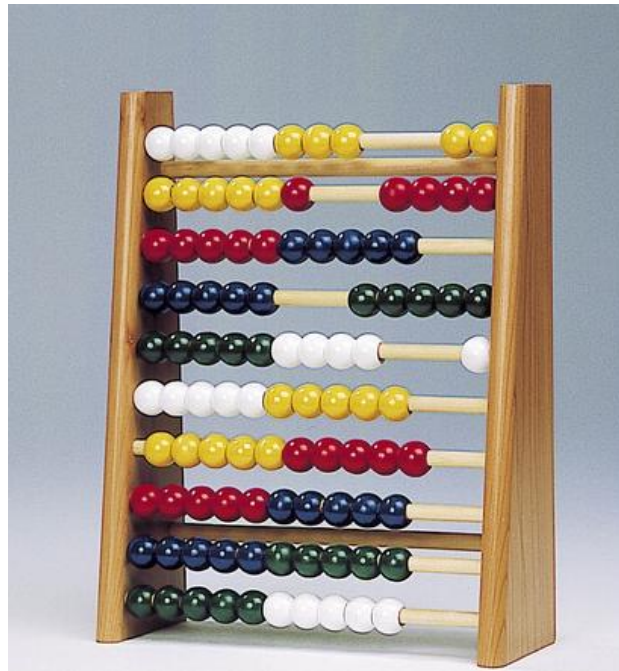
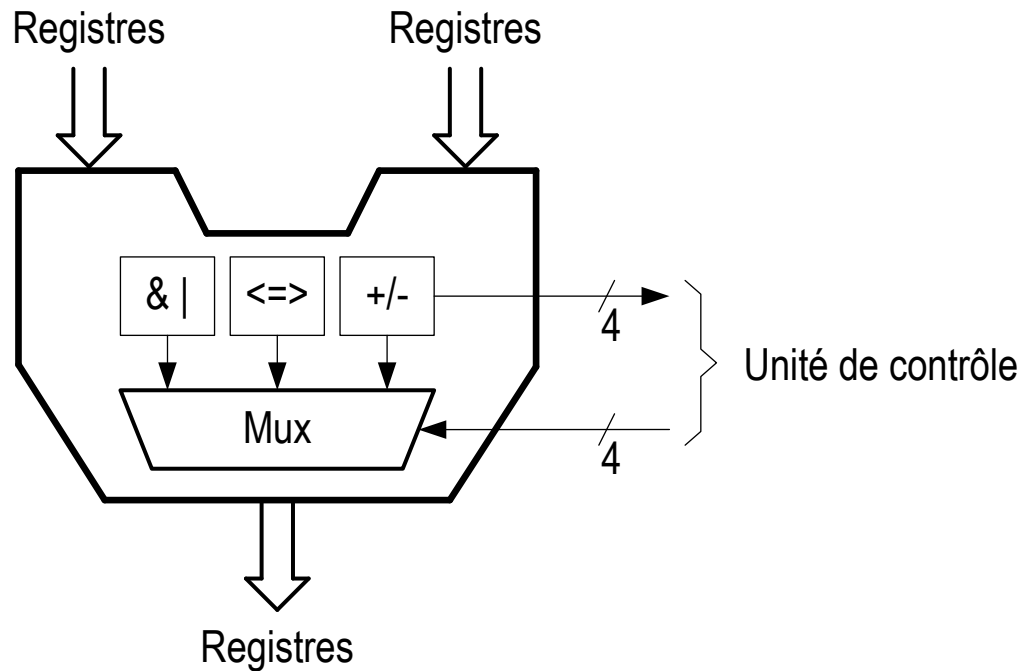


