

2 – L'architecture des microprocesseurs

1. L'ordinateur l'aspect matériel

C'est une machine - le matériel (**hardware**) - capable de réaliser une liste d'opérations élémentaires - le logiciel ou programme (**software**) - de façon séquentielle à une très grande vitesse. L'ordinateur n'est pas intelligent mais c'est l'homme qui lui dicte comment utiliser sa base de connaissance - les **instructions** -

Le **micro-ordinateur** est un ordinateur dont le cerveau ou unité centrale est composé d'un ou plusieurs microprocesseurs.

2. Le Microprocesseur

Le **microprocesseur** est un composant électronique qui possède la bibliothèque des fonctions à réaliser -Unité Arithmétique et Logique (**ALU**) - et qui dialogue avec les autres composants du micro-ordinateur par un certain nombre de fils électrique appelés **BUS**

Les opérations qu'il peut réaliser sont élémentaires (chargement, rangement, addition, soustractions,...). Pour réaliser des opérations plus complexes il faut assembler plusieurs instructions. Encore une fois, il ne possède pas d'intelligence, c'est à dire qu'il ne sait pas comment ni dans quel ordre il faut employer ses instructions. Cette intelligence est le fait du programme qui est écrit par un Homme (expert !)

3. Les Bus

Le microprocesseur doit communiquer avec d'autres composants par des BUS (canaux de communication). Les ordres passés au microprocesseur sont rangés dans des cases mémoires. Chacune de ces cases mémoire contient une instruction. Pour réaliser un programme, donc donner au microprocesseur plusieurs ordres à la suite, il faut avoir plusieurs cases mémoires disponibles. A tout moment le microprocesseur doit pouvoir sélectionner - **adresser** - une case mémoire pour en lire (ou éventuellement écrire) sont contenu.

Le contenu des informations lues est transféré dans les **registres** internes du microprocesseur par des fils : le **bus des données**.

La sélection des cases mémoires est réalisé par d'autres fils le **bus des adresses**.

D'autres ordres - lecture-écriture - sont échangés avec le microprocesseur par le **bus de contrôle**.

