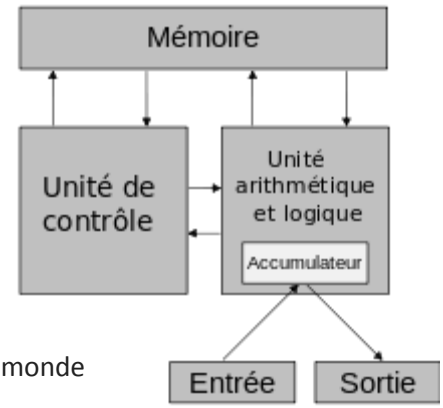
Première	Cours : Constitution et fonctionnement d'une machine numérique	

A- La machine de Von Neumann (source : Wikipedia)

Von Neumann a donné son nom à l'architecture de Von Neumann utilisée dans la quasi-totalité des ordinateurs modernes en 1945. Le modèle de calculateur à programme auquel son nom reste attaché et qu'il attribuait lui-même à Alan Turing, possède une unique mémoire qui sert à conserver les instructions et les données. Ce modèle, extrêmement innovant pour l'époque, est à la base de la conception de nombre d'ordinateurs.

L'architecture de Von Neumann décompose l'ordinateur en quatre parties distinctes :

1. l'unité arithmétique et logique (UAL) ou unité de traitement, qui effectue les opérations de base ;
2. l'unité de contrôle, qui est chargée du séquençage des opérations ;
3. la mémoire, qui contient à la fois les données et le programme qui indique à l'unité de contrôle quels calculs faire sur ces données. La mémoire se divise en mémoire vive (programmes et données en cours de fonctionnement) et mémoire de masse (programmes et données de base de la machine) ;
4. les dispositifs d'entrées-sorties, qui permettent de communiquer avec le monde extérieur.

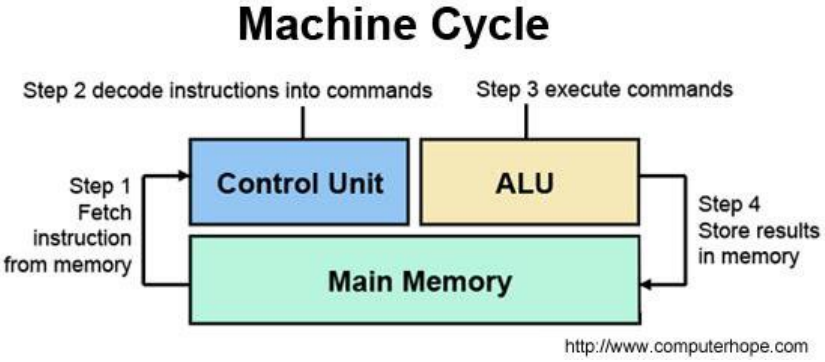


B- Cycle machine d'exécution d'une instruction.

Une instruction s'exécute en quatre temps :

Etape 1 : Récupération de l'instruction pointée par le « Program Counter ».
Etape 2 : Décodage de l'instruction.
Etape 3 : Exécution de l'opération.
Etape 4 : Mémorisation du résultat.

On peut observer ce fonctionnement sur le site http://www.peterhigginson.co.uk/AQA/ qui simule le fonctionnement d'une machine de type Von Neumann.
Le tuto de LMC.

