

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIVERSITE Mouloud MAMMARI DE TIZI-OUZOU



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'ELECTRONIQUE

Mémoire de Fin d'Etudes
De MASTER ACADEMIQUE

Domaine : Sciences et Technologies

Filière : Génie électrique

Spécialité : **Instrumentation**

Présenté par :

Khendriche Ishak

Thème

**Implémentation de systèmes à microprocesseur dans
l'informatique embarquée**

Encadré par : Mr Lazri.M

Promotion 2017/2018

REMERCIEMENT

Je remercie dieu le puissant de m'avoir donné les aptitudes pour mener à bien ce
modeste travail

Un grand merci pour mon promoteur **Mr Lazri.M** pour sa disponibilité, son suivi et
ses conseils tout au long de notre travail

Je tiens également à remercier les membres du jury qui me feront l'honneur de juger
notre travail

Je remercie aussi Mr Oualouche.F pour son soutien et tous les enseignants du
département électronique

Pour finir je remercie mes parents et mes amis qui m'ont soutenue durant tous mon
cursus universitaire

DEDICACE

A mon très cher père (Arezki Khendriche),

Autant de phrases et d'expressions aussi éloquentes soit-elles ne sauraient exprimer ma gratitude et ma reconnaissance. Tu as su m'inculquer le sens de la responsabilité, de l'optimisme et de la confiance en soi face aux difficultés de la vie. Tes conseils ont toujours guidé mes pas vers la réussite. Ta patience sans fin, ta compréhension et ton encouragement sont pour moi le soutien indispensable que tu as toujours su m'apporter. Je te dois ce que je suis aujourd'hui et ce que je serai demain et je ferai toujours de mon mieux pour rester ta fierté et ne jamais te décevoir.

Que Dieu, le miséricordieux, t'accueille dans son éternel paradis.

Table de matières

Introduction générale :10

Chapitre I : Electronique numérique :

I.1 : Préambule : 12

I.2 : Définition de l'électronique numérique..... **Erreur ! Signet non défini.**

I.3 : Système binaire (bits et octets) :..... **Erreur ! Signet non défini.**

I.4 : Le principe fondamental de l'électronique numérique : **Erreur ! Signet non défini.**

I.5 : Discussion :..... **Erreur ! Signet non défini.**

Chapitre II Système à microprocesseur

II.1 Préambule :..... **Erreur ! Signet non défini.**

II.2 : définition du microprocesseur :..... **Erreur ! Signet non défini.**

II.2 : Architecture et conception d'un microprocesseur : ... **Erreur ! Signet non défini.**

II.2.1 : Architecture d'un microprocesseur :..... **Erreur ! Signet non défini.**

II.2.2 : Architecture d'un CPU :..... **Erreur ! Signet non défini.**

II.2.2.1- Les registres: **Erreur ! Signet non défini.**

II.2.2.2 : Unité arithmétique et logique :..... **Erreur ! Signet non défini.**

II.2.2.3 : Unité de contrôle et de commande :..... **Erreur ! Signet non défini.**

II.3 : Fonctionnement d'un système à base de microprocesseur : ... **Erreur ! Signet non défini.**

II.3.1 : Les interruptions :..... **Erreur ! Signet non défini.**

II.3.2 : L'écriture en mémoire (WRITE) :41

II.3.3 : La lecture de la mémoire (READ) :42

II.3.4 : Wait Stat (temps d'attente pour la synchronisation) : **Erreur ! Signet non défini.**

II.3.5 : Communication avec les entrées/sorties : **Erreur ! Signet non défini.**

II.3.6 : Accès direct à la mémoire (DMA) : **Erreur ! Signet non défini.**

II.6 : discussion : **Erreur ! Signet non défini.**

Chapitre III : Electronique et informatique embarquée

III.1 : préambule :.....50

III.2 : Définition de l'informatique embarquée :50

III.3 : Architecture d'un système embarqué :51

III.4 : Caractéristiques des systèmes embarqués:.....52

III.5 :Contraintes de conception des systèmes embarqués :53

III.6 : Domaines d'application des systèmes embarqués :54

III.7 : Discussion :55

Conclusion générale :.....56

Liste des figures

Chapitre I :

- Figure I.1 : Un exemple de nombre en notation binaire et sa valeur en notation décimale.....**Erreur ! Signet non défini.**
- Figure I.2 : Numération décimal et binaire**Erreur ! Signet non défini.**
- Figure I.3 : Exemple sur la décomposition du nombre binaire 1011 16
- Figure I.4 : Exemple sur les chiffres des jours divins en code binaire 18
- Figure I.5 : Exemple sur une page de texte brut..... 19
- Figure I.6 : une mémoire Ram d'un PC**Erreur ! Signet non défini.**
- Figure I.7 : Une disquette (1.44Mo)**Erreur ! Signet non défini.**
- Figure I.8 : Une disquette de grande capacité (750Mo).....**Erreur ! Signet non défini.**
- Figure I.9 : Carte mémoire de type flash**Erreur ! Signet non défini.**
- Figure I.10 : Disque dur interne et disque dur externe**Erreur ! Signet non défini.**
- Figure I.11 : CD-ROM.....**Erreur ! Signet non défini.**
- Figure I.12 : DVD-ROM de 4,7Go.....**Erreur ! Signet non défini.**
- Figure I.13 : passage du signal analogique au signal numérique.....**Erreur ! Signet non défini.**
- Figure I.14 : Conversion binaire/ décimal/ hexadécimal**Erreur ! Signet non défini.**

Figure I.15 : Signal binaire transmis en bande de base.....**Erreur ! Signet non défini.**

ChapitreII :

II.2.1 : Architecture d'un microprocesseur :	36
Figure II.2 : Architecture interne d'un microprocesseur	36
Figure II.3 : Schéma sur la répartition du bus de données et le bus d'adresse	38
Figure II.4 : Structure interne d'un CPU	40
Figure II.5 : Représentation des registres dans un microprocesseur.....	41
Figure II.6 : Instruction informatique dans U.A.L.....	43
Figure II.7 : exemple sur l'unité de commande sur ordinateur.....	44
Figure II.8 : Schéma explicatif sur les interruptions.....	46
Figure II.9 : L'écriture en mémoire	46
Figure II.11 : Circuits intermédiaires entre la CPU et la mémoire.....	48

Chapitre III :

Figure III.1 : Architecture d'un système embarqué	50
Figure III.2 : Un système embarqué dans son environnement.....	51
Figure III.3 : exemples de systèmes embarqués	53
Figure III.4 : multiples domaines d'applications des systèmes embarqués.....	55

Liste des abréviations

PC *Personal Computer*

LED *Light Emitter Diode*

LCD *Liquid Crystal Display*

ASIC *Application Specific Integrated Circuits*

FPGA *Field Programmable Gate Array*

Mo Mégaoctet

Go Gigaoctet

ROM *Read Only Memory*

RAM *Random Access Memory*

BIOS *Basic Input Output system*

RI Registre d'instructions

ADC *Analog to digital converter*

DAC *Digital to analog converter*

uC Microcontrôleur

Introduction générale

L'informatique embarquée est basée sur un système complexe qui intègre du logiciel et du matériel conçus ensemble afin de fournir des fonctionnalités **données**. **Ce système contient généralement un ou plusieurs microprocesseurs** destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires. Le système matériel et l'application (logiciel) sont intimement liés et immergés dans le matériel et ne sont pas aussi facilement discernables comme dans un environnement de travail classique de type ordinateur de bureau PC (*Personal Computer*) .

Un système embarqué est autonome et ne possède pas des entrées/sorties standards tels qu'un clavier ou un écran d'ordinateur. Contrairement à un PC, l'interface IHM (Interface Homme machine) d'un système embarqué peut être aussi simple qu'une diode électroluminescente LED (*Light Emitter Diode*) qui clignote ou aussi complexe qu'un système de vision de nuit en temps réel ; les afficheurs à cristaux liquides LCD (*Liquid Crystal Display*) de structure généralement simple sont couramment utilisés.

Afin d'optimiser les performances et la fiabilité de ces systèmes, des circuits numériques programmables FPGA (*Field Programmable Gate Array*), des circuits dédiés à des applications spécifiques ASIC (*Application Specific Integrated Circuits*) ou des modules analogiques sont en plus utilisés.

Le logiciel a une fonctionnalité fixe à exécuter qui est spécifique à une application. L'utilisateur n'a pas la possibilité de modifier les programmes. Bien souvent, il n'a pas conscience d'utiliser un système à base des microprocesseurs.

Chapitre I

Électronique

numérique

I.1 Préambule

On évoque dans ce chapitre les généralités de l'électronique numérique, un domaine scientifique s'intéressant aux systèmes électroniques et présenter les fonctionnalités et les avantages offerts par ce domaine.

On aborde également les différents systèmes comprenant dans ce domaine et leurs principes de fonctionnement.

I.2 Définition de l'électronique numérique

L'électronique numérique regroupe les systèmes électroniques fonctionnant sur la base d'états électriques précis dont le nombre et les valeurs sont fixés à leur conception, à chaque état correspond une valeur numérique.

La prédétermination de ces états électriques permet de disposer de systèmes qui se comportent de manière stable et fiable. Elle permet en particulier de s'affranchir de bon nombre de parasites et autre déformation, se superposant à tout courant électrique et électronique.

Ce type d'électronique est opposé à l'électronique analogique, qui elle traite des systèmes électroniques opérant sur des grandeurs (tension, courant, charge) continues. En électronique numérique, l'information est représentée sous forme de bits et d'octets. Un PC est un exemple de système numérique.

Les notions de bit et d'octet y sont omniprésentes ("Mo" est l'abréviation de mégaoctet et "Go" de gigaoctet). Elles sont effectivement typiques de l'électronique numérique. Les notions de bit et d'octet sont liées à la manière dont l'information (à traiter par le système électronique) est représentée dans le signal électrique. La

différence fondamentale entre électronique analogique et électronique numérique, c'est la manière dont l'information est représentée dans le signal électrique.

L'électronique numérique a plusieurs avantages dont on peut citer quelques un comme :

a- Le maintien de l'information :

L'information ne se dégrade pas même après avoir été traitée et copiée plusieurs fois contrairement qu'en électronique analogique, tout traitement de l'information (y compris une simple copie) dégrade cette information.

b -Le traitement très complexe :

L'avantage d'une représentation numérique de l'information, c'est qu'on peut faire subir à cette information des traitements beaucoup plus complexes qu'en analogique, ceci est directement lié au fait qu'on peut copier l'information sans la dégrader.

c- la programmation :

Un "programme", c'est un ensemble d'instructions (c'est-à-dire d'informations) qui définissent la fonction d'un circuit. Un circuit qui peut être programmé n'a donc pas de fonction fixe, ce qui lui confère une flexibilité énorme en termes d'applications. Cette possibilité (même si elle est devenue banale) est remarquable: en numérique, on peut donc indiquer via un programme, c'est-à-dire après sa fabrication, ce qu'un circuit doit faire.

combinons plusieurs chiffres pour former des nombres appelés nombres décimaux.