4. Les interfaces ou périphériques = liaison avec l'environnement

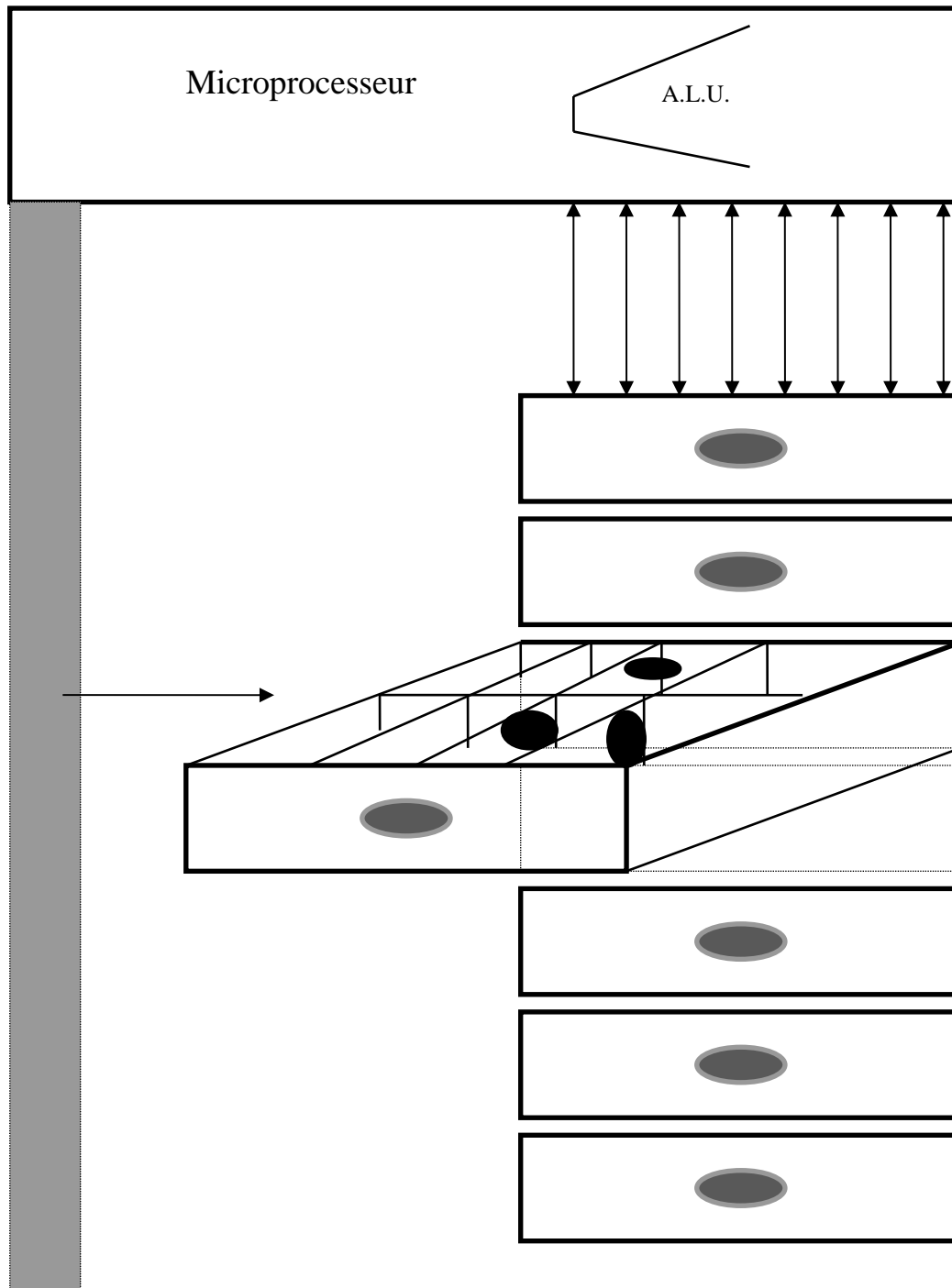
Pour rendre le micro-ordinateur utile, il doit pouvoir communiquer avec son environnement :

- un écran, un clavier et une souris pour un dialogue avec l'utilisateur;
- une imprimante, des disques et disquettes - mémoire de masse - pour archiver;
- un modem ou un réseau pour échanger des données avec d'autres ordinateurs;
- le son et l'image pour obtenir un aspect plus convivial;
- des entrées et des sorties pour réaliser des liaisons avec des machines extérieures.

2 – L'architecture des microprocesseurs

5. Le Microprocesseur

a. Représentation du microprocesseur et de sa mémoire



2 – L'architecture des microprocesseurs

6. Les différentes organisations de l'espace mémoire

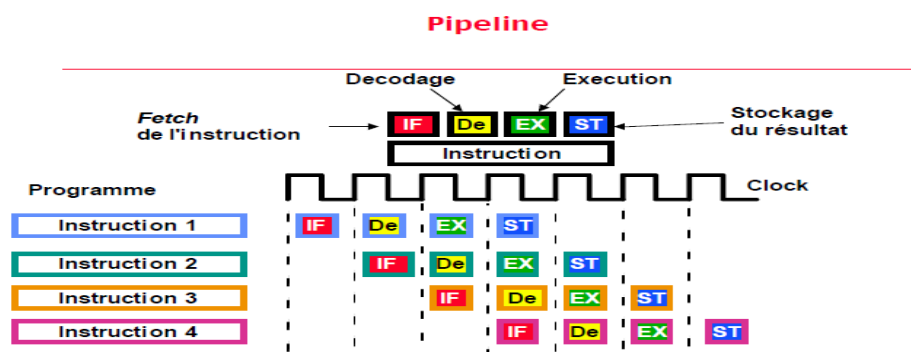
- L'architecture **HARVARD** dissocie la mémoire programme de la mémoire des données de ce fait le microprocesseur peut accéder en même temps aux instructions et aux données, La largeur du bus des instructions et des données peut être différent.
- Avec l'architecture **Von NEUMANN** le microprocesseur ne possède qu'un seul plan mémoire, il est plus facile à gérer et à organiser. Les données sont au même format que les instructions.

