

# INTRODUCTION

## 1 Définition d'un microprocesseur

---

On peut donner du terme microprocesseur la définition suivante : "Composant renfermant dans un seul boîtier l'unité de contrôle et l'unité de traitement d'une machine informatique de type VON NEUMANN"

Cette définition exclut volontairement :

- Les composants en plusieurs boîtiers (micros en tranches)
- Les architectures non VON NEUMANN (GAPP , Transputers)
- Les microcontrôleurs qui incluent d'autres composantes des systèmes informatiques (mémoire, entrées/sorties)

La vocation de ces composants est de constituer l'unité centrale de micro-ordinateurs et, à l'heure actuelle, de mini-ordinateurs. Toutefois leur prix raisonnable et leurs performances permettent leur utilisation dans des systèmes informatiques dédiés (concentrateurs de lignes, noeuds de réseaux, pilotes de robots etc).

## 2 Historique

---

L'histoire des microprocesseurs est intimement liée à celle de la technologie des semi-conducteurs dont voici les principales étapes :

- 1940 Invention du transistor
- 1950 Jack Saint CLAIR (TEXAS INSTRUMENTS) produit le 1er circuit intégré (CI)
- 1961 Début de la production en série des CI
- 1964 Intégration à petite échelle (SSI de 1 à 10 transistors)
- 1968 Intégration à moyenne échelle (MSI de 10 à 500 transistors)
- 1971 Intégration à grande échelle (LSI de 500 à 20 000 transistors)
- 1980 Intégration à très grande échelle (VLSI plus de 20 000 transistors)

Le premier microprocesseur a été fabriqué par INTEL en 1971. C'était un 4 bits baptisé 4004 destiné à équiper des calculatrices de bureau.

En 1972 INTEL produit le premier microprocesseur 8 bits baptisé 8008 par référence au précédent. Ce microprocesseur était destiné à répondre à un contrat de fabrication d'un terminal. En réalité le 8008 s'est avéré trop lent pour satisfaire au cahier des charges du terminal et INTEL a décidé de tenter le lancement de ce produit sur le marché grand public.

L'énorme succès de cette initiative fut à l'origine de la fabrication massive des microprocesseurs.

A la suite du succès du 8008, INTEL produisit dès 1974 le 8080 qui constituera le premier élément de la future famille de microprocesseurs de ce fabriquant.

En 1974, MOTOROLA, autre fondeur de silicium, décide de lancer le 6800 qui constituera lui aussi le début d'une grande famille.

Les années 70 voient alors apparaître de petites entreprises de fabrication de microprocesseurs souvent constituées par des transfuges des deux grandes compagnies.

On peut notamment citer MOSTEK avec son 6502 très inspiré du 6800 et ZILOG avec son Z80 qui constitue une amélioration technique du 8080 (augmentation du nombre de registres, simplification de l'alimentation...)

Les autres grands constructeurs (TEXAS INSTRUMENT, FAIRCHILD , RCA , SIGNETICS etc) hésitent encore à se lancer dans ce marché.

Ces composants seront à la base des premiers micro-ordinateurs tant personnels (TANDY avec son TRS80 muni d'un Z80 et le tout premier APPLE avec un 6502) que professionnels (EXORCISER de MOTOROLA et multiples constructeurs développant autour du format de carte S100 pour 8080 et Z80).

On peut remarquer que la conception même du composant avait été chez MOTOROLA plus guidée par des considérations informatiques qu'électroniques (possibilité de tests selon la représentation des nombres, sauvegarde automatique de contexte, distinction entre interruption matérielle et logicielle...). Tandis qu'INTEL et ZILOG avaient opté pour une approche plus matérielle (test de parité, pas de sauvegarde automatique de contexte pour prendre en compte les interruptions plus rapidement, rafraîchissement de mémoires dynamiques).

Paradoxalement la famille des 8 bits d'INTEL a bénéficié d'un grand effort de développement logiciel (écriture des systèmes d'exploitation CPM puis MPM et CCPM) alors que la famille MOTOROLA se voyait attribuer des systèmes d'exploitation plus "maison" et moins aptes à accueillir des progiciels.

Très vite les 8 bits commencent à atteindre la limite de leurs performances et les constructeurs se penchent sur de nouvelles solutions.

Les principaux reproches faits aux 8 bits sont :

- peu d'espace mémoire accessible (64K octets)
- peu de types d'informations manipulés (1 ou 2 octets)
- pas adaptés aux architectures multi-processeurs
- peu de modes d'adressage

Pour toutes ces raisons les constructeurs commencent à étudier une nouvelle génération de microprocesseurs. De sorte que l'on verra apparaître des microprocesseurs 8 bits plus puissants permettant de manipuler plus facilement des informations sur 16 bits et offrant de nouvelles possibilités d'adressage comme le 6809 de MOTOROLA ou le 8085 d'INTEL.

L'utilisation de plus en plus "informatique" des microprocesseurs (langages évolués , systèmes d'exploitation performants) a rapidement attiré les constructeurs vers des produits plus proches des unités centrales d'ordinateurs.

L'année 78 a vu donc apparaître sur le marché des microprocesseurs 16 bits offrant une capacité d'adressage de mémoire d'au moins 1M octets ( $2^{20}$ ), des performances plusieurs fois supérieures à celles des 8 bits, un large éventail de modes d'adressage (segments, bases, adresses relatives et, plus tard, mémoire virtuelle) et des possibilités plus importantes de manipulation de données (bit, chaîne de caractères, mots de 8, 16, 32 bits, quantités numériques en ASCII et en DCB...).

Force est de constater que l'idée première du petit composant à tout faire a fait place au microprocesseur unité centrale d'ordinateur.

Les deux grands constructeurs INTEL et MOTOROLA ont alors proposé les éléments de leur nouvelle génération de microprocesseurs.

## Microphotographie de la puce du 8080

# LES ANCÊTRES

Les microprocesseurs 4 et 8 bits font déjà partie du passé. Ils illustrent malgré tout les principes de fonctionnement de ce type de composants. Nous nous contenterons d'un petit tour d'horizon sous la forme des tableaux ci-après et de l'étude d'un microprocesseur de la 1<sup>ère</sup> génération le 6800 de MOTOROLA.

## 1 Les 4 bits

---

Constructeur Référence	INTEL 4004	INTEL 4040	ROCKWELL PPS4	FAIRCHILD PPS25	TEXAS INS TMS 1000
Nbre d'instr	46	60	50	95	petit
Temps d'addition entre regs en ms	8	8	5	3	15
Espace mémoire	4K	8K	4K	6,5K	1K
Registres d'usage général	16	24	4	1	4

Le 4004, apparu en 1970, intègre 2250 transistors. Il gère les appels de sous programmes par une pile interne à 4 niveaux.

Le 4040, datant de 1972, ajoute 4 niveaux à cette pile ainsi que la gestion des interruptions.

## 2 Les 8 bits

---

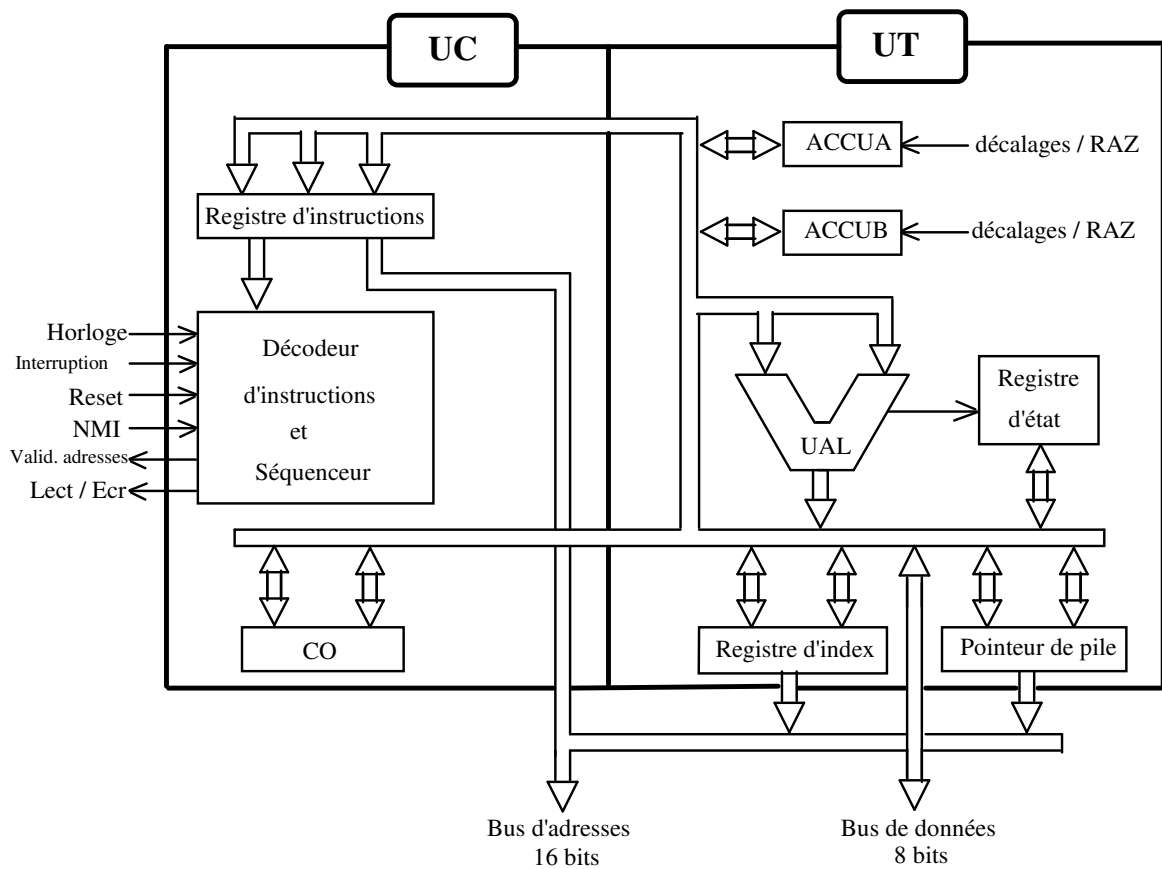
Constructeur Référence	INTEL 8008	INTEL 8080	INTEL 8085	MOTOROLA 6800
Nbre d'instr	48	69	71	71
Temps d'addition entre regs en ms	12,5 à 20	1,3 à 2	1,3	2
Espace mémoire	16K	64K	64K	64K
Registres d'usage général	7	7	7	3
Nombre de transistors	3300	4000	6200	
Horloge en MHz	0,3	2 2,67 ou 3,125	3,5 ou 6	1 1,5 ou 2
Année	1972	1974	1976	1974

Constructeur Référence	ZILOG Z80	MOSTEK 6502	ROCKWELL PPS8	NATIONAL SC/MP
Nbre d'instr	69	71	90	50
Temps d'addition entre regs en ms	1,6	2	4	5 à 25
Espace mémoire	64K	64K	32K	64K en pages de 4K
Regs généraux	17	3	3	6
Année	1976	1975		1976

L'architecture interne de ces microprocesseurs est très simple et directement calquée sur la structure VON NEUMANN. Nous nous contenterons d'analyser la structure de l'un d'entre eux.

### 3 Un exemple le 6800

#### 3.1 Architecture interne



### 3.2 Les registres

Le **compteur ordinal** (CO) sur 16 bits pointe sur l'instruction à traiter.

Le **pointeur de pile** (SP) sur 16 bits contient l'adresse du sommet de pile. Cette pile peut être placée n'importe où en mémoire.

Elle est utilisée de façon automatique pour sauvegarder l'adresse de retour lors de l'appel de sous programmes et pour sauvegarder le contexte (l'ensemble des registres) lors de la prise en compte d'une interruption.

Elle peut aussi être utilisée pour des sauvegardes temporaires ou des passages de paramètres grâce à des instructions spéciales (PUSH et PULL).

Le **registre d'index** (X) sur 16 bits est utilisé pour l'adressage indexé c'est à dire que l'adresse de l'opérande est obtenue en faisant la somme du contenu de ce registre avec un déplacement fixe placé dans l'instruction.

Les **accumulateurs** (A et B) sur 8 bits, ils servent d'opérandes à l'UAL et en réceptionnent les résultats. Ils permettent les opérations arithmétiques d'addition, de soustraction, d'incréméntation, de décréméntation et de changement de signe ainsi que les décalages et les opérations logiques (ET, OU, OU Exclusif, NON).

Le **registre de code de conditions** sur 8 bits contient les indicateurs de l'UAL (signe, résultat nul, débordement, retenue) ainsi qu'un bit indiquant si les interruptions doivent ou pas être prises en compte (masque d'interruptions).

### 3.3 Les opérandes

Le 6800 traite les opérandes sur 8 bits suivants :

- Entiers naturels (0 à 255)
- Entiers relatifs en représentation complément à 2 (-127 à 127)
- Décimaux en DCB sur 2 chiffres (8 bits)

Les opérandes sur 16 bits ne peuvent être traités que par "tranches" successives de 8 bits.

On peut toutefois noter que l'index X (sur 16 bits) peut subir des transferts vers et depuis 2 octets consécutifs en mémoire ainsi que des incrémentations, des décrémentations et des comparaisons d'égalité avec un mot constitué de 2 octets consécutifs en mémoire.

### 3.4 Les instructions

On peut les découper en 7 groupes :

#### - Arithmétiques

- |                       |                         |
|-----------------------|-------------------------|
| - Addition            | ADD et ADC <sup>1</sup> |
| - Soustraction        | SUB et SBC <sup>1</sup> |
| - Mise à zéro         | CLR                     |
| - Changement de signe | NEG                     |
| - Incréméntation      | INC                     |
| - Décréméntation      | DEC                     |
| - Ajustement décimal  | DAA                     |

Chacune de ces opérations peut porter sur l'un ou l'autre des accumulateurs (A ou B) ainsi que sur un opérande en mémoire à l'exception de l'ajustement décimal qui ne peut être effectué que sur l'accu A.

#### - Logiques

- Test de bit      BIT <sup>2</sup>

---

<sup>1</sup>Il existe outre les additions et soustractions habituelles (ADD et SUB) deux opérations (ADC et SBC) permettant de traiter le report de retenue lors de l'utilisation d'opérandes par "tranches" de 8 bits.

