

Nom :
Prénom :
N° d'ordre :

ENSICAEN
1^{ère} année
informatique

Examen de circuits logiques

durée : 90 minutes

Les documents et calculatrices ne sont pas autorisés.

Les réponses seront données sur ces feuilles à l'intérieur des cadres prévus à cet usage.

1- Conversion numérique

- ◆ Compléter le tableau ci-dessous

Base 2*	Base 10	Base 16	DCB
		1AC	
	10,56		
	-72,35		

* les nombres binaires seront représentés en complément à deux sur 12 bits et les valeurs non entières en virgule fixe Q_3 . (Rappel représentation Q_k sur N bits: $b_{N-1}b_{N-2}.....b_{k+1}b_k.....b_2b_1b_0$)

- ◆ Coder les valeurs suivantes sur 10 bits virgule flottante (E sur 4 bits ; F sur 5 bits) suivant la norme IEEE 754.

A=5

B = 0,35

C = -88

Rappel : représentation en virgule flottante norme IEEE 754

La valeur X est représentée suivant la forme : $X = (-1)^S \cdot 2^{E-7} \cdot 1, F$

X s'écrit alors en binaire virgule flottante : $\underbrace{s}_{\text{signe}} \underbrace{e_3 e_2 e_1 e_0}_E \underbrace{f_4 f_3 f_2 f_1 f_0}_F$

E et F sont codés en binaire non signé.

2- Opérateurs arithmétiques

Questions :

- Réaliser à base de portes NAND, un bloc logique fournissant deux sorties binaires s et c en fonction de trois entrées binaires x , y et z , telles que $x+y+z = 2c+s$ (équivalent du bloc additionneur complet).

- Réaliser à base de portes la cellule de base (rectangle $i-j$ dans le schéma ci-contre) du multiplieur classique. Cette cellule comporte 4 entrées (les entrées a_i et b_j , le terme provenant de la ligne supérieure e et la retenue d'entrée venant de la droite r_e) et 2 sorties (la sortie à propager verticalement s et la retenue r_s à propager horizontalement).

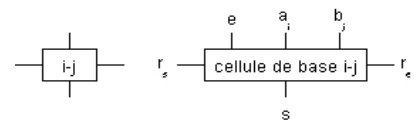
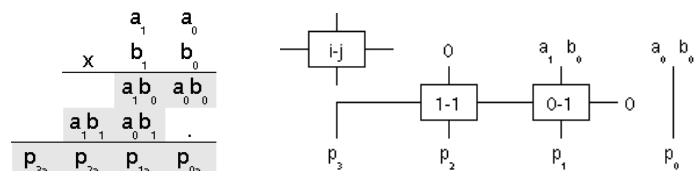


Illustration 1: cellule de base du multiplieur classique et symbole associé

Exemple : multiplication de nombres de 2 bits (a_1a_0 multiplié par b_1b_0) : opération posée sur papier à gauche et structure associée à droite.



Multiplieur de Wallace

Considérons la décomposition d'une multiplication des nombres non signés en plusieurs multiplications élémentaires sur un nombre réduit de bits. Comme exemple prenons le cas d'une multiplication de deux termes de 4 bits ($A=a_3a_2a_1a_0$ et $B=b_3b_2b_1b_0$).

On pose : $A_{msb} = a_3a_2$, $A_{lsb} = a_1a_0$; $B_{msb} = b_3b_2$, $B_{lsb} = b_1b_0$.
(*msb* « most significant bits » bits de poids fort, *lsb* « least significant bits » bits de poids faible).

Le produit peut alors se décomposer en :

$$AB = 2^4 A_{msb} B_{msb} + 2^2 (A_{msb} B_{lsb} + A_{lsb} B_{msb}) + A_{lsb} B_{lsb}$$

Les produits partiels peuvent s'effectuer en parallèle sur des multiplieurs 2 bits (structure vue à la question 2). Les multiplieurs 2 bits étant plus rapides que ceux de 4 bits, on peut gagner du temps. Les résultats de ces multiplications élémentaires doivent être combinés pour obtenir le résultat. Cette décomposition et à la base de la structure de Wallace pour les multiplieurs, elle s'accompagne de l'utilisation de blocs particulier appelés arbres de Wallace permettant de combiner plusieurs termes partiels en ne faisant qu'une seule véritable addition finale.

L'élément de base est l'arbre de Wallace à 3 entrées qui réalise $x+y+z = 2c+s$ permettant ainsi de passer de 3 à 2 entrées pour l'addition (cellule réalisée à la question 1).

Finalement, un multiplieur de Wallace 4 bits est composé de 4 multiplieurs classiques 2 bits, d'un arbre de Wallace à 3 entrées et d'un additionneur. Nous nous proposons de réaliser en partie ce multiplieur.

Les multiplieurs d'ordre supérieur feront eux appel à des arbres de Wallace plus importants à 5, 7, 9 ... entrées.

Question :

3. Confirmer le fonctionnement de l'architecture de Wallace pour la multiplication en remplissant le schéma à trou suivant.

