

CHAPITRE

7

LE PROCESSEUR

I- Introduction

II- Structure de l'unité centrale de traitement

III- Principe de fonctionnement d'un processeur

IV- Les critères de performance d'un microprocesseur

V- Exemples de microprocesseurs

I- Introduction:

Un processeur est un circuit électronique qui peut compter des millions de transistors.

L'unité centrale de traitement (UCT ou CPU : Central Processing Unit), encore dénommée processeur ou microprocesseur, est l'élément de l'ordinateur qui interprète et exécute les instructions d'un programme.

Les instructions à exécuter par le microprocesseur doivent être tout d'abord transférées de la mémoire centrale pour être logées dans l'UCT.

Une instruction est composée de deux parties : l'**action** et les **opérandes**.

- L'action est l'opération à exécuter telle que l'addition, la soustraction...
- Les opérandes sont les données sur lesquelles l'action va opérer.

Le **format d'une instruction** est sa spécification pour un microprocesseur donné sous forme de:

Action	Opérandes
--------	-----------

Exemple :

Un microprocesseur dispose d'une instruction d'addition dont le format est le suivant :

Add opérande1 opérande2

Et d'une instruction d'affichage dont le format est le suivant :

Aff opérande

L'ensemble des actions que peut exécuter le microprocesseur avec leurs formats spécifiques est appelé **jeu d'instruction** de l'UCT. Le jeu d'instruction diffère d'un microprocesseur à un autre : On peut trouver des actions exécutées par un microprocesseur et pas par un autre ou possédant un format différent.

Le microprocesseur doit connaître l'action demandée par l'instruction afin de préparer tout le nécessaire pour qu'elle soit bien exécutée.

II- Structure de l'unité centrale de traitement :

Le processeur se compose essentiellement des parties suivantes :

- L'unité de commande
- L'unité arithmétique et logique
- Les registres
- Et éventuellement une mémoire cache

1. L'unité de commande (UC) :

L'unité de commande dirige le fonctionnement de tous les autres éléments de l'unité centrale en leur envoyant des signaux de commande.

Elle gère le séquençement des étapes de l'exécution d'une instruction et elle active les lignes de l'opération à exécuter.

Pour que l'instruction soit interprétable par le microprocesseur elle doit être codée. On appelle **codage d'une instruction** la transformation d'une instruction d'un programme du format texte vers un format binaire qui respecte le format d'une instruction machine (action + opérandes). Cette opération est effectuée par le compilateur¹.

Ensuite, L'unité de commande doit décoder l'instruction, c'est à dire connaître de quelle opération s'agit il.

Exemple : S'il s'agit d'une opération d'addition, l'unité arithmétique et logique sera chargée par les opérande de l'opération et l'unité de commande va activer les lignes responsables de l'exécution de l'action d'addition.

¹ Le compilateur transforme un fichier écrit en un langage de programmation évolué en un fichier exécutable en langage machine. Sous DOS, les fichiers exécutables portent l'extension .EXE

Le travail de l'unité de commande est réalisé à l'aide de plusieurs composants tels que le décodeur, le séquenceur, l'horloge et différents registres tels que le compteur ordinal et le registre d'instruction que nous détaillerons ultérieurement dans ce chapitre au dans les sections circuits de l'UCT et registres de l'UCT.

2. L'unité arithmétique et logique (UAL) :

A l'origine, cette unité est chargée de l'exécution des opérations booléennes et des opérations arithmétiques (addition, soustraction, multiplication, division, comparaison, etc.) pour des entiers. En parallèle à cette unité, on peut trouver une unité de calcul sur les réels ainsi qu'une unité de traitement dédiée aux opérations multimédia (traitement des images et du son).

L'UAL est alimentées par les registres généraux et le registre d'état. Le résultat d'une opération est transmis au registre accumulateur.