- Test de signe    TST
- Comparaison    CMP
- Non              COM
- Ou                ORA
- Et                AND
- Ou exclusif    EOR

Ces opérations, à l'exception de COM et TST qui peuvent traiter des opérandes en mémoire, ne peuvent porter que sur les accumulateurs A et B.

#### - Transferts

- Chargement d'un accu            LDAA    LDAB
- Rangement d'un accu            STAA    STAB
- Empilement d'un accu           PSHA    PSHB
- Dépilement d'un accu          PULA    PULB
- Transfert entre accus            TBA    TAB

#### - Décalages

- Logique à droite et à gauche    LSR    LSL
- Arithmétique à droite et à gauche ASR    ASL
- Circulaire à droite et à gauche    ROR    ROL

Chacune de ces opérations peut porter sur l'un des accus ainsi que sur un opérande en mémoire.

#### - Opérations sur l'index et le pointeur de pile

- Incrémentation            INX    INS
- Décrémenter            DEX    DES
- Chargement            LDX    LDS<sup>3</sup>
- Rangement            STX    STS<sup>2</sup>
- Transferts            TXS    TSX<sup>4</sup>
- Comparaison            CPX<sup>5</sup>

#### - Ruptures de séquence

- Branchements inconditionnels    BRA    JMP
- Branchements conditionnels :
  - si égal                    BEQ
  - si différent              BNE
  - si <                      BCS entre entiers naturels    BLT entre relatifs
  - si < ou =                BLS entre entiers naturels    BLE entre relatifs
  - si >                      BHI entre entiers naturels    BGT entre relatifs
  - si > ou =                BCC entre entiers naturels    BGE entre relatifs
  - si débordement            BVS

---

<sup>2</sup> BIT est un ET logique sans production de résultat CMP est une soustraction sans production de résultat TST se contente d'une comparaison à 0 (nul, signe) Ces instructions permettent de positionner les indicateurs du registre de code de conditions utilisés lors des ruptures de séquence conditionnelles (branchements conditionnels).

<sup>3</sup> Le chargement et le rangement en mémoire s'effectuent sur 2 octets consécutifs de la façon suivante : écriture / lecture de l'octet de fort poids à l'adresse désignée comme opérande puis écriture / lecture de l'octet de faible poids à l'adresse suivante.

<sup>4</sup> Le transfert entre X et SP n'est pas direct mais tient compte du fait que SP désigne le sommet de pile sous la forme de la première place disponible et non de la dernière occupée ainsi : TSX transfère SP+1 dans X et TXS transfère X-1 dans SP.

<sup>5</sup> La comparaison n'existe pas sur SP et celle qui existe sur X ne permet que de tester l'égalité.

si non débordement	BVC	
si positif	BPL	
si négatif	BMI	
- Appel de sous programme	BSR	JSR
- Retour de sous programme	RTS	
- Appel d'interruption logicielle	SWI <sup>6</sup>	
- Retour d'interruption	RTI	
- Attente d'interruption	WAI <sup>7</sup>	

#### - Opérations sur le registre code condition

- Mise à 0/1 de la retenue	CLC	SEC
- Mise à 0/1 du débordement	CLV	SEV
- Masquage/démasquage des interruptions	SEI	CLI
- Transferts avec l'accu A	TAP	TPA

### 3.5 La mémoire et l'adressage

Le bus d'adresses est de 16 bits donc l'espace mémoire ne peut dépasser 64K octets. Le 6800 possède les modes d'adressage suivants :

- **inhérent** : L'instruction n'a pas d'opérande (RTS) ou bien l'opérande est dans un registre (CLRA)
- **immédiat** : La valeur de l'opérande est dans l'instruction (ex : ADDA #3 qui ajoute 3 à l'accumulateur A)
- **direct et étendu** : L'opérande est désigné par son adresse en mémoire soit sur 8 bits (direct) soit sur 16 bits (étendu).
- **indexé** : L'opérande est désigné par une adresse obtenue par addition du contenu du registre d'index X et de la valeur placée dans l'instruction. Le déplacement est fixe et constitué d'un entier naturel sur 8 bits (0 à 255).
- **relatif** : Ce mode d'adressage est réservé aux instructions de rupture de séquence. L'instruction contient un déplacement fixe constitué d'un entier relatif sur 8 bits qui sera ajouté au contenu du compteur ordinal (CO) pour calculer l'adresse de la prochaine instruction. Le déplacement ne peut dépasser les limites de + à -127.

### 3.6 Quelques remarques

#### 1°) L'Horloge

L'horloge de séquençement du 6800 est de 1MHz (des modèles ultérieurs 68A00 et 68B00 supportent des horloges de 1,5 et 2MHz).

Toute instruction dure au moins 2 périodes d'horloge et aucune ne dépasse 12 périodes (la plus grande partie des instructions se situant entre 2 et 6 périodes).

Contrairement aux microprocesseurs d'INTEL et ZILOG le fonctionnement du 6800 est biphasé c'est à dire que pendant la 1<sup>ère</sup> moitié de la période d'horloge s'effectuent les opérations internes (décodage de l'instruction,

<sup>6</sup> Le 6800 possède 2 lignes d'interruptions matérielles :  
 IRQ qui peut être masquée par un bit du registre de code de conditions  
 NMI qui ne peut pas être ignorée  
 et une interruption logicielle : SWI

<sup>7</sup> L'instruction WAI provoque l'arrêt du processeur jusqu'à ce qu'arrive une interruption physique.



UAL, transferts de registres) et pendant la 2<sup>ème</sup> moitié les échanges avec la mémoire. Le fonctionnement est totalement synchrone c'est à dire que la mémoire doit pouvoir "répondre" pendant la demi- période qui lui est allouée et il n'existe aucun moyen simple de faire attendre le microprocesseur si la mémoire est trop lente.

## 2°) Les Entrés/Sorties

Le 6800 ne possède aucune instruction spécifique d'entrée/sortie (contrairement aux 8080 et Z80 qui ont des instructions IN et OUT). Les registres des contrôleurs de périphériques apparaissent comme des zones de mémoire classiques et peuvent être traités comme des opérandes par toutes les instructions.

# LES MICROPROCESSEURS ACTUELS

## 1 Présentation générale

---

La génération actuelle de microprocesseurs a su tirer les leçons des problèmes rencontrés par les utilisateurs de 8 bits entre 1974 et 1980. On peut dégager deux grands axes d'évolution des microprocesseurs :

### Evolution logicielle

Les nouveaux microprocesseurs sont dotés d'instructions et surtout de modes d'adressage permettant l'utilisation de langages évolués et la réalisation de systèmes d'exploitation multitâches modernes :

- Instructions permettant le passage de paramètres
- Instructions permettant la manipulation de données complexes ( tableaux, chaînes de caractères...)
- Mémoire virtuelle
- Niveaux de privilèges et protection
- Adressage de mémoire par base, segments, index avec ou sans indirection
- Instructions de changement de tâche

### Evolution matérielle

Le degré d'intégration plus élevé permet d'avoir plus de registres et des unités de traitement plus puissantes. Cependant la principale évolution technique concerne la vitesse de traitement :

- Technologies rapides permettant des horloges dépassant les 100 MHz
- Parallélisme de fonctionnement
- Intégration des antémémoires
- Prédiction d'instructions
- Gestion de co-processeurs et de multi-processeurs

La période 1978-1993 se caractérise par une mise à profit des progrès des technologies des circuits intégrés (technologie CISC<sup>8</sup>) tandis que la période suivante met en évidence la nécessité de remettre en question des principes même de ces processeurs (technologie RISC<sup>9</sup>).

Nous allons maintenant essayer de décrire les membres des deux grandes familles de microprocesseurs de la nouvelle génération qui se partagent à l'heure actuelle le marché (INTEL et MOTOROLA).

Dans un premier temps, INTEL proposera les successeurs du 8080 : 8086, 80286, 80386, 80486 et Pentium qui sont une évolution logique du 8080 (passage à 16 puis 32 bits, gestion de mémoire virtuelle et paginée, intégration d'antémémoires, mise en place de protections et d'une arithmétique réelle) puis, l'apparition du P6, constitue un passage à la technologie RISC rendu difficile par la nécessité de conserver la compatibilité au niveau du code avec les processeurs antérieurs. INTEL sera donc conduit, à partir de cette époque à intégrer un compilateur permettant la traduction des instructions de type 80x86 en instructions RISC exécutées par le processeur. De plus INTEL introduira des instructions issues des processeurs de signal (DSP) de façon à accélérer les traitements d'image et de sons imposés par les environnements multimédia.

MOTOROLA connaîtra le même type d'évolution issue du 6800 : 68000, 68010, 68020, 68030, 68040 et 68060 (passage à 16 puis à 32 bits, gestion de mémoire virtuelle et paginée, intégration d'antémémoires,

---

<sup>8</sup> CISC Complex Instruction Set Computer, processeur doté d'un jeu d'instructions complexes tentant de se rapprocher des instructions des langages évolués.

<sup>9</sup> RISC Reduced Instruction Set Computer processeur à jeu d'instruction réduit. On cherche à réaliser des instructions simples pouvant être exécutées très rapidement et se prêtant facilement à une exécution en parallèle ou en pipe line.

mise en place de protections et d'une arithmétique réelle) puis, la définition de la famille Power PC, constitue un passage à la technologie RISC sans souci de compatibilité avec les processeurs antérieurs. Cette rupture de compatibilité donnera, au début, à MOTOROLA une avance sensible en terme de performance.

## 2 La famille INTEL

---

Elle est constituée de 6 generations :

- Le 8086 avec sa version 8 bits le 8088 et son "grand frère" le 80186 ayant lui aussi sa version 8 bits le 80188
- Le 80286 qui ajoute au 8086 une gestion de mémoire virtuelle et de tâches
- Le 80386 le premier 32 bits de la famille possédant aussi une gestion de mémoire virtuelle et de tâches.
- le 80486 qui ajoute au 80386 une arithmétique réelle intégrée et un contrôleur d'antémémoire.
- Le P5 (Pentium) qui ajoute au 80486 un traitement en parallèle de certaines instructions.
- Le P6 (Pentium Pro) est un processeur RISC doté d'un compilateur câblé pour transformer les instructions de type 80x86 en instructions RISC. Le Klamath ou Pentium II connu selon les modèles sous plusieurs appellations (Deschutes, Celeron, Xeon, Katmai ...).

Des versions MMX, c'est à dire intégrant des instructions de type traitement de signal, des processeurs P5 et P6 sont proposées à partir de 1997 (Tillamook).

Les générations futures (Merced puis McKinley) de processeurs 64 bits devraient apparaître à l'horizon 2000 en collaboration avec HP (architecture IA-64).

Le tableau suivant établit une comparaison entre les principaux éléments de cette famille :

Modèle	8086	80286	80386	80486	Pentium (P5)	Pentium Pro (P6)	Pentium II
Année	1978	1982	1985	1989	1993	1995	1997
Nbre de transistors	29 000	134 000	275 000	1,2 M à 1,6 M	3,1 M à 4,4 M	5,5 M	7,5 M
Performance	0,3	1,2	5 à 10	20 à 50	80 à 280	250 à 350	380 à 510
Unités d'exécution	1	1	1	1	3	5	5

Cependant, à l'intérieur de ces grands types sont apparus des quantités importantes de variantes (surtout pour les 80386 et 80486). Le tableau suivant tente de situer la plupart de ces processeurs :

## Les principaux éléments de la famille INTEL et compatibles

Processeur	Horloge (MHz)	Bus de données	Données internes	Bus d'adresses	Adresse virtuelle	Anté-mémoire	Arith. réelle
8086	4 5 8 10 12	16 bits	16 bits	20 bits	Non	Non	Ext.
80286	6 8 10 12 16 20 25	16	16	24	30 bits	Non	Ext.
80386 SX	16 20 25 33 40	16	32	24	46	Non	Ext.
80386 SL	20 25 16 33	16	32	24	46	Non	Ext.
80386 SLC	16 20 25	16	32	24	46	8K	Ext.
80386 DX	12 16 20 25 33 40	32	32	32	46	Non	Ext.
80486 SLC	16 20 25 33 40	16	32	24	46	1 ou 16K	Ext.
80486 SLC/e	20 25	16	32	24	46	1K	Ext.
80486 SLC2	16/32 <sup>(10)</sup> 20/40 25/50 33/66 40/80	16	32	24	46	16K	Ext.
80486 SL	25 33	32	32	32	46	8K	Oui
80486 DLC	25 33 40	32	32	32	46	1K ou 16K	Ext.
80486 SX	16 20 25 33 40	32	32	32	46	8K	Ext.
80486 SX2	25/50 <sup>(1)</sup> 33/66	32	32	32	46	8K	Ext.
80486 DX	20 25 33 40 50	32	32	32	46	8K	Oui
80486 DX2	20/40 <sup>(1)</sup> 25/50 33/66 40/80	32	32	32	46	8K	Oui
80486DX4	25/75 <sup>(1)</sup> 33/99 40/120	32	32	32	46	8 ou 16K	Oui
Am5x86	33/133 40/160	32	32	32	46	16K	Oui
Overdrive	16/33 <sup>(1)</sup> 20/40 25/50 33/66	32	32	32	46	8K	Oui
Pentium (P5 ou P54C)	60 66 75 90 100 120 133 150 166 200	64	32	32	46	8K (code) + 8K (données)	Oui
Pentium Pro (P6)	120 133 150 166 180 200 233	64	64	32	46	8K + 8K + 256 ou 512K	Oui
Pentium MMX (P55)	166 200 233 266	64	32	36	46	16K (code) + 16K (données)	Oui + MMX
Pentium II	233 266 300 350 400	64	32	36	46	16K + 16K + 512 Ko	Oui+ MMX
Merced	500 1000	64	64	36	46	8K + 8K + 256K	Oui + MMX

<sup>10</sup> Ces processeurs divisent par 2 ou 3 leur fréquence d'horloge lors des accès à la mémoire.