Les phases de traitement d'une instruction

1. Lecture
2. Décodage
3. Exécution
4. Accès mémoire
5. Résultat
6. Mise à jour « Status »
7. PC+1

7. Les structures : CISC et RISC, ...ARM

Un microprocesseur **CISC** (Jeu d'instructions complexes) permet de réaliser des instructions qui peuvent nécessiter plusieurs cycles machine (microcode) dans la phase de traitement (multiplication par exemple), Un microprocesseur **RISC** (Jeu d'instructions réduits) permet de réaliser des instructions qui seront traitées en un seul cycle machine grâce à sa structure « **PipeLine** ».



8. Synoptique d'une carte à microprocesseur

2 – L'architecture des microprocesseurs

9. Les mémoires

Elle permet de stocker les instructions du microprocesseur pour pouvoir être relue par la suite lors de l'exécution du programme.

Le deuxième rôle des mémoires est de stocker des résultats de calcul et des données temporaires.

On peut classer en trois catégories les mémoires : les mémoires vives, mortes et les mémoires de masse

9.1- Les mémoires vives :

Elles sont appelées ainsi (par abus de langage) les **RAM** ce qui signifie mot à mot mémoire à accès aléatoire. Ceci pour les différencier des bandes magnétiques de l'époque ... (qui étaient séquentielles).

N'importe quelle case de ces mémoires peuvent être écrite ou lue à n'importe quel instant. C'est pour cela qu'on les appelle des mémoires vives. Elles ont besoin d'énergie électrique pour conserver leurs informations. Plusieurs technologies existent pour réaliser ces composants (transistor Nmos, Cmos,...). Certaines d'entre elles ont besoin d'un cycle de rafraîchissement pour garder leur contenu ; ce sont les RAM dynamiques ou les **DRAM** ou encore **VRAM**.

9.2- Les mémoires mortes :

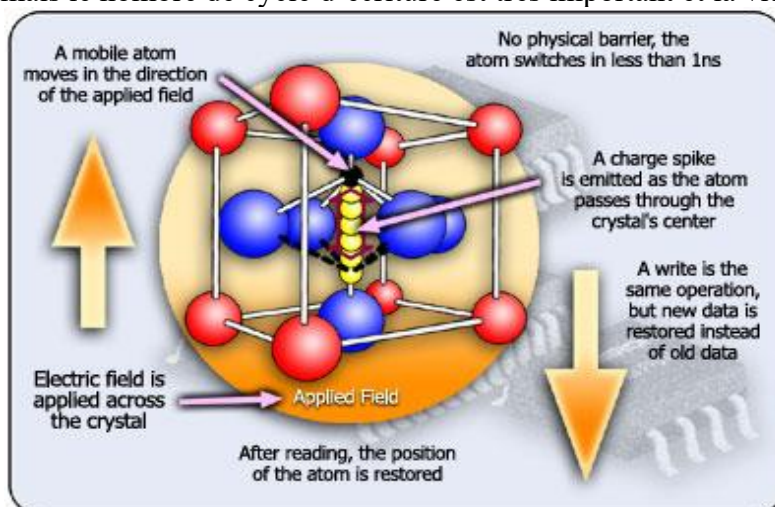
Elles sont appelées ainsi car elles ne peuvent pas être modifiées par une simple écriture. On les appelle d'un terme générique des **ROM** (mémoire uniquement en lecture).