L'UAL présente se présente comme suite :

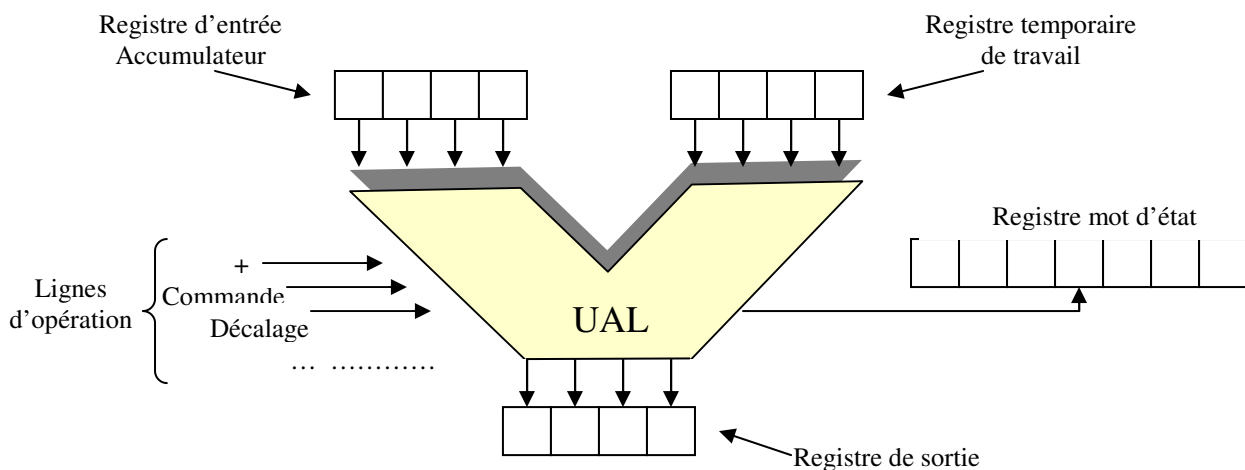


Figure 12 : Schéma de l'unité arithmétique et logique

3. Les registres:

Les registres sont des mémoires locales très rapides qui permettent de stocker des résultats temporaires ou des informations de commande.

L'unité centrale comprend un certain nombre de registres pour stocker des données à traiter, des résultats intermédiaires ou des informations de commande. Les registres ont une taille réduite qui peut être de l'ordre de 1,2,4 ou 8 octets.

Chaque registre est dédié à un traitement particulier. Parmi ceux-ci on distingue les registres suivants :

- Le registre instruction (RI) :

Il contient l'instruction à exécuter. Sa taille dépend du format des instructions machines. Le programmeur n'a pas accès au registre instruction. Ce registre est accessible uniquement par le microprocesseur.

- Le compteur ordinal (CO) ou encore Program Counter (PC):

Il contient l'adresse de la prochaine instruction à exécuter. Il est souvent noté par le symbole $C\emptyset$.

Après chaque exécution d'une instruction, il est automatiquement incrémenté du nombre de mots correspondant à la longueur de l'instruction traitée si le programme est exécuté en séquence. En cas de rupture de séquence (branchement conditionnel ²ou non, appel à une routine³,...), il est chargé avec la nouvelle adresse.

La taille du compteur ordinal dépend de l'espace adressable. Ce compteur n'est généralement pas accessible directement au programmeur.

- Le registre d'état (PSW : Processor Status Word) :

Il contient des informations sur l'état du système (retenue, dépassement, etc.). Une partie des bits de ce registre, aussi appelé registre condition, constitue des drapeaux (flags) qui indiquent certains états particuliers.

Par exemple à la fin de chaque opération on peut y trouver le signe du résultat (Négatif, Zéro ou Positif), ainsi qu'une éventuelle retenue (Carry) ou un dépassement de capacité (Overflow).

Ces bits indicateurs peuvent être testés pour déterminer la suite du déroulement du programme lors des branchements conditionnels. Ces bits sont accessibles au programmeur.

Registre mot d'état :

CF	BF	PF	ZF	SF	OF	IF
----	----	----	----	----	----	----

² Le branchement conditionnel consiste à exécuter, si une certaine condition est vérifiée, une portion du programme qui n'est située forcément à la suite de l'instruction courante.

³ L'appel à une routine ou traitement itératif est un traitement qui se répète et par la suite une fois le bloc d'instruction est exécuté une fois, on retourne à l'instruction du début du bloc.

CF	BF	PF	ZF	SF	OF	IF
Carry Flag	Borrow Flag	Parity Flag	Zero Flag	Sign Flag	Overflow flag	Interrupt Flag
Exprime une retenue	Exprime un emprunt lors de la soustraction	Exprime la parité	Vérification de l'annulation du résultat d'une opération.	Exprime le signe si négatif ou positif	Indique le dépassement de capacité	Autorisation ou non des interruptions

- L'accumulateur (ACC) :

C'est un registre de l'unité arithmétique et logique. Il a de nombreuses fonctions. Il peut contenir un des deux opérandes avant l'exécution et recevoir le résultat après. Il peut servir de registre tampon pour les opérations d'entrées/sorties. Sa taille est égale à la longueur des mots en mémoire.