### 3 La famille MOTOROLA

---

#### 1°) La famille des processeurs CISC (famille 680xx)

Elle est constituée de 5 générations :

- Le 68000 le 16 bits et sa version 8 bits le 68008.
- Le 68010 qui ajoute au 68000 des signaux de gestion de mémoire virtuelle et sa version étendue en adressage: le 68012.
- Le 68020 le 32 bits qui, outre les signaux de gestion de mémoire virtuelle, intègre une antémémoire pour les instructions.
- Le 68030 le 32 bits qui, outre les signaux de mémoire virtuelle, intègre un dispositif de pagination ainsi qu'une antémémoire pour les instructions et une autre pour les données.
- Le 68040 qui intègre l'unité de calcul en réels virgule flottante.
- Le 68060 qui parallélise l'exécution de certaines instructions

TYPE	HORLOGE (Fréquence)	Cycles UAL/instr	MIPS	MFLOPS	1er échantillon
68000	8 - 16,7	5,2	0,7 - 1,3		1979
68010	8 - 12,5	5,1	0,8 - 1,2		1982
68020	12,5 - 33,3	3,3	2 - 5,5	0,1 - 0,25	1984
68030	16,6 - 50	2,6	5 - 12	0,25 - 0,5	1987
68040	25 - 40	1,3	19 - 29	3,5 - 5,6	1989

Remarque : Le 6809, souvent considéré comme le premier élément de la nouvelle génération de microprocesseurs de MOTOROLA, n'est en réalité, qu'une évolution du 6800 :

- permettant le traitement d'opérandes sur 16 bits
- possédant un registre d'index supplémentaire et un pointeur de pile de l'utilisateur.
- permettant les adressages indirect et relatif au compteur ordinal ainsi que la pré-décrémentation et la post-incrémentation d'index.

Il est environ 3,5 fois plus performant que le 6800 mais ne constitue tout de même pas un vrai 16 bits (il est plus comparable au 8088 qu'au 8086).

#### 2°) La famille des processeurs RISC (famille Power PC)

L'architecture Power PC développée par MOTOROLA et IBM n'assure aucune compatibilité avec les 68xxx. Le choix a été délibérément fait de rompre cette filiation et de réaliser des processeurs RISC.

Toutefois afin que cette étude des principaux microprocesseurs soit la plus complète possible le Power PC sera étudié en fin de ce chapitre.

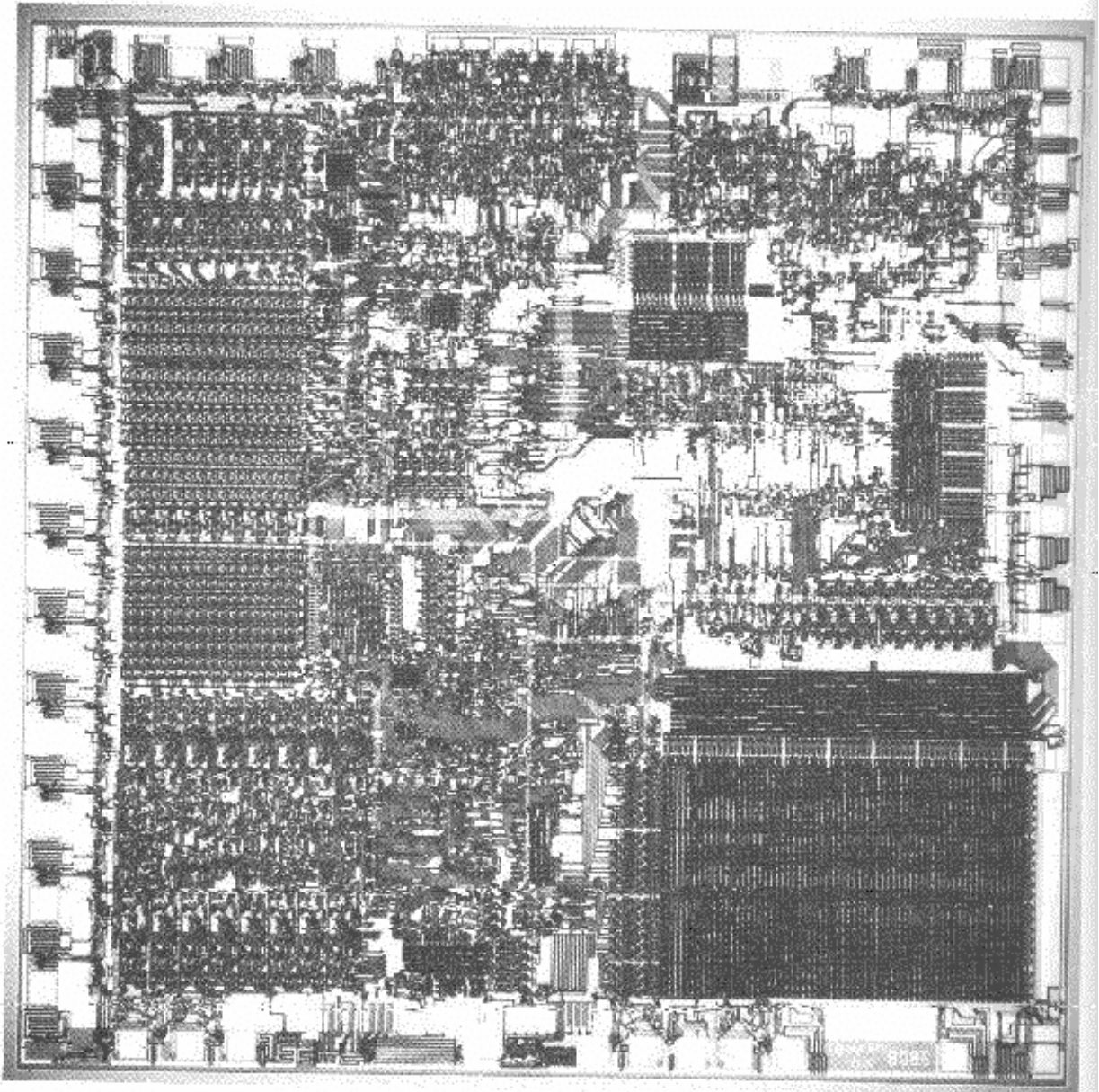
On y trouve 5 générations de processeurs :

- La première génération est constitué par les MPC 601 et MPC 602.
- La deuxième génération est constituée par les MPC 603, 603e, 604, 604e et 620.
- La troisième génération (G3) en 1998.
- La quatrième génération (G4) en 1999.
- Enfin, la cinquième génération, appelée 2K en 2001.

### Les principaux microprocesseurs MOTOROLA

Processeur	Année	Horloge (Mhz)	Bus de données (bits)	Données internes (bits)	Bus d'adresses (bits)	Adresse virtuelle (bits)	Anté-mémoire (octets)	Arithm. réelle
68000	1979	8 10 12,5 16	16	32	24	Non	Non	Ext.
68010	1982	8 10 12,5	16	32	24	Non	Non	Ext.
68012		8 10 12,5	32	32	30	Non	Non	Ext.
68020	1984	12,5 16,7 20 25 33	32	32	32	Non	64	Ext.
68030	1987	16,7 20 25 33 50	32	32	32	Non	64 + 64	Ext.
68040	1989	25 33 50	32	32	32	Non	4K + 4K	Oui
68LC040	1991	25 33 40	32	32	32	Non	64 + 64	Ext
68060	1994	50 66	32	32	32	Non	8K + 8K	Oui
Power PC 601 602	1994	50 60 66 80	64	32	32	52	32K	Oui
Power PC 603	1994	50 60 66 80	64	32	32	52	8K+8K	Oui
Power PC 603e	1994	100 200 240	64	32	32	52	16K+16K	Oui
Power PC 604	1995	75 100 120 133	64	32	32	52	16K+16K	Oui
Power PC 604e	1995	75 100 120 133	64	32	32	52	32K+32K	Oui
Power PC 620	1997	80 130 133	128	64	40	80	32K+32K	Oui
G3	1998	250	64	32	32	52	32K + 32K	Oui
G4	1999	500		32 et 64				Oui
2K (G5)	2001	1000		64				Oui

**Microphotographie de la puce du 8086.**



# LES MICROPROCESSEURS INTEL

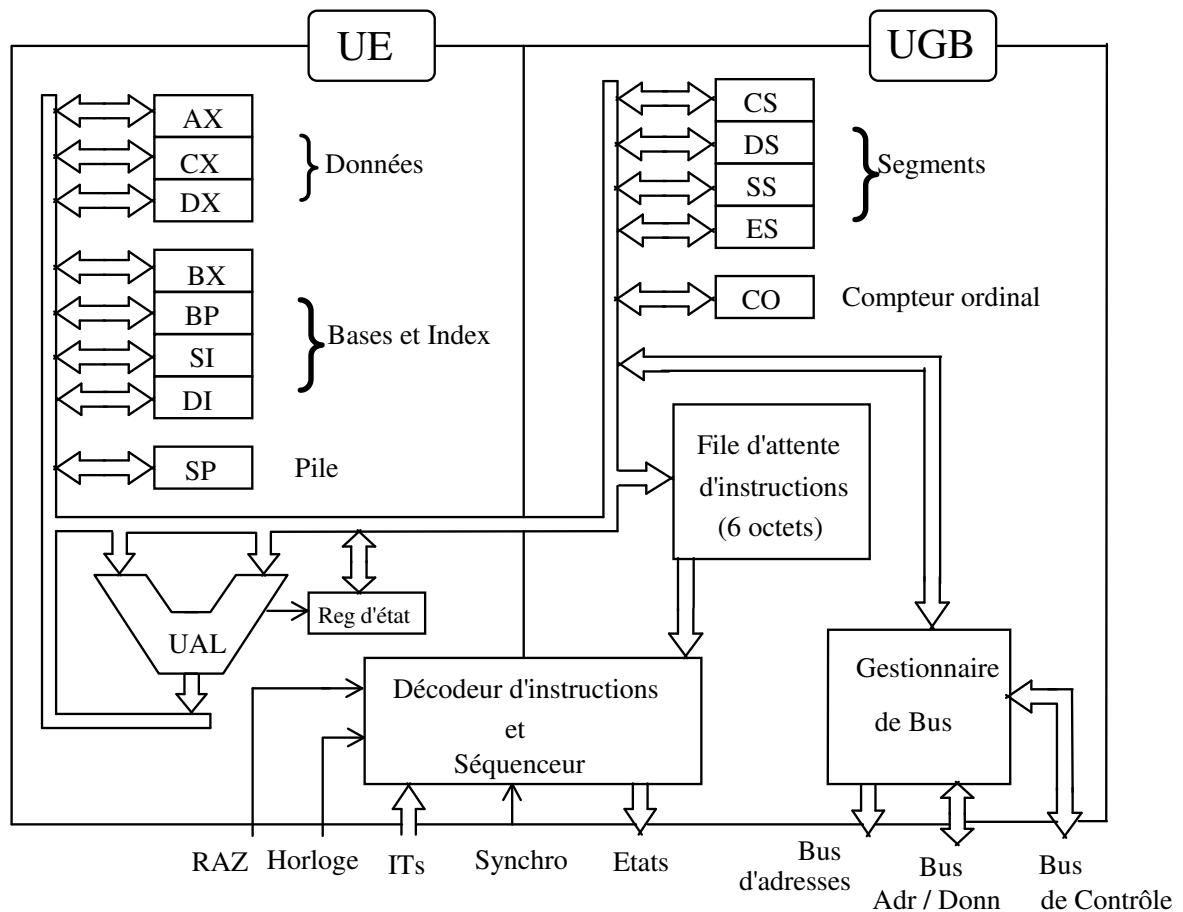
## 1 Les 8086 / 8088 / 80186 et 80188

### 1.1 Architecture interne

Le 8086 est constitué de 2 unités :

- L'unité de gestion des bus **UGB**
- L'unité d'exécution **UE**

Il peut être représenté par le schéma suivant :



**Erreur ! Les arguments du commutateur ne sont pas spécifiés.**L'UGB se charge :

- de la recherche des instructions et de leur mise en file d'attente
- de la lecture et l'écriture des opérandes
- du calcul des adresses en mémoire
- du contrôle physique de la mémoire

Elle fonctionne en parallélisme total avec l'UE.

L'UE puise les instructions dans la file d'attente et utilise l'UGB pour obtenir ses opérandes depuis la mémoire et pour y ranger ses résultats. Tous les problèmes d'adressage sont donc gérés par l'UGB.



L'UGB divise la mémoire en 4 segments de 64K octets chacun désigné par l'un des registres de segment CS , DS , SS et ES.

**CS** désigne le segment contenant le code du programme c'est donc dans ce segment que sont recherchées les instructions

**DS** désigne le segment contenant les opérandes

**SS** désigne le segment associé à la pile dont le sommet est pointé par le registre SP. La pile est utilisée pour les adresses de retour et les paramètres de sous programme ainsi que pour sauvegarder et restituer les contextes lors d'interruptions.

**ES** désigne un segment supplémentaire de données.

Ces 4 segments peuvent se chevaucher et même être confondus. Les opérandes et les instructions sont référencées par un déplacement à l'intérieur de l'un de ces 4 segments. L'adresse réelle en mémoire est obtenue en ajoutant à ce déplacement le contenu du registre de segment multiplié par 16. Cet artifice permet d'accéder à 1M octet ( $2^{20}$ ) de mémoire.

La mémoire est accessible octet par octet ou par mots de 16 bits. Toutefois l'accès à un mot de 16 bits se fait toujours en 2 transferts sur les 8088 et 80188 mais aussi sur les 8086 et 80186 lorsque l'adresse du mot est impaire. Il est donc utile, sur ces 2 derniers microprocesseurs, de veiller à "l'alignement" en adresses paires des opérandes 16 bits si l'on ne veut pas voir les performances chuter de façon importante.