Décimal	Binaire			
0	0	0	0	0
1	0	0	0	1
2	0	0	1	0
3	0	0	1	1
4	0	1	0	0
5	0	1	0	1
6	0	1	1	0
7	0	1	1	1
8	1	0	0	0
9	1	0	0	1
10	1	0	1	0

Figure I.2 : Numération décimal et binaire

Il est possible d'utiliser d'autres bases: rien n'empêche de compter et de calculer en utilisant des chiffres qui peuvent chacun prendre davantage ou au contraire moins de 10 valeurs. C'est ce qu'on fait en électronique numérique, où l'on utilise principalement le système binaire (ou base 2) dans lequel chaque chiffre ne peut prendre que deux valeurs différentes: 0 et 1. Un bit n'est rien d'autre qu'un chiffre en base 2 ("Bit" est d'ailleurs l'acronyme de "binary digit", ce qui signifie bien "chiffre binaire"). En combinant plusieurs bits, on forme des nombres binaires, ce qui est bien utile pour compter au-delà de... 1.

Un bit est aussi une grandeur logique Indépendamment du fait qu'un bit est un chiffre, un bit ne peut prendre que deux valeurs différentes. Un bit est donc également une variable logique. Bien que le bit (valant 0 ou 1) et la variable booléenne (valant

avec N bits? Il est facile de montrer (par dénombrement: voir image ci-dessus) que N bits permettent de représenter 2^N états différents. Il est intéressant dans ce contexte de mémoriser les principales puissances de 2.

JOURS DIVINS	CODE BINAIRE
1	1
2	10
3	11
4	100
5	101
6	110
7	111

Figure I.4 : Exemple sur les chiffres des jours divins en code binaire

Un octet est un mot de 8 bits. En pratique, on utilise très souvent des mots de 8 bits. Ceux-ci ont donc reçu le nom particulier d'octet. Un octet permet de représenter $2^8=256$ états différents, qui peuvent être par exemple: -les nombres entiers de 0 à 255 -les nombres entiers de -128 à 127 -un jeu complet de caractères (majuscules, minuscules, ponctuation, etc.) sur un PC -etc. Le terme "byte" est la traduction anglaise de "octet". Un byte vaut donc 8 bits: attention à ne pas confondre les deux notions (bit et byte) ! L'unité "Kb" désigne un kilobyte. On utilise également souvent

des groupes d'octets: typiquement des mots de 16, 24, 32 ou 64 bits.

L'octet sert d'unité de mesure des quantités d'information numérique. En pratique, un octet peut aussi être vu comme l'unité de base permettant d'évaluer une quantité d'information mise sous forme numérique. On voit ci-dessus des supports numériques courants et la quantité d'information nominale qu'ils peuvent contenir.

Voici quelques exemples de quantités de données tirées du monde de l'informatique (chiffres valables en 2004) :

- une page de texte "brut" (sans mise en page): environ 2Ko

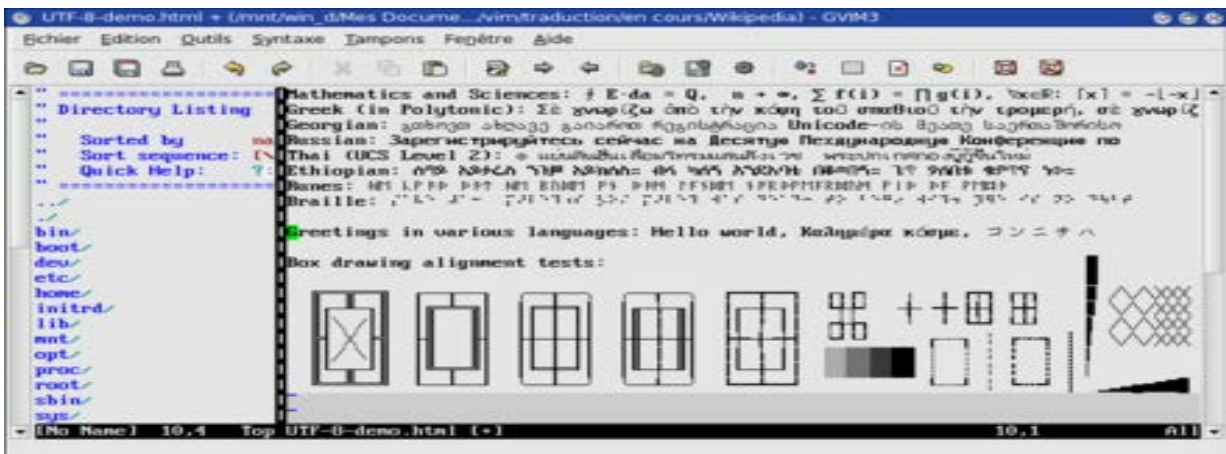


Figure 1.5 : Exemple sur une page de texte brut

- mémoire RAM d'un PC: typiquement 256Mo ou 512Mo



Figure I.6 : une mémoire Ram d'un PC

- disquette (en voie de disparition): 1,44Mo



FigureI.7 : Une disquette (1.44Mo)

- disquette de grande capacité (type "Zip"): 750Mo



FigureI.8 : Une disquette de grande capacité (750Mo)

- carte mémoire de type "flash" (p.ex. pour appareil photo numérique): entre 16Mo et 1Go



FigureI.9 : Carte mémoire de type flash

- disque dur: quelques dizaines à quelques centaines de Go



Figure I.10 : Disque dur interne et disque dur externe

•CD-ROM: 650Mo



Figure I.11 : CD-ROM

•DVD: 4,7Go à 17Go (suivant la technologie)



Figure I.12 : DVD-ROM de 4,7Go

I.4 Le principe fondamental de l'électronique numérique

Le principe fondamental de l'électronique numérique consiste à...

- 1) représenter toute information sous forme de bits ...que cette information soit logique, numérique ou analogique
- 2) représenter chaque bit sous forme de signal électrique Chaque bit issu de l'étape 1 est traduit sous forme de signal électrique. La traduction se fait en associant à chaque état du bit (0 ou 1) une valeur bien précise du signal électrique, par exemple 0V et 5V respectivement pour les états 0 et 1.

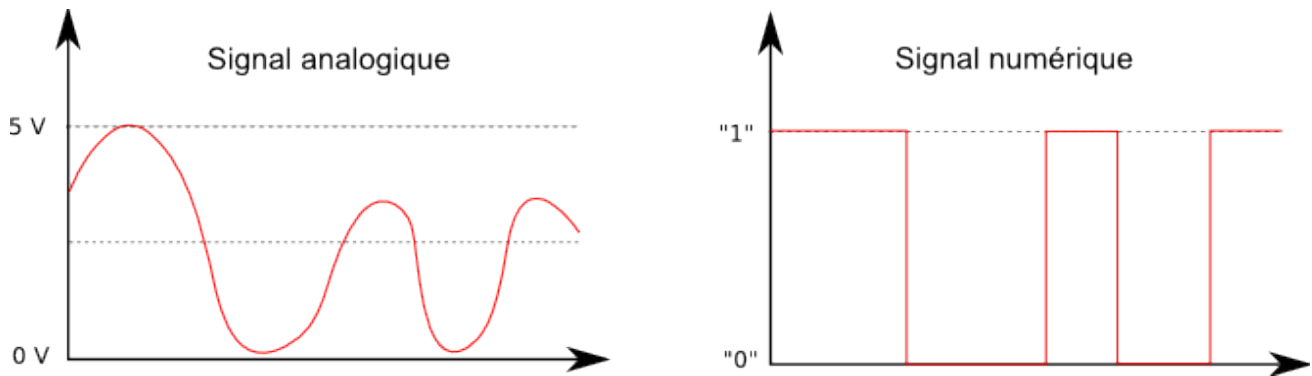


Figure I.13 : passage du signal analogique au signal numérique

1) représenter une information numérique:

N bits + 1 code Supposons maintenant que l'information est numérique, c'est-à-dire que la variable à coder possède un nombre d'états fini mais supérieur à deux (2). Pour coder cette information sous forme de bits, il suffit de prendre suffisamment de bits. Puisque un mot de N bits permet de représenter 2^N états: •pour une information possédant 3 ou 4 états: il faut 2 bits ($2^2=4$) •pour une information possédant entre 5 et 8 états: il faut 3 bits ($2^3=8$) •pour une information possédant entre 9 et 16 états: il faut 4 bits ($2^4=16$) •... •pour une information possédant entre 129 et 256 états: il faut 8 bits ($2^8=256$) •... •pour une information possédant entre $(1+2^{N-1})$ et 2^N états: il faut N bits Exemple: pour représenter la valeur d'une carte dans un jeu de cartes (52 états), il faut 6 bits puisque: $2^5 < 52 < 2^6$ Dans ce cas-ci aussi, il faut préciser le code utilisé.

en puissance	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
en décimal	128	64	32	16	8	4	2	1
nombre à convertir	0	0	0	1	1	0	1	0
résultat				16	8		2	

Résultat :

$$16 + 8 + 2 = 26$$

Figure I.14 : Conversion binaire/ décimal/ hexadécimal

2) Représenter un bit sous forme de signal électrique :

La seconde étape de la représentation d'une information en électronique numérique consiste à traduire chaque bit issu de l'étape 1 en signal électrique. Il suffit pour cela d'associer à chacun des deux états du bit une valeur précise d'un signal électrique. Il s'agit également à cette étape de convenir d'un "code" ou d'une convention entre la représentation "logique" (bit) et la représentation "électrique" de l'information. Plusieurs codes sont possibles. Le plus connu est celui de la "logique normale", qui associe un niveau de tension élevé à l'état logique 1 et un niveau de tension faible à l'état logique 0. Les niveaux les plus typiques sont 5V et 0V, mais d'autres valeurs peuvent être utilisées (par exemple 3V/0V ou encore +15V/-15V, etc.) On peut également travailler en "logique inverse", c'est-à-dire associer l'état logique 1 au niveau de tension le plus faible. Cette convention est évidemment moins intuitive. Enfin on peut utiliser d'autres signaux électriques que la tension: on peut par exemple utiliser un courant. Remarque: lors de cette opération, on fixe par convention le fait

que le signal électrique ne peut prendre que deux valeurs. On choisit donc d'interpréter le signal électrique (fondamentalement analogique) comme une valeur... logique. C'est le principe fondateur de l'électronique numérique. C'est également ce principe qui permet la non-dégradation de l'information et confère des avantages majeurs à l'électronique numérique.