Le registre ACC, appelé aussi registre AX pour les microprocesseur de type INTEL, est accessible au programmeur et très sollicité. Certaines machines possèdent plusieurs accumulateurs.

- Les registres généraux ou registres de travail :

Ils permettent de limiter les accès à la mémoire, ce qui accélère l'exécution d'un programme. Ils peuvent conserver des informations utilisées fréquemment, des résultats intermédiaires, etc. Ils sont accessibles au programmeur.

On peut citer par exemple pour les microprocesseur de type INTEL les registre de travail suivants :

1. Le registre de base BX:

En général à l'intérieur d'un programme, on fait référence à des adresses relatives ou virtuelles. L'unité centrale a besoin cependant de connaître les adresses physiques où se situent réellement les instructions et les données. Celles-ci dépendent de l'endroit où a été chargé le programme en mémoire. L'espace physique occupé par un programme peut ne pas être contigu. Le rôle des registres de base est de permettre le calcul des adresses effectives(réelles). Un registre de base contient une adresse de référence, par exemple l'adresse physique correspondant à l'adresse virtuelle 0 du programme (première instruction du programme). L'adresse physique est obtenue en ajoutant au champ adresse de l'instruction le contenu du registre de base.

Il peut aussi venir en appui à l'accumulateur si ce dernier n'est pas disponible.

Exemple :

Soit l'instruction suivante :

Add AX, BX

Cette instruction veut dire additionner le contenu du registre BX à celui de AX et mettre le résultat dans AX.

2. Le registre compteur CX:

Ce registre sert à stocker les compteurs dans une boucle, le nombre de positions lors d'une opération de décalage...Il est accessible par le programmeur.

Exemple :

Algorithme	Code
AX ← 2	mov AX,2 ;affectation de la valeur 2 à AX
Pour de 1 à 5 faire	mov CX,5 ;affectation de la valeur 5 à CX
AX ← AX+1	boucle : ;étiquette de la boucle
Fin pour	ADD AX,1 ;ajout de la valeur 1 à AX
	Loop boucle ;Décrémenter de CX et retour au niveau de l'étiquette de la boucle si CX est non nul.

3. Le registre de données DX:

Ce registre est souvent utilisé pour manipuler les chaînes de caractères et pour l'affichage des données.

- Le registres d'indice ou d'index (XR) :

Ils peuvent être utilisés comme des registres généraux mais ils ont une fonction spéciale utilisée pour l'adressage indexé. Dans ce cas l'adresse effective d'un opérande est obtenue en ajoutant le contenu du registre d'index à l'adresse contenue dans l'instruction. Ce type d'adressage et de registre est très utile pour manipuler des tableaux. Le programmeur dispose alors d'instructions permettant l'incrémenter ou la décrémenter du registre d'index. En particulier les registres d'index peuvent être incrémentés ou décrémentés automatiquement après chaque utilisation.

Dans les microprocesseur INTEL on trouve particulièrement 2 registres d'index : SI(Source Index et DI : Destination Index) .

- Le registre pointeur de pile (PP) :

Une pile est une zone mémoire dans laquelle les informations sont rangées de façon contiguë. Le pointeur de pile (Stack Pointer : SP) indique le sommet de la pile : la position de la dernière information enregistrée. Dans certaines machines le pointeur de pile indique la position où sera mémorisée la prochaine donnée. Le fonctionnement d'une pile est du type Dernier Entré Premier Sorti (LIFO : Last In First Out). Les deux principales opérations liées à la pile concernent l'ajout d'un élément dans la pile ou le retrait, souvent nommées respectivement PUSH et PULL. Lorsqu'une donnée est enregistrée dans la pile elle est placée à l'adresse qui suit celle du dernier mot stocké. Après l'opération le pointeur de pile est incrémenté. Lorsque un mot est retiré de la pile il correspond à la dernière information qui y a été entrée. Après l'opération le pointeur est décrémenté. Une pile est réservée à l'usage de l'unité centrale, en particulier pour sauvegarder les registres et l'adresse de retour en cas d'interruption ou lors de l'appel d'une procédure. Le pointeur de pile est accessible au programmeur, ce qui peut engendrer souvent des erreurs faute de mauvaise manipulation.