# Unité arithmétique et logique



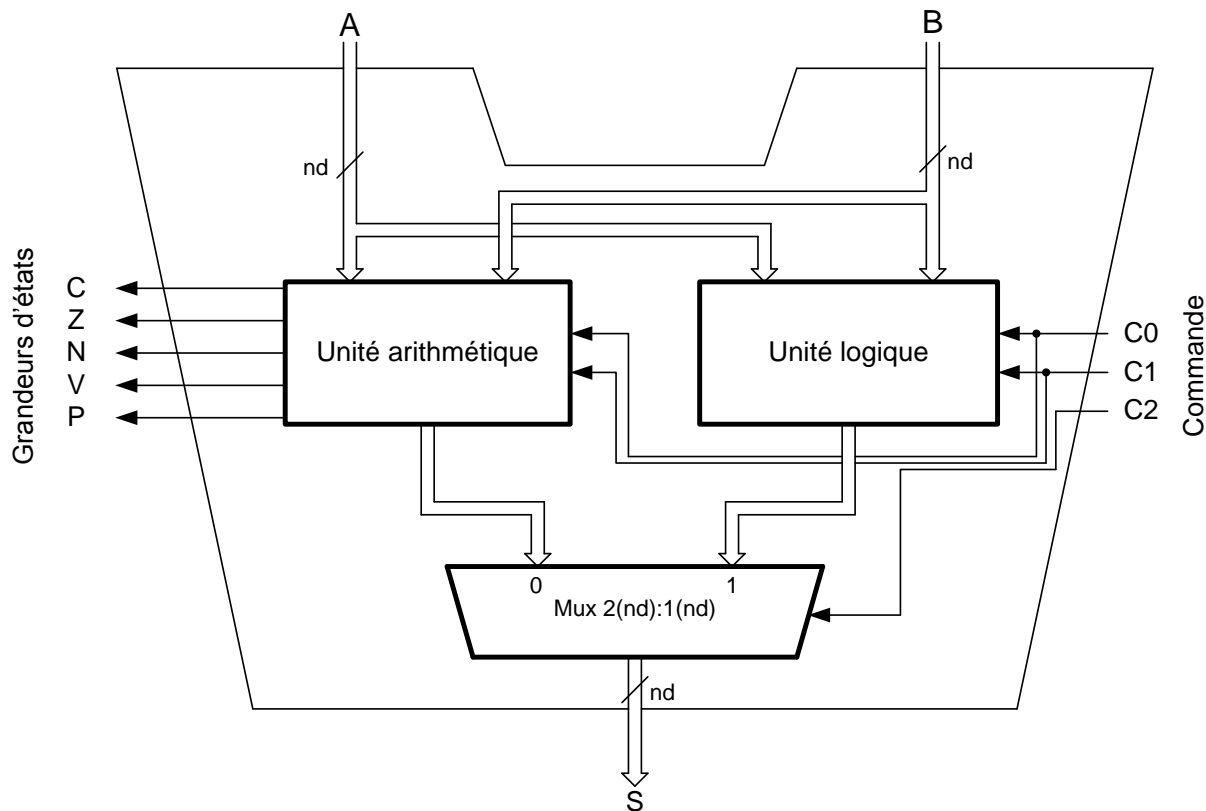
## Unité logique et arithmétique

L'Unité Arithmétique et Logique (ALU : arithmetic and Logic unit) est la partie de l'ordinateur qui effectue des opérations arithmétiques et logiques sur les données



## Unité logique et arithmétique

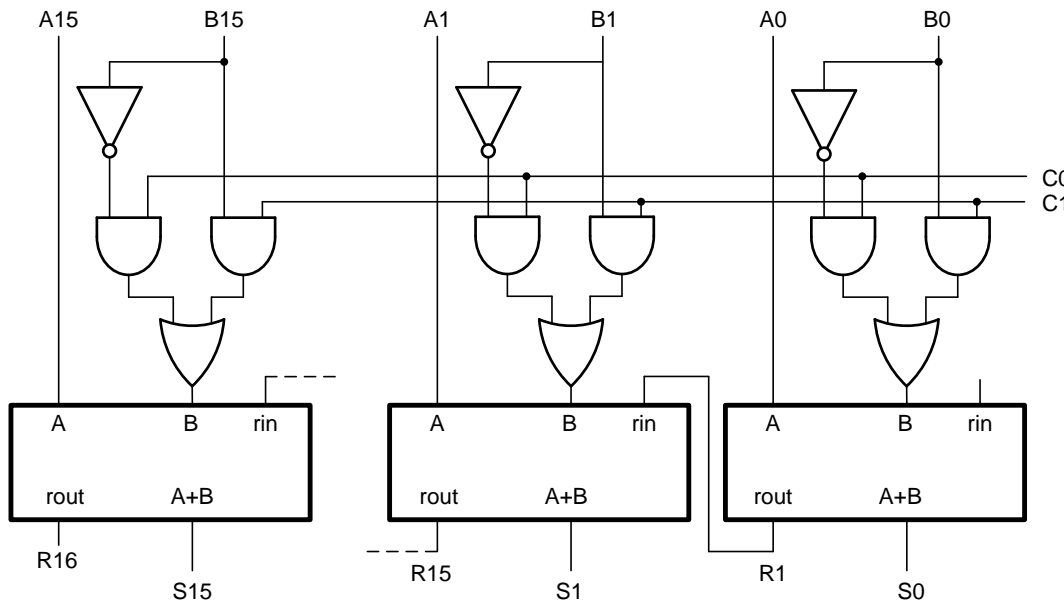
L'unité logique et l'unité arithmétique sont intégrées au sein d'une même unité dédiée au traitement des entiers. Pour chacune des unités, le choix de la fonction se fait par les commandes C0 et C1. Les deux unités fonctionnent en parallèle, et la sélection de la sortie d'une unité (logique ou arithmétique) se fait par la commande de multiplexage C2



## Unité logique et arithmétique

### Structure d'une unité arithmétique

La valeur de commande C0, C1 permet de réaliser, au choix, l'opération d'addition, de soustraction, voire de comparaison

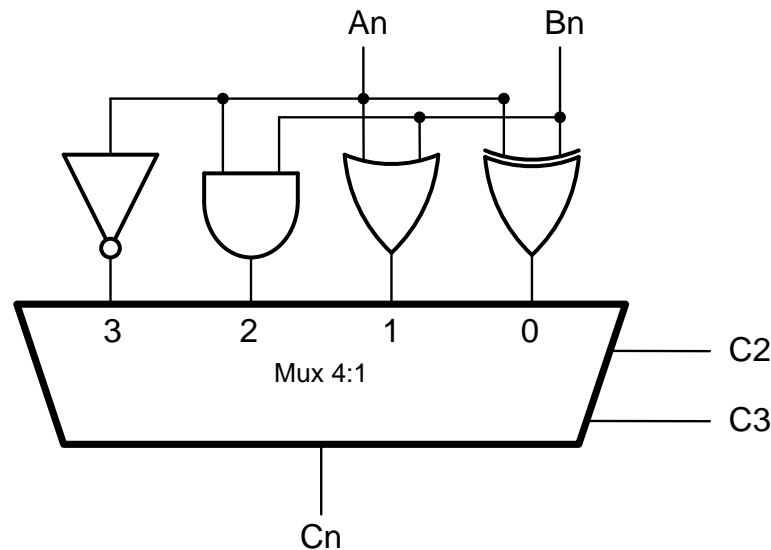


C0	C1	rin	Sn	Opération
0	0	0	A	
0	0	1	A + 1	Incrémentation
0	1	0	A + B	Addition
0	1	1	A + B + 1	Addition + retenue
1	0	0	A + $\bar{B}$	
1	0	1	A + $\bar{B}$ + 1	Soustraction
1	1	0	A - 1	Décrémentation
1	1	1	A	

## Unité logique et arithmétique

Structure de l'unité logique

L'unité logique doit permettre de réaliser au choix plusieurs opérations logiques élémentaires telles que le **OU**, le **ET**, le **OU EXCLUSIF** et le **NON**. Ceci doit être réalisé sur des mots de n bits.