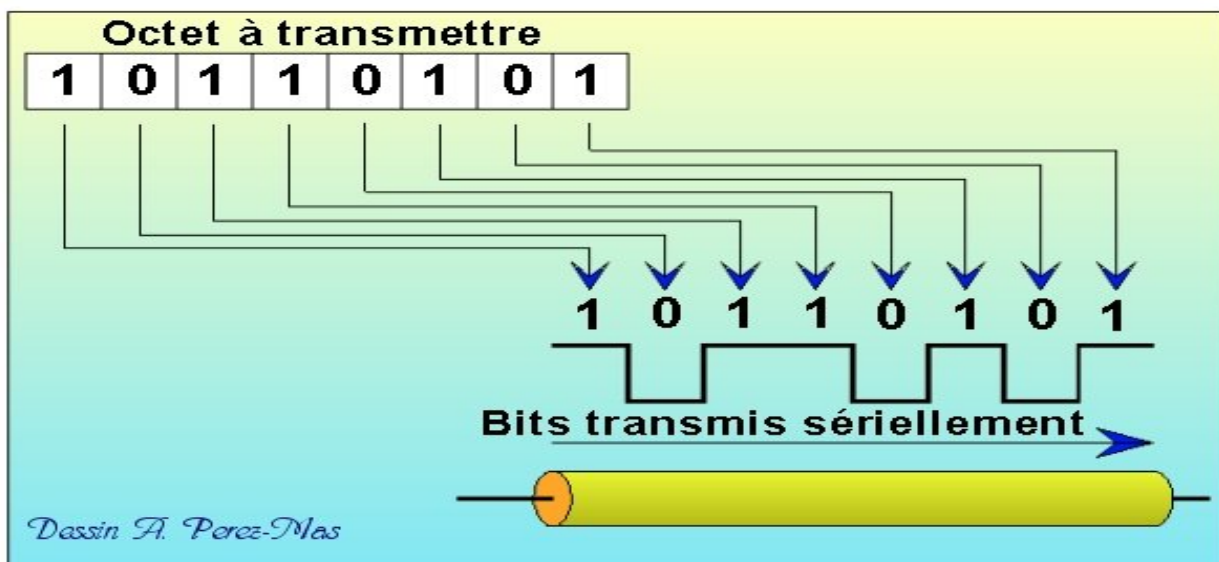


Figure I.15 : Signal binaire transmis en bande de base

I.5 Discussion

On a présenté dans ce chapitre l'électronique numérique et la fonctionnalité de ce domaine dans les systèmes électronique en expliquant d'avantage les systèmes inclus dans ce domaine comme le système binaire et définir l'électronique numérique et son principe fondamentale ainsi que son utilité et son importance dans la plupart des systèmes électronique.

Chapitre II

Systeme à microprocesseur

II.1 Préambule

Ce chapitre illustre l'étude du microprocesseur ainsi que ses caractéristiques et son architecture contenant des mémoires (ROM et RAM) et des portes d'entrée/sortie et sa fonctionnalité en expliquant les différentes étapes des opérations effectuées par le microprocesseur, en évoquant aussi la manière dont il a révolutionné le marché actuelle en facilitant les taches effectuées par l'utilisateur en termes de rapidité d'exécutions et efficacités du travail.

II.2 Définition du microprocesseur

Le « micro-processeur » ou plus simplement « processeur » est le cœur des ordinateurs. Fruit de plus 50 années de recherche et de développement, le processeur détermine les capacités ainsi que les performances des machines numériques actuelles. À l'origine, outils de calcul scientifique, les ordinateurs – et donc les processeurs – sont devenus des produits « grand public » avec l'explosion des jeux vidéo et des applications multimédia. Bien que cela puisse paraître paradoxal, le modèle de fonctionnement n'a pas changé depuis son invention. Un programme est toujours constitué d'instructions qui agissent sur des données. Ces instructions sont assemblées suivant un schéma séquentiel et leurs exécutions se font au travers du processeur qui les exécute les unes après les autres. Le modèle d'exécution fait aussi appel à des unités annexes telles que la mémoire ou les contrôleurs de périphériques, dont le principe de fonctionnement n'a pas beaucoup changé lui aussi. Ce qui a par contre fondamentalement changé est la méthode d'exécution. À l'origine, les processeurs étaient d'une architecture simple et le procédé de fabrication limitait la densité d'intégration.

II.3 Architecture et conception d'un microprocesseur

On entend par un processeur, un dispositif capable d'exécuter un certain nombre d'instructions une par une.

- Il nous faut donc un endroit pour stocker les instructions.
- Il faut pouvoir comprendre et exécuter les instructions.
- Il faut pouvoir stocker des données dans un endroit.

Puisque nous ne travaillons qu'avec des circuits électroniques, chaque instruction peut être codée en tension électrique. Pour simplifier des choses, on prend uniquement deux niveaux de tension. Ainsi chaque instruction est codée sur une combinaison de ces deux niveaux de tension (bit). Par exemple l'opération

mathématique d'addition peut être codée sur "00100110" .Par convention, le niveau haut de tension se présente par un "un" et le niveau bas par un "zéro". Alors, une succession d'instructions n'est qu'une suite des données binaires sur n bits. Ensuite, il faudrait décoder l'instruction courante et l'exécuter, puis passer à l'instruction suivante. Nous allons détailler chaque tâche en essayant d'inventer l'architecture nécessaire pour l'accomplir.

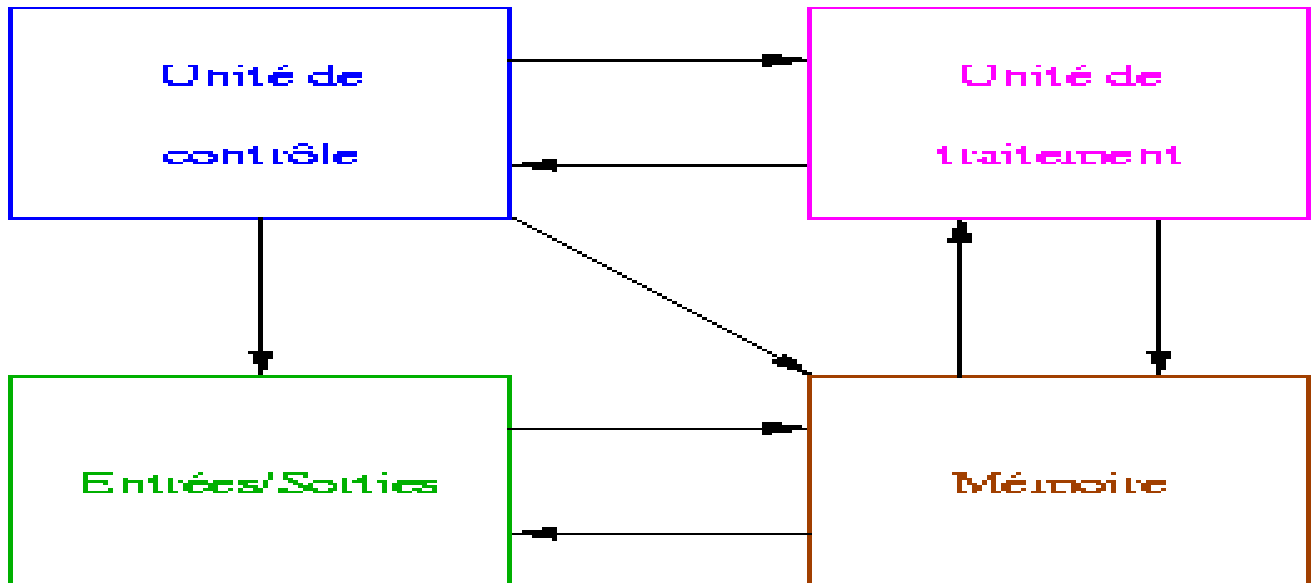


Figure II.1 : architecture d'un microprocesseur

II.3.1 Architecture d'un microprocesseur

Un système à base de microprocesseur est formé des trois éléments :

- Une unité CPU (central processing unit)
- Une mémoire (ROM et RAM)
- Des ports d'entrées/sorties.

Les trois modules sont interconnectés comme le montre la figure suivante autour de trois bus : bus de données, bus d'adresses et bus de contrôles et commandes

Bus : Il s'agit de plusieurs pistes électroniques qui sont reliées au microprocesseur. Ces bus assurent la communication interne et externe du microprocesseur.

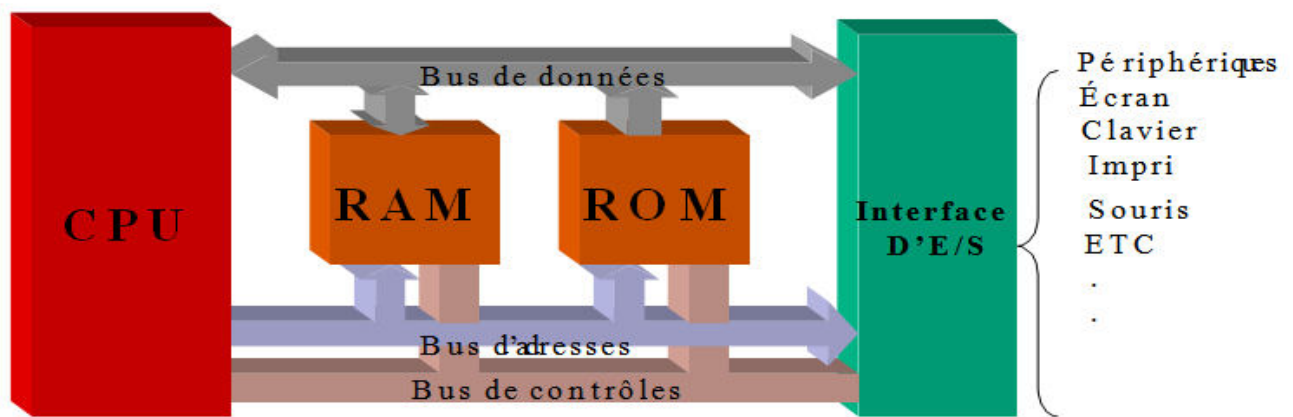


Figure II.2 : Architecture interne d'un microprocesseur

- Le bus de données : c'est un ensemble de fils bidirectionnels qui va permettre le transfert de données entre les différents éléments du système. C'est par ce bus que sont transmises les données qui doivent être traitées par le microprocesseur. A l'inverse, c'est également par ce bus que transitent les résultats en sortie du microprocesseur. Autrement dit, toutes les données entrantes et sortantes du microprocesseur sont véhiculées par le bus de données qui fixe la longueur du mot échangé avec la mémoire.