Il en existe de plusieurs technologies :

- **ROM** les informations inscrites à la fabrication
- **PROM** des mémoires programmables une fois « **OTP** » généralement à fusibles.
- **UVPROM** ou **EPROM** des mémoires effaçables par une lampe à ultra violet qui peuvent être programmées plusieurs fois, mais nécessite de retirer la mémoire de son emplacement.
- **EEPROM** ou **E2PROM** des mémoires effaçables électriquement qui peuvent être programmées plusieurs fois in situ. On les appelle également **Mémoire FLASH**
- Il existait dans les années 80 des mémoires à « **tore de ferrite** » et des mémoires à « **bulles magnétiques** ».

9.3 - Un nouveau type de mémoire est apparu : « **FRAM** » ou « **EERAM** »

Les « **FRAM** » pour mémoires ferroélectriques peuvent être lues et écrites un peu comme les mémoires flash, mais le nombre de cycle d'écriture est très important et la vitesse d'écriture est accrue.



Quand un champ électrique est appliqué à un cristal ferroélectrique, l'atome central se déplace dans la direction du champ. Il passe la barrière d'énergie causant une pointe de charge. Des circuits internes captent le sens de cette charge et activent la cellule de la mémoire. Quand le champ électrique est supprimé l'atome reste dans cette position gardant l'information stockée.

2 – L'architecture des microprocesseurs

9.4- Les mémoires de masse :

Le **disque dur** représente actuellement un excellent support d'informations et de grande capacité. Actuellement leur capacité varie de 2To à 100To.

La **disquette** était un support mobile qui pouvait être facilement déplacé d'un ordinateur à l'autre, elle permet également des copies de logicielle pour effectuer des sauvegardes.

Les mémoires flash (E2prom à connexion USB) appelée « Clé USB » remplacent avantageusement ces supports. 1Go à 256Go.

Le disque laser ou **CD-ROM** ainsi que le disque ré-inscriptible DVD ont actuellement de grandes capacités.

Les **cartouches magnétiques** permettent d'effectuer des sauvegardes plus rapides et plus condensées; différentes technologies existent et leur capacité varie de 200Mo à 200Go.

Les **cartes perforées** étaient utilisées dans les années 70, aujourd'hui nous avons des cartes à pistes magnétiques. Les disques **Bernouilli** et les disques **magnéto-optiques** sont aussi des mémoires de masse.

D'autres cartes couramment utilisées sont les cartes dites « **cartes à puce** ». En fait elle comporte une mémoire de type E2Prom, ou carrément un microcontrôleur intégré.

!!! N'oublions pas un support qui peut être considéré comme mémoire de masse : *le papier...*

UM2-L2 EEA : HLEE407 – Programmation des microcontrôleurs

2 – Notions de physique appliquée (Temps et Fréquence)

10. La vitesse des signaux

Quelques notions à connaître :

- La vitesse d'un électron libre dans un conducteur est de 1mm/s.
- La vitesse d'agitation thermique des électrons est de 120 000m/s.
- La vitesse de la lumière $C = 299\,792\,458\text{ m/s}$ soit environ $= 3 \times 10^8\text{ m/s}$ (300.000 km/s)
- La vitesse du son dans le vide = 0 ; dans l'air = 340 m/s ; dans l'eau = 1435 m/s ; dans la mer = 1500 m/s ; dans le béton = 3100 m/s ; dans l'acier = 5200 m/s
- La vitesse du signal électrique (de la lumière) est environ $C = 300\,000\,000\text{ m/s}$.

10.1 Les relations entre le temps, la fréquence et la longueur d'onde

$F = 1/T$ la fréquence en Hz T la période en s

$\lambda = C/F$ la longueur d'onde en m C la célérité en m/s F la fréquence en Hz

10.2 Domaine du spectre électromagnétique

Fréquence du secteur (en France)	= 50Hz	
Spectre sonore (Basse fréquence)	= 20Hz à 20KHz	
Ondes radio (Radiofréquence)	= 100KHz à 300MHz	3000m à 1m
Micro-Ondes	= 300MHz à 300GHz	1m à 1mm
Téra hertz (Submillimétrique)	= 300GHz à 3THz	1mm à 100µm
Infrarouge	= 3THz à 400THz	100µm à 750nm
Visible	= 400THz à 770THz	750nm à 390nm
Ultraviolet	= 750THz à 30PHz	390nm à 10nm
Rayon X	= 30PHz à 30EHz	10nm à 10pm
Rayon Gamma	> 30EHz	< 10pm