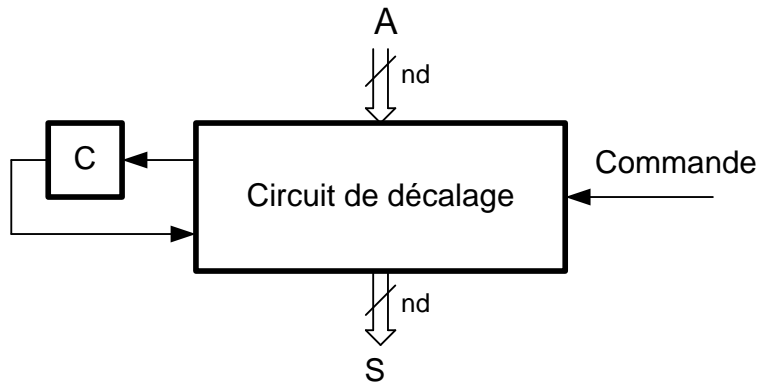


Il est assez aisé de comprendre que la juxtaposition de cette unité 1 bit permet de réaliser une unité logique n bits

## Unité de décalage

Structure du circuit de décalage

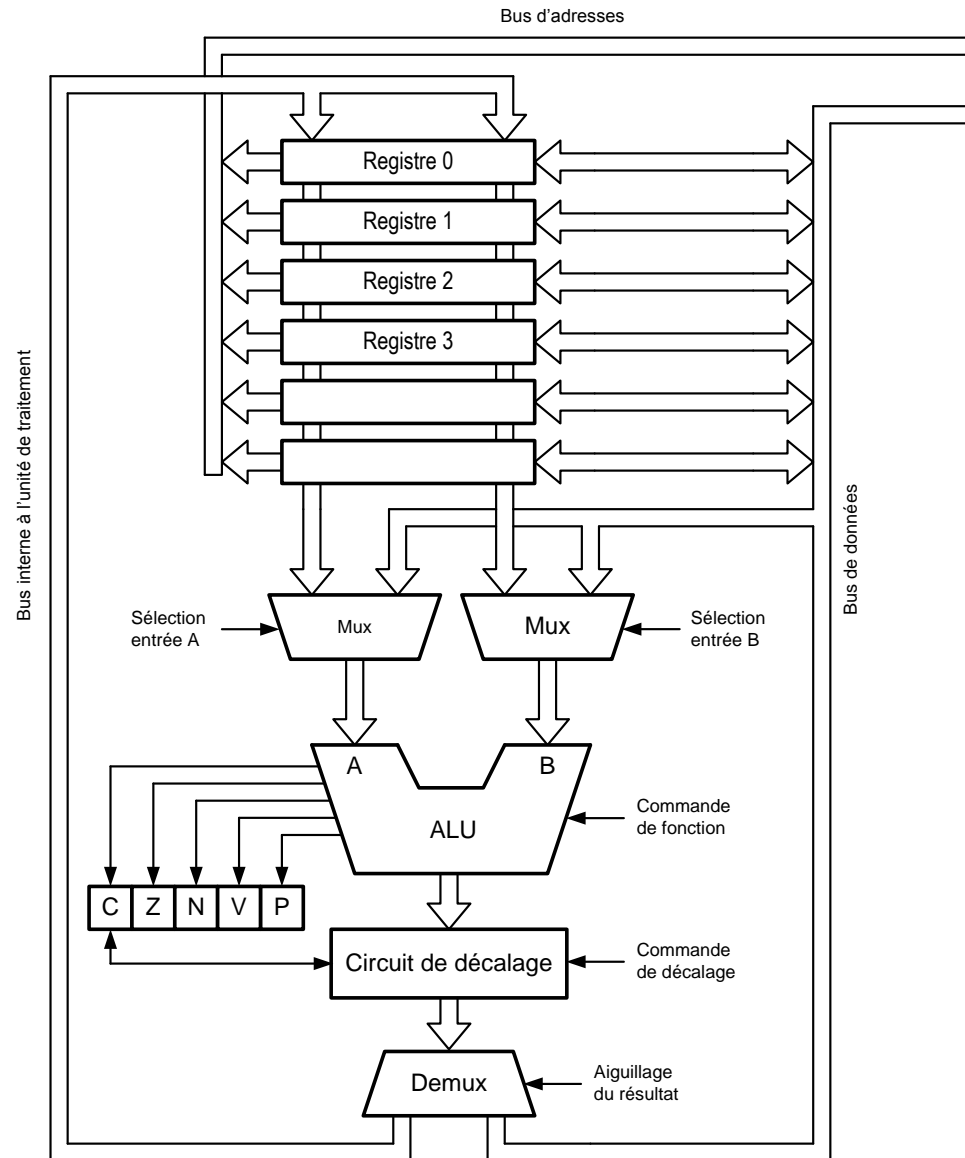
Le circuit de décalage (shifter) permet les opérations de décalage. Ces opérations sont réalisées au moyen d'un multiplexeur situé à la sortie de l'unité arithmétique et logique