- Le bus d'adresses : il permet d'adresser un élément par le microprocesseur .il est unidirectionnel .il détermine la capacité maximale d'adressage du système, c'est à dire le nombre maximum de mots de la mémoire associée (ex : 16 bits "adresses" 64 K mots).

- Le bus de commandes et de contrôle : c'est un bus qui permet de véhiculer les signaux de contrôles et de commandes tels que l'horloge les signaux Rd/Wr etc. ... Ce bus sert à coordonner tous les échanges d'informations décrits précédemment. Il véhicule des données qui valident la mémoire et les ports d'entrées) sorties. Il introduit des délais d'attente lorsque des informations sont envoyées à un périphérique qui présente une vitesse de traitement réduite. Le bus de commandes évite les conflits de bus lorsque deux éléments cherchent à communiquer en même temps.

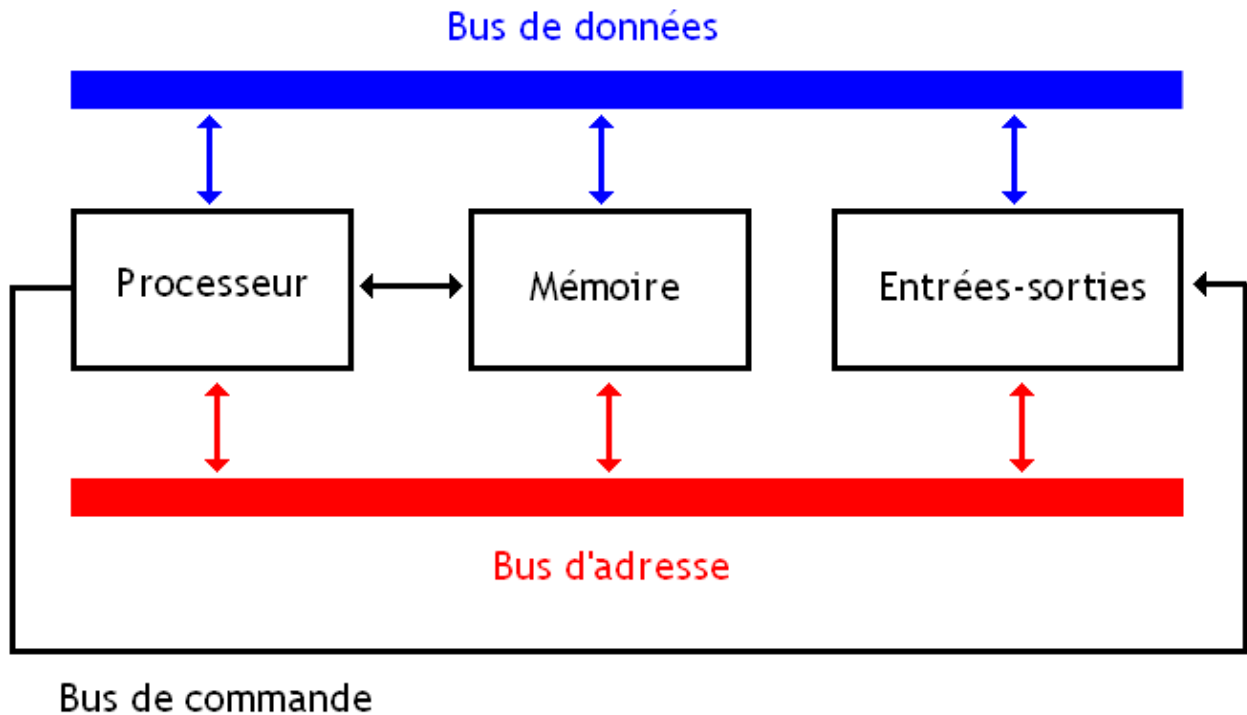


Figure II.3 : Schéma sur la répartition du bus de données et le bus d'adresse

Remarque 1

Dans certains cas, le bus de données et le bus d'adresses sont multiplexés sur un seul bus. Une logique externe doit alors effectuer le démultiplexage.

- La mémoire sert au rangement de deux types d'informations :
- Des données : les informations traitées par le microprocesseur.
- Des instructions : ensemble d'informations codées qui gère l'activité du microprocesseur.

Remarque 2

La mémoire morte (ROM : Read Only Memory) range en général le programme d'initialisation du système (exemple dans le PC elle range le BIOS : Basic Input Output

système).

La mémoire vive (RAM : Random Axes Memory) sert au rangement des programmes utilisateurs c'est une mémoire volatile.

- Les interfaces d'entrées sorties vont permettre au microprocesseur de communiquer avec le monde extérieur, Nous trouvons des ports utilisés exclusivement pour l'entrée, et d'autres ports exclusivement pour la sortie. Il existe aussi des ports bidirectionnels. Donc le microprocesseur peut lire des données à partir d'une interface d'entrée (exemple souris, clavier disque dur, Etc. ...) de même il peut restituer le résultat de son traitement au monde extérieur en adressant des interfaces de sortie (tel que les imprimantes le clavier etc. ...) donc les interfaces d'entrées / sorties vont soulager le microprocesseur pour la communication avec le monde extérieur.

Le microprocesseur doit Donc contrôler les fonctions effectuées par les autres modules, il doit chercher ainsi que décoder des instructions ranger en mémoire, et il doit adresser des interfaces d'entrées/sorties pour lire des données du monde extérieur, et restituer le résultat de son traitement.

II.3.2 Architecture d'un CPU

Une CPU est formée par les trois éléments fonctionnels interconnectés suivants :

- ▶ Registres.
- ▶ UAL : Unité arithmétique et logique.
- ▶ Circuit de contrôle.

Structure interne d'un CPU

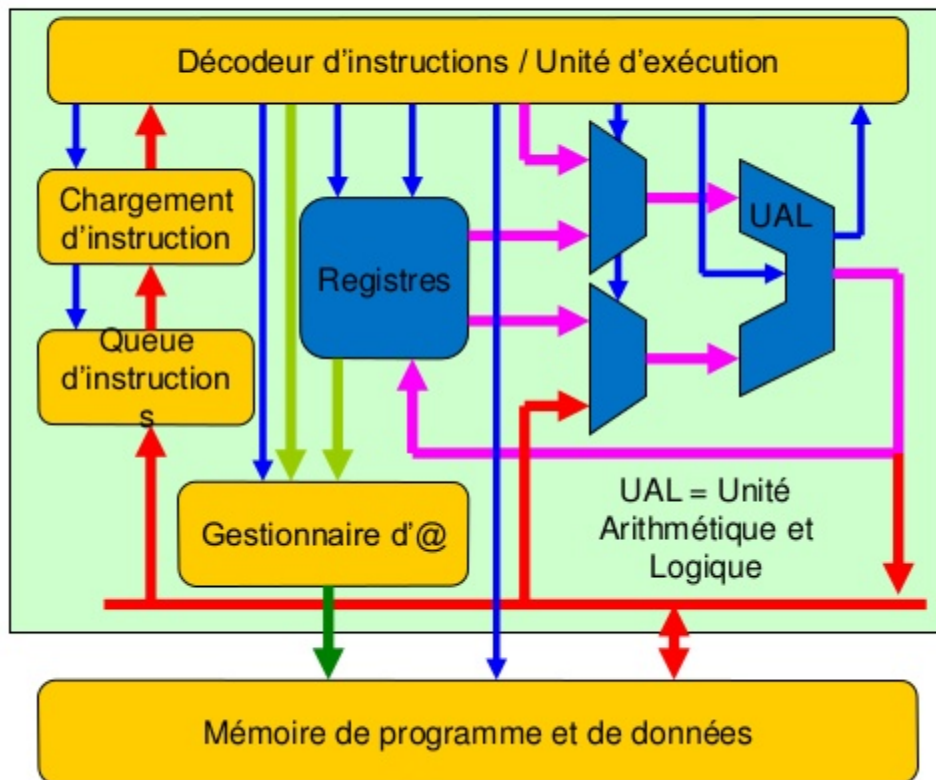


Figure II.4 : Structure interne d'un CPU

II.3.2.1-Les registres

-L'accumulateur

Il s'agit d'un registre d'usage général recevant des opérandes, des résultats intermédiaires ou des résultats provenant de l'unité arithmétique et logique. Ils évitent des appels fréquents à la mémoire, réduisant ainsi les temps de calcul. Donc la plupart des opérations arithmétiques et logiques se font dans l'accumulateur.

- Le compteur de programme

Le compteur de programme contient l'adresse (décalage ou offset) de l'instruction suivante en mémoire qui doit être exécutée. Autrement dit, il doit indiquer au processeur la prochaine instruction à exécuter. Le registre compteur de programme est constamment modifié après l'exécution de chaque instruction afin qu'il pointe sur l'instruction suivante. Les microprocesseurs de la famille x86 dépendent entièrement du registre compteur de programme pour connaître l'instruction suivante.

- Registre d'instruction et décodeur d'instruction

➤ Le registre d'instruction

Chaque opération que le microprocesseur va effectuer est codée (c'est-à-dire pour chaque instruction on assigne un code qui ne peut pas être modifié ni changé par un autre code) appelé « instruction code » ou « opération code », pour exécuter une instruction le microprocesseur transmet l'adresse se trouvant dans le registre compteur de programme à la mémoire, la mémoire retourne au microprocesseur l'octet adresse par ce dernier (le code de l'instruction) celui-ci sera stocker dans un registre appelé registre d'instructions (RI), donc Le registre d'instructions contient la prochaine instruction à être exécutée par le processeur. Cette instruction sera acheminée (par un bus de données) au décodeur d'instructions qui sera chargé de l'interpréter.

➤ Le décodeur d'instruction

C'est lui qui va interpréter l'instruction contenue dans le registre d'instruction (RI). C'est-à-dire qu'elle est l'opération à effectuer (Addition, branchement etc...) Et comment aller chercher les opérandes requises pour cette opération (par exemple, les nombres à additionner). Le décodeur d'instructions communique alors avec l'unité de commandes et de contrôles qui pourra déclencher

les événements en conséquence. Par exemple, si le décodeur d'un 8086 reçoit l'octet 10000001 (le "Add mem/reg data"), il sait que le processeur doit aller chercher 3 autres octets en mémoire vive pour compléter l'instruction.

-Registres d'adresses

Ces registres servent à gérer l'adressage de la mémoire. En effet le processeur peut utiliser un registre ou une paire de registres pour accéder à un emplacement mémoire, et puisque les registres peuvent être incrémenter ou décrémenter donc on peut accéder facilement à des données qui se trouvent en mémoire d'une manière adjacente (tel que les tableaux).

- Registre d'état

Le registre d'état sert à contenir l'état de certaines opérations effectuées par le processeur. Par exemple, quand le résultat d'une opération est trop grand pour être contenu dans le registre cible (celui qui doit contenir le résultat de l'opération), un bit spécifique du registre d'état (le bit OF) est mis à 1 pour indiquer le débordement.