4. Les circuits de l'unité centrale de traitement:

▪ Le décodeur :

il détermine quelle est l'opération à effectuer afin d'activer les circuits des microprogrammes correspondants à l'action voulue et quels sont les opérandes nécessaire à son exécution.

Les microprogrammes sont définis par le constructeurs de la machines. Ils représentent un ensemble de signaux câblées ayant pour rôle de commander l'exécution de l'action détectée par le décodeur.

Exemple :

Soit l'instruction ADD AX,5

Le décodage de cette instruction va déterminer qu'il s'agit d'une opération d'addition et va effectuer les instructions suivantes :

1. Activer les signaux de commandes relatifs au stockage de la valeur 5 dans un registre temporaire,
2. Activer la ligne d'opération responsable de l'exécution de l'addition,

3. Active les signaux de commandes relatifs au stockage du résultat dans AX.

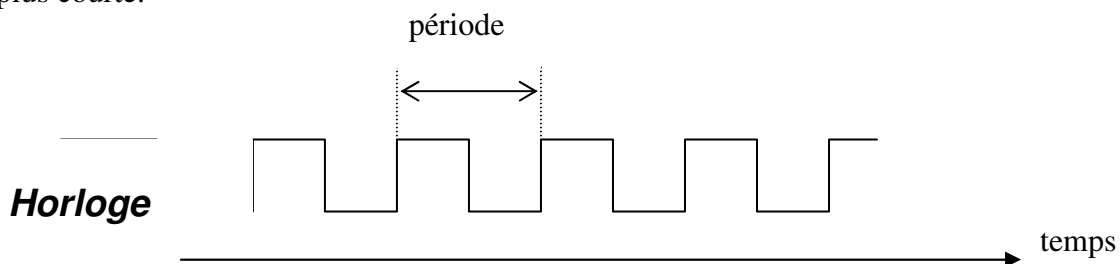
Tous ces signaux constituent le microprogramme associé à l'action ADD.

▪ **L'horloge (interne ou externe) :**

C'est un circuit qui émet des impulsions permettant la synchronisation de tous les éléments de l'unité centrale.

L'horloge est souvent construite à base de quartz, substance qui émet naturellement des signaux périodiques. Ces impulsions périodiques constituent le cycle de base ou cycle machine. L'intervalle de temps entre deux impulsions représente le cycle ou la période de l'horloge.

Une instruction peut consommer 1 ou plusieurs cycles machines. On rencontre parfois le terme de cycle CPU (Central Process Unit) pour indiquer le temps d'exécution de l'instruction la plus courte.



▪ **Le séquenceur :**

Il génère les signaux de commande aux différents composants. À partir de l'horloge, il contrôle le fonctionnement des circuits de l'unité de calcul nécessaires à l'exécution de chaque instruction et gère de cette façon de tous les signaux émis par l'unité de commande.

Exemple :

Après 2 top d'horloge, le séquenceur comprend que l'instruction d'addition est exécutée et envoie un signe au compteur ordinal pour passer à l'instruction suivante.

II- Principes de fonctionnement d'un processeur :

1. Les étapes d'exécution d'une instruction:

Le schéma suivant illustre la communication de l'unité de commande avec la mémoire centrale en vue d'exécuter une instruction.