Type de décalage	C	Opération
Pas de décalage	c	a7 a6 a5 a4 a3 a2 a1 a0
Logique droite	c	0 a7 a6 a5 a4 a3 a2 a1
Logique gauche	c	a6 a5 a4 a3 a2 a1 a0 0
Arithmétique droite	c	a7 a7 a6 a5 a4 a3 a2 a1
Arithmétique gauche	c	a6 a5 a4 a3 a2 a1 a0 0
Circulaire droite	c	a0 a7 a6 a5 a4 a3 a2 a1
Circulaire gauche	c	a6 a5 a4 a3 a2 a1 a0 a7
Circulaire droite avec C	c	c a7 a6 a5 a4 a3 a2 a1
Circulaire gauche avec C	a7	a6 a5 a4 a3 a2 a1 a0 c

L'entrée est un mot binaire A de n bits. La sortie B du circuit est aussi un mot de n bits. Plusieurs types de décalages sont possibles, certains utilisent l'indicateur d'état C

## Exemple d'unité de traitement



**Calcul arithmétique et logique**Les indicateurs d'états (status flag)

**Z** : indicateur de résultat nul (Zero Flag)

Cet indicateur est mis à 1 lorsque le résultat de la dernière opération est zéro.

Lorsque l'on vient d'effectuer une fonction logique, Z=1 indique que le résultat bit à bit est nul pour chaque couple

$$( 1 \quad 0 \quad 0 \quad 1 ) \text{ .OR. } ( 1 \quad 0 \quad 1 \quad 0 ) \rightarrow Z=0$$

$$( 1 \quad 0 \quad 0 \quad 1 ) \text{ .AND. } ( 0 \quad 1 \quad 1 \quad 0 ) \rightarrow Z=1$$



**Calcul arithmétique et logique**Les indicateurs d'états (status flag)

**C** : indicateur de report (Carry Flag)

C'est l'indicateur de report (retenue), qui intervient dans les opérations d'addition et de soustractions sur des entiers naturels. Il est positionné en particulier par les instructions ADD, SUB et CMP. C = 1 s'il y a une retenue après l'addition ou la soustraction du bit de poids fort des opérandes

0	1	0	0		1	1	0	0		1	1	1	1	
+	0	1	1	0	+	0	1	1	0	+	0	0	0	1
<hr/>					<hr/>					<hr/>				
<b>C=0</b>	1	0	1	0	<b>C=1</b>	0	0	1	0	<b>C=1</b>	0	0	0	0

## Calcul arithmétique et logique

Les indicateurs d'états (status flag)

**N** : indicateur de résultat négatif (Negativ Flag)

N est positionné à 1 si le bit de poids fort du résultat d'une addition ou soustraction est 1, sinon N=0. N est utile lorsque l'on manipule des entiers relatifs, car le bit de poids fort donne alors le signe du résultat.

$\begin{array}{rcccc} & 0 & 1 & 0 & 0 \\ + & 0 & 1 & 1 & 0 \\ \hline \boxed{N=1} & 0 & 1 & 0 & \end{array}$	$\begin{array}{rcccc} & 1 & 1 & 0 & 0 \\ + & 0 & 1 & 1 & 0 \\ \hline \boxed{N=0} & 0 & 1 & 0 & \end{array}$	$\begin{array}{rcccc} & 1 & 1 & 1 & 1 \\ + & 0 & 0 & 0 & 1 \\ \hline \boxed{N=0} & 0 & 0 & 0 & \end{array}$
---	---	---

## Calcul arithmétique et logique

Les indicateurs d'états (status flag)

**V** : indicateur de dépassement (Overflow Flag)

Indicateur de débordement  $V=1$  si le résultat d'une addition ou soustraction donne un nombre qui n'est pas codable en relatif dans l'accumulateur (par exemple si l'addition de 2 nombres positifs donne un codage négatif) .

<b>V=1</b>	0	1	0	0		1	1	0	0		1	1	1	1		
+	0	1	1	0		+	0	1	1	0		+	0	0	0	1
<b>V=0</b>	1	0	1	0		<b>V=0</b>	0	0	1	0		<b>V=0</b>	0	0	0	0

**Instructions Arithmétiques : addition**Addition de deux nombres positifs en simple précision

Dans ce cas le bit de poids fort correspond au bit de signe. Le signe du résultat doit être identique aux signes des opérandes

Pour deux nombres positifs dont la somme est strictement inférieure à  $2^{n-1}$ , il n'y a pas de dépassement. Le bit de signe du résultat reste à 0 et par conséquent le résultat est correct

Pas de dépassement ( $V=0$ )  
Pas de report ( $C=0$ )

	0	0	1	1	0	1	0	1	53
+	0	1	0	0	0	1	1	0	+ 70
0	0	1	1	1	1	0	1	1	123

**Instructions Arithmétiques : addition**Addition de deux nombres positifs en simple précision

Dans ce cas le bit de poids fort correspond au bit de signe. Le signe du résultat doit être identique aux signes des opérandes

Si le résultat dépasse  $2^{n-1}-1$ , le bit de signe passe à 1 et par conséquent le résultat (en complément à deux est faux. Il y a donc dépassement

Dépassement (V=1 : Overflow)  
Pas de report (C=0)

	0	0	1	1	0	1	0	1	53
+	0	1	1	0	0	1	1	0	+102
—	1	0	0	1	1	0	1	1	-101

**Instructions Arithmétiques : addition**Addition de deux nombres négatifs en simple précision

Le résultat de l'addition de deux nombres négatifs doit être négatif.