Le 8088 ne diffère du 8086 que par les points suivants :

- La mémoire est organisée en octets et les accès aux opérandes 16 bits se font en 2 fois celle que soit l'adresse.
- La file d'attente d'instructions est composée de 4 octets au lieu de 6.
- Le nombre d'accès à la mémoire étant presque multiplié par 2 les performances sont bien inférieures à celles du 8086

Le 80186 possède outre l'architecture interne du 8086 les éléments suivants :

- Un générateur d'horloge
- 2 canaux DMA (Direct Memory Acces) permettant de faire des transferts de mémoire à mémoire ou de mémoire à périphérique de façon totalement automatique.
- Un encodeur de priorité acceptant 4 lignes d'interruption (INT0 à INT3)
- 3 compteurs programmables autonomes (TIMERS) permettant la génération d'horloges ou de délais (watchdog par exemple).
- 6 lignes programmables de pré-décodage de boîtiers de mémoire permettant de simplifier la circuiterie externe.
- 7 lignes de pré-décodage de contrôleurs de périphériques programmables offrant les mêmes avantages que les précédentes.
- Un générateur programmable de cycles d'attente pour les mémoires et les contrôleurs de périphériques lents.

Les différences entre le 80188 et le 80186 sont les mêmes que celles décrites entre le 8088 et le 8086.

## **1.2 Registres**

Les 8086 et 8088 possèdent 14 registres de 16 bits :

**AX** accumulateur accessible en 2 fois 8 bits AH et AL il sert aux opérations arithmétiques et d'entrées/sorties

**DX** registre de données accessible en 2 fois 8 bits DH et DL, il est utilisé comme AX et pour les adresses d'entrées/sorties.

**CX** compteur accessible en 2 fois 8 bits CH et CL, il sert de comptage lors des instructions répétitives.

**BX** base accessible en 2 fois 8 bits BH et BL, il est registre de base dans le segment de données (DS).

**BP** registre de base dans le segment de pile (SS).

**SI** index de préférence associé au segment de données (DS).

**DI** index de préférence associé au segment de données supplémentaire (ES).

**SP** pointeur de pile associé au segment de pile (SS).

**CS** registre de segment de code.

**DS** registre de segment de données.

**SS** registre de segment de pile.

**ES** registre de segment de données supplémentaire.

**CO** compteur ordinal associé au segment de code (CS)

**SR** registre d'état contenant les indicateurs de l'UAL (signe, retenue, débordement, résultat nul, parité) ainsi que les bits associés aux interruptions et aux instructions répétitives.

Les 80186 et 80188 possèdent en outre :

- 2 fois 6 registres associés aux 2 canaux DMA
- 3 fois 4 registres associés aux 3 compteurs programmables (TIMERS)
- 5 registres associés aux pré-décodages et aux générateurs de cycles d'attente
- 16 registres associés au contrôleur d'interruptions
- Il faut ajouter à cette liste un registre permettant d'adresser les précédents.

### **1.3 Les opérandes**

Les 8086 , 8088 , 80186 et 80188 traitent les opérandes suivants :

- Entiers naturels sur 8 et 16 bits
- Entiers relatifs sur 8 et 16 bits en complément à 2
- Caractères en ASCII sur 8 bits
- Chaînes de caractères de 1 à 64K codes ASCII
- Décimaux en représentation DCB sur 8 bits (1 ou 2 chiffres)
- Pointeurs constitués soit d'un déplacement sur 16 bits soit, sur 32 bits, d'un segment et d'un déplacement.

### **1.4 Les instructions**

Les 8086 et 8088 possèdent les instructions suivantes :

#### **- Arithmétiques**

- |                  |  |
|------------------|--|
| - Addition       | <b>ADD</b> et <b>ADC</b> sur 8 et 16 bits  |
| - Soustraction   | <b>SUB</b> et <b>SBB</b> sur 8 et 16 bits  |
| - Multiplication | <b>MUL</b> et <b>IMUL</b> sur 8 et 16 bits avec ou sans signe                    |
| - Division       | <b>DIV</b> et <b>IDIV</b> sur 16 et 32 bits sans signe et sur 16 bits avec signe |
| - Incrémentation | <b>INC</b> sur 8 et 16 bits  |

- Décrémentation            **DEC** sur 8 et 16 bits
- Changement de signe      **NEG** sur 8 et 16 bits
- Correction décimale        **AAA DAA AAS DAS AAM AAD** pour les 4 opérations
- Extension de signe         **CBW** et **CWD** sur 16 et 32 bits

#### - Logiques

- Comparaison                **CMP**
- Test de bit                 **TEST**
- Non                         **NOT**
- Ou                         **OR**
- Et                         **AND**
- Ou exclusif                **XOR**

Toutes ces opérations portent sur des opérands 8 ou 16 bits. On se référera au chapitre 3.3.4 pour plus de détail sur les opérations TEST et CMP.

#### - Transferts

- Chargement d'un registre            **MOV** 8 ou 16 bits selon le registre
- Rangement d'un registre            **MOV** 8 ou 16 bits selon le registre
- Empilement de 16 bits                **PUSH**
- Dépilement de 16 bits                **POP**
- Echange entre registres et mémoire      **XCHG** 8 ou 16 bits selon le registre
- Lecture d'un caractère dans une chaîne      **XLAT** (le début de la chaîne est pointé par BX et le rang du caractère est dans AL)
- Chargement d'une adresse dans un registre      **LEA** (au lieu de recevoir la valeur comme avec MOV le registre reçoit l'adresse)
- Chargement d'une adresse de segment dans DS et d'un déplacement dans un registre      **LDS**
- Chargement d'une adresse de segment dans ES et d'un déplacement dans un registre      **LES**

#### - Décalages

- Logique à droite et à gauche      **SHR**    **SHL**
- Arithmétique à droite et à gauche      **SAR**    **SAL**
- Circulaire à droite et à gauche      **ROR** **ROL** **RLC** **RCR**

Le nombre de décalages est spécifié dans CL, l'opérande est sur 8 ou 16 bits.

#### - Ruptures de séquence

- Branchement inconditionnel      **JMP**
- Branchements conditionnels :
  - si égal                    **JE**
  - si différent              **JNE**
  - si <                        **JB** entre entiers naturel    **JL** entre entiers relatifs
  - si < ou =                **JBE** entre entiers naturels   **JLE** entre entiers relatifs
  - si >                        **JA** entre entiers naturels   **JG** entre entiers relatifs
  - si > ou =                **JAE** entre entiers naturels   **JGE** entre entiers relatifs
  - si débordement            **JO**
  - si non débordement        **JNO**
  - si négatif                 **JS**
  - si positif                 **JNS**
  - si parité fausse          **JNP**
  - si parité juste            **JP**
  - si CX est nul             **JCXZ**

- Appel de sous programme           **CALL**
- Retour de sous programme       **RET**
- Appel d'interruption           **INT**
- Retour d'interruption           **IRET**
- Appel de l'interruption relative au débordement       **INT0**
- Attente d'interruption           **WAIT**
- Répéter en décrémentant CX tant que CX est différent de zéro   **LOOP**
- Répéter en décrémentant CX tant que CX est différent de zéro et que l'indicateur d'égalité n'est pas positionné :           **LOOPNE**
- Répéter en décrémentant CX tant que CX est différent de zéro et que l'indicateur d'égalité est positionné :           **LOOPE**

#### - Opérations sur le registre d'état

- Transferts avec AH               **LAHF SAHF**
- Empilement                       **PUSHF**
- Dépilement                       **POPF**
- Mise à 0/1 de la retenue       **CLC STC**
- Complémentation de la retenue   **CMC**
- Mise à 0/1 du bit de direction <sup>11</sup> **CLD STD**
- Masquage/démasquage des interruptions   **CLI STI**

#### - Instructions d'Entrée/Sortie

- Lecture sur 8 ou 16 bits dans AL ou AX :   **IN**
- Ecriture des 8 ou 16 bits de AL ou AX :   **OUT**

#### - Manipulation de caractères

- Transfert d'un ou 2 caractères pointés par SI dans celui ou ceux pointés par DI et mise à jour de SI et DI :           **MOVS**
- Idem avec répétition tant que CX est différent de 0. CX étant automatiquement décrémenté :       **REP MOVS**
- Chargement d'un ou 2 caractères pointés par SI dans AL ou AX et mise à jour automatique de SI :   **LODS**
- Rangement d'un ou 2 caractères contenus dans AL ou AX dans la zone pointée par DI et mise à jour de DI :   **STOS**
- Comparaison d'un ou 2 caractères pointés par DI avec le contenu de AL ou AX et mise à jour de DI :   **SCAS**
- Idem avec répétition tant que CX est différent de 0 et non égalité. CX étant automatiquement décrémenté :       **REPNE SCAS**
- Idem avec répétition tant que CX est différent de 0 et égalité. CX étant automatiquement décrémenté :       **REPE SCAS**
- Comparaison d'un ou 2 caractères pointés par DI avec un ou 2 caractères pointés par SI avec mise à jour de DI et SI : **CMPS**

---

<sup>11</sup> La valeur de ce bit détermine le sens dans lequel sont traitées les chaînes de caractères (droite à gauche ou gauche à droite) par les instructions décrites plus loin.

- Idem avec répétition tant que CX est différent de 0 et non égalité. CX étant automatiquement décrémenté : **REPE CMPS**

- Idem avec répétition tant que CX est différent de 0 et égalité. CX étant automatiquement décrémenté : **REPE CMPS**

La mise à jour des pointeurs SI et DI est faite par addition ou soustraction selon la valeur du bit de direction du registre d'état de 1 ou 2 selon que l'instruction traite 1 ou 2 caractères (8 ou 16 bits)

#### - Instructions de contrôle

- Synchronisation avec un coprocesseur **ESC**
- Verrouillage du bus dans un système multi-microprocesseurs **LOCK**
- Arrêt du microprocesseur **HLT**
- Attente d'interruption **WAIT**

#### - Instructions supplémentaires

Les 80186 et 80188 possèdent quelques possibilités supplémentaires :

- Empilement de tous les registres **PUSHA**
- Dépilement de tous les registres **POPA**
- Lecture de caractères depuis un périphérique **INS**<sup>12</sup>
- Ecriture de caractères vers un périphérique **OUTS**<sup>1</sup>
- Préparation de la pile pour passage de paramètres **ENTER**
- Restitution de la pile après passage de paramètres **LEAVE**
- Vérification de non dépassement de limite (taille de tableau ou de chaîne de caractères par exemple) **BOUND**

### 1.5 La mémoire et l'adressage

Le bus d'adresses est de 20 bits donc l'espace mémoire ne peut excéder 1M octets (2<sup>20</sup>). Les bus d'adresses et de données partagent les mêmes lignes physiques et sont donc multiplexés dans le temps.

Le fonctionnement lors d'un accès à la mémoire est le suivant :

- Durant la 1<sup>ère</sup> période d'horloge l'adresse est émise accompagnée d'un signal (ALE) permettant d'en mémoriser la valeur dans un registre externe (les lignes d'adresses deviendront des lignes de données par la suite en raison du multiplexage).
- Durant la 2<sup>ème</sup> période d'horloge le signal indiquant s'il s'agit d'une lecture ou d'une écriture est émis. Lors d'une écriture c'est aussi pendant cette période que le bus de données reçoit la valeur à écrire.
- La 3<sup>ème</sup> période d'horloge voit s'effectuer le transfert avec la mémoire. Si celle-ci est trop lente un signal spécial (READY) doit être envoyé au microprocesseur de façon à ce qu'il insère des périodes d'horloge d'attente entre cette 3<sup>ème</sup> période et la suivante.
- Durant la 4<sup>ème</sup> période tous les signaux et le bus retrouvent leur état de repos.

La possibilité d'insérer des cycles d'attente entre la 3<sup>ème</sup> et la 4<sup>ème</sup> période d'un accès à la mémoire est la caractéristique des microprocesseurs à bus asynchrone (ce qui n'était pas le cas de la plupart des 8 bits).

On a déjà vu en 1.1 que la mémoire est ensuite divisée en segments de 64K octets et que l'accès est souvent fait en 2 fois 8 bits même sur un 8086 ou un 80186 en raison des problèmes d'alignement en adresses paires.

Les 8086 , 8088 , 80186 et 80188 possèdent les modes d'adressage suivants :

---

<sup>12</sup> Les instructions INS et OUTS fonctionnent comme STOS et LODS à la différence que le rôle de AL ou AX est tenu par le périphérique dont l'adresse est dans DX. Elles peuvent être répétées tant que CX est différent de 0 avec décrémentement de CX par : REP INS ou REP OUTS

- **inhérent ou registre** : Soit l'instruction n'a pas d'opérande soit l'opérande est dans un registre
- **immédiat** : La valeur de l'opérande est dans l'instruction sur 8 ou 16 bits
- **direct ou étendu** : L'opérande est désigné par son adresse relative au début du segment soit sur 8 bits (direct) soit sur 16 bits (étendu).
- **avec base** : L'opérande est désigné par son adresse relative au début du segment obtenue soit par le contenu d'un registre de base BP ou BX soit par la somme d'un registre de base BP ou BX et d'un déplacement sur 8 ou 16 bits contenu dans l'instruction
- **avec index** : L'opérande est désigné par son adresse relative au début du segment obtenue soit par le contenu d'un registre d'index SI ou DI soit par la somme d'un registre d'index SI ou DI et d'un déplacement sur 8 ou 16 bits contenu dans l'instruction
- **avec base et index** : L'opérande est désigné par son adresse relative au début du segment obtenue soit par la somme d'un registre de base BP ou BX et d'un registre d'index SI ou DI, soit par la somme d'un registre de base BP ou BX, d'un registre d'index et d'un déplacement sur 8 ou 16 bits contenu dans l'instruction
- **relatif** : Ce mode n'est utilisable qu'avec les instructions de rupture de séquence. L'instruction contient un déplacement fixe constitué d'un entier relatif sur 8 ou 16 bits qui sera ajouté au contenu du compteur ordinal pour calculer l'adresse de la prochaine instruction.
- **indirect mémoire** : Ce mode n'est utilisable qu'avec les instructions de branchement inconditionnel. L'instruction désigne un opérande 16 bits qui est placé dans le CO (saut intra-segment) ou un opérande 32 bits dont le premier mot est placé dans CS et le deuxième dans CO (saut extra-segment).

Remarque La littérature consacrée au 8086 (même celle du constructeur) ne décrit généralement pas ces 2 derniers modes dans la mesure où ils ne sont liés qu'aux instructions de rupture de séquence et par conséquent ne sont pas disponibles pour l'accès aux opérandes.

## **1.6 L'horloge**

L'horloge de séquençage des 8086 et 8088 peut être de 4, 5, 8, 10 ou 12 MHz alors que le 80186 peut utiliser une horloge de 6, 8, 10, 12.5, 16 ou 20 MHz.