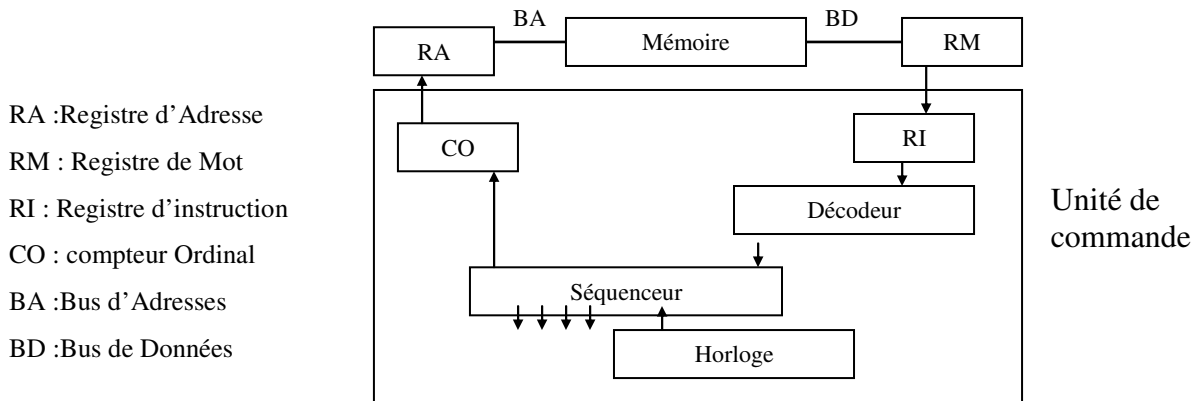


Figure 13 : les étapes d'exécutions d'une instruction

Cependant, l'exécution des instructions peut se découper en trois étapes qui :

1^{ère} étape : Instruction Fetch (Recherche de l'instruction) :

- émettre un ordre de lecture du mot mémoire dont l'adresse se trouve dans le compteur ordinal ;
- lire l'instruction fournie par la mémoire centrale ;
- placer l'instruction dans le registre d'instruction ;
- Incrémentation du compteur ordinal par l'adresse de la prochaine instruction ;

2^{ème} étape : Analyse et décodage de l'instruction :

Le décodage de l'instruction se trouvant dans le registre d'instruction par le décodeur permet de :

- connaître la nature de l'opération à réaliser et donc les circuits à activer;
- connaître la ou les adresses effectives des opérandes nécessaires à l'exécution de l'instruction.

3^{ème} étape : Exécution de l'instruction

- émettre un ordre de lecture de la donnée située en MC et dont l'adresse est fournie par l'étape2;

Architecture des systèmes informatiques

- lire la donnée fournie par la MC;
- placer cette donnée dans un registre interne du CPU ou bien dans l'accumulateur selon le travail demandé par l'instruction
- envoyer les microcommandes via le séquenceur;
- stocker le résultat dans un registre de sortie, l'accumulateur par exemple.
- affectation des bits du registre d'état en fonction du résultat obtenu.

Ensuite, l'adresse de la prochaine instruction étant dans le compteur ordinal, on reprend à la première étape et ainsi de suite jusqu'à l'exécution de tout le programme.

Application :

Soit le pseudo-code d'une portion d'un programme suivant :

Mul A,B ; multiplication de A et B

Ran C ; rangement du résultat dans la variable C

le processeur avec lequel nous travaillons possède le format d'instruction suivant :

Code opération (COP)	adresse1 (@ 1)	adresse2 (@ 2)
-------------------------	---------------------	---------------------

- Une case mémoire est sur 4 octets et une adresse mémoire est sur 12 bits.

- COP (multiplication) = $C6_{(hexa)}$

- COP (rangement en mémoire centrale à partir de l'accumulateur) = $B3_{(hexa)}$

L'adresse de la variable A en mémoire est $64_{(hexa)}$ et elle contient la valeur $2_{(10)}$.

- L'adresse de la variable B en mémoire est $78_{(hexa)}$ et elle contient la valeur $4_{(10)}$.

- L'adresse de la variable C en mémoire est $79_{(hexa)}$.

- Le programme est stocké en mémoire à partir de l'adresse $14_{(hexa)}$.

Donner la forme hexadécimale de ce programme en mémoire centrale.

Quelle est la taille de ce programme ?

Architecture des systèmes informatiques

Citer les étapes d'exécution de ce programme en précisant le contenu du CO, RI, AX, BX et la variable C en mémoire.

Solution :

Adresse de A = $64_{(\text{hexa})}$

Adresse de B = $78_{(\text{hexa})}$

Adresse de C = $79_{(\text{hexa})}$

MC	
14	C6 064 078
15	B3 079 000
64	00 00 00 02
78	00 00 00 04
79	
...	...

La taille de la première instruction est 4 octets, La taille de la deuxième instruction est 4 octets → Ce programme a une taille de 8 octets.

Exécution de l'instruction (Mul A,B) :

- Première étape :

Mettre l'adresse de la première instruction dans le compteur ordinal.

CO	014
----	-----

cette adresse sera transportée par le bus d'adresse pour lire une case mémoire et transmettre son contenu au registre d'instruction.

RI	C6 064 078
----	------------

Ensuite incrémentation du compteur ordinal

CO

015

- Deuxième étape :

Décodage de l'instruction dans RI : Il s'agit d'une opération de multiplication sur les opérandes d'adresses 064 et 078.