Le bit de signe du résultat doit rester à 1. Dans tout les cas il y a un bit supplémentaire qui passe à 1. Ce bit peut être considéré comme bit de report pour les calculs en multiprécision

Pas de dépassement (V=0)  
Report (C=1)

	1	1	0	0	1	0	1	1	-53
+	1	0	1	1	1	0	1	0	+ -70
	1	0	0	0	0	1	0	1	-123

**Instructions Arithmétiques : addition**Addition de deux nombres négatifs en simple précision

Le résultat de l'addition de deux nombres négatifs doit être négatif.

Si le bit de signe du résultat est 0, alors il y a dépassement. Ce cas peut se produire lorsque la somme de deux nombres négatifs est inférieure à  $2^{n-1}-1$

Dépassement (V=1 : Overflow)  
Report (C=1)

	1	1	0	0	1	0	1	1	-53
+	1	0	0	1	1	0	1	0	+ -102
1	0	1	1	0	0	1	0	1	101

**Instructions Arithmétiques : addition**Addition d'un nombre positif et un nombre négatifs en simple précision

L'addition deux nombres de signes opposés ne provoque pas de dépassement.

Si le résultat est négatif, le report est égal à 0

Pas de dépassement (V=0)

Pas de report (C=0)

1	0	0	1	1	0	1	0	-102
+	0	0	1	1	0	1	0	+ 53
0	1	1	0	0	1	1	1	-49

Si le résultat est positif, le report est égal à 1

Pas de dépassement (V=0)

Report (C=1)

1	1	1	1	1	1	1	1	-1
+	0	1	1	1	1	1	1	+ 127
1	0	1	1	1	1	1	0	126



**Instructions Arithmétiques : addition**Addition de deux nombres en multi-précision

Par addition multiprécision on entend une addition sur un nombre de bits supérieur aux nombres de bits avec lesquels les opérations de l'ALU sont effectuées.

Dans ce cas l'addition des poids forts détermine s'il y a dépassement ou non.

En effet, les bits de poids faibles ne contiennent pas de signe

Cas sans dépassement

		Report : →							1										
	0	0	0	1	0	1	0	1		0	1	1	1	0	1	0	1	5493	
+	0	1	0	0	0	1	1	0		+	1	1	0	0	0	1	1	0	+ 18118
<hr style="border: 0; border-top: 1px solid black;"/>																			
	0	1	0	1	1	1	0	0			0	0	1	1	1	0	1	1	23611

V=1 (n'a pas de signification)

## Instructions Arithmétiques : addition

### Addition de deux nombres en multi-précision

Par addition multiprécision on entend une addition sur un nombre de bit supérieur aux nombre de bits avec lequel les opérations de l'ALU sont effectuées.

Dans ce cas l'addition des poids forts détermine s'il y a dépassement ou non.

En effet, les bits de poids faibles ne contiennent pas de signe

### Cas avec dépassement

		Report : → 0 ←																		
		0	0	1	1	0	1	0	1			0	1	1	1	0	1	0	1	13685
+		0	1	1	0	0	1	1	0	+		0	1	0	0	0	1	1	0	+ 26182
		1	0	0	1	1	0	1	1			1	0	1	1	1	0	1	1	-25669

V=1 (dépassement)
V=1 (n'a pas de signification)

### Instructions Arithmétiques : soustraction

La soustraction est effectuée (dans le microcontrôleur) par une addition après avoir pris le complément à deux du second opérande.

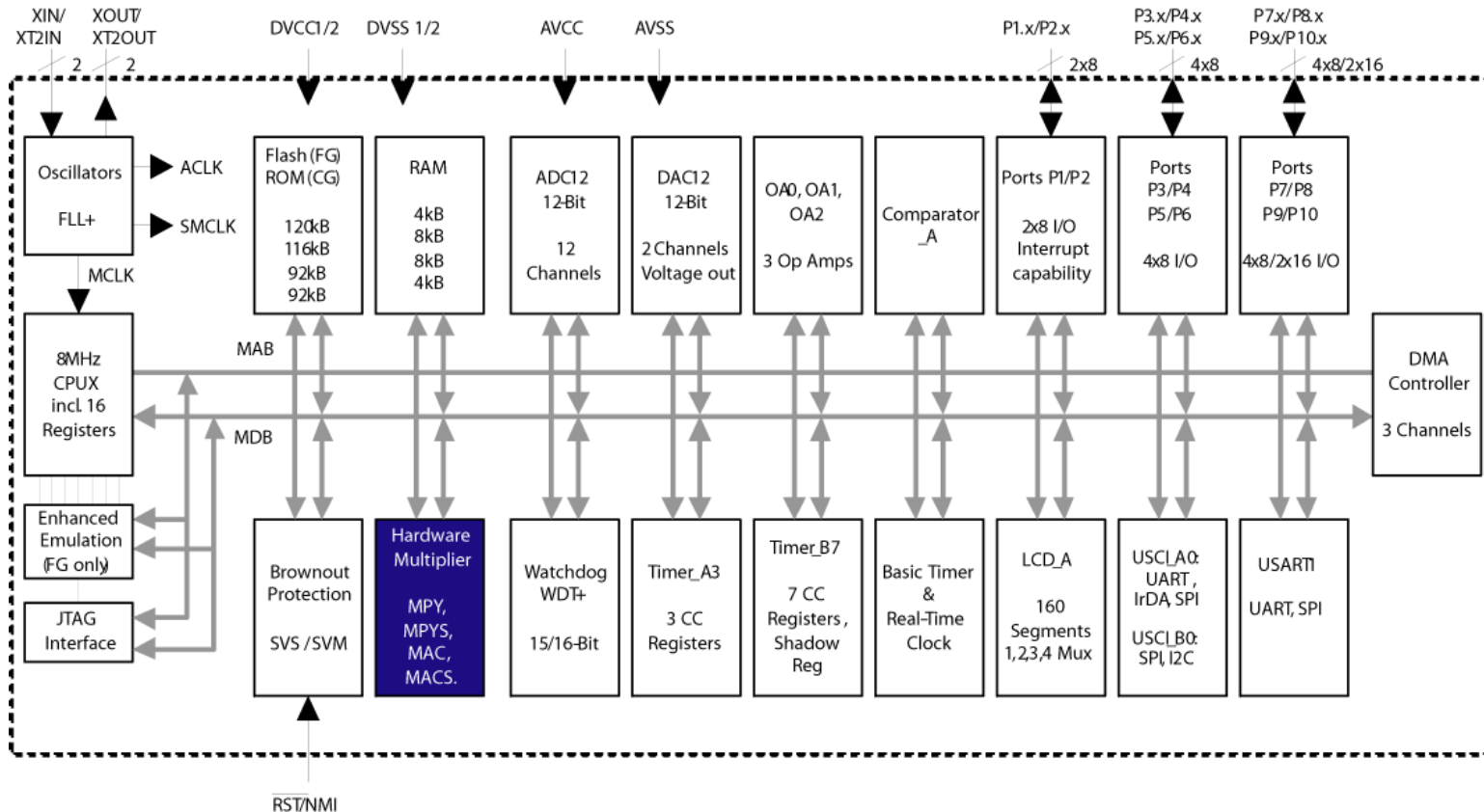
Les calculs se conduisent donc de la même manière que pour l'addition.

