

Architecture et programmation des ordinateurs :

5- Microprocesseurs & jeu d'instruction

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (1)

1- Description d'un microprocesseur

Un microprocesseur se présente sous la forme d'un circuit intégré muni d'un nombre généralement important de broches.

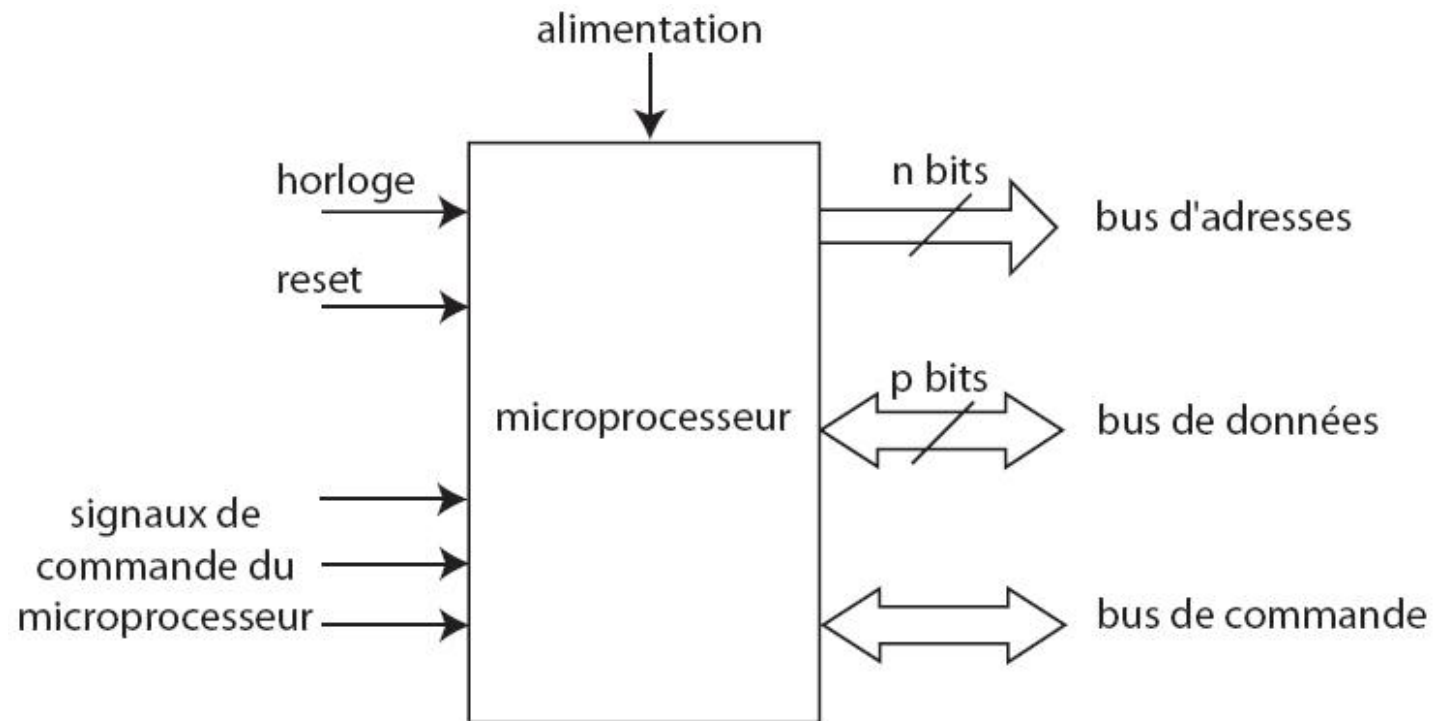
Exemples :

- *Intel 8085, 8086 : 40 broches, DIP (Dual In-line Package) ;*
- *Motorola 68000 : 64 broches, DIP ;*
- *Intel 80386 : 196 broches, PGA (Pin Grid Array).*

Technologies de fabrication : NMOS, PMOS, CMOS.

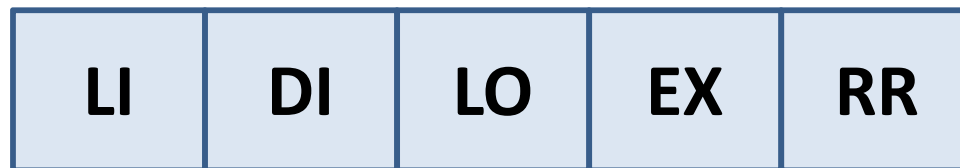
Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (2)

2- Organisation fonctionnelle externe d'un microprocesseur



Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (3)

3- Organisation fonctionnelle interne d'un microprocesseur



LI : Lecture de l'Instruction codée

DI : Décodage de l'Instruction

LO : Lecture de des Opérandes

EX : Exécution

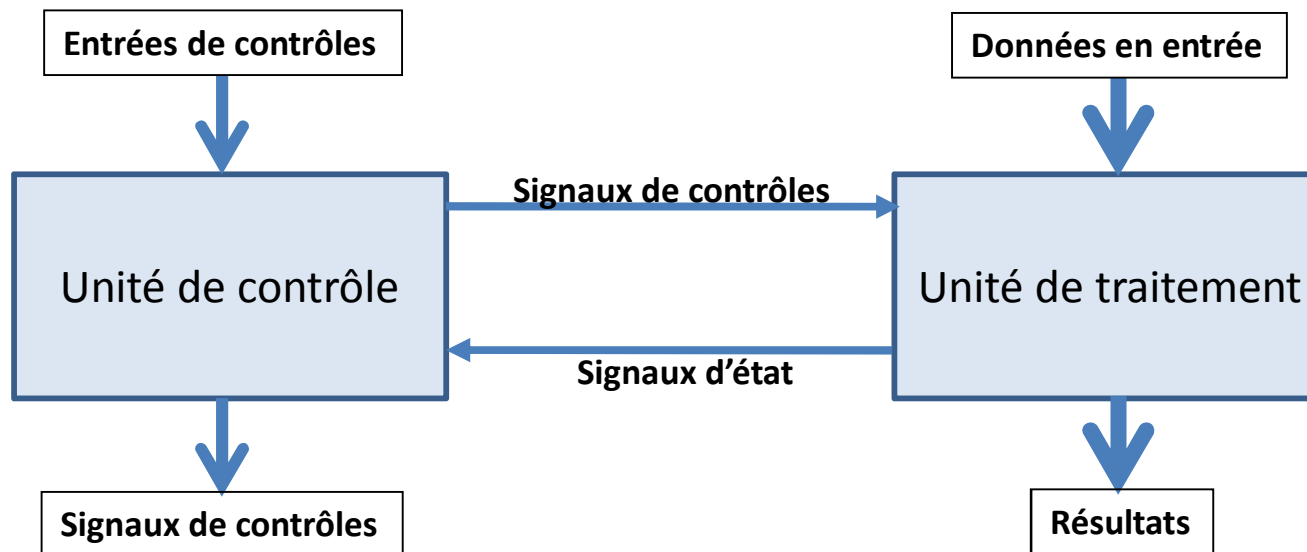
RR : Rangement du résultat

Remarque :

Unité de contrôle a en charge les différentes phases de l'exécution d'une instruction

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (4)

3- Organisation fonctionnelle interne d'un microprocesseur



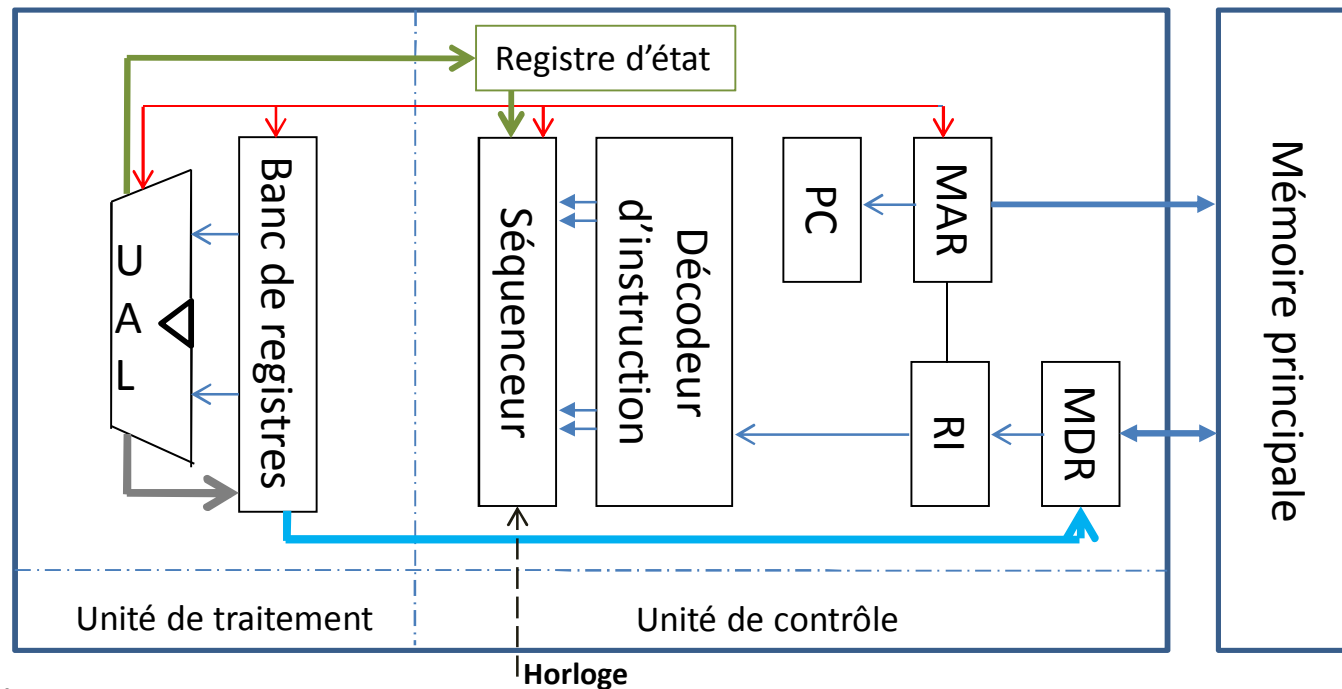
Remarques :

L'unité de traitement = éléments de mémorisation (registres internes) + UAL

Il s'agit d'une machine séquentielle

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (5)