- troisième étape :

- lire la donnée de l'adresse 064 dans la MC;
- placer cette donnée dans l'accumulateur Ax
- lire la donnée de l'adresse 078 dans la MC;
- placer cette donnée dans un autre registre tel que BX vu que l'accumulateur est occupée.

AX

00 00 00 02

BX

00 00 00 04

- envoyer les microcommandes de multiplication via le séquenceur vers l'UAL;
- stocker le résultat dans un registre de sortie, l'accumulateur par exemple.

AX

00 00 00 08

Exécution de l'instruction (RAN C) :

- Première étape :

Le compteur ordinal contient l'adresse 015.

CO

015

Architecture des systèmes informatiques

Cette adresse sera transportée par le bus d'adresse pour lire une case mémoire et transmettre son contenu au registre d'instruction.

RI

B3 079 000

Ensuite le compteur ordinal contiendra l'adresse de la prochaine instruction.

CO

016

▪ Deuxième étape :

Décodage de l'instruction dans RI : Il s'agit d'une opération d'écriture de la valeur contenue dans l'accumulateur à la case mémoire d'adresse 079.

▪ Troisième étape :

- lire le contenu de l'accumulateur;
- envoyer une microcommande d'écriture ;
- envoyer l'adresse 079 à travers le bus d'adresse ;
- envoyer la données de l'accumulateur via le bus de données.
- Ecrire la donnée 00 00 00 08 dans la case mémoire 078 destinée à la variable C.

AX

00 00 00 08

MC

078

00 00 00 08

2. Le cycle d'exécution d'une instruction

Un cycle d'instruction est la période de temps nécessaire pour effectuer les trois étapes de recherche, de décodage et d'exécution. Ce cycle peut se décomposer en un cycle de recherche (instruction et opérandes), un cycle de décodage et un cycle d'exécution.

Exemple :

Soit l'instruction suivante : ADD AX, 5 qui ajoute au contenu de l'accumulateur la valeur 5.

Supposons que cette instruction nécessite 3 tops d'horloges pour l'étape de recherche, 1 top d'horloge pour le décodage et 2 tops pour l'exécution. Le chronogramme d'exécution de cette instruction peut se présenter comme suit :