Pour deux nombres de même signe il n'y a jamais de dépassement sur le résultat alors qu'il peut y en avoir un si les deux nombres ont des signes complémentaires .

$$A - B = A + \bar{B} + 1$$



# Le multiplicateur



## Multiplication 16 bits x 16 bits et addition (accumulation)

Le **multiplicateur hardware** est un périphérique, il ne fait donc pas partie de l'unité de traitement du MSP430.

Les **activités du multiplicateur hardware** n'interfèrent pas avec les **activités de l'unité de traitement**.

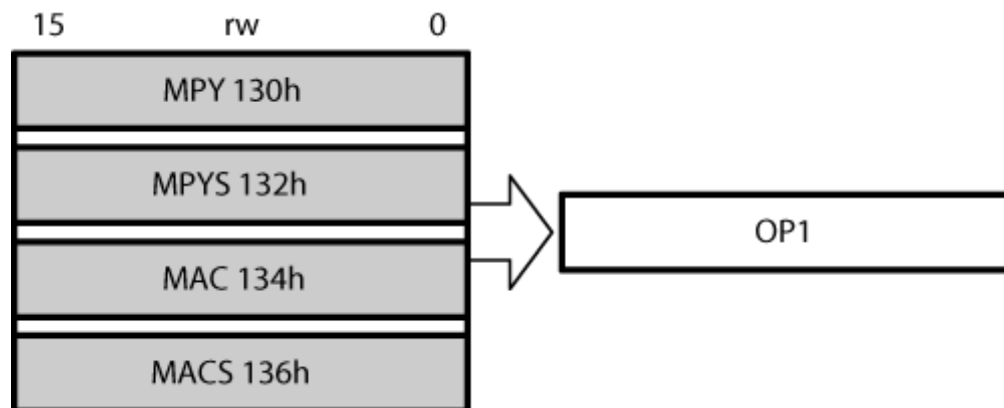
Les **registres de multiplicateur sont des registres périphériques** qui sont écrits et lus avec l'unité de traitement instructions.

### Description des opérations

- multiplication non signés
- multiplication signés
- multiplication et accumulation non signée
- multiplication et accumulation signée
- 16×16 bits, 16×8 bits, 8×16 bits, 8×8 bits

## Type d'opérations, première opérande

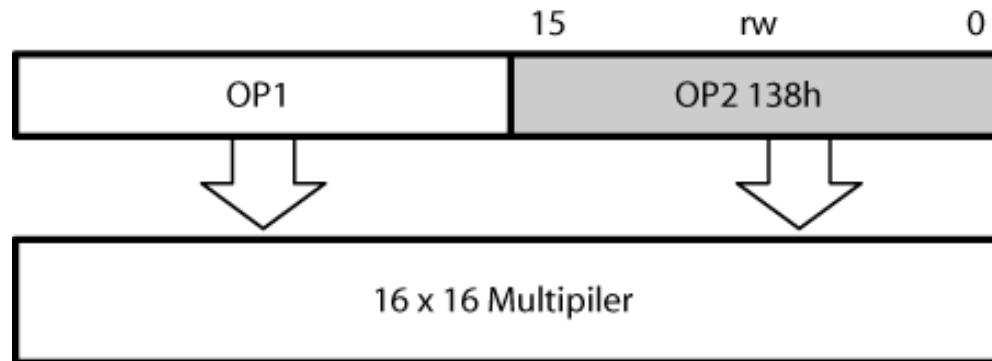
Le type d'opération est choisi en fonction de l'adresse dans laquelle est écrit le premier opérande.