10.3 Notions pratiques

Les calculs de propagation des signaux électriques sur une carte numérique :

30m	10MHz
10m	30MHz
3m	100MHz
1m	300MHz
30cm	1GHz
10cm	3GHz
3cm	10GHz
1cm	30GHz
3mm	100GHz
1mm	300GHz

UM2-L2 EEA : HLEE407 – Programmation des microcontrôleurs

2 – L'organisation logicielle

11- Software ou l'aspect logiciel

Un **PROGRAMME** est une suite logique et chronologique de fonctions élémentaires à réaliser appelées **INSTRUCTIONS**.

Chaque instruction représente un ordre précis indiquant ce que doit faire le microprocesseur.

Un ensemble d'instructions peuvent être regroupées ensemble pour réaliser une action précise. Si cette action peut servir plusieurs fois dans le programme il est judicieux d'en faire une **PROCEDURE**.

11.1 Les routines

Un **SOUS-PROGRAMME** est une partie d'un programme qui peut être appelé de plusieurs endroits du programme principal. Sa particularité est que la dernière instruction permet de part l'utilisation de la sauvegarde dans la pile opérationnelle « l'adresse de retour », de revenir au programme appelant.

Une **PROCEDURE** est un sous-programme qui ne redonne pas de paramètre au programme appelant.

Une **FONCTION** est un sous-programme qui rend un paramètre au programme appelant.

La **RE-ENTRANCE** d'une routine permet l'appel de cette routine soit par elle-même soit par un autre programme avant que la première exécution ne soit terminée, sans perdre les informations du premier appel.

11.2- Les FIFO (les files)

Les **FILES** correspondent à une **LISTE** de cases mémoire où les informations qui rentrent les premières seront les premières utilisées (First In - First out). C'est l'application directe d'un Registre à Décalage.

11.3- Les LIFO (les piles)

Les **PILES** permettent de ranger des informations et de pouvoir les récupérer dans **ORDRE INVERSE** du rangement (Last In - First out). L'application de la pile opérationnelle d'un microprocesseur.

11.4- Les Pointeurs

Le pointeur d'adresse est une mémoire ou un registre qui contient l'adresse d'une case mémoire, c'est à dire qu'il montre cette case.

12- Les Masques

Le microprocesseur ne peut traiter qu'un mot entier ; dans certains logiciels il est parfois utile de pouvoir tester un bit d'un mot pour en connaître l'état '0' ou '1'.

De même il est utile de pouvoir modifier l'état d'un seul bit sur un mot sans en changer la valeur des autres.

12.1- Le filtre en ET l'instruction « ANDA # »

Le filtre en « ET » permet d'extraire l'information d'un ou plusieurs bits et d'affecter une variable à 0 si ce ou ces bits sont à '0' ou différente de 0 si au moins un de ces bits est à '1'

N° Bit= 7 6 5 4 3 2 1 0

Mem= X X X X X X X X

ET 0 0 0 1 0 0 0 0 ← Filtre en « ET »

Résultat= 0 0 0 X 0 0 0 0

Dans cet exemple le Résultat sera **nul** si le bit 4 est =0 ou **non nul** si le bit 4 est =1

12.2- Le forçage à '0' par un masque en ET l'instruction « ANDA # »

Le forçage à '0' par un masque en « ET » permet de positionner à '0' le ou les bits ainsi affectés.

N° Bit= 7 6 5 4 3 2 1 0

Mem= X X X X X X X X

ET 1 1 1 1 0 1 1 1 ← Masque en « ET »

Resultat= X X X X 0 X X X

Dans cet exemple le bit 3 du Résultat sera =0 et les autres bits resteront inchangés.