Il devient très difficile, sur ce type de microprocesseurs, de mesurer la durée d'une instruction en effet plusieurs paramètres entrent en jeu :

- L'instruction est ou pas déjà dans la file d'attente (il faut tenir compte du fait que l'UGB fait de la prédiction d'instructions "en aveugle" et ne prévoit pas les ruptures de séquence. A chaque fois qu'un saut est effectué il faut remplir à nouveau la file d'instructions).
- Des cycles d'attente ont lieu ou pas lors des accès à la mémoire.
- L'accès à la mémoire est fait en 1 ou 2 transferts (problème des adresses impaires).

## **1.7 Les Entrées/Sorties**

Les instructions d'entrée/sortie activent des lignes de contrôle spécifiques aux périphériques et permettent d'adresser jusqu'à 64K octets (un contrôleur de périphérique occupe entre 2 et 32 octets en moyenne).

## **1.8 Les interruptions**

Il existe une table (placée en mémoire à partir de l'adresse physique 0) contenant 256 vecteurs (un vecteur est formé de 2 mots de 16 bits : segment et déplacement) associée aux interruptions.

Une interruption peut être provoquée de façon logicielle ou matérielle :

- logicielle, c'est l'instruction INT suivie du n° de l'interruption à traiter

- matérielle, c'est l'activation de la ligne IRQ suivie de l'émission sur le bus de données du n° de l'IT (cette valeur est placée sur le bus par l'auteur de l'interruption en réponse à une sollicitation du microprocesseur).

La prise en compte d'une IT se traduit par la sauvegarde du point de retour et du registre d'état dans la pile suivie du branchement au sous-programme désigné dans la table de vecteurs.

Certains vecteurs de cette table sont associés à des événements particuliers :

- 0 : Division par zéro
- 1 : Mode pas à pas, lorsqu'il est mis dans ce mode le microprocesseur exécute une interruption de type 1 après chaque instruction (trace)
- 2 : Associé à la ligne NMI (Non Maskable Interrupt)
- 3 : Utilisé lors des mises au point pour insérer des points d'arrêt
- 4 : Associé à l'instruction INTO, elle est exécutée si l'indicateur de débordement est positionné

Les suivants n'existent que sur les 80186 et 80188

- 5 : Associé à l'instruction BOUND de vérification de limites
- 6 : Généré lors de la découverte d'un code opération inconnu
- 7 : Associé à l'instruction ESC lorsqu'il n'y a pas de coprocesseur
- 8 : Associé au 1er compteur (TIMER)
- 10 : Associé au 1er canal DMA
- 11 : Associé au 2<sup>ème</sup> canal DMA
- 12 : Ligne INT0 externe
- 13 : Ligne INT1 externe
- 14 : Ligne INT2 externe
- 15 : Ligne INT3 externe
- 18 : Associé au 2<sup>ème</sup> compteur (TIMER)
- 19 : Associé au 3<sup>ème</sup> compteur (TIMER)

## 1.9 Sémaphores

L'instruction LOCK permet de verrouiller les bus dans un système multiprocesseurs (autre microprocesseur, unité d'échange ou canaux DMA). Grâce à cette instruction on peut réaliser des séquences lecture/modification/ écriture en mémoire sans risquer de conflits (sémaphores).

## 1.10 Compatibilité

Marque	Référence	Horloge (MHz)
AMD	Am8086	5 8 10
Harris	HS80C86	5 8 10
Siemens	SAB8086	8 10
AMD	Am8088	5 8 10
Harris	HS80C88	5 8 10
AMD	Am80L188	16
NEC	V30	10 12 16
Siemens	SAB80186	8 10 16
AMD	Am80L188	16
NEC	V20	8 10
Siemens	SAB80188	8 10

**Microphotographie de la puce du 80286.**



## **2 Le 80286**

---

### **2.1 Architecture interne**

On retrouve dans le 80286 les 2 unités du 8086 :

- Unité de gestion de bus           **UGB**
- Unité d'exécution               **UE**

auxquelles ont été ajoutées 2 nouvelles unités :

- Unité d'instructions           **UI**
- Unité d'adresses               **UA**

L'unité d'instructions UI puise dans la file d'attente de 6 octets de l'UGB pour maintenir à jour une file d'attente de 3 instructions décodées.

L'unité d'adresses UA assure le calcul des adresses physiques tant en mode réel qu'en mode virtuel. Le 80286 peut fonctionner en adressage réel il est alors totalement compatible au niveau du code avec les 8086, 8088, 80186 et 80188.

Il peut aussi fonctionner en adressage virtuel avec protections.

#### 1°) Mode réel

Dans ce mode, le fonctionnement est identique à celui décrit pour le 8086. On retrouve les 4 segments de 64K octets. Bien que le 80286 puisse adresser 16M octets ( $2^{24}$ ) de mémoire seuls 1M octets sont accessibles en mode réel (pour la compatibilité avec le 8086).

On peut considérer qu'en mode réel le 80286 n'est qu'un 8086 amélioré et environ 2,5 fois plus rapide.

#### 2°) Mode virtuel protégé

Le 80286 passe en mode virtuel protégé par la mise à 1 d'un bit du registre mot d'état.

Chaque tâche dispose alors d'un giga octets ( $2^{30}$ ) de mémoire virtuelle placée dans 16M octets ( $2^{24}$ ) de mémoire réelle.

L'adresse est constituée d'un sélecteur et d'un déplacement.

Le sélecteur (16 bits) désigne une table de descripteurs et une entrée dans celle-ci.

Le descripteur (48 bits) contient, outre l'adresse de base en mémoire et la taille de la page de mémoire virtuelle, des indications concernant les droits d'accès à cette page.

Le déplacement est ajouté à cette adresse de base pour constituer l'adresse physique.

Le 80286 utilise comme sélecteurs les 4 registres de segments CS DS SS et ES. Il possède en outre 4 registres de 48 bits contenant le descripteur associé à chacun de ces sélecteurs. La mise à jour de ces registres est automatiquement faite par le 80286 à chaque modification des sélecteurs par accès aux tables en mémoire.

Il existe 3 tables de descripteurs décrites chacune par un registre de 40 bits dont un champ de 24 bits pointe sur le début de la table en mémoire et un second champ de 16 bits donne la taille de la table. Les 3 tables sont les suivantes :

- Table globale GDT contenant les descripteurs des pages accessibles par toutes les tâches.
- Table locale LDT contenant les descripteurs des pages propres à une tâche.
- Table d'interruptions IDT contenant les descripteurs des procédures d'interruption (256 interruptions)

## 2.2 Les registres

Le 80286 possède les 14 registres de 16 bits du 8086 dont le rôle est le même, exception faite de l'utilisation en tant que sélecteurs des registres de segments en mode virtuel.

Le 80286 contient de plus les registres suivants :

**MSW** : Mot d'état (16 bits) 4 bits seulement sont utilisés pour le passage en mode virtuel, le changement de tâche et pour indiquer la présence ou l'absence d'un coprocesseur (instruction ESC).

**GDTR** : Descripteur de la table globale (24 bits d'adresse et 16 bits de longueur)

**LDTR** : Descripteur de la table locale (24 bits d'adresse et 16 bits de longueur)

**IDTR** : Descripteur de la table d'interruption (24 bits d'adresse et 16 bits de longueur)

**CSDC** : copie du descripteur dont le sélecteur est CS (48 bits)

**DSDC** : copie du descripteur dont le sélecteur est DS (48 bits)

**SSDC** : copie du descripteur dont le sélecteur est SS (48 bits)

**ESDC** : copie du descripteur dont le sélecteur est ES (48 bits)

**TR** : sélecteur associé à la tâche courante (16 bits) désigne dans la table locale ou globale un descripteur de tâche.

**TRB** : Adresse de base de la table d'état de la tâche courante (24 bits)

**TRL** : Taille de la table d'état de la tâche courante (16 bits)

Remarque : Le registre d'état SR utilise 3 bits supplémentaires :

- 2 pour décrire le niveau de privilège nécessaire à l'utilisation des instructions d'entrée/sortie (voir 2.7).

- 1 (bit NT) pour l'enchaînement des tâches (voir 2.8).

## 2.3 Les opérandes

Le 80286 traite les mêmes opérandes que le 8086 (cf 1.3)

## 2.4 Les instructions

On retrouve tout le jeu d'instruction du 80186 (cf 1.4) auquel sont venues s'ajouter les instructions relatives aux tâches et à la mémoire virtuelle.

**CTS** : Remise à 0 du bit TS du mot d'état. Ce bit est mis à 1 par le 80286 lors d'un changement de tâche de façon à noter qu'il sera probablement nécessaire de changer le contexte des coprocesseurs s'ils doivent être utilisés par la nouvelle tâche. Si ce bit est à 1 et qu'un coprocesseur doit être utilisé le 80286 exécutera une interruption de type 7.

CTS sera donc utilisée par le programmeur après avoir mis à jour le contexte des coprocesseurs.

**LGDT** : Chargement du descripteur de table globale dans GDTR

**LIDT** : Chargement du descripteur de table globale dans IDTR

**LLDT** : Chargement du descripteur de table globale dans LDTR

**SGDT** : Rangement du descripteur de table globale GDTR

**SIDT** : Rangement du descripteur de table globale IDTR

**SLDT** : Rangement du descripteur de table globale LDTR

**LTR** : Chargement du registre TR

**STR** : Rangement du registre TR

Toute modification de TR provoque un accès à la table locale ou globale permettant de lire le descripteur de tâche et de mettre à jour les registres TRB et TRL

**LMSW** : Chargement du mot d'état MSW

**SMSW** : Rangement du mot d'état MSW

**LAR** : Place dans l'opérande l'octet du descripteur associé au sélecteur spécifié décrivant les droits d'accès  
**LSL** : Place dans l'opérande le double octet du descripteur associé au sélecteur spécifié contenant la taille du segment  
**ARPL** : Ajuste le niveau de privilège du sélecteur désigné à celui décrit dans l'opérande (cette instruction ne peut que diminuer le niveau initial)  
**VERR** : Vérifie si le segment décrit par le sélecteur spécifié peut être lu  
**VERW** : Vérifie si le segment décrit par le sélecteur spécifié peut être écrit

Remarque : Pour les 5 dernières instructions se reporter à la description des descripteurs en 2.5 et des privilèges en 2.6

## **2.5 L'adressage de la mémoire**

Les principes d'accès à la mémoire ont été décrits en 2.1.

En mode réel le 80286 possède exactement les mêmes types d'adressage que le 8086 (cf 1.5).

En mode virtuel les modes d'adressage sont encore les mêmes à la seule différence que les registres de segments sont utilisés comme sélecteurs pour accéder à des descripteurs de segments.

Nous allons maintenant décrire plus en détail le contenu des sélecteurs et des descripteurs.

### 1°) Sélecteurs

Ils sont constitués de 3 champs :

- Champ RPL, sur 2 bits, décrit le niveau de privilège (0 à 3)
- Champ TI sur 1 bit désigne la table de descripteurs globale ou locale
- Champ index sur 13 bits contient l'entrée dans la table

### 2°) Descripteurs

Un descripteur est constitué de 64 bits dont 48 seulement sont utilisés. On distingue 3 types de descripteurs :

#### **- Descripteur de segment de code et données**

Il est utilisé pour les segments de code, de données et de pile. Il est constitué de 3 champs :

- Adresse de base du segment en mémoire (24 bits)
- Taille du segment (16 bits)
- Droits d'accès au segment (8 bits) formé de :
  - P : un bit de présence du segment en mémoire
  - DPL : 2 bits, privilège associé au segment (0 à 3)
  - S : un bit, mis à 1 pour distinguer les descripteurs de segment de codes et données des descripteurs de segments du système (voir ci-dessous)
  - Type : 3 bits, protections en lecture/écriture et exécution
  - A : un bit, utilisation du segment, permet la mise en place de stratégies de remplacement de segments

#### **- Descripteur de segment du système**

Il est utilisé pour décrire les segments contenant les tables de descripteurs locales et les tables d'état de tâches.

Il est constitué de 3 champs :

- Adresse de base du segment en mémoire (24 bits)
- Taille du segment (16 bits)
- Droits d'accès au segment (8 bits) formé de :
  - P : un bit de validité du contenu du descripteur
  - DPL : 2 bits, privilège associé au segment (0 à 3)
  - S : un bit, mis à 0 (voir ci-dessus)

- Type : 4 bits :
  - 1 Table d'état de tâche disponible
  - 2 Descripteur de table locale
  - 3 Table d'état de tâche occupée

### - Descripteur de point d'entrée

Il est utilisé pour contrôler des points d'entrée liés à des sous-programmes, des tâches, des interruptions ou des erreurs (traps ou exceptions). Il est constitué de 4 champs :

- Sélecteur du segment de destination (16 bits)
- Déplacement dans le segment de destination (14 bits)
- Compte de mots (sur 5 bits) : nombre de mots à transférer de la pile d'origine à celle de destination lors de l'appel de sous-programme
- Droits d'accès au segment (8 bits) formé de :
  - P : un bit de validité du contenu du descripteur
  - DPL : 2 bits, privilège associé au segment (0 à 3)
  - S : un bit, mis à 0 (voir ci-dessus)
- Type : 4 bits :
  - 4 Point d'entrée de sous-programme
  - 5 Point d'entrée de tâche
  - 6 Point d'entrée d'interruption
  - 7 Point d'entrée d'exception

## 2.6 Les privilèges

Le 80286 gère 4 niveaux de privilège qui contrôlent l'utilisation d'instructions particulières et l'accès aux descripteurs et à leurs segments associés

Ces 4 niveaux sont prévus pour l'utilisation suivante :

- niveau 0 : noyau du système d'exploitation
- niveau 1 : primitives du système d'exploitation
- niveau 2 : extensions du système d'exploitation
- niveau 3 : applications

Les niveaux de privilèges apparaissent en 3 lieux :

### 1°) Privilèges de tâche

Il est défini par le champ RPL du registre CS. Il ne peut être modifié que par le passage à travers un point d'entrée (appel de sous programme , changement de tâche , interruption , exception).