3- Organisation fonctionnelle interne d'un microprocesseur



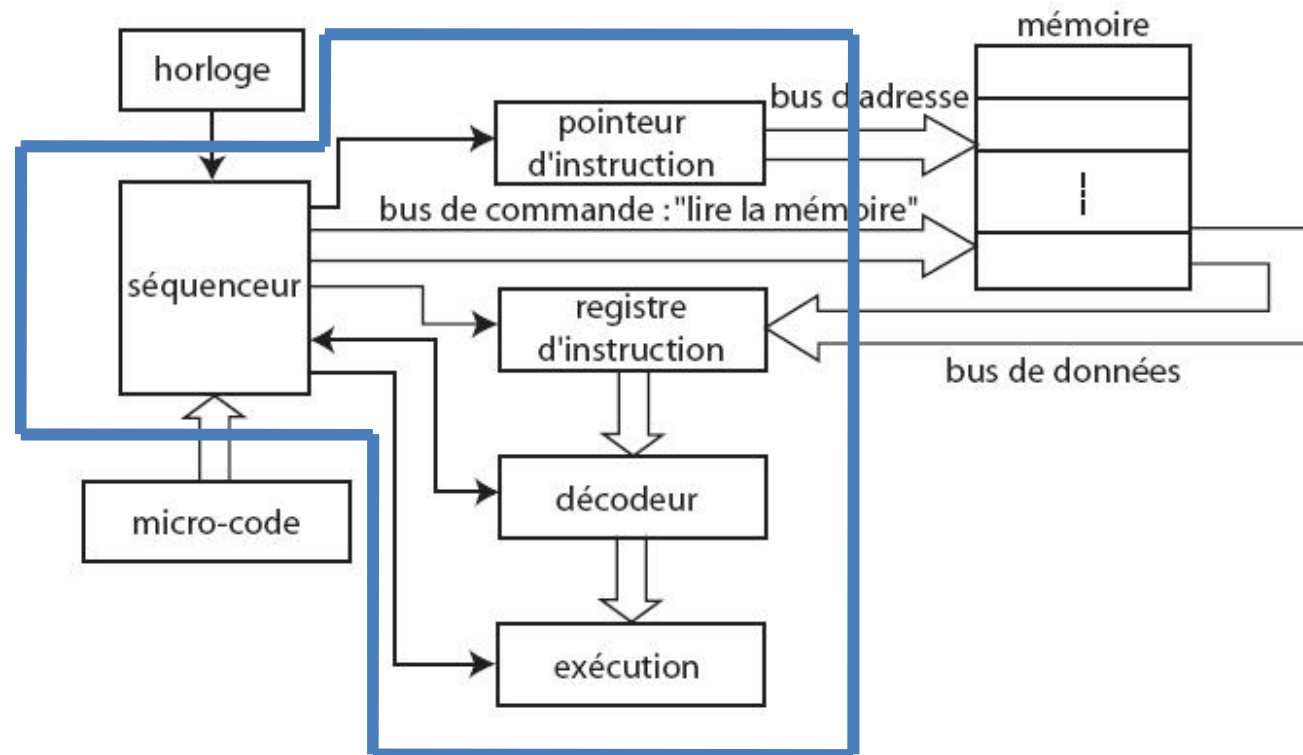
Remarques :

Banc de registres :: opérandes ; MAR -> RI :: adresse opérande

PC :: adresse instruction ; MDR -> MP :: Résultats

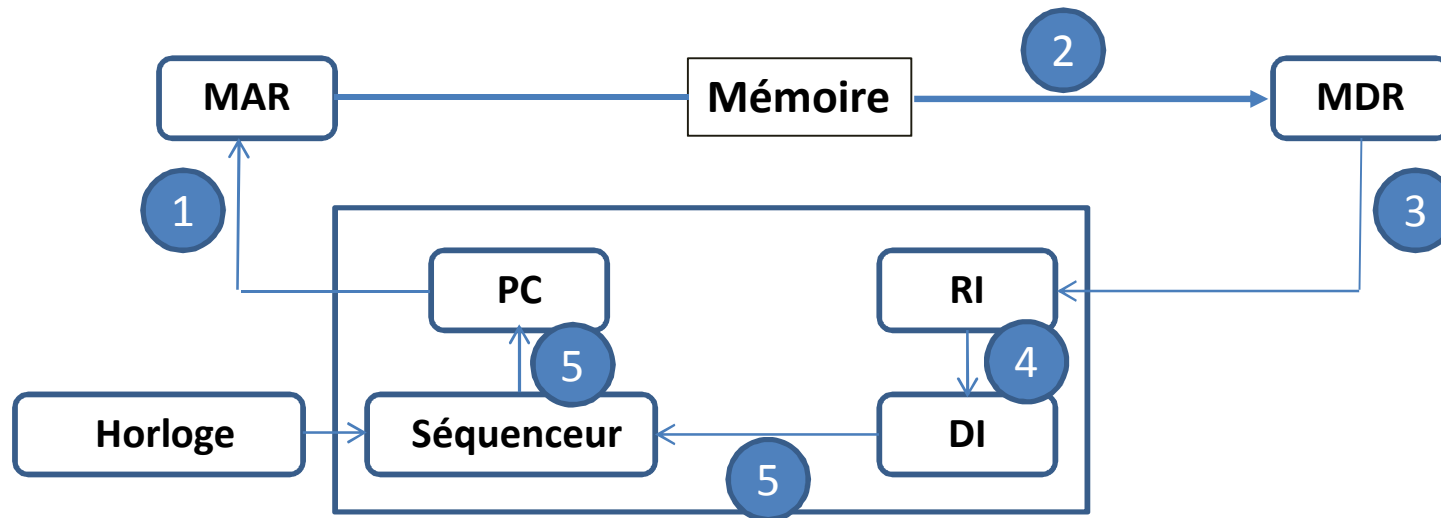
Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (6)

3- Organisation fonctionnelle interne d'un microprocesseur



Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (7)

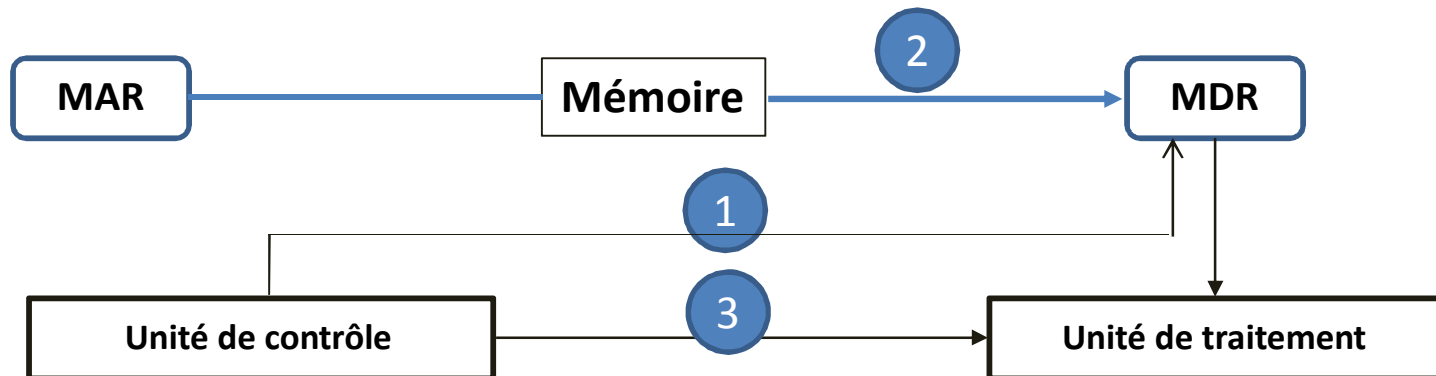
4- Cycle de recherche d'une instruction



MAR : Memory Adress Register ; RI : Registre instruction PC : Registre Compteur
MDR : Memory Data Register ; DI: Décodeur instruction

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (8)

5- Cycle d'exécution d'une instruction



Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (9)

5- Fonctionnement d'un microprocesseur

Un microprocesseur exécute un programme. Le programme est une suite d'instructions stockées dans la mémoire. Une instruction peut être codée sur **un ou plusieurs octets**.
Format d'une instruction :

Opération à effectuer

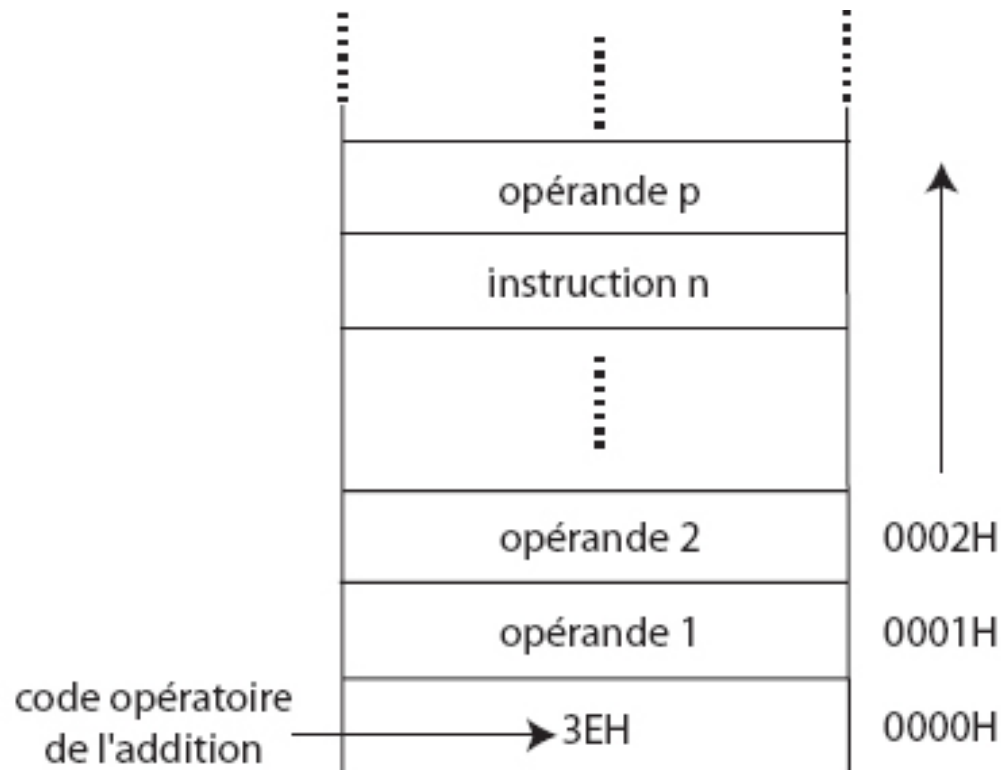
Opérande 1 , Opérande 2

ADDITIONNER

Case mémoire 1 , Case mémoire 2

Architecture des ordinateurs : Microprocesseur & jeu d'instructions (10)