Adresse d'OP1	Registre	Operation
0130h	MPY	Multiplication non-signée
0132h	MPYS	Multiplication signée
0134h	MAC	Multiplication et accumulation non signée
0136h	MACS	Multiplication et accumulation signée

**Deuxième opérande**

L'écriture du second opérande dans le registre OP2 provoque l'exécution de la multiplication ou de la multiplication et accumulation



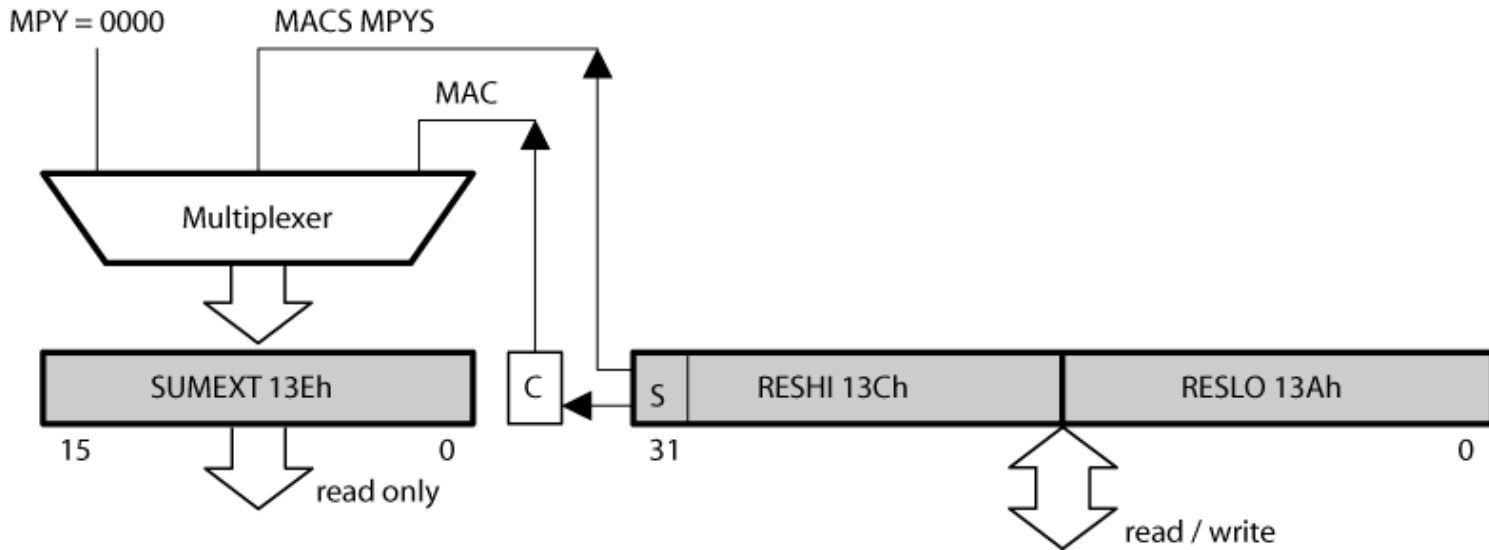
Le résultat est disponible après trois cycles d'horloge MCLK.

Le résultat peut être lu par l'instruction suivant l'écriture du registre OP2

Lorsque le résultat est accédé par le mode d'adressage indirect, une instruction NOP est requise avant de lire le résultat



Le multiplicateur hardware a **trois registres de résultat: RESLO, RESHI, et SUMEXT.**



Mode	SUMEXT
MPY	SUMEXT est toujours à 0000h
MPYS	SUMEXT contient l'extension du signe du résultat 00000h Résultat nul ou positif 0FFFFh Résultat négatif
MAC	SUMEXT contient la retenue (carry) du résultat 00000h Pas de retenue sur le résultat 00001h Retenue sur le résultat
MACS	SUMEXT contains the extended sign of the result 00000h Résultat nul ou positif 0FFFFh Résultat négatif

Mode	RESHI Contents
MPY	16 bits de poids fort du résultat
MPYS	16 bits de poids fort en complément à deux
MAC	16 bits de poids fort du résultat
MACS	16 bits de poids fort en complément à deux