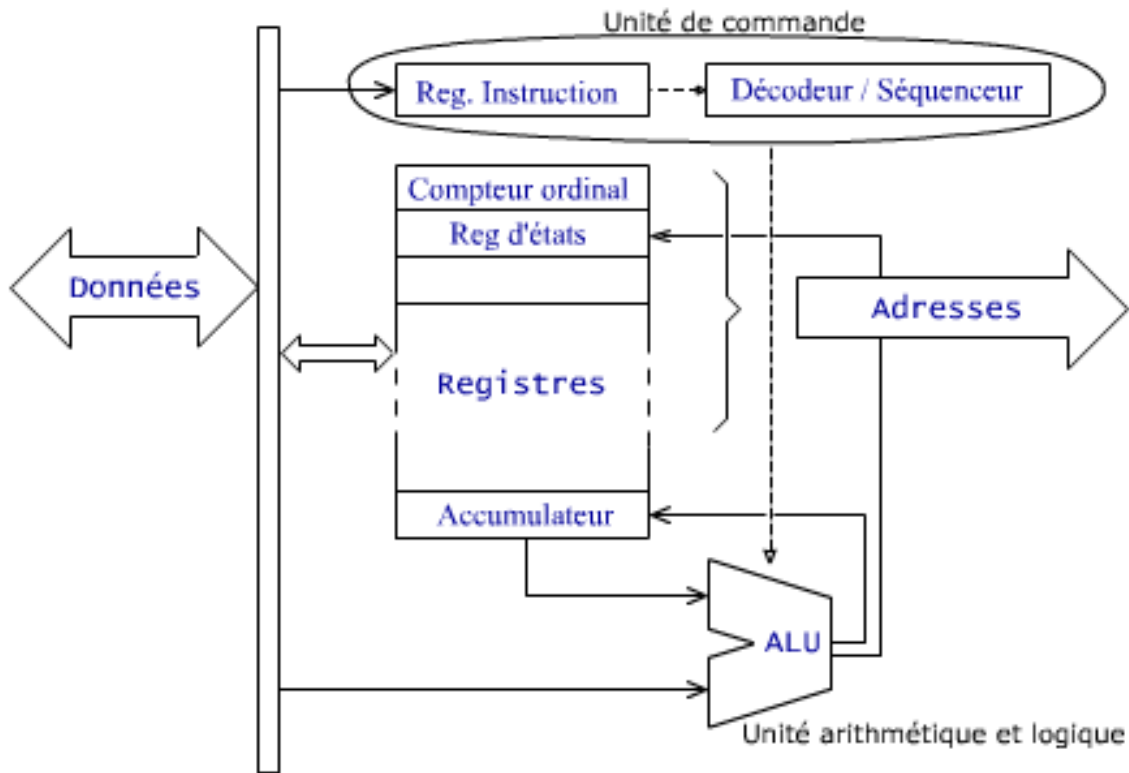


Figure II.5 : Représentation des registres dans un microprocesseur

II.3.2.2 Unité arithmétique et logique

Comme son nom l'indique, cette unité peut exécuter deux types d'opérations.

- Opérations arithmétiques :

Elles incluent l'addition et la soustraction qui sont des opérations de base (une soustraction est une addition avec le complément à deux), la multiplication et la division. Les données traitées sont considérées dans des représentations entières.

- Opérations logiques :

Ces opérations sont effectuées bit à bit sur les bits de même poids de deux mots,

tel que ET, OU, NOT OU EXCLUSIF, de même les opérations de rotation et de décalage (arithmétique et logique)

Elle reçoit ses opérandes (les octets qu'elle manipule) du bus de données. Celles-ci peuvent provenir de registres ou de la mémoire. A la fin d'une opération, l'UAL peut aller modifier certains bits du registre d'état.

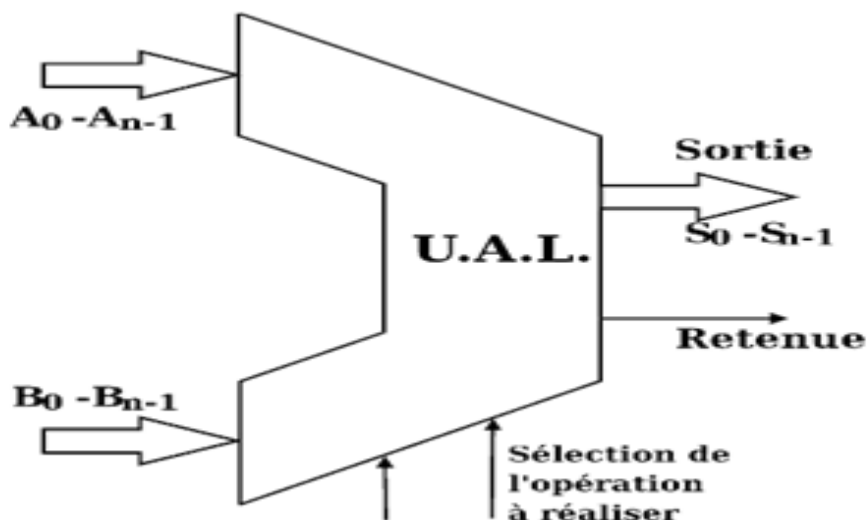


Figure II.6 : Instruction informatique dans U.A.L

II.3.2.3 Unité de contrôle et de commande

Synchronisée par le signal de l'horloge, c'est elle qui déclenche les événements dans le processeur (on peut remarquer à ce sujet qu'elle est connectée à toutes les autres composantes du processeur). Par exemple, quand une information passe dans un bus, cette information est destinée à un seul endroit (par exemple, un registre). C'est donc l'unité de commande et de contrôle qui va "déverrouiller" l'entrée de cette destination pour que l'information qui circule sur le bus puisse y entrer (et ne pas entrer ailleurs en même temps).

Il s'agit donc essentiellement d'un automate exécutant les différentes séquences

propres à chaque instruction. Cet automate peut être réalisé de plusieurs façons (câbler ou micro-programmer et dans les deux cas le jeu d'instructions est fixe). La plupart des unités de traitement sont micro-programmées et donc à jeux d'instructions fixes.

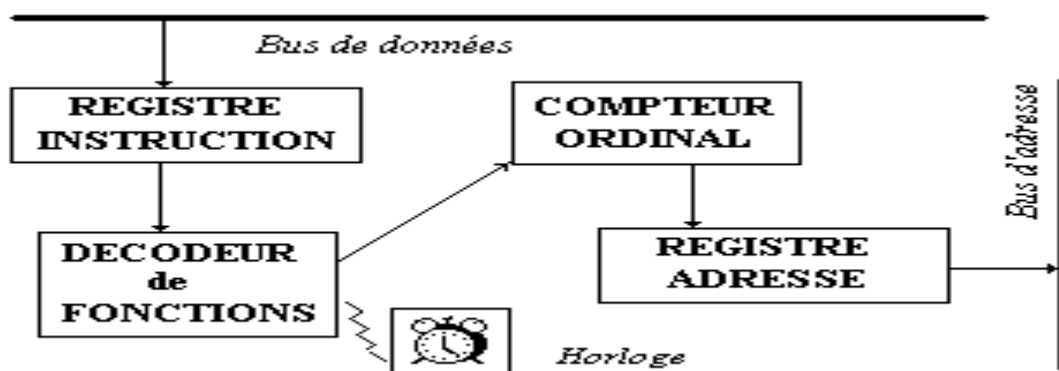


Figure II.7 : exemple sur l'unité de commande sur ordinateur

II.4 Fonctionnement d'un système à base de microprocesseur

Un microprocesseur contrôle toutes les fonctions de la CPU ou l'unité centrale de traitement, d'un ordinateur ou autre appareil numérique. Les fonctions du microprocesseur comme un cerveau artificiel. La fonction entière de la CPU est commandée par un circuit intégré unique. Il comporte dans son fonctionnement :

II.4.1 Les interruptions

Les interruptions permettent au matériel (périphérique) de communiquer avec le processeur. Dans certains cas, on désire que le processeur réagisse

rapidement à un évènement extérieur : arrivée d'un paquet de données sur une connexion réseau, frappe d'un caractère au clavier, modification de l'heure. Les interruptions sont surtout utilisées pour la gestion des périphériques de l'ordinateur, en effet les systèmes à base de microprocesseurs peuvent comporter plusieurs éléments matériels tels que l'écran, les lecteurs de CD, lecteurs de DVD, les ADC (Analog to digital converter) et DAC (digital to analog converter) etc...Mais la majorité de ces périphériques n'ont besoin du microprocesseur qu'à certains moments. Si un périphérique nécessite une intervention, il génère lui-même une demande d'interruption.

Une interruption est signalée au processeur par un signal électrique sur une borne spéciale. Lors de la réception de ce signal, le processeur (traite) l'interruption dès la fin de l'instruction qu'il était en train d'exécuter. Le traitement de l'interruption consiste soit :

- à l'ignorer et passer normalement à l'instruction suivante : c'est possible uniquement pour certaines interruptions, nommées interruptions masquables. Il est en effet parfois nécessaire de pouvoir ignorer les interruptions pendant un certain temps, pour effectuer des traitements très urgents par exemple. Lorsque le traitement est terminé, le processeur démasque les interruptions et les prend alors en compte.
- à exécuter un traitant d'interruption (interrupt handler). Un traitant d'interruption est un programme qui est appelé automatiquement lorsqu'une interruption survient. L'adresse de début du traitant est donnée par la table des vecteurs d'interruptions.

Remarque :

Parfois le microprocesseur est sollicité par plusieurs interruptions en même temps, pour répondre à ces appels un ordre de priorité est souvent pris en compte pour

leurs traitements.

Les interruptions augmentent considérablement l'efficacité du processeur.

Les interruptions sont de deux types :

- * Interruption matérielle.
- * Interruption logicielle.