5- Fonctionnement d'un microprocesseur



Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (11)

5- Fonctionnement d'un microprocesseur

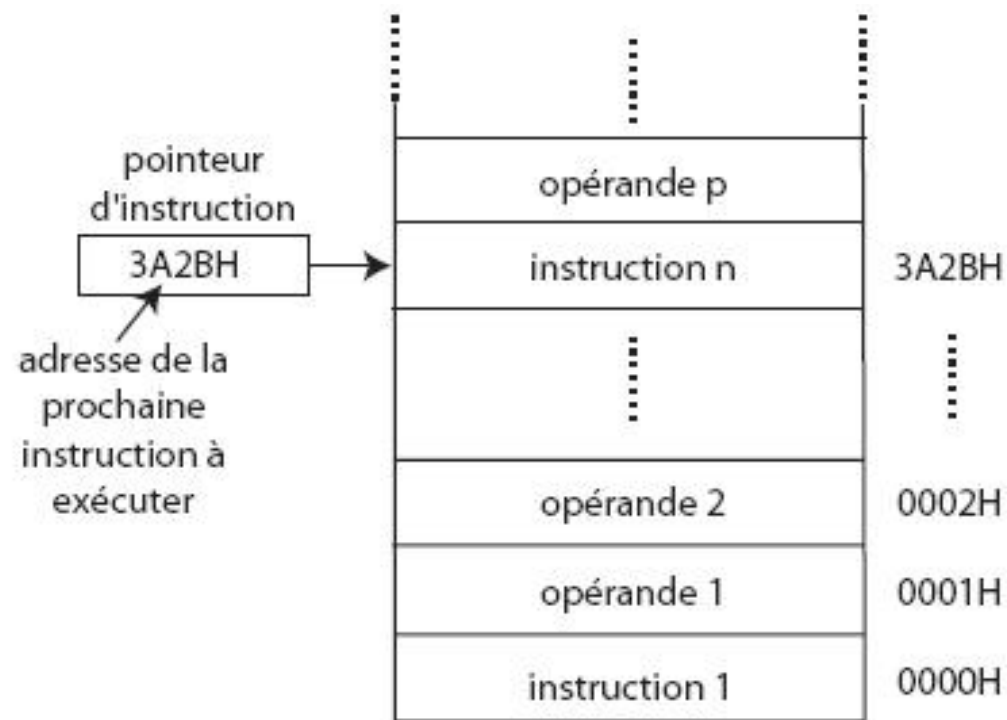
Pour exécuter les instructions dans l'ordre établi par le programme, le microprocesseur doit savoir à chaque instant l'adresse de la prochaine instruction à exécuter.

Le microprocesseur utilise un registre contenant cette information.

Ce registre est appelé **pointeur d'instruction (IP : Instruction Pointer) ou compteur d'instructions ou compteur ordinal.**

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (12)

5- Fonctionnement d'un microprocesseur



Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (13)

5- Fonctionnement d'un microprocesseur

Remarque : la valeur initiale du pointeur d'instruction est fixée par le constructeur du microprocesseur. Elle vaut une valeur bien définie à chaque mise sous tension du microprocesseur ou bien lors d'une remise à zéro (reset).

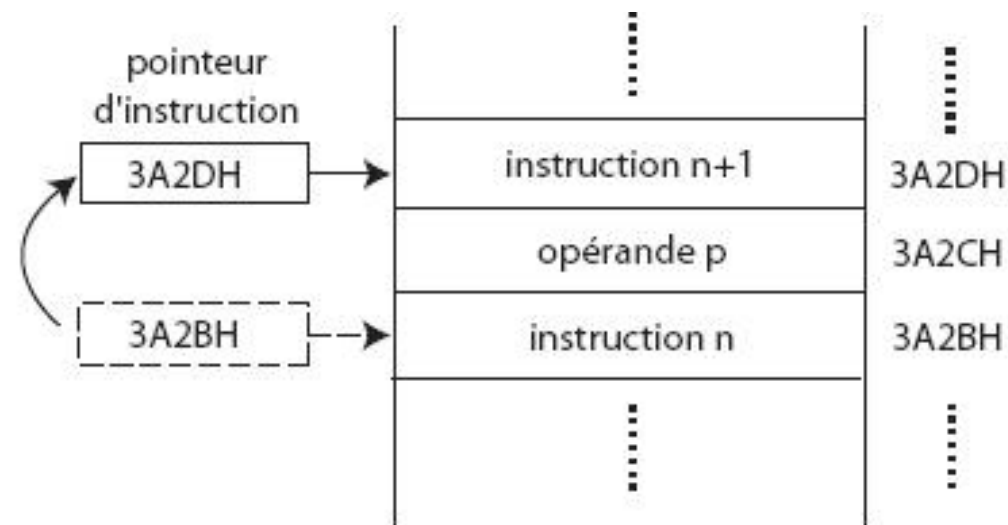
Pour savoir quel type d'opération doit être exécuté (addition, soustraction, ...), le microprocesseur lit le premier octet de l'instruction pointée par le pointeur d'instruction (code opératoire) et le range dans un registre appelé registre d'instruction. Le code opératoire est décodé par des circuits de décodage contenus dans le microprocesseur. Des signaux de commande pour l'UAL sont produits en fonction de l'opération demandée qui est alors exécutée.

Remarque : pour exécuter une instruction, l'UAL utilise des registres de travail, exemple : l'accumulateur, registre temporaire recevant des données intermédiaires.

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (14)

5- Fonctionnement d'un microprocesseur

Pendant que l'instruction est décodée, le pointeur d'instruction est incrémenté de façon à pointer vers l'instruction suivante :



Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (15)

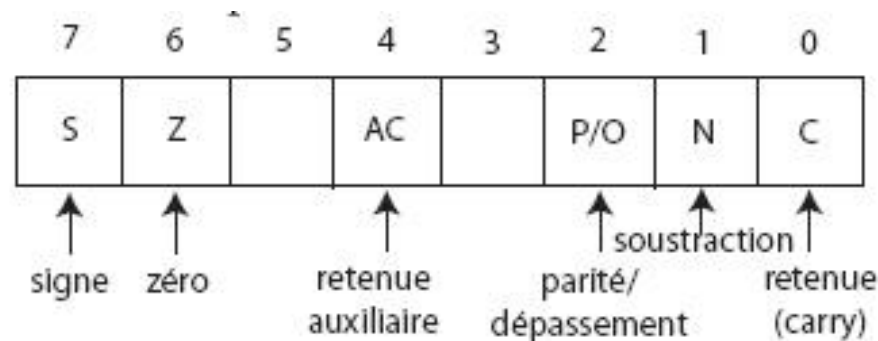
5- Fonctionnement d'un microprocesseur

puis le processus de lecture et de décodage des instructions recommence.

A la suite de chaque instruction, un registre du microprocesseur est actualisé en fonction du dernier résultat : c'est le **registre d'état** du microprocesseur. Chacun des bits du registre d'état est un **indicateur d'état** ou **flag (drapeau)**.

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (16)

5- Fonctionnement d'un microprocesseur



Exemple : registre d'état du microprocesseur Z80

Les indicateurs d'état sont activés lorsqu'une certaine condition est remplie,
Exemple : le flag Z est mis à 1 lorsque la dernière opération a donné un résultat nul,
le flag C est mis à un lorsque le résultat d'une addition possède une retenue, ...

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17)

6- Le microprocesseur MOTOROLA 68000 - Introduction

Le 68000 est un microprocesseur CISC 16/32 bits sorti en 1979. il devint rapidement le processeur le plus prisé de sa génération. Le formidable succès des machines comme le Mac, l'Amiga ou l'Atari est en grande partie due à cet innovant processeur.

Voici ces caractéristiques :

- Mémoire adressable de 16Mo (24 lignes d'adresse).
- Jeu d'instruction très complet (56 Instructions).
- 14 modes d'adressage différents.
- 8 registres de donnée 32 bits.
- 7 registres d'adresse 32 bits.
- Gestion de privilèges (Utilisateur / Superviseur).
- Deux pointeurs de pile 32 bits différents.

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-1)

6- Le microprocesseur MOTOROLA 68000 - Introduction

Le terme 16/32 bits résume la façon dont le circuit manipule les données.
Le bus de données interne est sur 32 bits mais le processeur communique avec le reste du monde sur un bus externe de 16 bits.

Aujourd'hui la série "microprocesseur 680x0" se compose du 68010, 68020, 68030, 68040 et 68060. Ces processeurs constituent la base des versions microcontrôleurs 683xx. Les caractéristiques de ces composants ainsi que de nombreux manuels utilisateurs peuvent être télécharger sur le site de [MOTOROLA](http://www.motorola.com).

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-2)

6- Le microprocesseur MOTOROLA 68000 - Introduction

Tous les registres du 68000 sont sur 32 bits excepté le registre d'état CCR 16 bits.

Petit rappel sur les formats de données :

Quartet = 4 bits

Octet = 8 bits

Mot / Word = 16 bits

Long mot / Long Word = 32 bits

Les registres de données et d'adresses sont banalisés à l'exception de A7 aussi appelé SP (Stack pointer) qui représente le pointeur de pile.

En interne, il existe en fait deux pointeurs différents l'USP (User Stack Pointer) en mode utilisateur et le SSP (Supervisor Stack Pointer) en mode superviseur.

Il s'agit toujours du registre A7, le 68000 utilisant implicitement l'un ou l'autre en fonction de son mode de fonctionnement.

