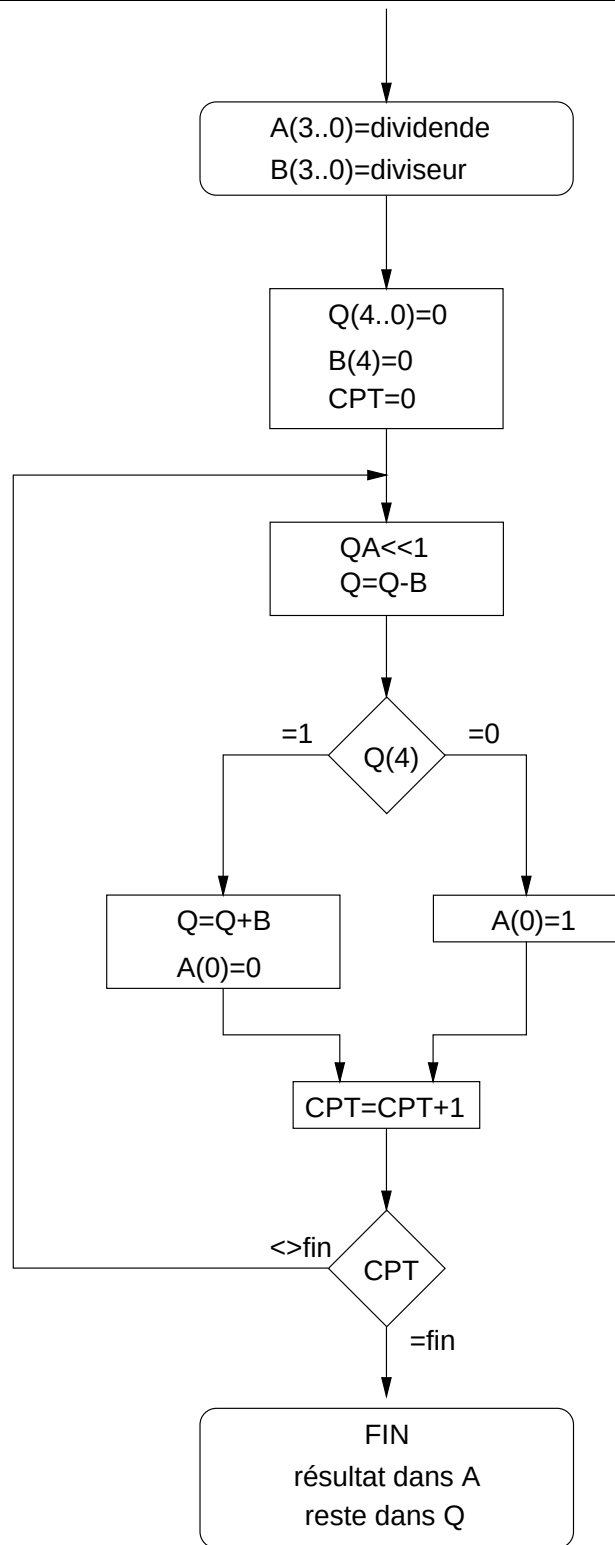
3. En utilisant les questions précédentes, donner l'organigramme de fonctionnement du diviseur de la figure 4.

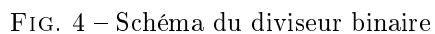
Réponse _____
Voir organigramme Figure-Réponse 2

4. Dérouler les divisions $14/3$ et $4/2$.

Réponse _____

FigRep 2 Organigramme de la division binaire avec restauration



13

CPT	Q(4)	Q(3..0)	A(3..0)	B(4..0)	état
00	0	0000	0100	00010	init.
00	0	0000	100X	00010	dec.
00	1	1110	100X	00010	Q=Q-B
00	0	0000	100X	00010	Q=Q+B
00	0	0000	1000	00010	rés.
01	0	0001	000X	00010	dec.
01	1	1111	000X	00010	Q=Q-B
01	0	0001	000X	00010	Q=Q+B
01	0	0001	0000	00010	rés.
10	0	0010	000X	00010	dec.
10	0	0000	000X	00010	Q=Q-B
10	0	0000	0001	00010	rés.
11	0	0000	001X	00010	dec.
11	1	1110	001X	00010	Q=Q-B
11	0	0000	001X	00010	Q=Q+B
11	0	0000	0010	00010	rés.
-	0	0000	0010	00010	FIN



Exercice 6 : Division binaire sans restauration

On se propose d'éliminer la phase de restauration.