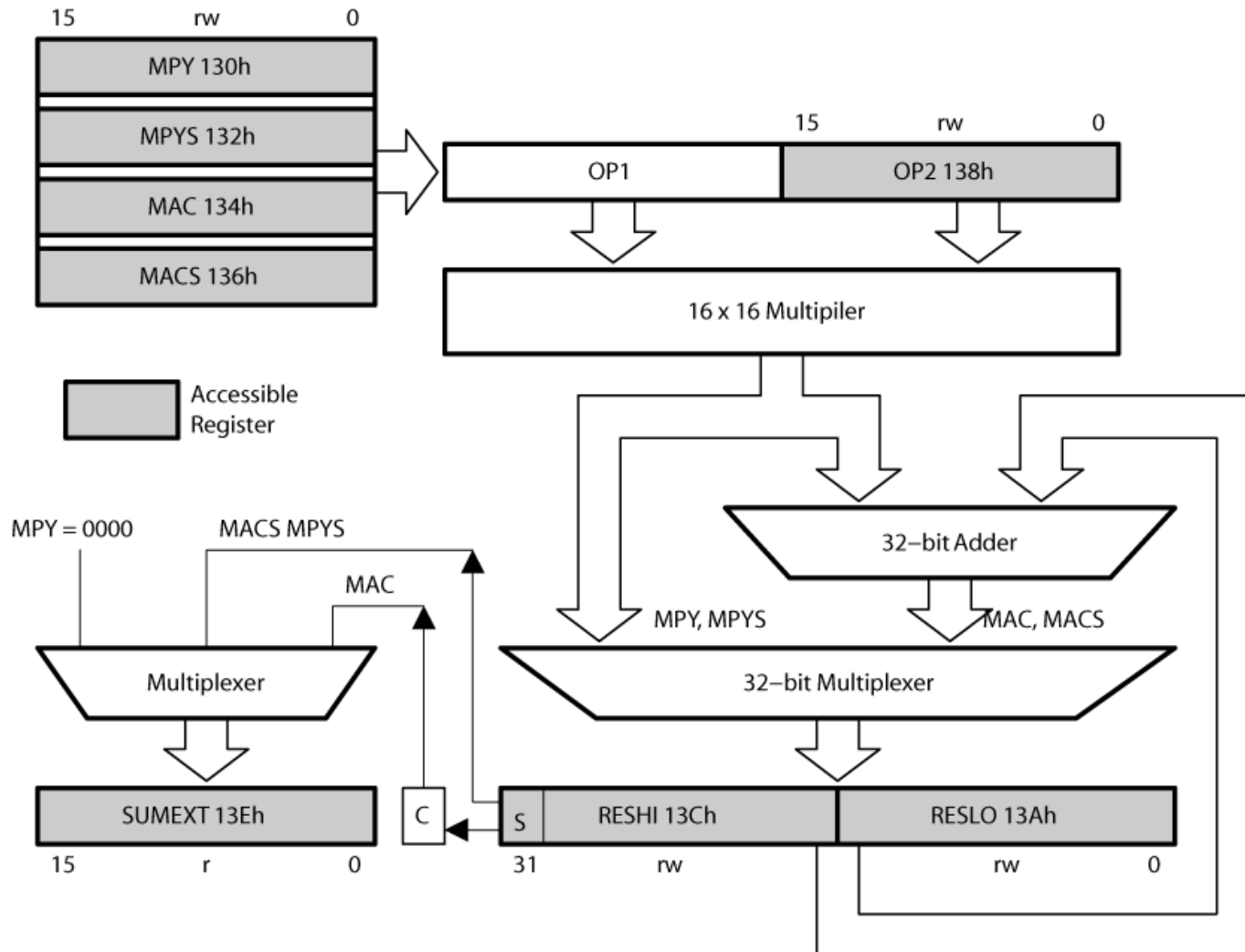
## Dépassement pour les opérations de multiplication et accumulation

Le multiplicateur ne détecte pas automatiquement les dépassements (overflow and underflow) en mode MACS.

- La plage des nombres nuls et positifs va de 0 à 7FFF FFFFh
- La plage des nombre négatifs va de 0FFFF FFFFh à 8000 0000h

C'est à l'utilisateur de vérifier le dépassement

Schéma bloc du multiplicateur accumulateur



**Multiplication de deux nombres entiers non signés 1234h x ABCDh = C374FA4h**

Multiplication entière non signée 4660 x 43981 = 204951460

MOV #01234h,&MPY ; premier opérande  
MOV #0ABCDh,&OP2 ; second opérande } RESHI = 0C37h RESLO = 4FA4h

**Multiplication de deux nombres entiers signés 1234h x ABCDh = FA034FA4h**

Multiplication entière signée 4660 x (-21555)=-100446300

MOV #01234h,&MPYS ; premier opérande  
MOV #05678h,&OP2 ; second opérande } SUMEXT FFFFh RESHI = FA03h RESLO = 4FA4h

**Multiplication de deux nombres entiers non signés avec accumulation :**  
 **$1234h * 8765h + 1234 * 5678h = 4660 * (-30875) + 4660 * 22136 = -40723740$**

MOV.W #1234h,&MPYS ; Première opérande (4660), indication d'une multiplication signée  
 MOV.W #8765h,&OP2 ; Deuxième opérande (-30875), déclenchement de la multiplication  
 ; résultat intermédiaire :  
 ; SUMEXT RESHI RESLO : FFFFh F76Ch 9A84h (= -143877500)

MOV.W #1234h,&MACS ; Première opérande (4660), indication d'une multiplication signée avec  
 accumulation (addition du résultat précédent)

MOV.W #5678h,&OP2 ; Deuxième opérande (22136), déclenchement de la multiplication  
 ; résultat intermédiaire de la multiplication 6260060h (103153760)  
 ; résultat final :  
 ; SUMEXT RESHI RESLO : FFFFh FD92h 9AE4h (= -40723740)

## Limitation de la multiplication

Les multiplications peuvent être effectuées avec les deux opérandes non signées ou signées.

La multiplication, d'une opérande signée par un opérande non signée n'est pas prévue.

## Code assembleur généré pour une multiplication de type long en C

```
long A=12345;  
long B = -54321;  
long E;  
  
...  
E=A*B;
```

```
MOV.W  #0x3039,R10  
CLR.W  R11  
  
MOV.W  #0x2BCF,R8  
MOV.W  #0xFFFF,R9  
  
MOV.W  R8,R12  
MOV.W  R9,R13  
MOV.W  R10,R14  
MOV.W  R11,R15  
CALL   #?Mult32Hw
```

```
?Mult32Hw:
```

```
PUSH.W  SR  
DINT  
MOV.W  R12,&MPY  
MOV.W  R14,&OP2  
MOV.W  R12,&MAC  
MOV.W  &RESLO,R12  
MOV.W  &RESHI,&RESLO  
MOV.W  R15,&OP2  
MOV.W  R13,&MAC  
MOV.W  R14,&OP2  
MOV.W  &RESLO,R13  
RETI
```