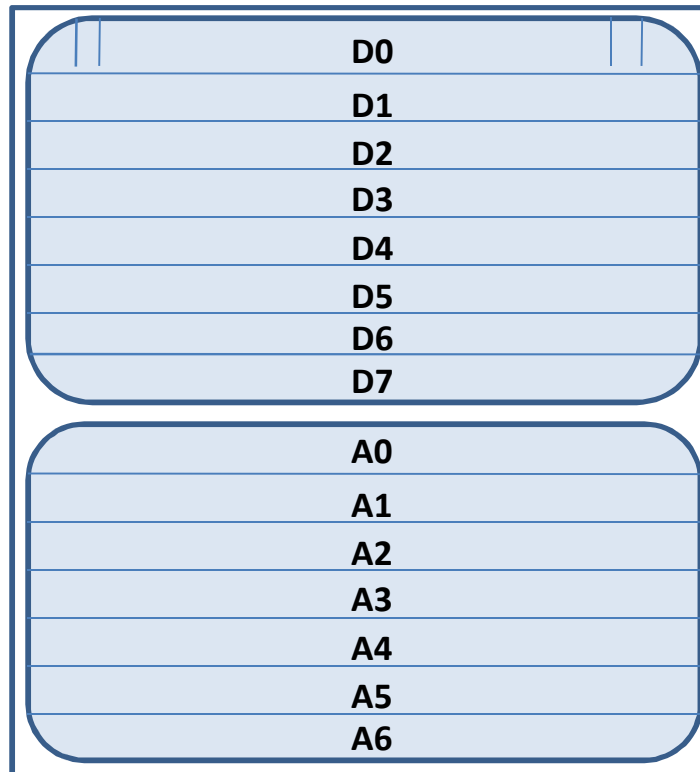
Architecture des ordinateurs :

Microprocesseur & jeu d'instructions ó (17-3)

6- Le microprocesseur MOTOROLA 68000 – Les registres

b31 : bit de poids fort

8 Registre de données
sur 32 bits



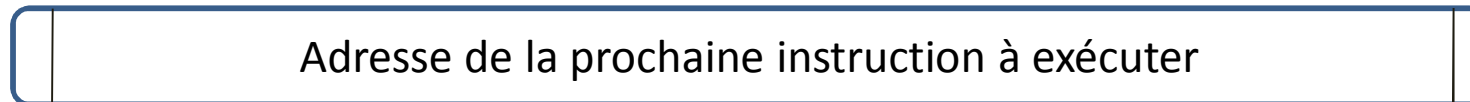
b0 : bit de poids faible

7 Registre d'adresse
sur 32 bits

Architecture des ordinateurs : Microprocesseur & jeu d'instructions (17-4)

6- Le microprocesseur MOTOROLA 68000 – Les registres

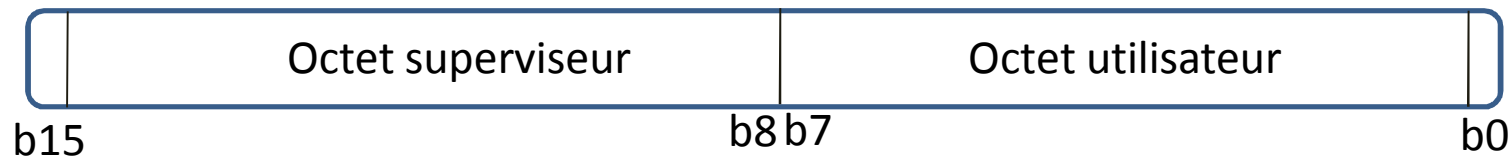
Registre PC : Programm Counter ou compteur de programme



b31 : bit de poids fort

b0 : bit de poids faible

Registre SR : Status Register ou registre d'état

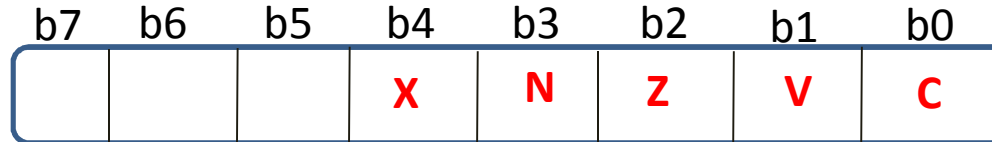


Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-5)

6- Le microprocesseur MOTOROLA 68000 – Les registres

Registre SR : Status Register ou registre d'état

Registre CCR : condition Code Register ou Octet utilisateur



Bit Z: positionné si le résultat d'une opération est nul.

Bit V: positionné s'il y a dépassement arithmétique.

Bit C: positionné si une retenue est générée.

Bit X: bit de retenue pour les opérations de calculs étendues.

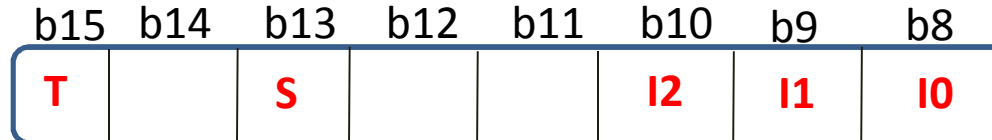
Bit N: positionné si le résultat d'une opération est

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-6)

6- Le microprocesseur MOTOROLA 68000 – Les registres

Registre SR : Status Register ou registre d'état

Octet Superviseur :



bit T : s'il est positionné met le processeur en mode trace.
Dans ce mode, le processeur s'arrête après l'exécution de chaque instruction et génère une exception permettant, par exemple, la mise en place d'un programme de déboguage.

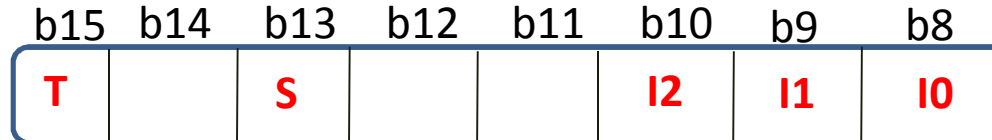
bit S : définit le mode de fonctionnement du processeur,
0 pour le mode Utilisateur et 1 pour le mode Superviseur.

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-7)

6- Le microprocesseur MOTOROLA 68000 – Les registres

Registre SR : Status Register ou registre d'état

Octet Superviseur :



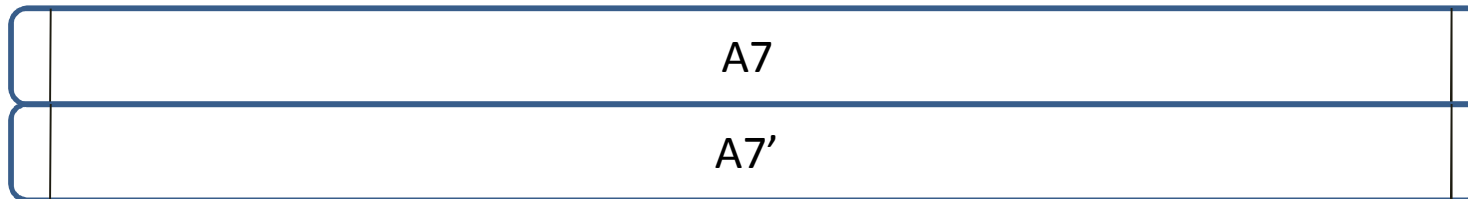
Les bits **I0** , **I1** et **I2** représentent le niveau du masque d'interruption. Les interruptions dont le niveau est inférieur ou égal au masque ne sont pas prisent en compte.

une exception est un évènement matériel (interruptions) ou logiciel qui dérouté le processeur vers une procédure particulière.

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-8)

6- Le microprocesseur MOTOROLA 68000 – Les registres

Registre SP : Stack Pointer ou registre pointeur de pile



Si **S=1**, le microprocesseur est en **mode superviseur**

-> le pointeur de la pile utilisé est **A7'**, appelé **SSP : Supervisor Stack Pointer**

Si **S=0**, le microprocesseur est en **mode utilisateur**

-> le pointeur de la pile utilisé est **A7**, appelé **USP : User Stack Pointer**

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-9)

6- Le microprocesseur MOTOROLA 68000 – Gestion de la pile

le fonctionnement de la pile est un élément fondamental pour le fonctionnement du processeur. La pile est en fait une portion de mémoire dont l'adresse se trouve dans A7 et qui permet la sauvegarde temporaire de données.

On parle d'empilement lors d'une écriture dans la pile et de dépilement lors d'une lecture.

Ce processus est utilisé, soit automatiquement par le micro lors de l'appel d'un sous-programme par exemple, soit par le programmeur.

Pour bien comprendre le fonctionnement d'une pile, on peut imaginer la pile comme une pile d'assiettes de différentes couleurs. En prenant comme condition initiale A7=10, on empile une assiette bleu, une assiette rouge puis une assiette verte. L'assiette bleu sera stockée en position 10, la rouge en 9 et la verte en 8, au final A7 vaut 8.

L'opération de dépilement est identique à l'empilement mais à l'envers, on récupère l'assiette verte, la rouge puis la bleu et A7 revient à sa position initiale soit 10

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-10)

6- Le microprocesseur MOTOROLA 68000 – Gestion de la pile

On peut remarquer que lors du processus d'empilement la première assiette empilé est bleu, lors du dépilement c'est la dernière.

C'est pour cette raison que l'on parle de pile **LIFO (Last In First Out)** le dernier élément empilé sera le premier dépilé.

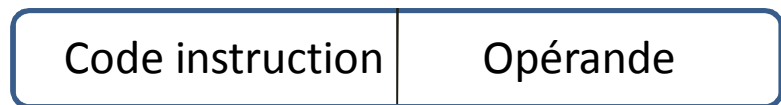
Pour information, il existe un autre type de pile appelée pile **FIFO (First In First Out)**. Dans ce cas le premier élément empilé sera le premier dépilé.

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-11)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage – Contexte général

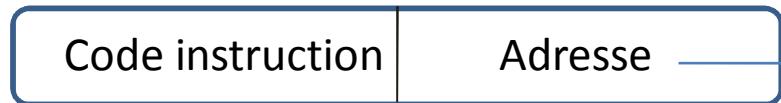
Le type de donnée contenu dans le champ opérandes caractérise le mode d'adressage.