Dans le cas de la division avec restauration, si on a besoin d'une restauration à l'étape $i + 1$, on a :

$$\begin{aligned} \text{étape } i + 1 & \begin{cases} Q_{i+1} = 2Q_i & \text{décalage} \\ Q_{i+1} = 2Q_i - B & \text{soustraction du diviseur} \\ Q_{i+1} = 2Q_i - B + B & \text{restauration (car } Q_{i+1} \text{ est négatif)} \end{cases} \\ \text{étape } i + 2 & \begin{cases} Q_{i+2} = 4Q_i & \text{décalage} \\ Q_{i+2} = 4Q_i - B & \text{soustraction du diviseur} \\ \dots \end{cases} \end{aligned}$$

On obtient le même résultat par la méthode suivante :

$$\begin{aligned} \text{étape } i + 1 & \begin{cases} Q_{i+1} = 2Q_i & \text{décalage} \\ Q_{i+1} = 2Q_i - B & \text{soustraction du diviseur} \end{cases} \\ \text{étape } i + 2 & \begin{cases} Q_{i+2} = 4Q_i - 2B & \text{décalage} \\ Q_{i+2} = 4Q_i - 2B + B & \text{addition du diviseur (car } Q_{i+1} \text{ est négatif)} \end{cases} \end{aligned}$$

Dans le cas de la division avec restauration, si on n'a pas besoin d'une restauration à l'étape $i + 1$, on fait :

$$\begin{aligned} \text{étape } i + 1 & \begin{cases} Q_{i+1} = 2Q_i & \text{décalage} \\ Q_{i+1} = 2Q_i - B & \text{soustraction du diviseur} \end{cases} \\ \text{étape } i + 2 & \begin{cases} Q_{i+2} = 4Q_i - 2B & \text{décalage} \\ Q_{i+2} = 4Q_i - 2B - B & \text{soustraction du diviseur} \\ \dots \end{cases} \end{aligned}$$

1. Que se passe-t-il dans le cas de la dernière étape?

Réponse _____

On compte sur l'itération suivante pour faire la restauration le cas échéant. Le problème est qu'à la dernière itération, on n'a pas d'étape suivante donc on n'a pas restauration (même si il en faut une). Il faut donc regarder si le reste final est négatif et si oui faire une restauration en ajoutant le diviseur.



2. En utilisant la méthode nouvellement définie, donner un nouvel organigramme de fonctionnement du diviseur de la figure 4.

Réponse _____

Voir organigramme Figure-Réponse 3



3. Dérouler les divisions 14/3 et 6/2.

Réponse _____

CPT	Q(4)	Q(3..0)	A(3..0)	B(4..0)	état
00	0	0000	1110	00011	init.
00	0	0001	110X	00011	dec.
00	1	1110	110X	00011	Q=Q-B
00	1	1110	1100	00011	rés.
01	1	1110	100X	00011	dec.
01	0	0000	100X	00011	Q=Q+B
01	0	0000	1001	00011	rés.
10	0	0001	001X	00011	dec.
10	1	1110	001X	00011	Q=Q-B
10	1	1110	0010	00011	rés.
11	1	1100	010X	00011	dec.
11	1	1111	010X	00011	Q=Q+B
11	1	1111	0100	00011	rés.
-	0	0010	0100	00011	Q=Q+B
-	0	0010	0100	00011	FIN
CPT	Q(4)	Q(3..0)	A(3..0)	B(4..0)	état
00	0	0000	0110	00010	init.
00	0	0000	110X	00010	dec.
00	1	1110	110X	00010	Q=Q-B
00	1	1110	1100	00010	rés.
01	1	1101	100X	00010	dec.
01	1	1111	100X	00010	Q=Q+B
01	1	1111	0000	00010	rés.
10	1	1111	000X	00010	dec.
10	0	0001	000X	00010	Q=Q+B
10	0	0001	0001	00010	rés.
11	0	0010	001X	00010	dec.
11	0	0000	001X	00010	Q=Q-B
11	0	0000	0011	00010	rés.
-	0	0000	0011	00010	FIN



FigRep 3 Organigramme de la division binaire sans restauration