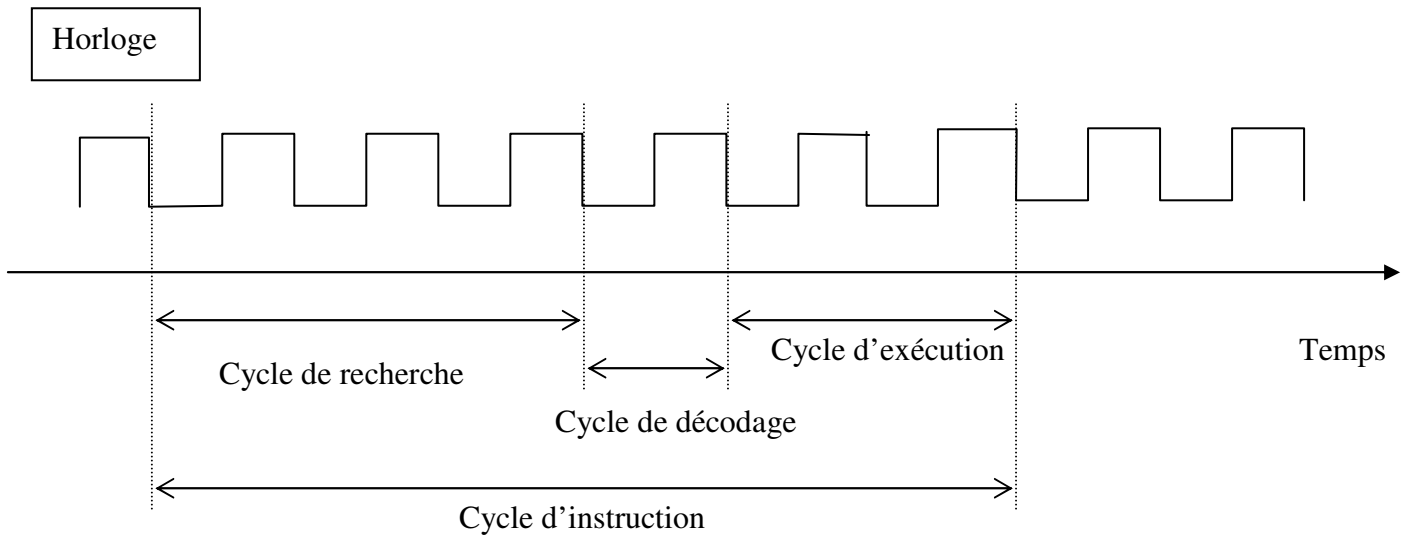


Figure14 :Le cycle d'exécution d'une instruction

3. La technique de pipeline :

Le microprocesseur est chargé d'exécuter les diverses instructions d'une programme le plus rapidement possible. Face à un ensemble d'instructions le microprocesseur peut fonctionner de deux manières possibles : séquentielle ou selon la technique de pipeline.

- Fonctionnement séquentiel :

Lors d'une exécution séquentielle, un microprocesseur exécute une seule instruction à la fois : Il doit achever toutes les étapes relatives à l'exécution d'une instruction (Recherche, décodage et exécution) pour pouvoir passer à l'instruction suivante.

Exemple :

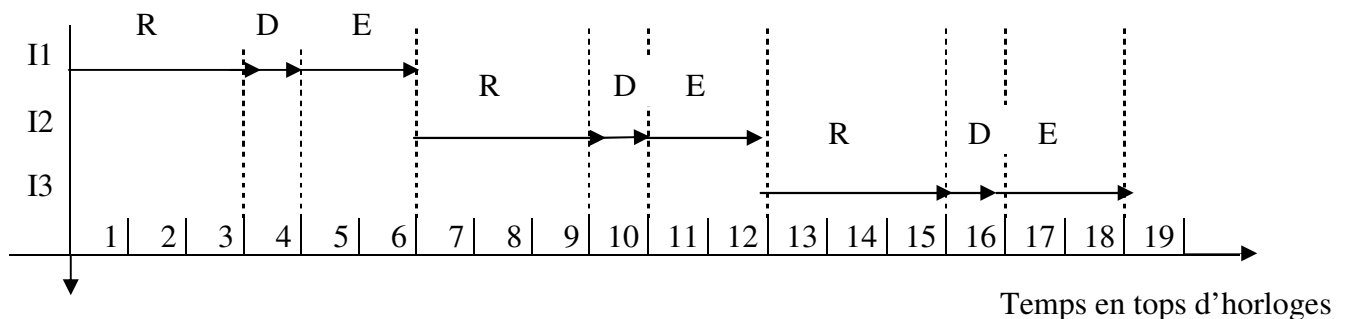
Soit les trois instructions suivantes : I1, I2 et I3.

Le cycle de recherche (R) = 3 tops d'horloges,

Le cycle de décodage (D) = 1 top d'horloges,

Le cycle d'exécution (E) = 2 tops d'horloges.

L'exécution séquentielle de ces instructions dans le microprocesseur se déroule de la façon suivante :



➔ Durée totale = (3 + 1 + 2) * 3 = 18 tops d'horloges.

- Fonctionnement pipeliné ou parallèle :

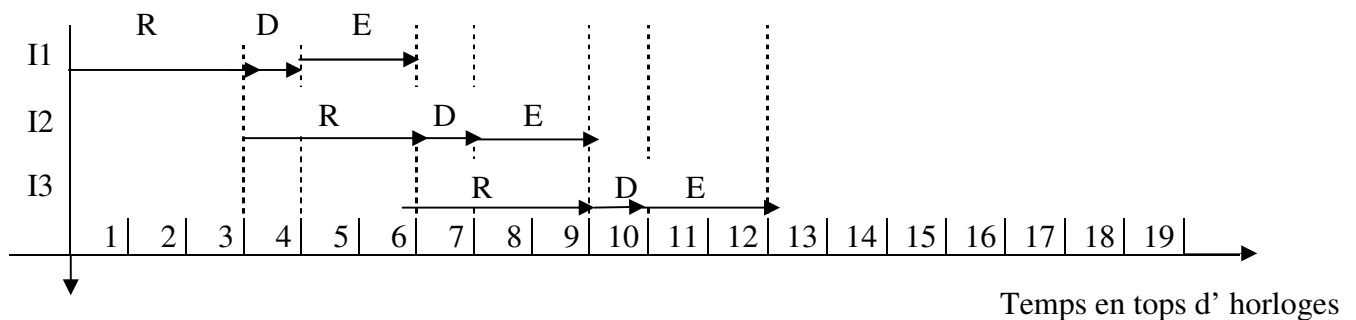
La technique de pipeline consiste à faire fonctionner plusieurs instructions en parallèle. Une fois, le microprocesseur achevé l'exécution d'une étape d'une instruction, il peut entamer son exécution pour l'instruction suivante.