1 Immédiat



Opérande = Constante

2 Direct absolu

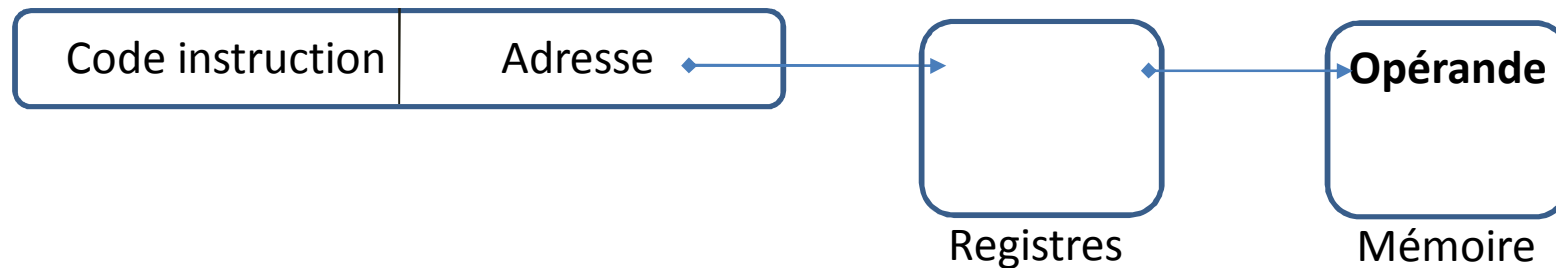


Mémoire

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-12)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage – Contexte général

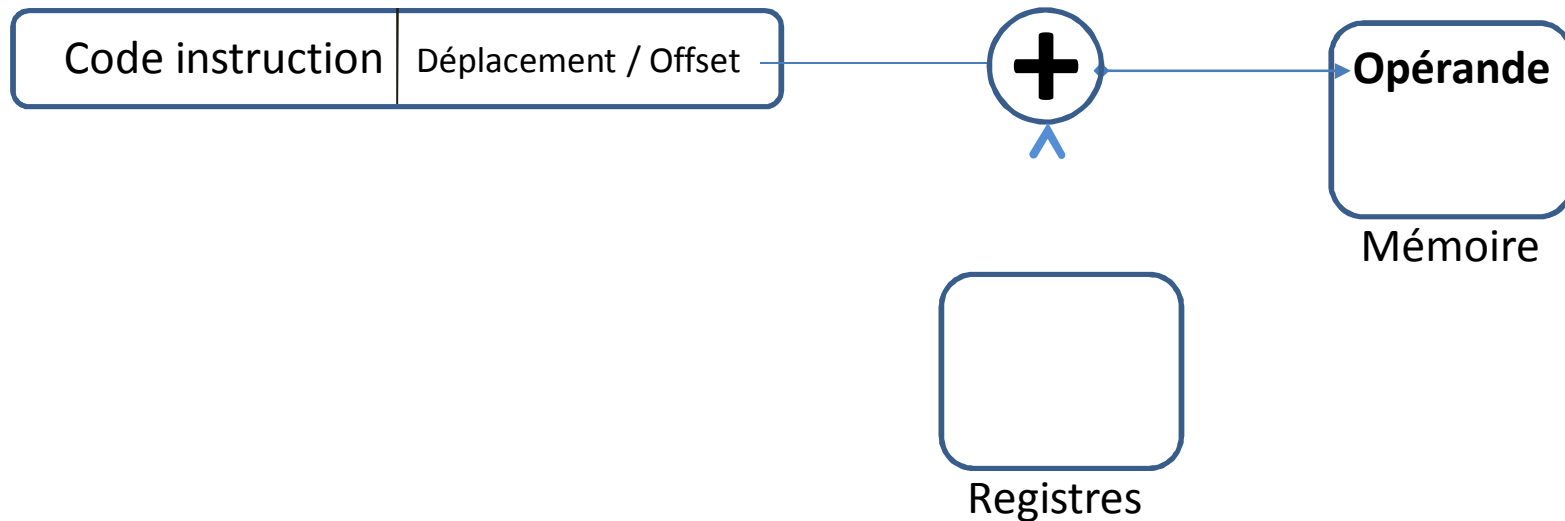
3 Indirect par registre



Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-13)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage – Contexte général

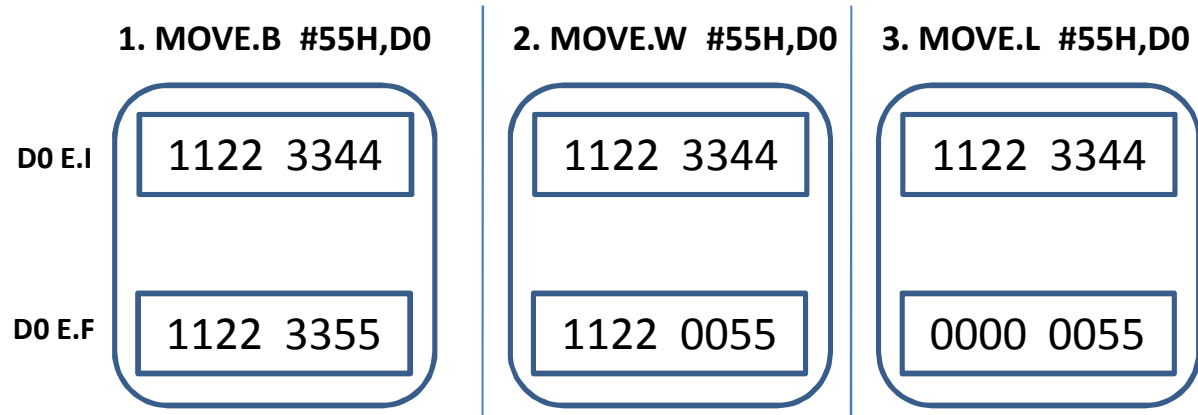
4 Relatif



Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-14)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

A Adressage immédiat : destination : registre de données



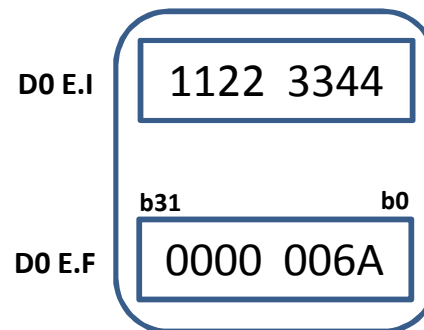
Taille de données transférées : .B : Byte : un octet / .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-15)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

A Adressage immédiat rapide : destination : registre de données

4. MOVEQ #6AH,D0



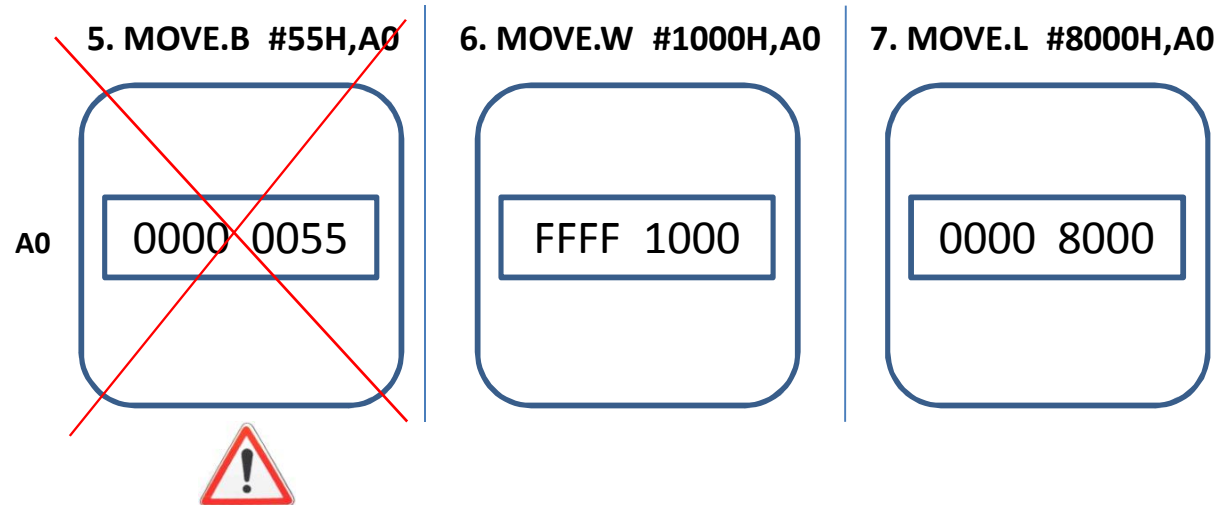
Taille de données transférées : .B : Byte : un octet **UNIQUEMENT**



Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-16)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

A Adressage immédiat : destination : registre d'adresse



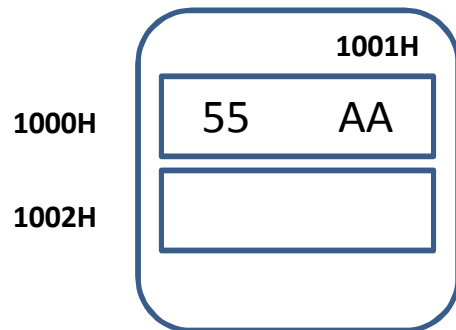
Taille de données transférées : .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits **UNIQUEMENT**

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-17)

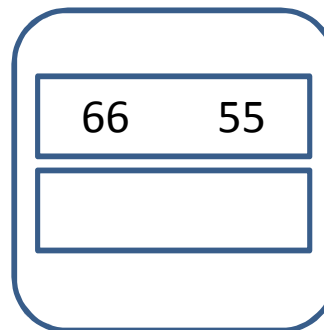
6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

A Adressage immédiat : destination : mémoire principale

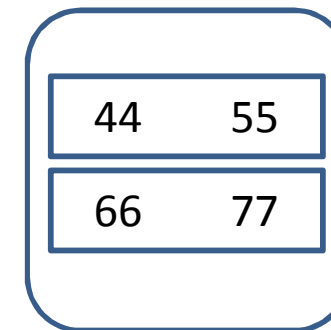
8. I1. MOVE.B #55H,1000H
I2. MOVE.B #AAH,1001H