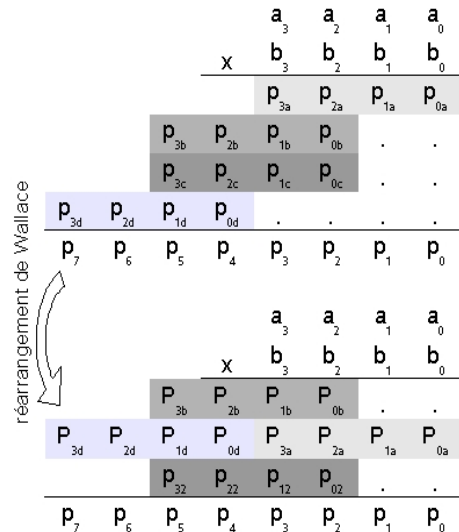
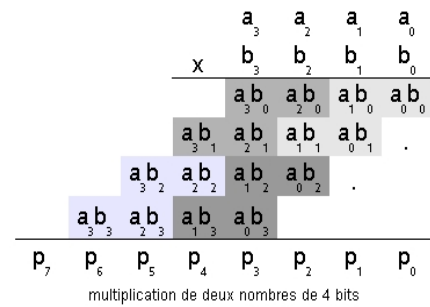
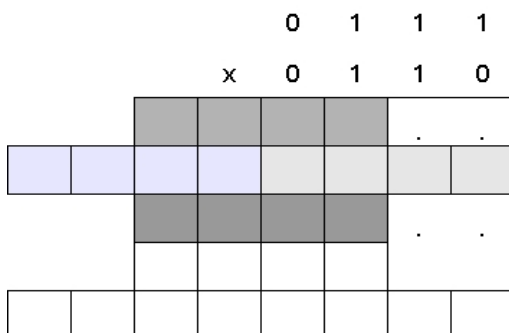


Illustration 2: réarrangement de Wallace

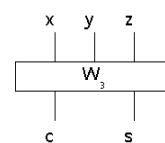


Illustration 3: arbre de Wallace à 3 entrées

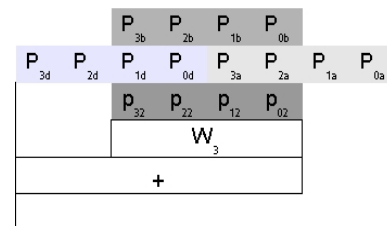
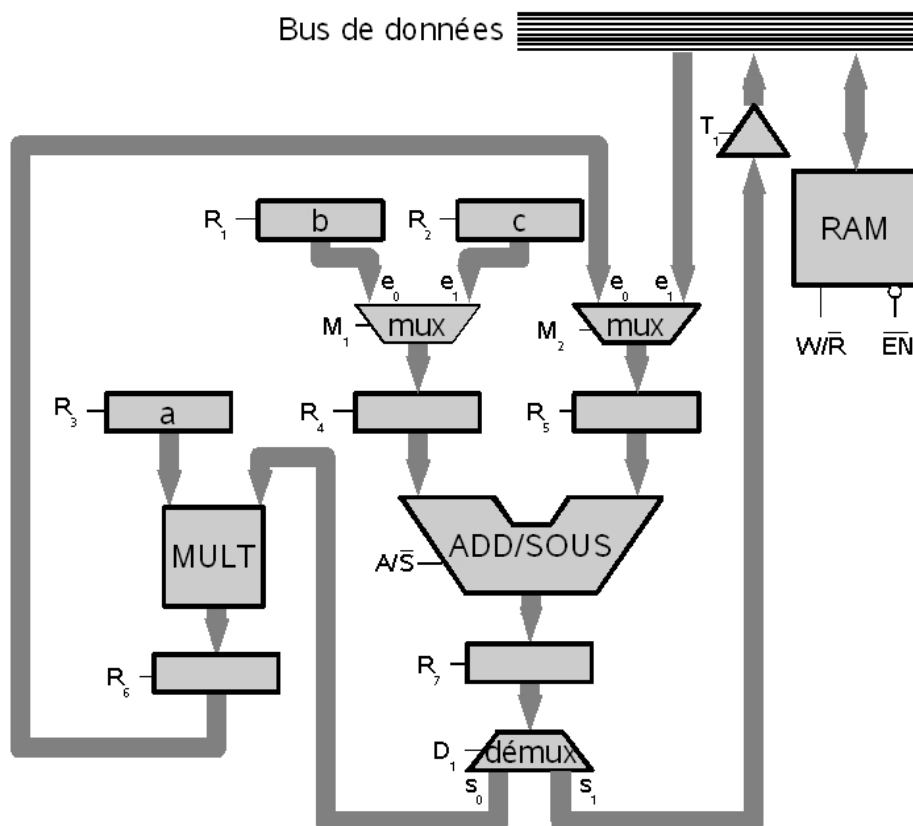


Illustration 4: structure du multiplieur Wallace 4 bits

3- Machine à états finis

- ♦ L'objectif est de réaliser l'unité de contrôle sous forme de machine à états finis permettant d'effectuer le calcul $y=a(x-b)+c$ sur l'unité de traitement présentée ci-dessous



Remarques :

La multiplication prend **deux temps** de cycle, tandis que l'addition en prend un seul.

Il faut que la donnée soit présente sur le bus avant de pouvoir être chargée dans le registre ou écrite en mémoire. On supposera que les adresses mémoire pour la lecture et l'écriture sont gérées automatiquement.

On supposera que les valeurs a, b et c sont préchargées.

On utilisera des bascules D pour le registre d'état.

Les signaux R_i commande le chargement des registres lorsqu'ils sont à l'état 1. Les sorties du *tristate* sont connectées au bus de données lorsque T_1 est à l'état 1, elles en sont isolées sinon.

Questions :

1. Établir le tableau d'évolution de la machine d'états.
2. Donner les équations logiques régissant l'évolution des états.
3. Déterminer les équations des signaux de commandes des composants de l'architecture.