Quand une tâche est activée, son contexte est pris dans la table d'état de tâche (TSS). Cette table est décrite par un descripteur de segment du système (type 3) placé dans la table globale ou locale et pointé par le registre TR. En réalité, ce descripteur est recopié par le 80286 dans les registres TRB et TRL chaque fois que le contenu de TR est modifié.

Le contexte associé à une tâche contient les valeurs de chacun des registres du 80286 (y compris CS dont le champ RPL détermine le privilège de la tâche) et le descripteur de la table locale (LDT) appartenant à la tâche.

### 2°) Privilège de descripteur

Il est contenu dans les 2 bits DPL du champ "droits d'accès du descripteur".

Il définit le niveau de privilège nécessaire pour accéder à ce descripteur (le niveau de privilège de la tâche courante doit être supérieur ou égal à DPL).

Les descripteurs de la table locale (LDT) échappent à cette règle puisqu'ils ne sont accessibles que par la tâche propriétaire de cette table locale.

### 3°) Privilège de sélecteur

Il est défini par le champ RPL d'un sélecteur. Il détermine le niveau de privilège nécessaire pour utiliser ce sélecteur.

Remarque : Le privilège du sélecteur CS est le privilège de la tâche.

## **2.7 La protection**

Le 80286 assure la sécurité en interdisant l'utilisation des instructions LIDT , LLDT , LGDT , LTR , LMSW , CTS et HLT aux tâches dont le niveau de privilège est non nul.

De même, les instructions INS , IN , OUTS , OUT , STI , CLI et LOCK sont interdites aux tâches dont le niveau de privilège est inférieur à la valeur contenue dans le registre d'état.

## **2.8 Le changement de tâche**

Il peut être obtenu par les instructions JMP ou CALL faisant référence à la table d'état des tâches ou à un descripteur de point d'entrée de tâche (type 5) dans la table locale (LDT) ou globale (GDT).

Il peut aussi être obtenu par une interruption logicielle (INT) ou matérielle (ligne physique) ou d'exception (erreur) qui fera référence à un descripteur de point d'entrée de tâche dans la table d'interruptions (IDT).

Le changement de tâche consiste en une sauvegarde du contexte courant dans la table d'état de la tâche courante (TSS) et en un chargement du nouveau contexte depuis la TSS de la nouvelle tâche.

L'instruction IRET provoque le retour à la tâche qui a appelé la tâche courante ou qui a été interrompue. La valeur du bit NT du registre d'état permet au 80286 de distinguer ces 2 cas.

## **2.9 Les interruptions**

Comme le 8086, le 80286 connaît 256 interruptions pouvant être appelées de façon logicielle ou matérielle. Parmi ces interruptions certaines sont associées à des événements particuliers :

- 0 à 7 : même utilisation que dans le 8086 (voir 1.8)
- 8 : en mode réel : n° d'IT dépassant les limites de la table  
en mode virtuel : 2 exceptions ont été détectées au cours de la même instruction
- 9 : Débordement du segment par le coprocesseur
- 10 : En mode virtuel seulement : table d'état de tâche incorrecte
- 11 : En mode virtuel seulement : segment de mémoire virtuelle non présent en mémoire physique
- 12 : En mode virtuel seulement : débordement de la pile ou segment de pile non présent en mémoire physique.
- 13 : En mode réel : débordement de segment  
En mode virtuel : violation de protection ou tentative de passage à un niveau de privilège supérieur

## **2.10 L'horloge**

Le temps d'exécution d'une instruction est, en général, le même en mode réel et en mode virtuel en raison de l'existence de l'unité d'adresse qui effectue tous les calculs avec une architecture de pipe-line et des registres contenant des copies des descripteurs.

La fréquence d'horloge peut être 6 , 8 , 10 , 12 , 16 ou 20MHz.

## 2.11 Compatibilité

<b>Marque</b>	<b>Référence</b>	<b>Horloge (MHz)</b>
<b>AMD</b>	Am80286	8 10 12 16 20
<b>Harris</b>	HS80C286	10 12.5 16 20 25
<b>Siemens</b>	SAB80286	8 10 12 16

## **3 Le 80386**

---

### **3.1 Architecture interne**

Le 80386 est composé de 4 unités :

- **Unité de gestion de bus (UGB)** qui assure la prédiction d'instructions dans une file d'attente de 16 octets ainsi que le contrôle physique des bus
- **Unité d'exécution (UE)** qui contient les 8 registres de 32 bits d'usage général et l'UAL
- **Unité d'instructions (UI)** qui puise dans la file d'attente de l'UGB pour maintenir à jour une file d'attente de 2 instructions décodées
- **Unité de gestion de mémoire (MMU)** elle-même constituée de 2 unités :
  - l'**unité de segmentation** qui gère les adresses logiques et les protections
  - l'**unité de pagination** qui gère les adresses physiques et effectue les calculs associés

Comme le 80286, le 80386 peut fonctionner en 2 modes :

#### 1°) Mode réel

Dans ce mode, il apparaît comme un 8086 doté d'extensions 32 bits

Bien que le 80386 puisse adresser 4G octets ( $2^{32}$ ) de mémoire, seuls 1M octets sont accessibles en mode réel.

#### 2°) Mode virtuel protégé

Dans ce mode, le 80386 dispose de 64T octets ( $2^{46}$ ) de mémoire virtuelle placés dans les 4G octets ( $2^{32}$ ) de mémoire physique.

Le fonctionnement du mode virtuel du 80386 est semblable à celui de 80286 les seules différences étant les suivantes :

- Les descripteurs contiennent une adresse sur 32 bits au lieu de 24 et une taille sur 20 bits au lieu de 16
- Le nombre de types placés dans les droits d'accès des descripteurs de segments du système a été augmenté pour différencier les tables d'état de tâche au format 80286 de celles au format 80386 (ces dernières étant plus grandes puisque le 80386 a plus de registres et qu'ils sont de taille supérieure). INTEL a choisi de conserver la possibilité de gérer des tables de type 80286 afin de garder la plus totale compatibilité entre le 386 et le 286.
- Le nombre de types placés dans les droits d'accès des descripteurs de points d'entrée a été augmenté pour les mêmes raisons que précédemment.
- L'adresse calculée à partir du sélecteur et du déplacement peut ne pas constituer directement l'adresse réelle (comme c'était le cas dans le 80286) lorsque l'unité de pagination est mise en oeuvre (voir 3.6)

### **3.2 Les registres**

Les registres du 80386 sont les suivants :

#### \* 8 registres d'usage général

**EAX** : sur 32 bits, les 16 bits de faible poids étant désignés par AX (divisé en AH et AL) il a le même rôle que AX

**EDX** : sur 32 bits, même remarque que ci-dessus pour DX

**ECX** : sur 32 bits, même remarque que ci-dessus pour CX

**EBX** : sur 32 bits, même remarque que ci-dessus pour BX

**ESI** : sur 32 bits, les 16 bits de faible poids étant désignés par SI, il a le même rôle que SI

**EDI** : sur 32 bits, même remarque que ci-dessus pour DI

**EBP** : sur 32 bits, même remarque que ci-dessus pour BP

**ESP** : sur 32 bits, même remarque que ci-dessus pour SP

\* 6 registres de segments

**CS** : sur 16 bits, segment ou sélecteur de code

**SS** : sur 16 bits, segment ou sélecteur de pile

**DS** : sur 16 bits, segment ou sélecteur de données

**ES** : sur 16 bits, segment ou sélecteur de données supplémentaire

**FS** : sur 16 bits, segment ou sélecteur de données supplémentaire

**GS** : sur 16 bits, segment ou sélecteur de données supplémentaire

\* Le compteur ordinal et le registre d'état

**ECO** : sur 32 bits, compteur ordinal

**ESR** : sur 32 bits, registre d'état qui contient les mêmes indicateurs que celui du 80286 auxquels ont été ajoutés 2 bits:

- **VM** indiquant l'utilisation du mode virtuel en émulation du 8086
- **RF** utilisé en phase de mise au point

\* A ces extensions 32 bits du 80286, il faut ajouter les 4 registres de description de tables **GDTR** , **LDTR**, **IDTR** et **TR** et les copies associées **CSDC** , **DSDC** , **SSDC** , **ESDC** , **TRB** et **TRL** auxquels viennent encore s'ajouter les copies associées aux 2 registres de segments supplémentaires **FSDC** et **GSDC**.

\* Le 80386 contient en outre 4 registres de contrôle

**CR0** : registre de contrôle général qui inclut le registre MSW du 80286 plus un bit de mise en fonctionnement de l'unité de pagination (voir 3.6)

**CR1** : non utilisé (prévu pour des évolutions futures)

**CR2** : contient l'adresse ayant provoqué le dernier défaut de page mémoire virtuelle

**CR3** : pointeur sur le répertoire de pages utilisé par l'unité de pagination (voir 3.6)

\* 8 registres de mise au point

**DR0** à **DR3** : adresses de points d'arrêt

**DR4** et **DR5** : non utilisés (prévus pour des évolutions futures)



**DR6** : état relatif au point d'arrêt

**DR7** : condition de prise en compte des points d'arrêt

\* 2 registres de test de mémoire

**TR6** : registre de contrôle du test

**TR7** : registre d'état du test

Enfin le 80386 possède une mémoire associative de 32 mots contenant une copie des entrées les plus récemment utilisées (algorithme du LRU) du répertoire de la table des pages. Cette mémoire associative est entièrement gérée par l'unité de pagination (voir 3.6)

### **3.3 Les opérandes**

Le 80386 traite les opérandes suivants :

- Bit : un seul bit pris dans un mot
- Suite de bits : un groupe d'au plus 32 bits consécutifs
- Chaîne de bits : un groupe d'au plus 4G bits consécutifs ( $2^{32}$ )
  
- Entiers naturels : 8 16 32 ou 64 bits
- Entiers relatifs : 8 16 32 ou 64 bits en notation complément à 2
  
- Pointeur : constitué soit d'un déplacement sur 16 ou 32 bits soit d'un segment ou sélecteur sur 16 bits et d'un déplacement sur 16 ou 32 bits
  
- Caractère en ASCII sur 8 bits
  
- Chaîne d'octets, de mots (2 octets) ou de doubles mots (4 octets) pouvant aller de 1 à 1G octets ( $2^{30}$ )
  
- Entiers décimaux en DCB sur 8 bits (1 ou 2 chiffres)

### **3.4 Les instructions**

Le 80386 possède toutes les instructions du 80286 auxquelles viennent s'ajouter :

**MOVZX** : extension d'un entier naturel sur 8 ou 16 bits à un format 32 bits

**MOVSX** : extension d'un entier relatif sur 8 ou 16 bits à un format 32 bits

**CDWE** : conversion d'un mot (16 bits) en double mot (32 bits)

**CDQ** : conversion d'un double mot (32 bits) en quadruple mot (64 bits)

**LFS** : comme LDS pour le registre FS

**LGS** : comme LDS pour le registre GS

**LSS** : comme LDS pour le registre SS

**PUSHFD** : empilement de ESR

**POPFD** : dépilement de ESR

**SHRD/SHLD** : décalage à droite/gauche sur 32 bits

**BT** : test d'un bit

**BTS** : test d'un bit puis mise à 1

**BTR** : test d'un bit puis mise à 0

**BTC** : test d'un bit puis complémentation

**BSF** : déplacement en avant de n bits (n étant l'opérande de l'instruction)

**BSR** : déplacement en arrière de n bits (n étant l'opérande de l'instruction)

**IBTS** : insertion d'une chaîne de bits

**XBTS** : extraction d'une chaîne de bits

**SETxx** : positionnement d'un booléen sur 8 bits si la condition exprimée par xx est vraie :

xx	condition testée
O	débordement
B	< pour des entiers naturels
NB	> ou = pour des entiers naturels
E	égal
NE	différent
BE	< ou = pour des entiers naturels
NBE	> pour des entiers naturels
S	négatif
NS	positif
P	parité juste
NP	parité fausse
L	< pour des entiers relatifs
NL	> ou = pour des entiers relatifs
LE	< ou = pour des entiers relatifs
NLE	> pour des entiers relatifs

### **3.5 L'adressage de la mémoire**

Les principes d'accès à la mémoire ont été décrits en 3.1.

Les méthodes de pagination le seront en 3.6.

Les modes d'adressage des opérandes par le 80386 sont ceux du 8086 à la différence que les déplacements et les valeurs immédiates peuvent être exprimées sur 8 , 16 ou 32 bits et que les 8 registres généraux peuvent servir indifféremment de base et d'index.

Le 80386 offre de plus la possibilité de multiplier, lors du calcul d'adresse, le contenu de l'un des registres par un facteur de 1 , 2 , 4 ou 8 ce qui permet l'accès facile à des tableaux.

### **3.6 La pagination**

Le 80386 utilise 2 niveaux de tables pour transformer une adresse en adresse physique. Le mécanisme de pagination repose sur 3 éléments :

- Le répertoire de tables de pages
- Les tables de pages
- Les pages

#### 1°) Le répertoire

C'est une table de 4K octets pointée par CR3.

Chaque entrée contient l'adresse d'une table de pages et des indications de protection, de présence et d'utilisation de cette table.

#### 2°) Les tables de pages

Ce sont des tables de 4K octets. Chaque entrée contient l'adresse physique de la page et les indications de protection, de présence et d'utilisation relatives à cette page.

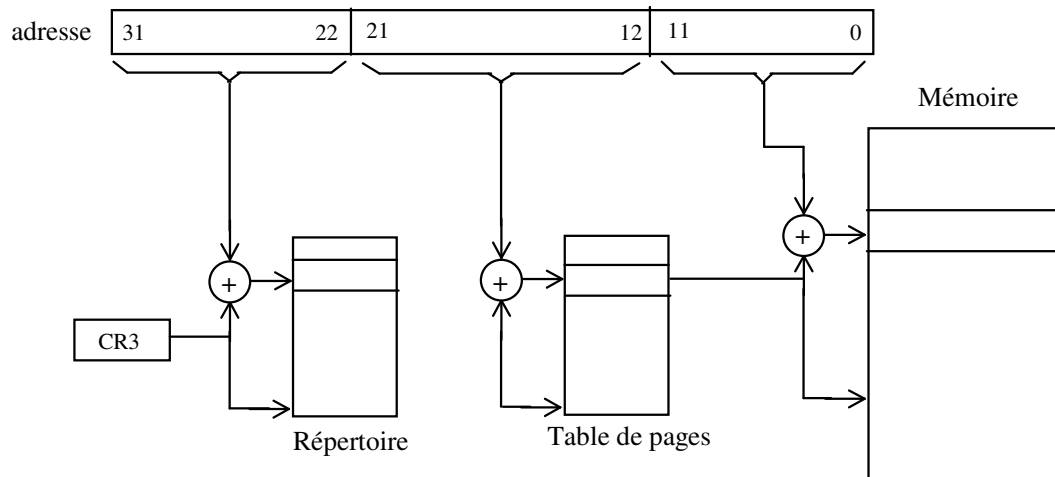
### 3°) Le calcul

Les bits 22 à 31 de l'adresse servent à désigner l'entrée dans le répertoire.