Note : ici le masque est le complément du bit ou des bits que l'on désire mettre à 0

12.4- Le forçage à '1' par un masque en OU l'instruction « ORAA # »

UM2-L2 EEA : HLEE407 – Programmation des microcontrôleurs

2 – L'organisation logicielle

13- Les interruptions

A tout instant une action extérieure au programme peut interrompre le déroulement d'un logiciel pour traiter une routine spécifique au phénomène qui a déclenché l'interruption.

Il en existe de plusieurs sortes et suivant le cas elles peuvent être hiérarchisées :

- le Reset -
- la NMI -
- l'IRQ -
- les SWI –

14- La pile opérationnelle

Elle permet le stockage des adresses de retour en cas d'appel de sous-programme ou d'interruption.

Il est également possible de stocker dans cette zone mémoire des valeurs intermédiaires de calcul.

15- Le multitâche

Le microprocesseur de par sa structure ne peut effectuer qu'un seul programme en même temps. Du fait de sa rapidité d'exécution, il est possible d'interrompre de façon cadencée le déroulement d'un programme pour en traiter un autre. L'utilisateur aura l'impression de voir « **tourner** » deux ou plusieurs programmes en même temps.

On peut effectuer ces coupures de plusieurs façons pour « donner la main » aux programmes les uns après les autres, c'est le rôle de l'organisateur de tâches « Scheduler »

- le temps partagé
- la tâche de fond

16- Les Langages

Pour programmer les microprocesseurs il faut ranger en mémoire une liste d'instructions qui seront comprises par les microprocesseurs. La première possibilité est de directement inscrire les **CODES OPERATOIRES** en mémoire en lieu et place en vue du fonctionnement du programme. Il est très difficile de connaître par coeur, pour un seul microprocesseur directement ces codes (valeurs numériques en hexadécimal); suivant les microprocesseurs il y a entre 200 et 5000 codes différents ! Ces codes s'appellent le **LANGAGE MACHINE**. Il est possible d'utiliser un langage plus près de l'utilisation de la fonction par exemple « ADD » pour signifier réaliser une addition. Cette façon de transcrire les textes **MNEMONIQUES** en langage machine s'appelle « **ASSEMBLER** ».

Par cette façon de procéder il faut que le **programmeur** connaisse correctement toutes les possibilités opératoires qu'offre le microprocesseur en question.

En vue de généraliser un logiciel, des **langages de programmation évoluée** ont été mis en place; ceux-ci ne dépendent pas du microprocesseur, mais ont une syntaxe standardisée.

- BASIC, PASCAL, ADA, FORTRAN, COBOL, ALGOL, MAPLE, langage C

Des ateliers de développement rapide d'application «**RAD**» tel que:

- VB, Window, Langage G, Labwiev

Tous ces langages ont leur propre **COMPILATEUR** pour fournir un programme compatible avec le microprocesseur utilisé et sous l'environnement souhaité.