9. I3. MOVE.W #6655H,1000H



10. I4. MOVE.L #44556677H,1000H



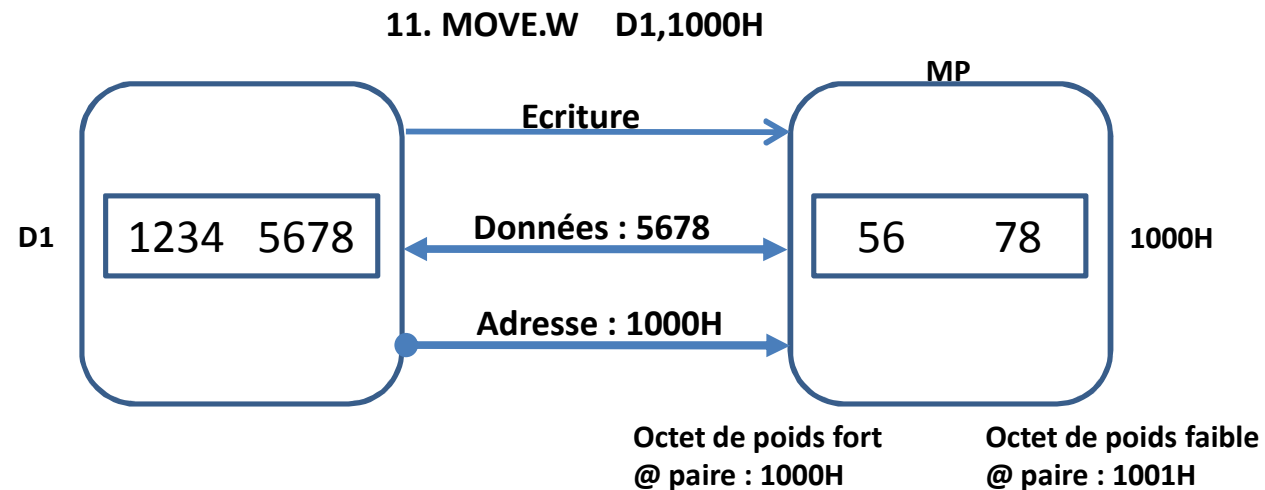
~~MOVE.W #55, 1001H~~
~~MOVE.L #55, 1001H~~

Taille de données transférées : .B : Byte : un octet / .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-18)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

B Adressage Direct : destination : registre de données

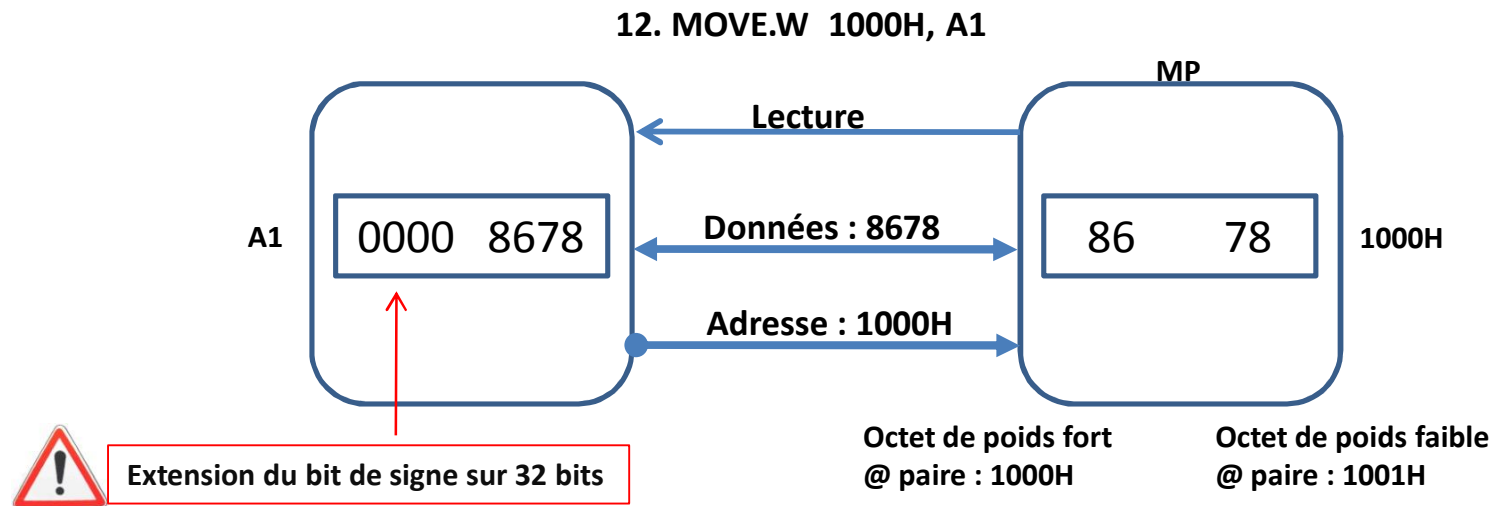


Taille de données transférées : .B : Byte : un octet / .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-19)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

B Adressage Direct : destination : registre d'adresse

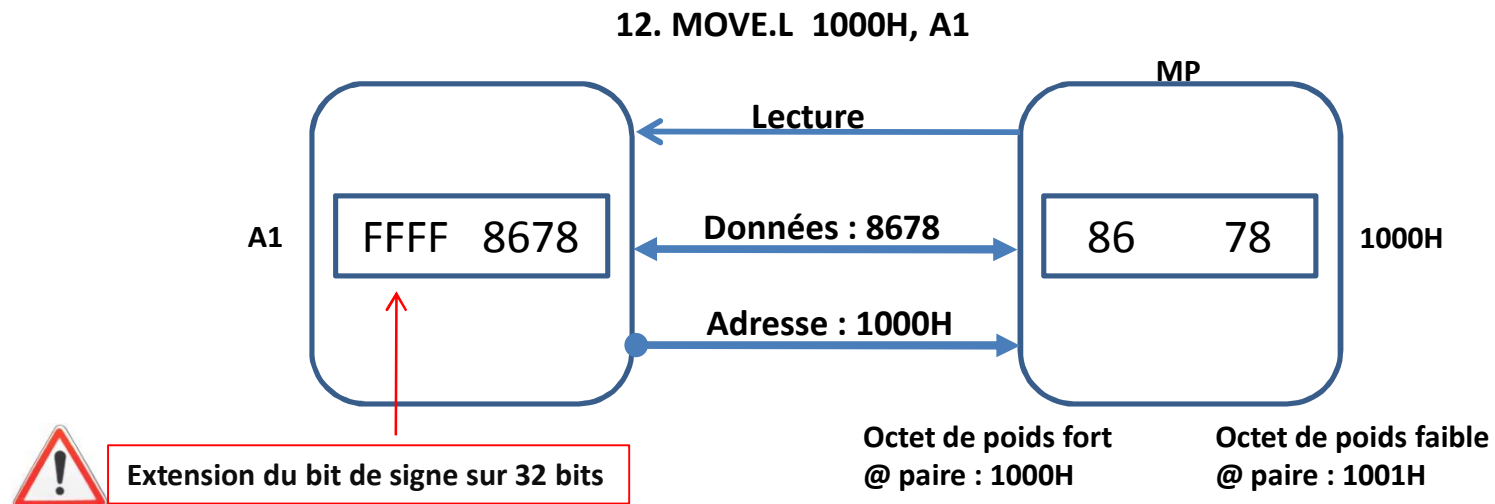


Taille de données transférées : .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits **UNIQUEMENT**

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-19)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

B Adressage Direct : destination : registre d'adresse



Taille de données transférées : .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits **UNIQUEMENT**

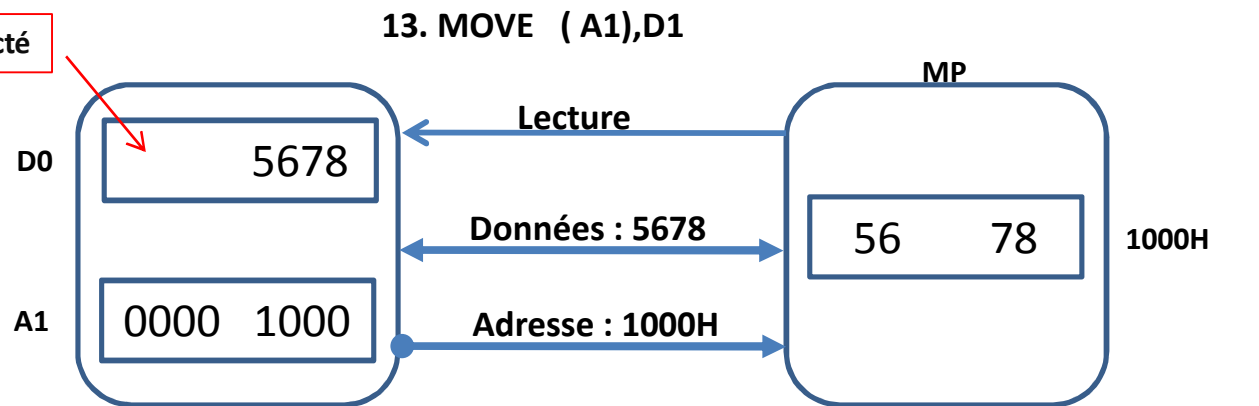
Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-20)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

B Adressage Indirect : Adressage Registre Indirect



Non affecté



Lecture par défaut d'un mot

Octet de poids fort
@ paire : 1000H

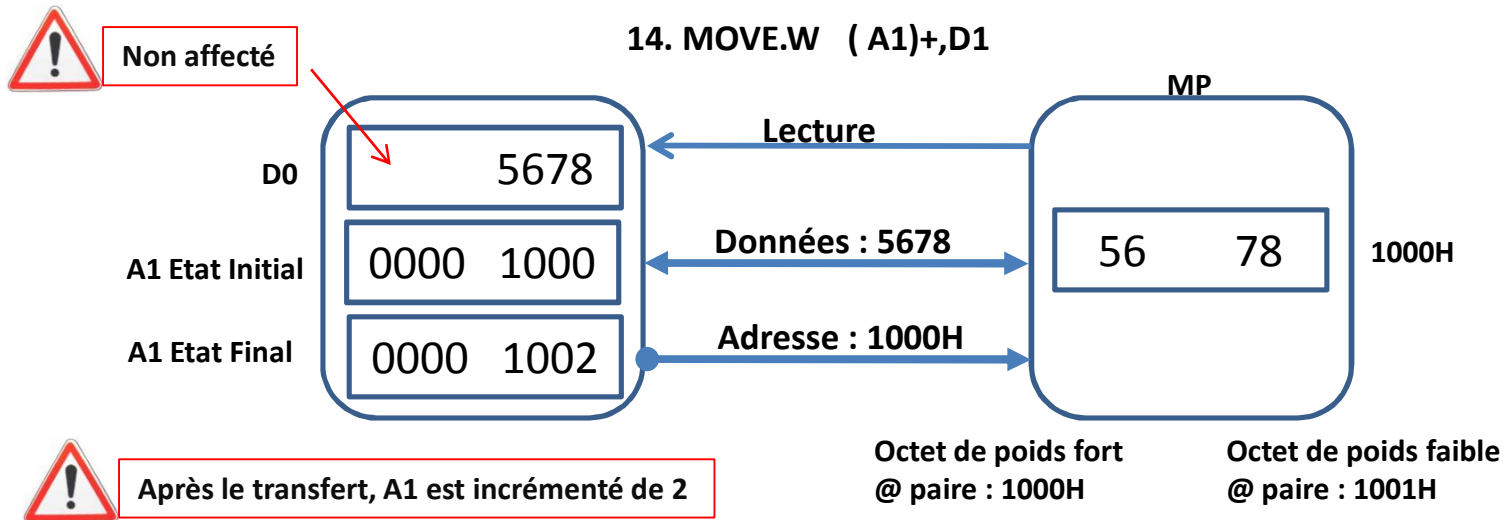
Octet de poids faible
@ paire : 1001H

Taille de données transférées : .B : Byte : un octet / .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-21)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