Remarque :

Le processeur ne peut pas exécuter la même étape (exemple le décodage) pour deux instructions en même temps.

Exemple :

Si on reprend le même exemple précédant pris pour le fonctionnement séquentiel, l'exécution pipelinée de ces instructions dans le microprocesseur se déroule de la façon suivante :



→ Durée totale = 12 tops d'horloges.

III- Les critères de performance d'un microprocesseur

1. La fréquence d'horloge :

La durée d'exécution d'une instruction est mesurée en tops d'horloge. Donc plus la période de cette horloge est réduite plus le temps d'exécution de l'instruction en secondes est court.

La fréquence de l'horloge est l'inverse de la période et elle est calculée en hertz (Hz).

Ce qui nous ramène à conclure que l'augmentation de la fréquence (C'est à dire la diminution de la période) influence le nombre d'instructions exécutées par secondes devenant plus élevé et ainsi rendre le microprocesseur plus performant.

Exemple :

Soit une instruction dont l'étape d'exécution nécessite 3 tops horloge, l'étape de décodage 1 top

et l'étape d'exécution 2 tops.

- En supposant la fréquence d'horloge de ce microprocesseur est 10MHz, quelle est la durée d'exécution de cette instruction ?

Architecture des systèmes informatiques

La fréquence $f = 10\text{MHz} = (10 \cdot 10^6)\text{Hz}$

La période $T = 1/f = 10^{-7}\text{s}$

La durée de l'instruction $= (3 + 1 + 2) \cdot 10^{-7}\text{s} = 0.6 \mu\text{s}$

Remarque :

- 1ms (milli-seconde) $= 10^{-3}\text{s}$
- $1\mu\text{s}$ (micro-seconde) $= 10^{-6}\text{s}$
- 1ns (nano-seconde) $= 10^{-9}\text{s}$

- En supposant la fréquence d'horloge de ce microprocesseur est 20MHz , quelle est la durée d'exécution de cette instruction ?

La fréquence $f = 20\text{MHz} = (20 \cdot 10^6)\text{Hz}$

La période $T = 1/f = 0.05 \cdot 10^{-6}\text{s} = 0.05\mu\text{s}$

La durée de l'instruction $= 6 \cdot 0.05\mu\text{s} = 0.3 \mu\text{s}$

➔ La durée d'exécution est inférieure pour une fréquence plus élevée.

2. La taille du bus de données :

Le temps le plus pénalisant au cours de l'exécution d'une instruction est le temps de recherche de d'une instruction ou d'une données dans la mémoire centrale.

La taille du bus de données agit sur le nombre des accès mémoires effectués au cours d'une instruction et par la suite il constitue l'un des critères de performance du microprocesseur les plus importants

Pour cette raison, On distingue souvent les microprocesseurs par la taille de leurs bus de données. Par exemple, on dit la famille des microprocesseurs INTEL 8 bits, la famille 16 bits ou la famille 32 bits.

3. La mémoire cache :

On rappelle que la mémoire cache est une mémoire de petite taille (quelques kilooctets) servant à stocker une portion du programme en exécution. Son rôle est minimiser les accès du microprocesseur à la mémoire centrale en les remplaçant par l'accès à la mémoire cache dont le temps d'accès est plus réduit. Plus la taille de la mémoire cache est grande moins on a

d'accès à la mémoire centrale et plus le gain en temps d'accès est important. Ce qui fait de la taille de la mémoire cache un autre critère de performance du microprocesseur.

IV- Exemples de microprocesseurs :

Les microprocesseurs ont connu une évolution au cours du temps. Aujourd'hui les deux grandes marques les plus commercialisés sur le marché sont : les processeur INTEL et AMD.

Le tableau ci-dessous présente certaines caractéristiques de ces processeurs :



Marque	AMD	INTEL	
Processeur	Athlon	Pentium 3	Pentium 4
Fréquence	1GHz	450MHz→1,2GHz	3GHz
Taille bus de données	64bits	64bits	64 bits
Figure			

Figure 15 : caractéristique des processeurs