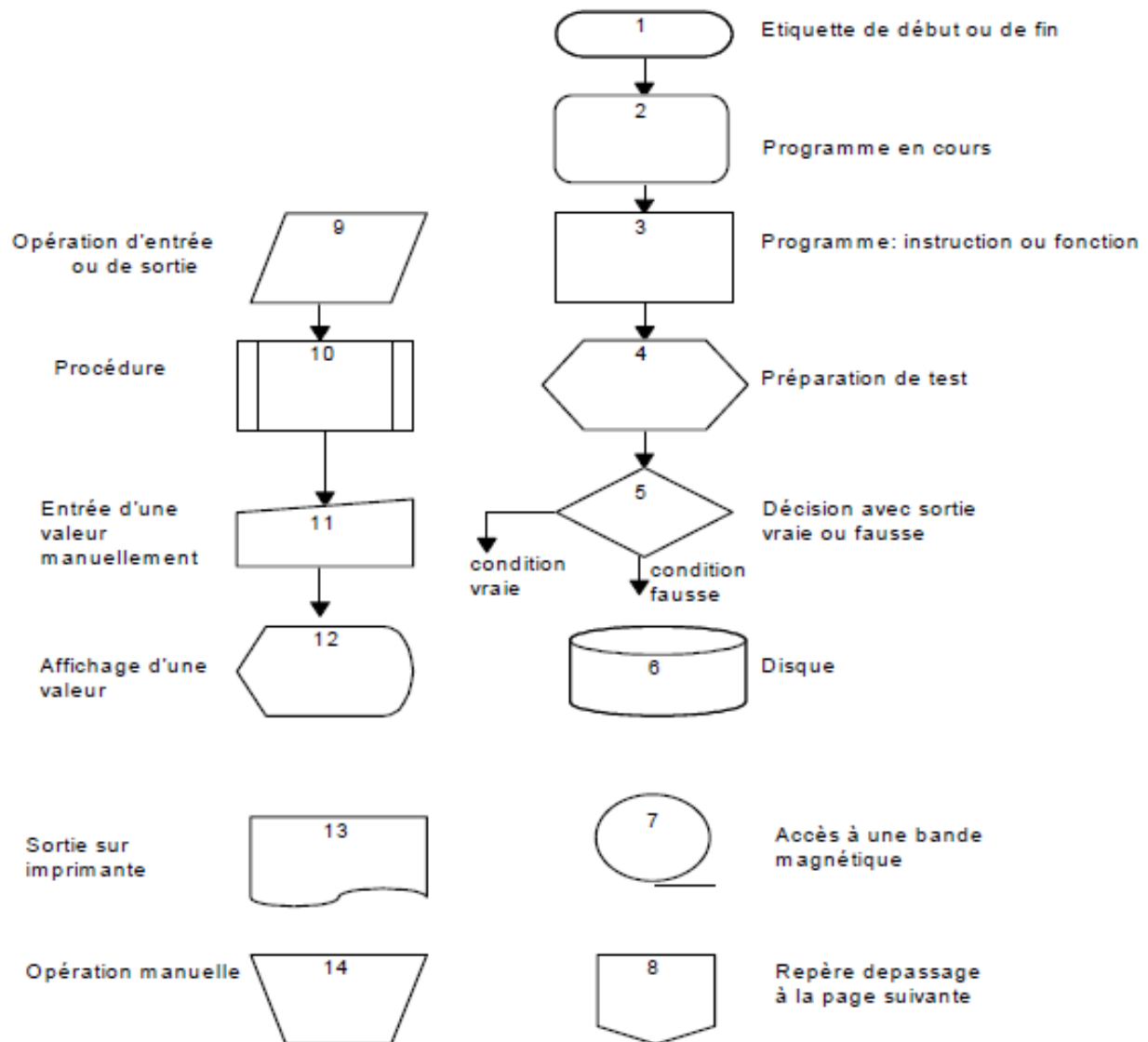
Certain de ces langages tels que le basic ou le pascal fournissent un code intermédiaire qui peut être lu et interprété par un **INTERPRETEUR**.

Dans tous les cas lorsque le programmeur édite la liste des instructions pour créer son programme, il le fait sur un éditeur de texte plus ou moins adapté à la programmation. Ce texte décrivant le logiciel s'appelle « **SOURCE** ».

Quand le programmeur a assemblé ou compilé son programme, le résultat est un fichier contenant les codes binaires compatibles au microprocesseur sur lequel le logiciel doit être implanté. Ce fichier s'appelle « **OBJET** ».

2 – L'organisation logicielle

17- Les symboles normalisés pour l'analyse



UM2-L2 EEA : HLEE407 – Programmation des microcontrôleurs**3 – L'architecture des microprocesseurs****Mots Clé:**

Software, Hardware, Firmware.

Alu, Bus, Registres, Accumulateur, Index.

Harvard, VonNeumann.

CISC, RISC, Pipeline.

Ram, Rom, UVProm, EEProm, Flash.

HD, FD, CD, DVD.

A Savoir:

Rôle des bus.

Le synoptique d'une carte à microprocesseur.

Les phases du traitement d'une instruction.

En Plus:

Les différents types de mémoire.