B Adressage Indirect : Adressage Registre Indirect Pos-incrémentation

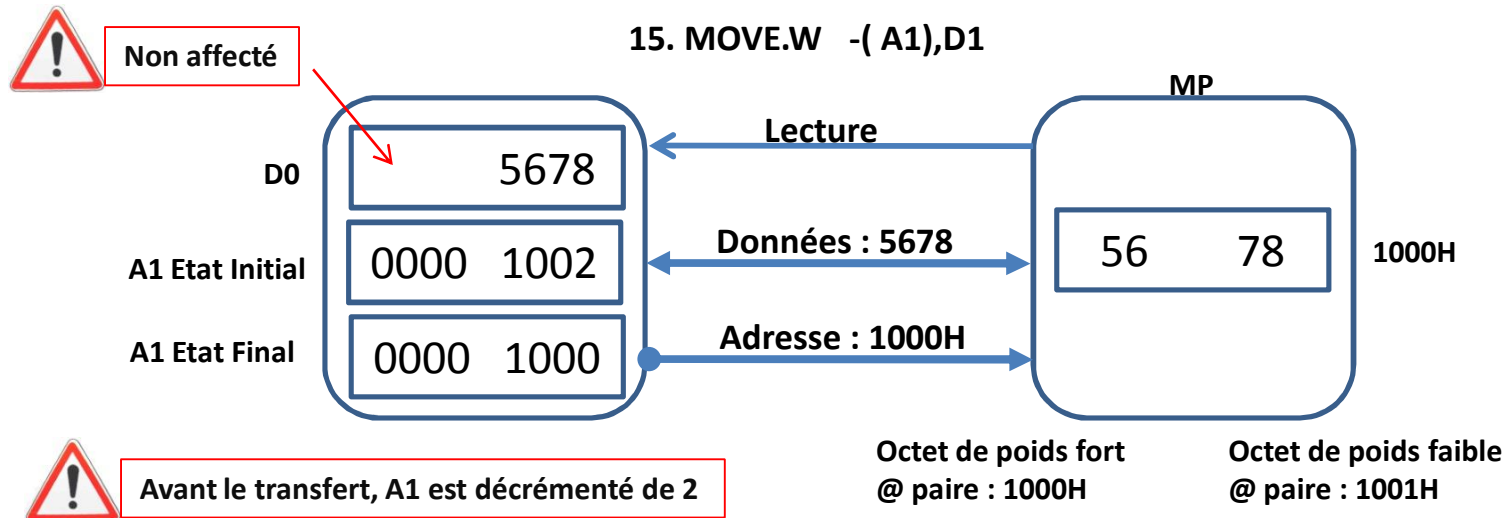


Taille de données transférées : .B : Byte : un octet / .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-22)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

B Adressage Indirect : Adressage Registre Indirect Pré-décrémentation

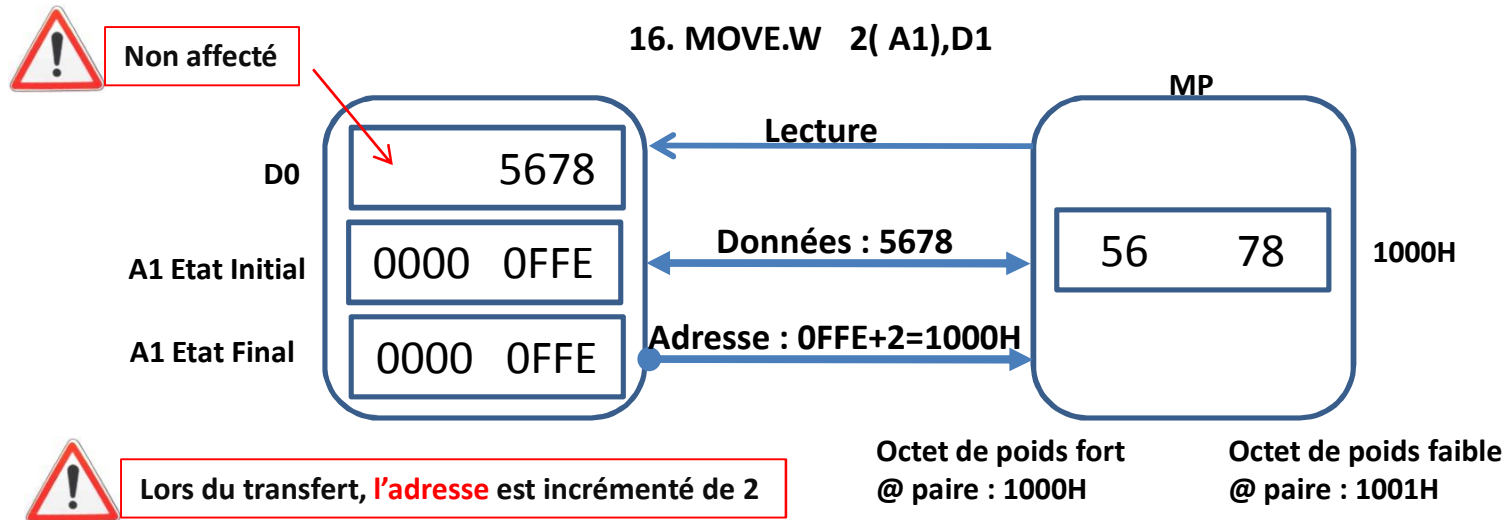


Taille de données transférées : .B : Byte : un octet / .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-23)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

B Adressage Indirect : Adressage Registre Indirect avec déplacement



Taille de données transférées : .B : Byte : un octet / .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-24)

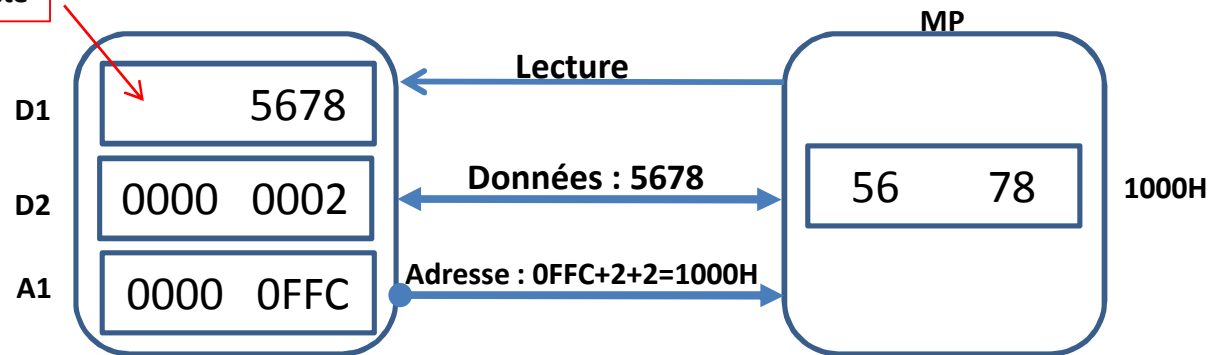
6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

B Adressage Indirect : Adressage Registre Indirect avec déplacement sur 8 bits signés et index court :



Non affecté

17. MOVE.W 2(A1,D2.W),D1



Lors du transfert, l'adresse est additionné avec le contenu de D2 puis incrémenté de 2

Octet de poids fort
@ paire : 1000H

Octet de poids faible
@ paire : 1001H

Taille de données transférées : .B : Byte : un octet / .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits

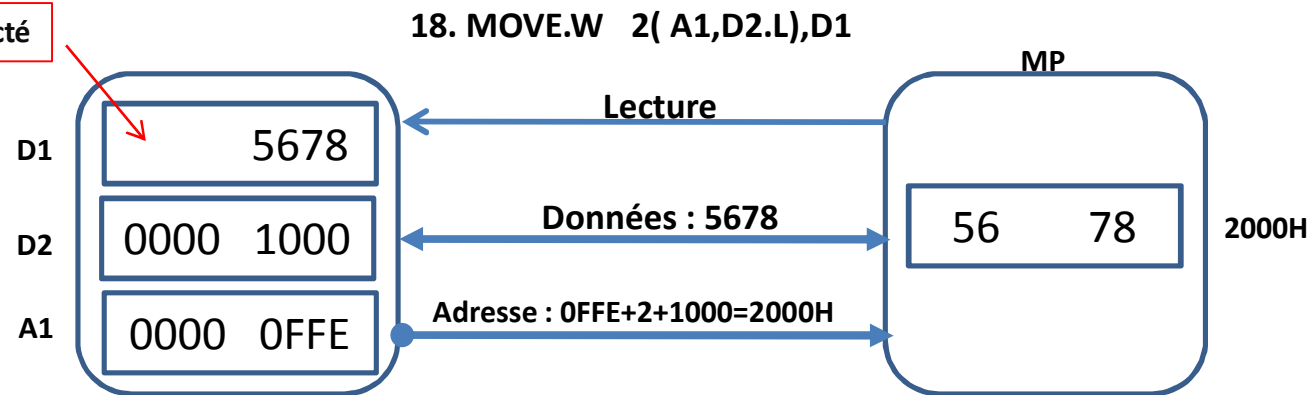
Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-25)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

B Adressage Indirect : Adressage Registre Indirect avec déplacement sur 8 bits signés et index long :



Non affecté



Lors du transfert, l'adresse est additionné avec le continue de D2 puis incrémenté de 2