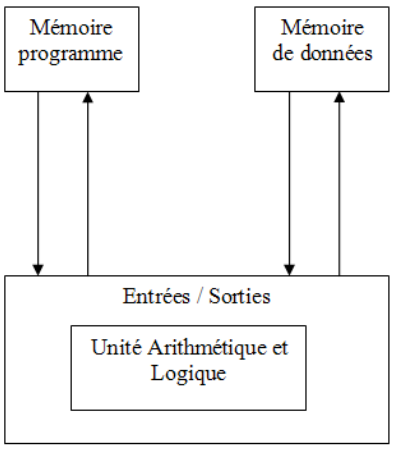


C- Architecture de type Harvard

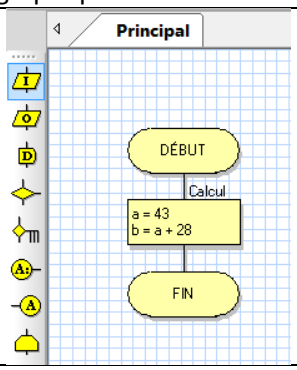
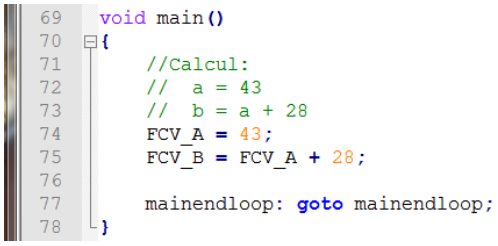
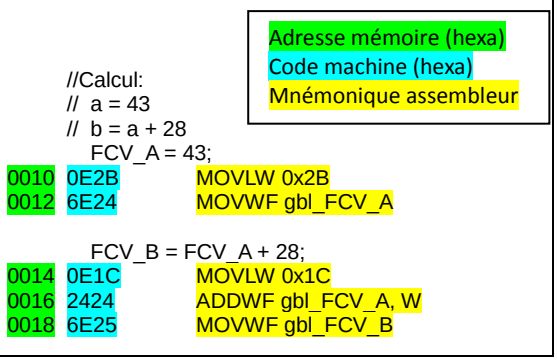
L'**architecture de type Harvard** est une conception des processeurs qui sépare physiquement la mémoire de données et la mémoire programme. L'accès à chacune des deux mémoires s'effectue via deux bus distincts.

Le nom de cette structure vient du nom de l'université Harvard où une telle architecture a été mise en pratique pour la première fois avec le Mark I en 1944.

Les µcontroleurs de la famille PIC18Fxxx de Microchip étudié dans ce qui suit respecte l'architecture Harward.



D- Etapes d'élaboration des codes machines à partir d'un langage de haut niveau.

Programmation graphique sous Flowcode	Traduction du langage graphique en langage C	Traduction du langage C en assembleur puis en langage machine.
		


```
69  void main()
70  {
71      //Calcul:
72      //  a = 43
73      //  b = a + 28
74      FCV_A = 43;
75      FCV_B = FCV_A + 28;
76
77      mainendloop: goto mainendloop;
78  }
```

```
Adresse mémoire (hexa)
Code machine (hexa)
Mnémonique assembleur

      //Calcul:
      // a = 43
      // b = a + 28
         FCV_A = 43;
0010  0E2B        MOVLW 0x2B
0012  6E24        MOVWF gbl_FCV_A

         FCV_B = FCV_A + 28;
0014  0E1C        MOVLW 0x1C
0016  2424        ADDWF gbl_FCV_A, W
0018  6E25        MOVWF gbl_FCV_B
```

E- Fonctionnement de l'architecture simplifiée d'un µcontrôleur PIC18F252

20.1 Instruction Set - PIC18FXX2 — Source : DS39564C-page 217 © 2006 Microchip Technology Inc.

ADDWF — ADD W to f

Syntax:	[*label*] ADDWF f [,d [,a]
Operands:	0 ≤ f ≤ 255 d ∈ [0,1] a ∈ [0,1]
Operation:	(W) + (f) → dest
Status Affected:	N, OV, C, DC, Z
Encoding:	0010 01da ffff ffff
Description:	Add W to register 'f'. If 'd' is 0, the result is stored in W. If 'd' is 1, the result is stored back in register 'f' (default). If 'a' is 0, the Access Bank will be selected. If 'a' is 1, the BSR is used.

MOVLW — Move literal to W

Syntax:	[*label*] MOVLW k
Operands:	0 ≤ k ≤ 255
Operation:	k → W
Status Affected:	None
Encoding:	0000 1110 kkkk kkkk
Description:	The eight-bit literal 'k' is loaded into W.

MOVWF — Move W to f

Syntax:	[*label*] MOVWF f [,a]
Operands:	0 ≤ f ≤ 255 a ∈ [0,1]
Operation:	(W) → f
Status Affected:	None
Encoding:	0110 111a ffff ffff
Description:	Move data from W to register 'f'. Location 'f' can be anywhere in the 256 byte bank. If 'a' is 0, the Access Bank will be selected, overriding the BSR value. If 'a' = 1, then the bank will be selected as per the BSR value (default).