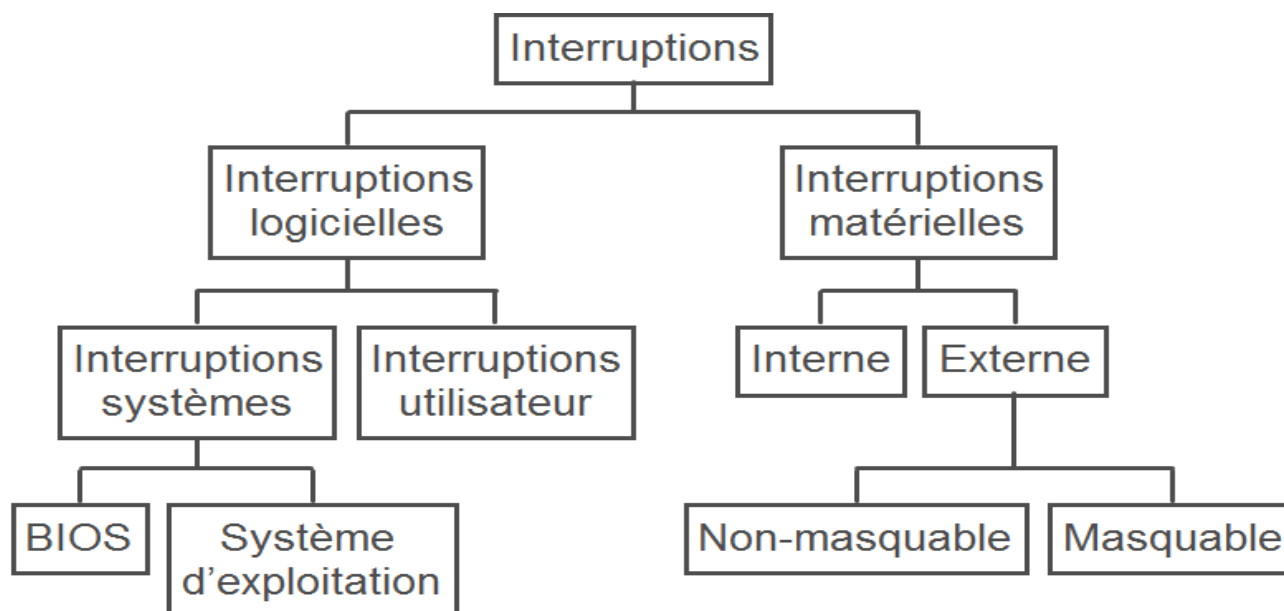


Figure II.8 : Schéma explicatif sur les interruptions

II.4.2 L'écriture en mémoire (WRITE)

Pour écrire une donnée dans la mémoire le microprocesseur doit placer l'adresse de la donnée sur le bus d'adresses (son emplacement dans la mémoire) puis il place la donnée sur le bus de données et enfin génère le signal WRITE (ordre d'écriture dans la mémoire).

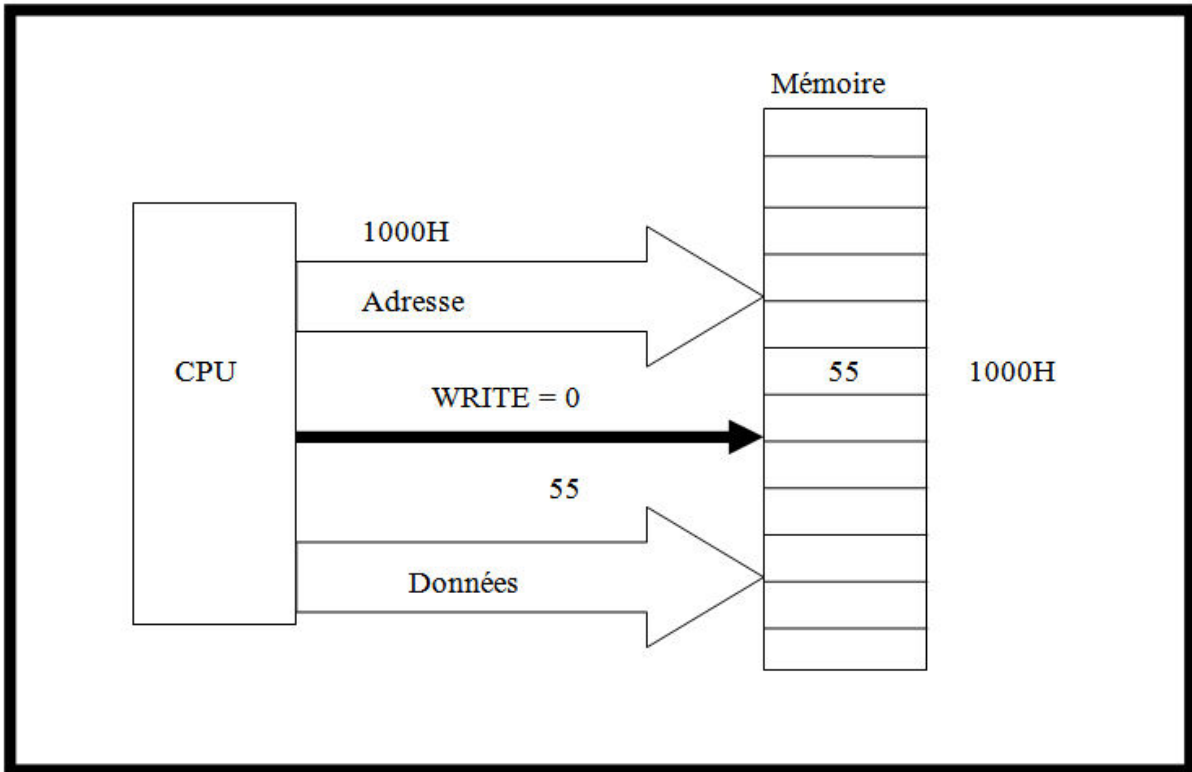


Figure II.9 : L'écriture en mémoire

II.4.3 La lecture de la mémoire (READ)

Pour lire une donnée de la mémoire le microprocesseur doit connaître son emplacement, en effet il dépose son adresse sur le bus d'adresses puis génère le signal READ (il demande une opération de lecture de la mémoire) alors la donnée sera acheminée vers le microprocesseur à travers le bus de données. La donnée sera stockée dans un registre dans le microprocesseur.

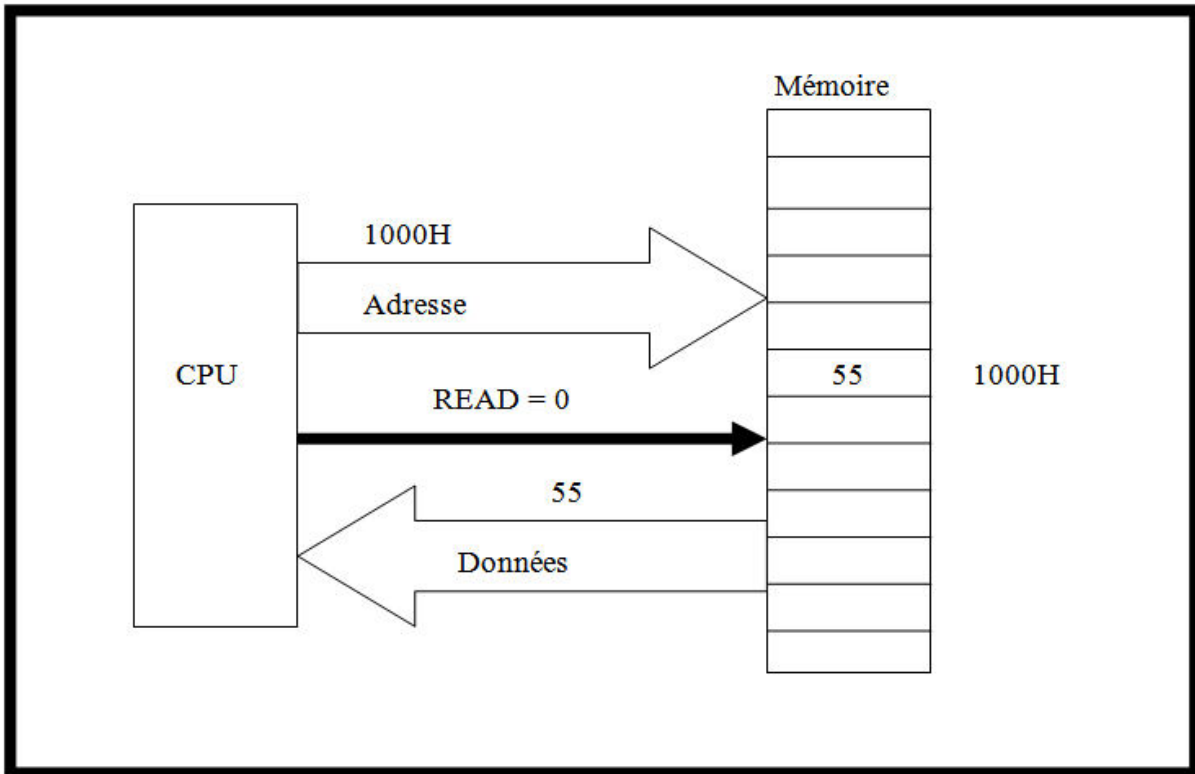


Figure II.10 : La lecture en mémoire

Remarque

Si la donnée est un code opératoire d'une instruction alors elle sera logée dans le registre d'instructions sinon elle sera logée dans un registre de données (en général l'accumulateur).

II.4.4 Wait Stat (temps d'attente pour la synchronisation)

Le Wait Stat n'est qu'un temps d'attente qu'il faut donner à certains périphériques pour terminer leurs opérations (Notamment la mémoire), en effet en général dans la plupart des systèmes à base de microprocesseur on trouve toujours des circuits intermédiaires entre la CPU et la mémoire comme le montre la figure suivante :