Octet de poids fort
@ paire : 1000H

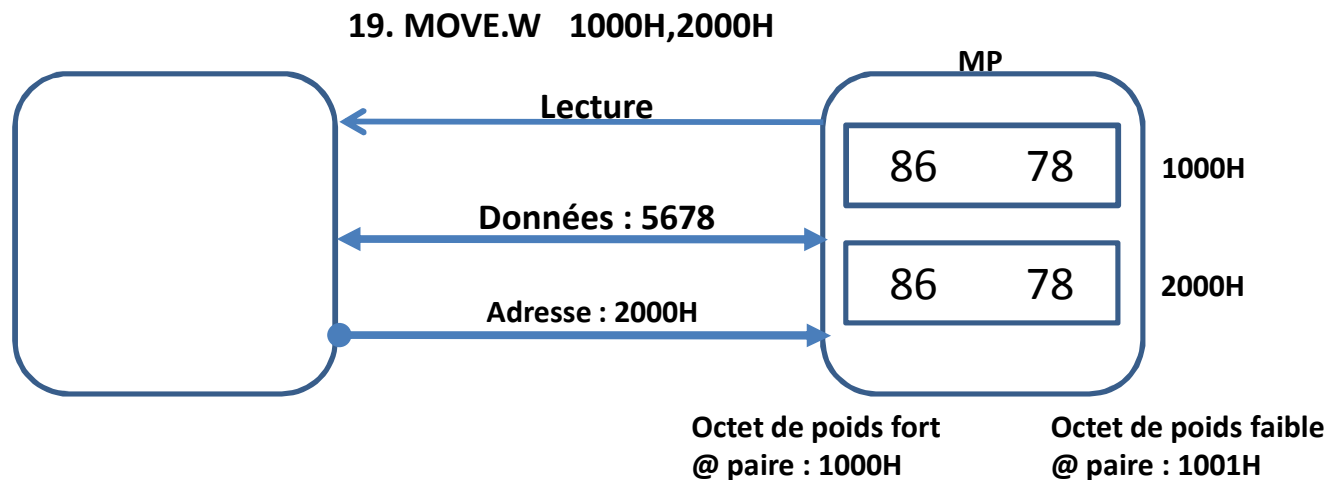
Octet de poids faible
@ paire : 1001H

Taille de données transférées : .B : Byte : un octet / .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-26)

6- Le microprocesseur MOTOROLA 68000 – Mode d'adressage

C Adressage Absolu: Adressage Absolu Court :



Taille de données transférées : .B : Byte : un octet / .W : Word : Mot de 16 bits / .L : Long word : Mot de 32 bits

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-27)

6- Le microprocesseur MOTOROLA 68000 – Jeu d'instructions – transfert de données

INSTRUCTION	TAILLE DE L'OPÉRANDE	OPÉRATION
EXG	32	R _x ← R _y
LEA	32	EA → A _n
LINK	—	A _n → SP @ – SP → A _n SP+d → SP
MOVE	8, 16, 32	(EA) _s → EA _d
MOVEM	16, 32	(EA) → A _n , D _n A _n , D _n → EA
MOVEP	16, 32	(EA) → D _n D _n → EA
MOVED	8	#xxx → D _n
PEA	32	EA → SP @ –
SWAP	32	D _n [31:16] ↔ D _n [15:0]
UNLK	—	A _n ← SP SP @ + → A _n

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-28)

6- Le microprocesseur MOTOROLA 68000 – Jeu d'instructions – Instruction arithmétique

INSTRUCTION	TAILLE DE L'OPÉRANDE	OPÉRATION
ADD	8, 16, 32	$DN + [AE] \rightarrow DN$ $[AE] + DN \rightarrow AE$ $[AE] + \# \text{ DONNÉE} \rightarrow AE$ $AN + [AE] \rightarrow AN$
ADX	8, 16, 32	$Dx + Dy + X \rightarrow Dx$ $[-(Ax)] + [-(Ay)] + X \rightarrow - [Ax]$
CLR	8, 16, 32	$0 \rightarrow AE$
CMP	8, 16, 32	$DN - [AE]$ $[AE] - \# \text{ DONNÉE}$ $[A \times 1 +] - [A_y + 1]$ $AN - [AE]$
DIVS	16, 32	$DN / [AE] \rightarrow DN$
DIVU	32 + 16	$DN / [AE] \rightarrow DN$
EXT	8 \rightarrow 16 16 \rightarrow 32	EXTENSION BIT 7 SUR BITS 8 A 15 EXTENSION BIT 15 SUR BITS 16 A 31
MULS	16 x 16 \rightarrow 32	$DN \times [AE] \rightarrow DN$
MULU	16 x 16 \rightarrow 32	$DN \times [AE] \rightarrow DN$
NEG	8, 16, 32	$0 - [AE] \rightarrow AE$
NEGX	8, 16, 32	$0 - [AE] - X \rightarrow AE$
SUB	8, 16, 32	$DN - [AE] \rightarrow DN$ $[AE] - DN \rightarrow AE$ $[AE] - \# \text{ DONNÉE} \rightarrow AE$ $AN - [AE] \rightarrow AN$
SUBX	16, 32 8, 16, 32	$Dx - Dy - X \rightarrow Dx$ $[-(Ax)] - [-(Ay)] - X \rightarrow - [Ax]$
TAS	8	$[AE] - 0$ et BIT 7 de AE = 1
TST	8, 16, 32	$[AE] - 0$

Architecture des ordinateurs :
Microprocesseur & jeu d'instructions ó (17-29)

6- Le microprocesseur MOTOROLA 68000 – Jeu d'instructions – Opérations décimales

INSTRUCTION	TAILLE DE L'OPÉRANDE	OPÉRATION
ABCD	8	$(Dz)_{10} + (Dy)_{10} + X \rightarrow Dz$ $-(Ax)_{10} + -(Ay)_{10} + X \rightarrow -(Az)$
SBCD	8	$(Dz)_{10} - (Dy)_{10} - X \rightarrow Dz$ $(Ax)_{10} - (Ay)_{10} - X \rightarrow -(Az)$
NBCD	8	$0 - (AE)_{10} - X \rightarrow AE$

Architecture des ordinateurs :
Microprocesseur & jeu d'instructions ó (17-30)

6- Le microprocesseur MOTOROLA 68000 – Jeu d'instructions – Opérations logiques

INSTRUCTION	TAILLE DE L'OPÉRANDE	OPÉRATION
AND	8, 16, 32	DN ET (AE) → DN (AE) ET DN → AE (AE) ET # DONNÉE → AE
OR	8, 16, 32	DN OU (AE) → DN (AE) OU DN → AE (AE) OU # DONNÉE → AE
EOR	8, 16, 32	(AE) OU exclusif Dy → AE (AE) OU exclusif # donnée → AE
NOT	8, 16, 32	COMPLÉMENT A UN DE (AE) → AE

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-31)

6- Le microprocesseur MOTOROLA 68000 – Jeu d'instructions – décalages & rotations

Instruction	TAILLE DE L'OPÉRANDE	Operation
ASL	B, 16, 32	
ASR	B, 16, 32	
LSL	B, 16, 32	
LSR	B, 16, 32	
ROL	B, 16, 32	
RO#	B, 16, 32	
ROXL	B, 16, 32	
ROXR	B, 16, 32	

Architecture des ordinateurs :
Microprocesseur & jeu d'instructions ó (17-32)

6- Le microprocesseur MOTOROLA 68000 – Jeu d'instructions – Test de bits

INSTRUCTION	TAILLE DE L'OPÉRANDE	OPÉRATION
BTST	8, 32	TEST D'UN BIT DE (AE) → Z
BSET	8, 32	TEST D'UN BIT DE (AE) → Z ET MISE A UN DE CE BIT
BCLR	8, 32	TEST D'UN BIT DE (AE) → Z ET MISE A ZÉRO DE CE BIT
BCHG	8, 32	TEST D'UN BIT DE (AE) → Z ET INVERSION DE CE BIT

Architecture des ordinateurs :
Microprocesseur & jeu d'instructions ó (17-33)

6- Le microprocesseur MOTOROLA 68000 – Jeu d'instructions – Contrôle de programme

INSTRUCTION	OPÉRATION
Bcc	BRANCHEMENT CONDITIONNEL · DÉPLACEMENT 8 OU 16 BITS
DBcc	TEST CONDITION, DÉCRÉMENTATION ET BRANCHEMENT DÉPLACEMENT 16 BITS
Scc	TEST CONDITION ET MISE A FF OU 00 DE L'OCTET
BRA	BRANCHEMENT · DÉPLACEMENT 8 OU 16 BITS
BSR	BRANCHEMENT SOUS-PROGRAMME DÉPLACEMENT 8 OU 16 BITS
JMP	SAUT
JSR	SAUT SOUS-PROGRAMME
RTR	RESTITUTION DES CODES CONDITIONS ET RETOUR
RTS	RETOUR DE SOUS-PROGRAMME

Architecture des ordinateurs : Microprocesseur & jeu d'instructions ó (17-34)

6- Le microprocesseur MOTOROLA 68000 – Jeu d'instructions – Contrôle système

INSTRUCTION	OPÉRATION
INSTRUCTIONS PRIVILÉGIÉES	
ANDI to SR	ET LOGIQUE AVEC LE REGISTRE D'ÉTAT
EORI to SR	OU EXCLUSIF LOGIQUE AVEC LE REGISTRE D'ÉTAT
ORI to SR	OU LOGIQUE AVEC LE REGISTRE D'ÉTAT
MOVE AE to SR	CHARGEMENT DU REGISTRE D'ÉTAT
MOVE USP	TRANSFERT DU POINTEUR DE PILE UTILISATEUR
RESET	INITIALISATION DES CIRCUITS EXTERNES
RTE	RETOUR D'EXCEPTION
STOP	ARRÊT DE L'EXÉCUTION DU PROGRAMME
GÉNÉRATIONS DE «TRAP»	
CHK	TEST DE LIMITES
TRAP	TRAP
TRAPV	TRAP SI DÉBOREMENT
REGISTRE CODES-CONDITION	
ANDI to CCR	ET LOGIQUE AVEC LE REGISTRE CODES-CONDITION
EORI to CCR	OU EXCLUSIF LOGIQUE AVEC LE REGISTRE CODES CONDITION
ORI to CCR	OU LOGIQUE AVEC LE REGISTRE CODES CONDITION
MOVE AE to CCR	CHARGEMENT DU REGISTRE CODES CONDITION
MOVE SR to AE	MÉMORISATION DU REGISTRE D'ÉTAT



Architecture des ordinateurs : BIBLIOGRAPHIE

1. Aho A, Concepts fondamentaux de l'informatique, Dunod, Paris, 1993
2. Tanenbaum A., Architecture des ordinateurs, InterEditions, Paris, 1996
3. Vieillfond C., Mise en œuvre du 68000, Sybex, 1984