Les bits 12 à 21 servent à désigner l'entrée dans la table des pages.

Les bits 0 à 11 servent de déplacement dans la page en mémoire.

On a le schéma suivant :



### Erreur ! Les arguments du commutateur ne sont pas spécifiés. 3.7 Les privilèges et les changements de tâche

Le fonctionnement est celui décrit pour le 80286 en 2.7 et 2.8

### 3.8 Les interruptions

Elles fonctionnent comme sur le 80286 (voir 2.9) mais 2 cas d'exception ont été ajoutés :

- 14 : Défaut de page
- 15 : Erreur de coprocesseur

De plus, INTEL réserve les interruptions 17 à 32 à de futures évolutions.

### 3.9 L'horloge

Le temps d'exécution d'une instruction est, en général, indépendant du mode d'accès à la mémoire (mode réel ou mode virtuel avec ou sans pagination).

Le 80386 offre une gestion en pipe-line du bus d'adresse qui consiste à placer sur le bus l'adresse de l'accès suivant avant que n'ait été terminé l'accès précédent. Cette méthode permet de démarrer un nouvel accès à la mémoire avant même que le précédent ne soit terminé ce qui autorise la gestion de façon optimale de bancs de mémoire concurrents.

Ainsi, avec une horloge à 16MHz et une mémoire en bancs ayant un temps d'accès de 100ns, le 80386 peut fonctionner sans cycle d'attente.

Les fréquences proposées sont 12 , 16 , 20 , 25 et 33 Mhz.

### 3.10 Compatibilité

Il faut distinguer le vrai 80386 aussi appelé 386DX des versions améliorées de 286 appelés 386SX dont le bus de données est sur 16 bits (32 bits en interne) et celui d'adresses sur 24 bits mais qui peuvent exécuter le jeu d'instruction de 80486.

<b>Marque</b>	<b>Référence</b>	<b>Horloge (MHz)</b>	<b>Remarques</b>
<b>AMD</b>	Am386DX	16 20 25 33 40	
	Am386DXL	20 25 33 40	faible consommation
	Am386DXLV	25 36	faible consommation et alim 3,3V
	Am386SX	16 20 25 33 40	
	Am386SXL	20 25 33 40	faible consommation
	Am386SXLV	20 25 33	faible consommation et alim 3,3V
<b>IBM</b>	386SLC	16 20 25	antémémoire de 8Ko
	486DLC	16 20 25	antémémoire de 16Ko
	486DLC2	33/66	double vitesse
	486SLC	16 20 25	c'est un 386SX avec 16Ko d'antémémoire
	486SLC2	16/32 20/40 25/50	comme le précédent avec double vitesse
		33/66 40/80	

## 4 Le 80486

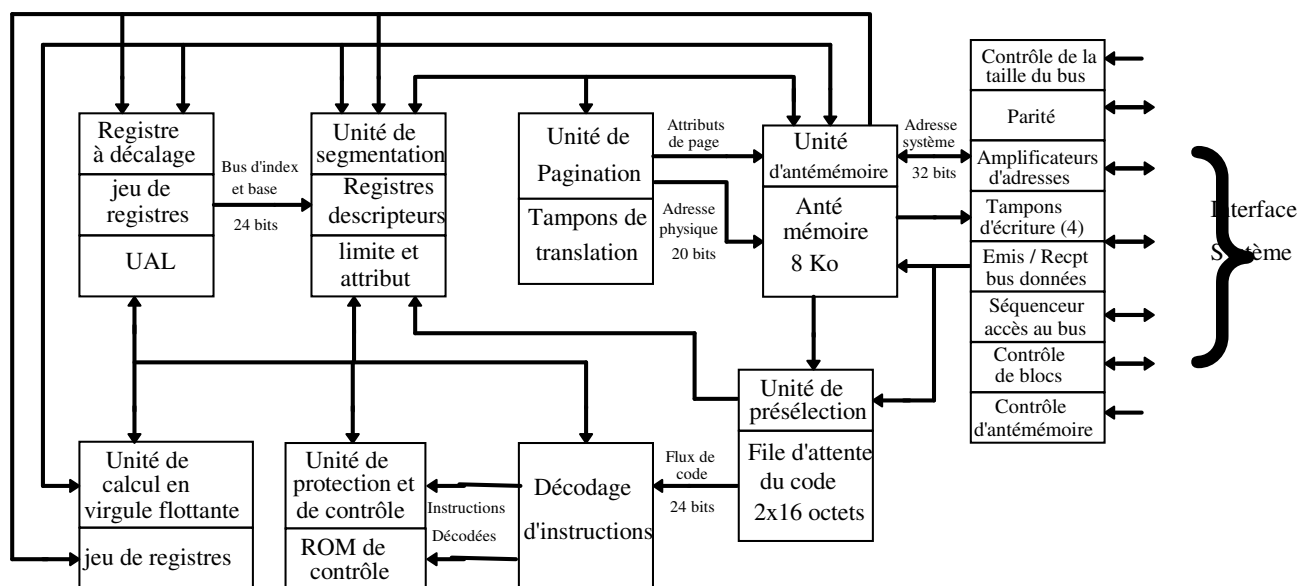
### 4.1 Architecture interne

L'architecture du 80486 est celle du 80386 auquel ont été rajoutés le coprocesseur arithmétique 80387 et une antémémoire de 8Ko dotée de son contrôleur.

La plupart des efforts ont porté sur les performances des accès à la mémoire.

Les opérations sur les entiers simples ont été accélérées et parfois même ramenées à un seul cycle d'horloge comme dans les processeurs RISC.

Parallèlement, l'intégration du processeur arithmétique rend le traitement des réels plus rapide car les transferts de données sont plus efficaces.



### 4.2 Les registres et les opérandes

Le 80486 possède les mêmes registres que le 80386 et traite les mêmes opérandes que lui. Seuls 3 registres de test de fonctionnement de l'antémémoire ont été ajoutés.

### 4.3 Les instructions

On retrouve tout le jeu d'instructions du 80386 auquel viennent s'ajouter les instructions suivantes :

**BSWAP** qui inverse l'ordre des octets dans un registre 32 bits (utilisé dans les codages de bases de données)

**XADD** qui fait un échange et une addition. Cette instruction est indivisible et permet la mise en oeuvre de sémaphores.

**CMPXCHG** qui fait une comparaison et un échange. Cette instruction est indivisible et permet la mise en oeuvre de sémaphores.

Les 3 instructions suivantes ne sont utilisables qu'en mode superviseur:

**INVD** qui invalide l'antémémoire pour les données.

**WBINVD** qui invalide l'antémémoire pour les données et génère un signal permettant la mise en oeuvre d'une antémémoire externe.

**INVPLG** qui invalide la recherche dans la table des correspondances d'adresses du gestionnaire de la mémoire.

#### **4.4 La gestion de la mémoire**

La mémoire est organisée en 1 giga ( $2^{30}$ ) mots de 64 bits ainsi un seul accès permet d'obtenir 2 mots de 32 bits.

Lorsque, pour un cycle de lecture, la donnée recherchée n'est pas dans l'antémémoire, le 80486 lance un accès permettant de remplir 4 mots (16 octets) de l'antémémoire.

Cet accès se fait en 5 cycles d'horloge :

1<sup>er</sup> cycle : envoi de l'adresse

2<sup>ème</sup> cycle : transfert du mot de 32 bits et démarrage de l'accès au mot de 64 bits suivant en mémoire.

3<sup>ème</sup> cycle : transfert du 2<sup>ème</sup> mot de 32 bits qui avait été lu en mémoire en même temps que le premier.

4<sup>ème</sup> cycle : transfert du 3<sup>ème</sup> mot de 32 bits.

5<sup>ème</sup> cycle : transfert du 4<sup>ème</sup> mot de 32 bits.

Des cycles d'attente peuvent être insérés entre chacune de ces étapes si la mémoire n'est pas assez rapide. Le signal (BRDY) du 80486 est prévu à cet effet.

Le 80486 génère automatiquement 4 bits de parité (1 par octet) qui sont ajoutés au bus de données et peuvent être mis en mémoire.

Lors d'une lecture, si une erreur de parité est détectée, le 80486 n'en est pas affecté mais active une ligne externe (PCHK) qui peut être utilisée pour générer une interruption physique.

Afin d'assurer la cohérence de l'antémémoire dans un système multiprocesseur (où plusieurs processeurs peuvent écrire en mémoire), le 80486 peut être informé lors d'une modification de la mémoire par l'activation de sa ligne EADS (external address). Il se chargera alors de comparer cette adresse à celles présentes dans l'antémémoire afin d'assurer les mises à jour éventuelles.

Il est, de plus, possible de déclarer certaines pages de la mémoire comme ne devant pas être gérées au travers de l'antémémoire de sorte que les transferts se font directement avec la mémoire centrale.

#### **4.5 L'horloge**

Les fréquences proposées sont 20, 25 et 33 Mhz. Il existe des versions appelées DX2 ou SX2 capables de travailler avec une fréquence élevée en la réduisant de moitié lors des accès à la mémoire. De même, les DX4 ou SX4 la réduisent du tiers.

#### **4.6 Compatibilité**

IL faut distinguer le vrai 80486 appelé 486DX des versions n'incluant pas l'unité de calcul en réels appelés 486SX.

Marque	Référence	Horloge (MHz)	Remarques
<b>AMD</b>	Am486DX	33 40	
	Am486DX2	20/50 33/66 40/80	double vitesse
	Am486DX4	33/99 40/120	triple vitesse
	Am486SX	33 40	
	Am486SX2	20/50 33/66	double vitesse
	Am5x86	33/133 40/160	antémémoire de 16Ko et quadruple vitesse
<b>Texas</b>	TI486SXL-S-GA	40	antémémoire de 8Ko
	TI486SXL2-S-GA	20/40 25/50	antémémoire de 8Ko et double vitesse
<b>Cyrix</b>	486DLC	25 33 40	pas d'unité en réel, antémémoire 1Ko
	486SLC	20 25 33 40	pas d'unité en réel, antémémoire 1Ko
	486SLC2	25/50	pas d'unité en réel, antémémoire 1Ko, double vitesse
	Cx486D	40	version avec ventilateur
	Cx486S	33 40 50	c'est un 486SX avec antémémoire de 2Ko
	Cx486S2	20/40 25/50	c'est un 486SX avec antémémoire de 2Ko en double vitesse
	Cx486DX	33 40 50	
	Cx486DX2	20/40 25/50	double vitesse
	Cx486DX4	25/75 33/99	triple vitesse
	<b>IBM</b>	486BLX	15 20 25 33
486BLX2		15/30 20/40 25/50 33/66	double vitesse
486BLX4		15/45 20/60 25/75 33/99	triple vitesse
486SBLDX2		33/66 40/80	double vitesse

## 5 Les coprocesseurs arithmétiques

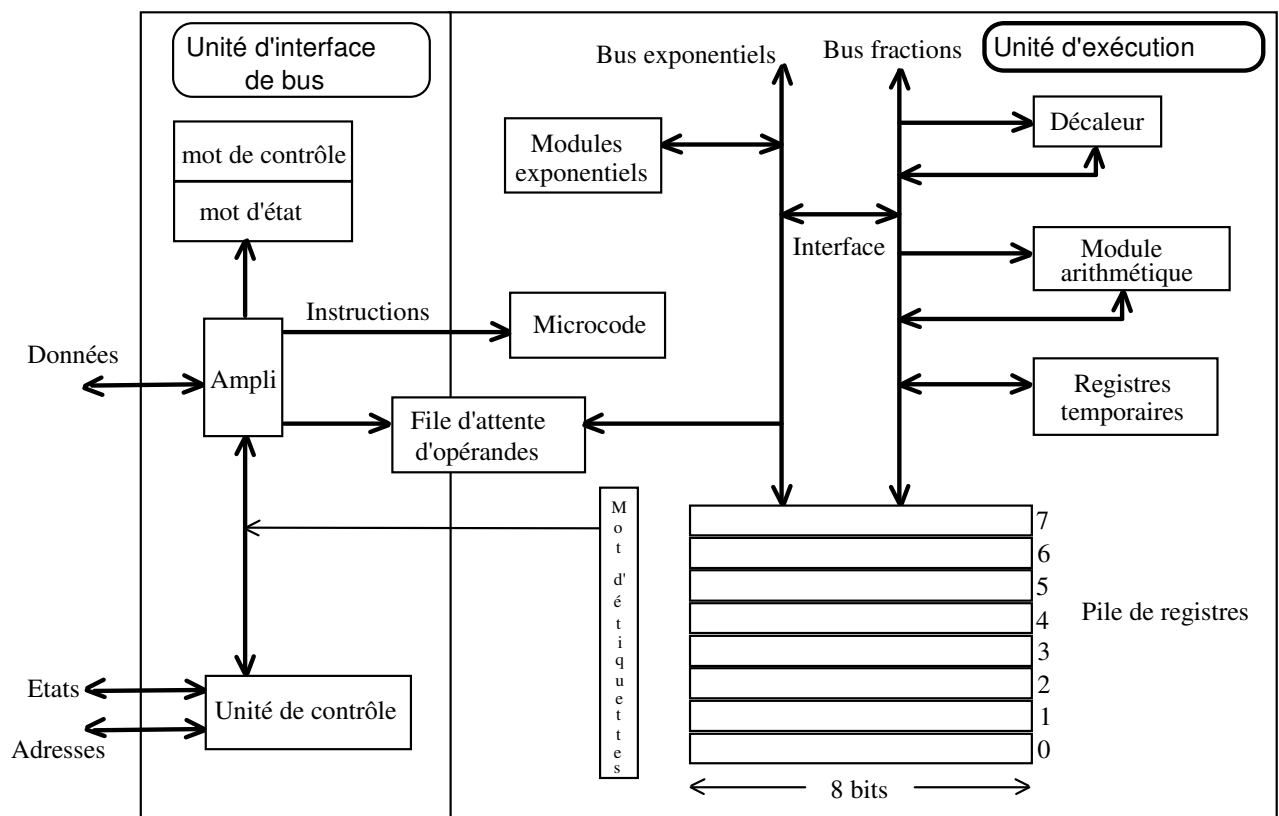
Les instructions arithmétiques directement disponibles sur un microprocesseur sont peu nombreuses. Il a donc été nécessaire d'utiliser des algorithmes permettant de calculer les fonctions mathématiques à partir de ces seules opérations de base (algorithmes de Cordic, polynômes de Tchébitchev...)