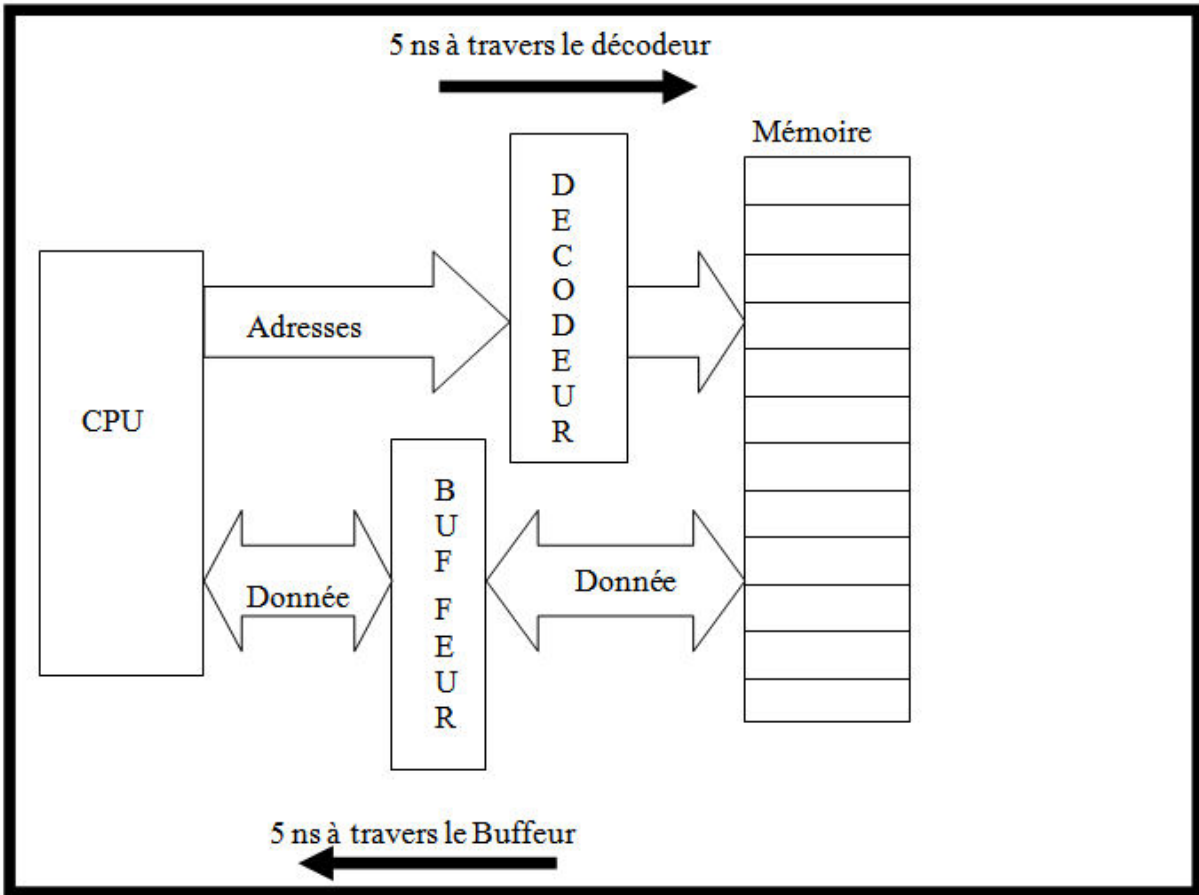


Figure II.11 :Circuits intermédiaires entre la CPU et la mémoire

Ces circuits entraînent toujours des temps de retard supplémentaires ce qui oblige le microprocesseur à utiliser des Wait Stat (temps de retard) pour synchroniser les transferts entre la mémoire et la CPU .la figure précédente montre que le microprocesseur perd environ 10 ns pour avoir l'information à cause du décodage et la bufférisations du bus.

Remarque

Parfois le microprocesseur utilise plus qu'un seul Wait Stat pour synchroniser avec ces périphériques (tout dépend du temps que le périphérique demande pour achever son fonctionnement)

II.4.4 Communication avec les entrées/sorties

Il peut s'agir d'un flux d'informations de l'extérieur vers l'ordinateur (Acquisition via le clavier, une connexion réseau, un disque dur, etc...), ou d'un flux de l'ordinateur vers l'extérieur (écran, réseau, disque, etc...).

Les données échangées entre un périphérique et le processeur transitent par l'interface (ou contrôleur) associé à ce périphérique. L'interface possède de la mémoire tampon pour stocker les données échangées (suivant le type d'interface, cette mémoire tampon fait de 1 seul octet à quelques Méga-octets). L'interface stocke aussi des informations pour gérer la communication avec le périphérique :

- des informations de commande, pour définir le mode de fonctionnement de l'interface: sens de transfert (entrée ou sortie), mode de transfert des données (par scrutation ou interruption), etc. Ces informations de commandes sont communiquées à l'interface lors de la phase d'initialisation de celle-ci, avant le début du transfert.

- des informations *d'état*, qui mémorisent la manière dont le transfert c'est effectué (erreur de transmission, réception d'informations, etc.). Ces informations sont destinées au processeur.

Lors de l'exécution des instructions d'entrées/sorties, le processeur met à 1 sa borne IO/M et présente l'adresse E/S sur le bus d'adresse. Le signal IO/M indique aux circuits de décodage d'adresses qu'il ne s'agit pas d'une adresse en mémoire principale, mais de l'adresse d'une interface d'entrées/sorties.

Remarque

La communication entre le microprocesseur et les interfaces d'entrées/sorties peut être série (sur un seul fil bit par bit) ou parallèle (sur plusieurs fils).

II.4.5 Accès direct à la mémoire (DMA)

Lorsqu'un transfert en mémoire est nécessaire de la mémoire RAM à un port d'E/S, la CPU lit le premier octet en mémoire et le charge dans l'un des registres du microprocesseur. La CPU écrit ensuite l'octet rangé précédemment sur le port d'E/S approprié.

Il en résulte que le microprocesseur effectue des opérations de lecture et d'écriture répétées. Ainsi un certain temps est perdu entre le traitement de chaque octet. Pour remédier à ce problème, une procédure est mise au point pour l'accès direct à la mémoire (Direct Memory Access), qui permet de transférer des données de la mémoire RAM au port d'E/S sans passer par le microprocesseur. Pour cela, un contrôleur DMA, qui reprend le rôle de la CPU, c'est à dire qu'il gère les transferts de la RAM aux ports d'E/S.

II.4.6 Discussion

On a présenté dans ce chapitre le microprocesseur et expliqué sa fonctionnalité et ses caractéristiques, où on a réalisé l'importance du microprocesseur dans divers calculs scientifique, et traité son contenu et les étapes de procédures.

Le microprocesseur est effectivement le cœur des ordinateurs qui explique son importance où maintenant est devenu un produit « grand publique ».

Chapitre III

Electronique et

informatique

embarquées

III.1 Préambule

A l'heure actuelle, l'informatique embarqués joue un rôle important dans notre vie de tous les jours, elle est présente dans plusieurs domaines, tel que l'astronomie, le médical, ou la télécommunication...

Ce chapitre présente tout d'abord la notion de l'informatique embarquée ainsi que ses caractéristiques. Dans ce contexte nous discuterons aussi de ses domaines d'applications puis des contraintes relatives à ce type de systèmes.

III.2 Définition de l'informatique embarquée

L'informatique embarquée est un système autonome, souvent temps réel, spécialisé dans une tâche bien précise. Ses ressources sont généralement limitées, ce système comprend une partie matérielle et une partie logicielle qui sont conçues spécifiquement pour réaliser une fonction dédiée. Il contient généralement un ou plusieurs microprocesseurs destinés à exécuter un ensemble de programmes définis lors de la conception et stockés dans des mémoires.

III.3 Architecture d'un système embarqué

L'architecture d'un système embarqué se définit par le schéma suivant :

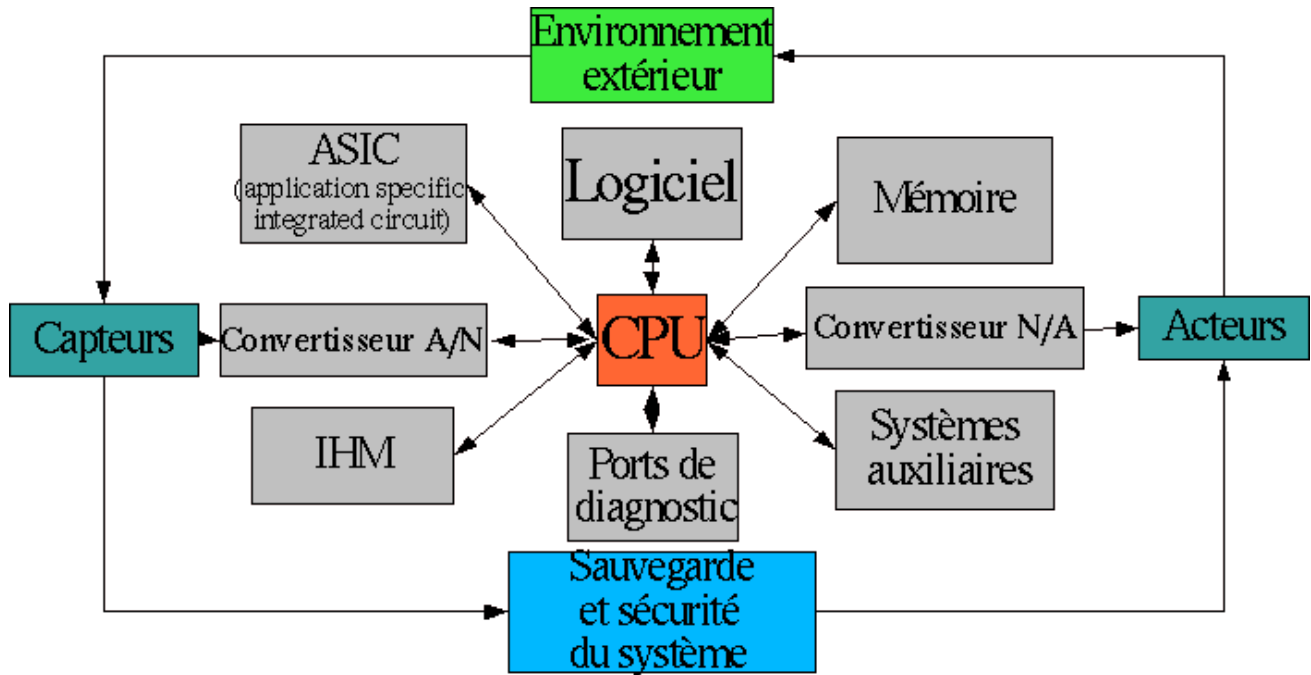


Figure III.1 : Architecture d'un système embarqué

- Réalisé autour d'un microcontrôleur (uC)
- uC est un système à processeur dans un seul chip (Soc)
- comprend: processeur, mémoire, GPIO (entrées/sorties simples configurables), contrôleur de bus, contrôleur d'interruption, contrôleur d'écran, USB, Ethernet, ...
- très bon rapport performance/prix et performance/consom
- Ensemble compact (volume optimisé)
- Démarrage autonome du système (boot)
- pas de disque dur, utilisation de mémoire flash, ...
- Généralement pas d'extension possible

III.4 Caractéristiques des systèmes embarqués

Les systèmes embarqués ont pour but de permettre aux objets usuels de réagir à l'environnement, ils peuvent aussi apporter une interface avec l'utilisateur. Généralement ils sont caractérisés par :

- Coût réduit, maximisation rapport performance/prix.
- Capacité mémoire adaptée.
- Volume restreint (compact, pas modulaire).
- Exécution temps réel (souvent).
- Fiabilité et sécurité de fonctionnement.
- Consommation d'énergie maîtrisée (très faible en cas d'utilisation sur batterie)
- Capacité de calcul approprié é à l'application.

La structure de base de ces systèmes est donnée par la figure 1 : l'environnement est mesuré par divers capteurs. L'information des capteurs est échantillonnée pour être traité par le cœur du système embarqué. Puis le résultat du traitement est converti en signaux analogiques qui génèrent les actions sur l'environnement (afficheur d'informations pour l'utilisateur, actionneurs, transmission d'informations ... etc.).

