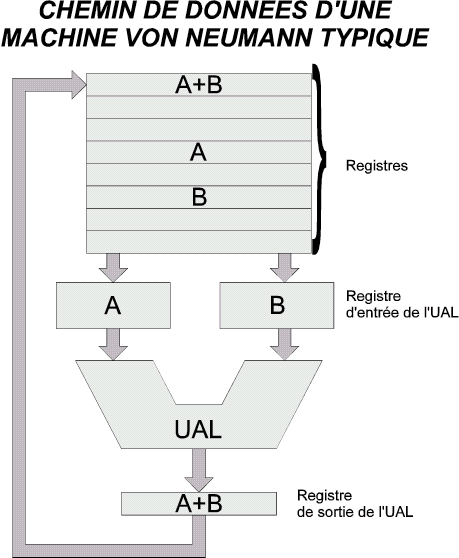
**Généralités sur les systèmes à microcontrôleur (µC) : le logiciel (notion de variable)**

**Correction du TD1**



**Connaissances**

Système de numération, codage.

**Mots-clés**

UAL(ALU) – variable – opération arithmétique – opération logique

**Activité 1 :** 38(10) = 00100110(2) 127(10) = 01111111(2) 218(10) = 11011010(2)

**Activité 2 :** var3 a été déclaré de type **octet non signé (8bits)**

Var3

Cas 1 : 38(10) + 127(10) donne 165(10)

00100110(2) + 01111111(2) = 10100101(2) = 165(2)

La valeur contenue dans var3 est juste, car le nombre de bits nécessaires au résultat est **compatible** avec son type.

Var3

Cas 2 : 38(10) + 218(10) donne 256(10)

00100110(2) + 11011010(2) = 100000000(2) = 256(2)

La valeur contenue dans var3 est fausse, car le nombre de bits nécessaires au résultat est **incompatible** avec son type. Dans le cas présent, le type de var3 doit être un entier non signé codé sur **16 bits**.

*Algorithme*

Var3 : entier non signé sur 16 bits

*C#*

ushort var3 = 0 ; ou UInt16 var3 = 0 ;

**Activité 3**

Solution : déclarer val en tant que variable codée sur 32 bits

*C#*

UInt32 val = 65538 ; ou uint val = 65538 ;

**Activité 4**

**Complément à 2 de -1(10)**

1(10) = 00000001(2)

Complément à 1 de 1(10) = 11111110(2)

Complément à 2 de 1(10) = [Complément à 1 de 1(10)] + 1] = 11111110(2) + 1

= 11111111(2)

Donc **-1(10) = 11111111(2) = FF(16)**

**Complément à 2 de -129(10)**

On sait que **-2n-1 ≤ N(10) ≤ 2n-1 – 1**

Donc -129 ne peut pas être codé sur 8 bits, car un nombre signé sur 8 bits est compris entre -128 et + 127.

***C#***

Sbyte var1 = -1 ; Int16 var2 = -129 ;