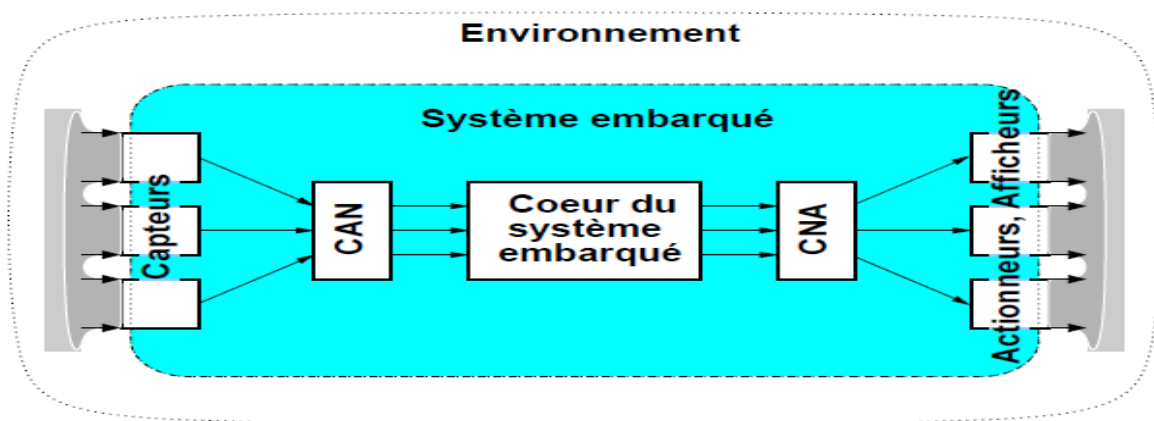


Figure III.2 : Un système embarqué dans son environnement

III.5 Contraintes de conception des systèmes embarqués

L'appellation système embarqué regroupe de nombreux systèmes clairement distincts, tant au niveau des applications qu'au niveau des performances requises. De ce fait, il est difficile de définir des contraintes universelles. Néanmoins, il est tout de même possible de lister un certains nombres de contraintes que l'on doit couramment respecter dans le cadre de la conception des systèmes embarqués, parmi lesquels nous trouvons :

- Le coût est, dans la plupart des cas, la plus importante des contraintes à prendre en compte lors de développement. Cette contrainte est particulièrement forte dans le cas d'application grand public.
- La limitation des ressources de traitement est liée à une limitation de la surface disponible pour le système et pour les composants qui le constitue. Il s'agit aussi d'une limitation des ressources de mémorisation.
- Le débit de données en entrée et en sortie et aussi limité. Néanmoins, la tendance actuelle des applications va vers une augmentation des besoins en termes de vitesse et de quantité des échanges de données, donc vers une augmentation forte des débits.
- Le nombre de connexion du système vers l'extérieur est limité.
- La limitation en termes de consommation de puissances et d'énergie est souvent importante, car un système embarqué n'est qu'une partie d'un système hôte qui peut être lui même limité en source d'énergie. C'est bien entendu le cas de tout système alimenté par une batterie.
- Les besoins de flexibilité logicielle et matérielle souvent grands. D'une part, la flexibilité permet une meilleure intégration du système dans un large domaine d'application. D'autre par, il permet au système d'évoluer dans le temps par des mises à jour logicielle, matérielle grâce à l'utilisation de mémoires programmables et de circuits matériels reconfigurables.

III.6 Domaines d'application des systèmes embarqués

Les systèmes embarqués sont désormais utilisés dans des applications diverses tels que le transport (avionique, espace, automobile, ferroviaire), dans les appareils électriques et électroniques (appareils photo, jouets, postes de télévision, électroménager, systèmes audio, téléphones portables), dans la distribution d'énergie, dans l'automatisation, etc.

Exemples de systèmes embarqués

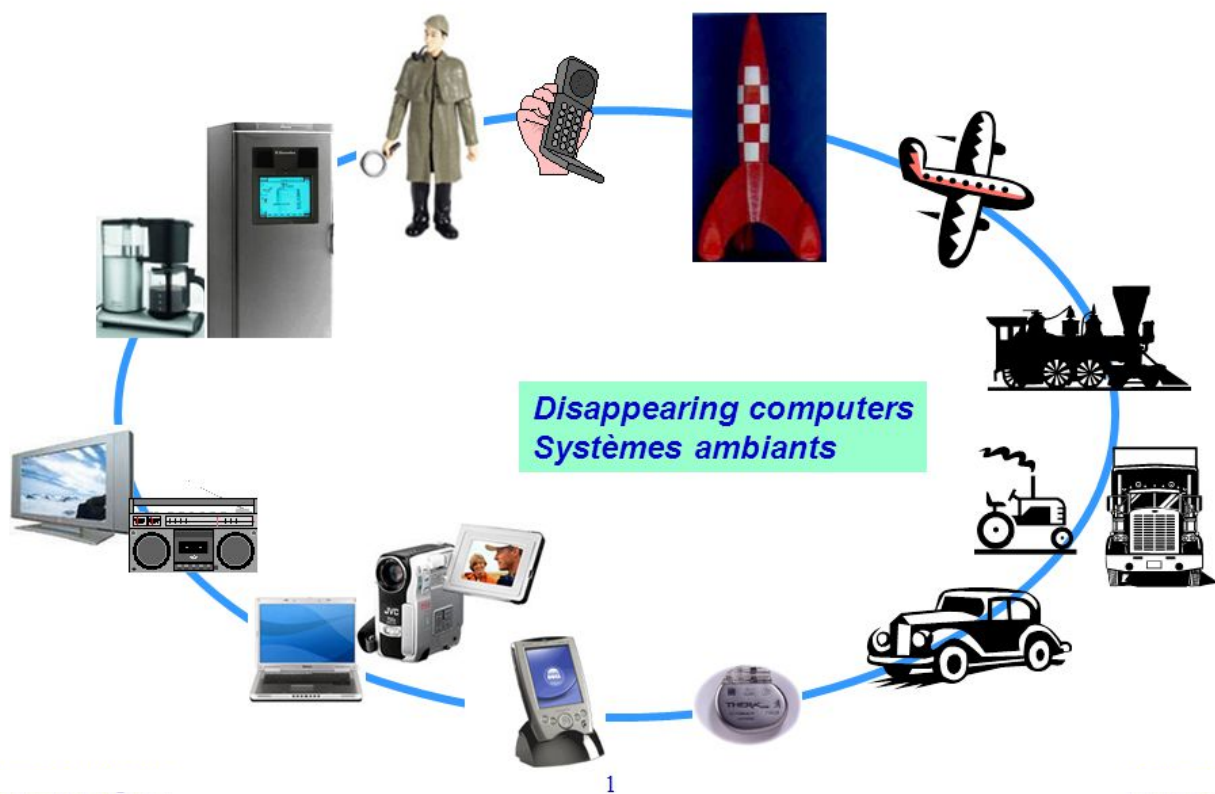


Figure III.3 :exemples de systèmes embarqués

Les systèmes embarqués sont définis comme des ensembles matériels logiciels intégrés dans le but d'effectuer des tâches fonctionnelles précises. Les systèmes et logiciels embarqués jouent un rôle majeur dans les quasi-intégralités des secteurs industriels et sont très présents au sein d'industries historiques : les transports terrestres et l'aéronautique, le nucléaire, la défense et l'espace, les télécommunications (téléphones portables, assistants personnels, application internes chez les opérateurs, ils jouent et joueront un rôle de plus important dans de nombreux autres secteurs comme :

- La production, la distribution et la gestion de l'utilisation de l'énergie.
- La production industrielle (automatique, systèmes discrets et continus).
- L'instrumentation médicale.
- Le bâtiment (domotique).
- L'électronique grand public (terminaux mobiles, multimédia, jeux et loisirs numériques).
- La logistique (commerce et distribution).
- Les infrastructures urbaines (eau, trafic, captation de la qualité de l'aire).
- La sécurité (vidéosurveillance, moyens d'identification)
- Les transactions bancaires et commerciales (terminaux de paiement, cartes à puce).

Domaines d'applications

- Industrie
 - Transports
 - Militaire
 - Robotique
- Grand public
 - Consoles de jeu portables
 - PDA
 - Téléphonie
 - Vers le multimédia



Figure III.4 : multiples domaines d'applications des systèmes embarqués

III.7 Discussion

Dans ce chapitre, nous avons présenté brièvement l'informatique embarquée, qui fait plus que jamais partie de notre vie quotidienne, elle nécessite une fiabilité irréprochable.

En effet, l'informatique se trouve embarquée dans de très nombreux objets et parmi ces objets, on trouve les téléphones mobiles.

Conclusion générale

Le travail réalisé dans le cadre de ce mémoire est l'implémentation de systèmes à microprocesseur dans l'informatique embarquée.

Dans le premier chapitre, une étude du domaine de l'électronique numérique qui regroupe les systèmes électroniques fonctionnant sur la base d'états électrique.

Nous avons présenté dans le deuxième chapitre les systèmes à microprocesseur qui est le cœur battant des ordinateurs, ses performances et son efficacité la laisse incontournable lors de l'étude sur l'informatique embarquée.

Dans le troisième chapitre, une présentation générale de l'électronique et l'informatique embarquées qu'on trouve dans plusieurs domaines, entres autres, l'astronomie, la télécommunication, le médical, les satellites...

Comme perspectives, nous recommandons de faire une étude des microprocesseurs en utilisant la programmation en langage C et l'implémentation des algorithmes permettant de perfectionner les applications des systèmes embarqués.

Bibliographie

- [1] Francine krief : « les systèmes embarqués communicant mobilité, sécurité, autonomie », Paris, 2008
- [2] Tadrict Nadia, diplôme d'ingénieur d'état : « implémentation d'un DPS TMS320LF240A », université de tizi ouzou, 2006
- [3] Arab Naima, diplôme d'ingénieur d'état : « implémentation d'un DSP TMS320LF2407A en vue d'autopilote une machine synchrone a aimants permanents », université de tizi ouzou, 2007
- [4] Chibane Mourad, diplôme d'études universitaire : « maintenance des téléphones portables », universite de tizi ouzou, 2008
- [5] Belkhiri Ahmed, diplôme d'ingénieur d'état : « étude et application d'un système à base du microprocesseur ARM », universite de batna, 2003
- [6] http://fabrice.sincere.pagesperso-orange.fr/cm_electronique/cours%20electronique/cours%20electronique%20numerique.pdf ,
- [7] https://www.researchgate.net/publication/323904482_systemes_a_microprocesseur,
- [8] <https://www.technologuepro.com/cours-systemes-embarques/cours-systemes-embarques-introduction.htm>,
- [9] <https://homepages.laas.fr/fcaignet/Cours/combinatoire.pdf>