L'inconvénient majeur de ces méthodes est le nombre important de calculs qu'elles nécessitent et qui les rend extrêmement lentes.

Les constructeurs se sont donc tout naturellement tournés vers des solutions câblées en fabriquant des extensions du microprocesseur que l'on a baptisées coprocesseurs.

Avec les progrès de l'intégration ces composants ont, par la suite, pu être joints au microprocesseur (80486 et 68040).

### 5.1 Architecture interne des coprocesseurs INTEL 8087, 80287 et 80387





## 5.2 Représentation des données

Il existe 7 formats de données numériques différents :

Format	Limites	Représentation
Mot entier	-32767 à 32767	2 octets en complément à 2
Entier court	$-2 \cdot 10^9$ à $2 \cdot 10^9$	4 octets en complément à 2
Entier long	$-9 \cdot 10^{18}$ à $9 \cdot 10^{18}$	8 octets en complément à 2
DCB compacté <sup>13</sup>	-9999999999999999 à 9999999999999999	1 octet pour le signe (seul le bit de gauche est utilisé). 9 octets pour les 18 chiffres décimaux
Réel court <sup>14</sup>	$-3,39 \cdot 10^{38}$ à $3,39 \cdot 10^{38}$	exposant sur 8 bits, mantisse sur 24 bits
Réel long <sup>2</sup>	$-1,8 \cdot 10^{308}$ à $1,8 \cdot 10^{308}$	exposant sur 11 bits, mantisse sur 53 bits
Réel temporaire <sup>2</sup>	$-1,1910^{4932}$ à $1,1910^{4932}$	exposant sur 16 bits, mantisse sur 64 bits

## 5.3 Les coprocesseurs arithmétiques de la famille INTEL

Chacun des membres de la famille INTEL s'est vu doter d'un coprocesseur arithmétique permettant des calculs rapides.

**Le 8087** : Disponible en 5 , 8 et 10 MHz, il est prévu pour fonctionner avec les 8086 , 8088 , 80186 et 80188.  
Il traite les opérations arithmétiques et les fonctions trigonométriques, exponentielles et logarithmiques.  
Il contient 8 registres de 80 bits qui viennent s'ajouter à ceux du microprocesseur.

**Le 80287** : Disponible en 5 , 6 , 8 et 10 MHz, il est prévu pour fonctionner avec le 80286.  
Il offre les mêmes possibilités que le 8087 mais présente une architecture directement compatible avec le 80286 et ses modes réel et virtuel.

**Le 80387** : Disponible en 16 et 20 MHz, il est prévu pour fonctionner avec le 80386.  
Il offre les fonctionnalités des 8087 et 80287 mais avec des performances 5 à 7 fois supérieures et permet de gérer des réels non normalisés.  
Il est compatible avec les modes réel et virtuel du 80386 ainsi qu'avec la gestion en pipe-line des bus.

---

<sup>13</sup>Format correspondant à la norme COBOL

<sup>14</sup>Format correspondant à la norme IEEE 754

## 6 Le Pentium (P5)

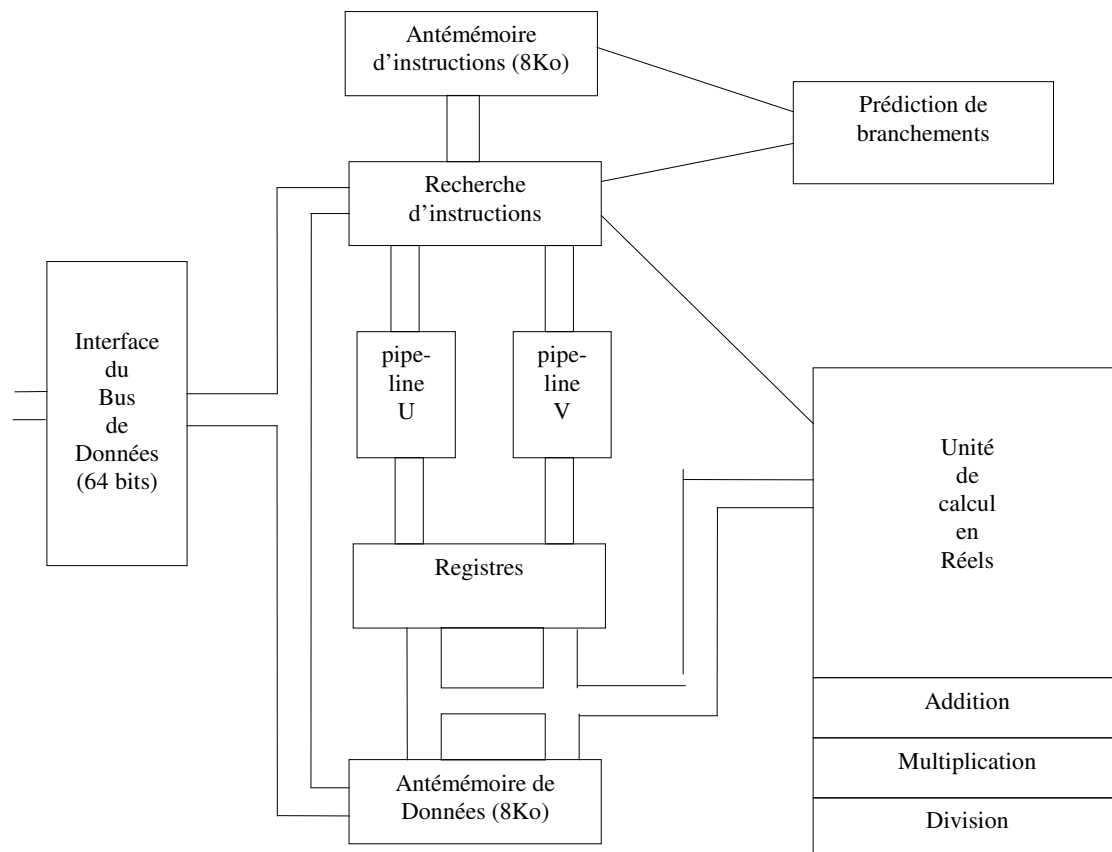
Le Pentium doit son nom au fait qu'il est le cinquième de sa famille (il aurait pu s'appeler 80586).

Les objectifs visés lors de la conception de ce processeur sont les suivants :

- Compatibilité avec les 80x86
- Augmentation des performances par l'augmentation de la taille des antémémoires, la mise en place d'un mécanisme de prédiction pour les instructions de branchements, l'utilisation d'un bus de données de 64 bits.
- Mémoire gérée en pages de 4Ko (comme sur le 80386) ou de 4Mo.
- Développement de compilateurs utilisant au mieux les possibilités de parallélisme du processeur et optimisant l'utilisation des antémémoires.

### 6.1 Architecture interne

Le schéma général du Pentium est le suivant :



Le décodage et l'exécution des instructions suit le principe suivant :

L'unité **IF** (Instruction Fetch) assure la lecture de l'instruction dans l'antémémoire

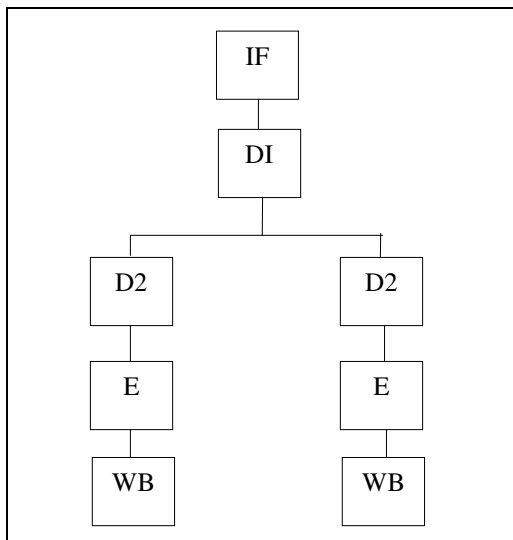
L'unité **DI** (Decode 1) fait le premier décodage de l'instruction et détecte son indépendance vis à vis de l'instruction suivante. Si elles sont indépendantes elles seront envoyées vers les pipes-lines U et V en parallèle pour être exécutées.

Chaque pipe-line (U et V) est constitué de :

L'unité **D2** (Decode 2) qui termine le décodage de l'instruction et prend en charge la génération des adresses nécessaires à l'exécution de l'instruction.

L'unité **E** (Exec) qui assure l'exécution proprement dite. Les instructions traitant des réels ne peuvent pas être prises en compte ici et doivent l'être dans l'unité spécialisée pour les flottants. En réalité les deux unités E ne sont pas totalement identiques et seule l'unité du pipe line U peut exécuter les opérations de décalage et celles portant sur des groupes de bits.

L'unité **WB** (Write Back) qui prend en charge le rangement du résultat.



Une logique spéciale gère la mise à jour des indicateurs et de la pile afin que tout se passe comme si les instructions étaient exécutées séquentiellement. En effet, lorsque deux instructions se déroulent en parallèle (une dans le pipe-line U l'autre dans le V) rien ne garantit que la seconde ne se termine pas avant la première (si par exemple elle utilise une opération plus rapide comme une addition vis à vis d'une multiplication). Il faut assurer que les indicateurs du registre d'état correspondent au résultat obtenu par la seconde instruction et non par la première.

## **6.2 Les registres et les instructions**

On retrouve tous les registres du 80486 et un registre de contrôle supplémentaire (CR4) permettant de gérer la pagination propre au Pentium (voir 6.3).

Le Pentium connaît toutes les instructions du 80486 auxquelles viennent s'ajouter quelques instructions relatives à la mise en cohérence des contenus des antémémoires dans un contexte multi-processeur. L'instruction multi-processeur CMPXCHG a été étendue aux opérandes sur 64 bits sous le nom : CMPCHG8B.

## **6.3 La gestion de la mémoire**

Le bus de données de 64 bits permet un débit de 528 Mo/s avec un Pentium à 66Mhz est . L'accès à la mémoire peut se faire en 8, 16, 32 ou 64 bits.

Un contrôle d'erreurs est mis en place sur les bus de données et d'adresse ainsi que sur les antémémoires et la table de translation de page.

Le bus des adresses est constitué de 29 lignes ( $A_3$  à  $A_{31}$ ) et de 8 lignes ( $BE_0$  à  $BE_7$ ) correspondant au décodage des bits 0 à 2 de l'adresse. Ainsi les 29 lignes d'adresse permettent de désigner un mot de 64 bits parmi 512 Méga tandis que les 8 lignes décodées permettent de désigner un octet parmi les 8 constituant le mot. On arrive ainsi à 4 Go. Le pentium met en outre en oeuvre un bit de parité sur le bus des adresses

Le pentium possède 2 antémémoires de 8Ko chacune. L'une est réservée aux instructions et l'autre aux données. Ces antémémoires sont divisées en blocs de 32 octets.

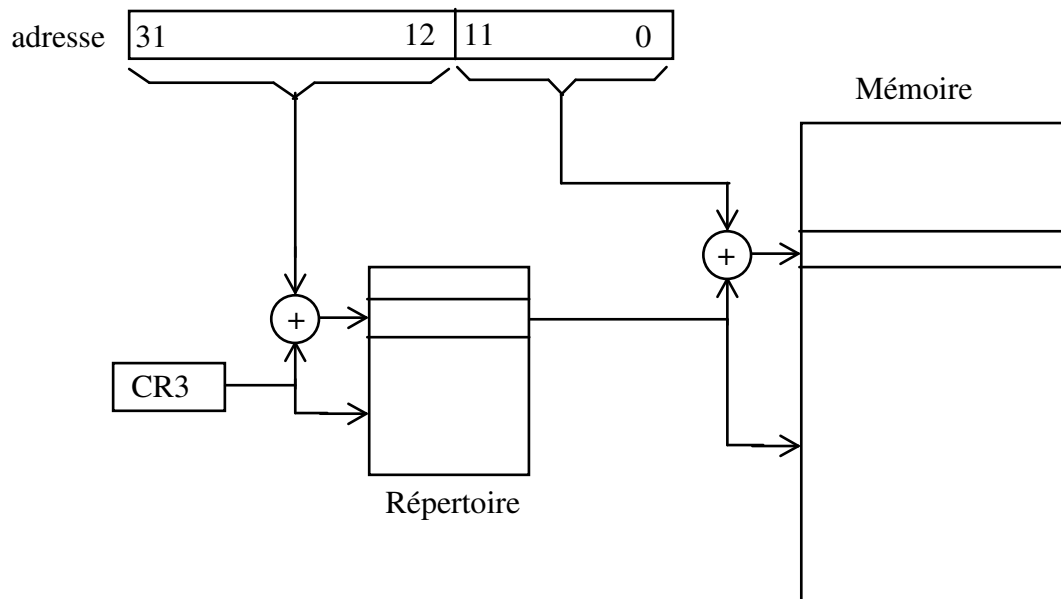
La mémoire est gérée, comme depuis le 80386 (voir 3.6) par le biais de tables de page.

Contrairement aux 80386 et 80486, le pentium permet de définir de pages de tailles différentes :

pour les données, 64 pages de 4Ko et 8 pages de 4Mo

pour les instructions, 32 pages de 4Ko.

On retrouve donc les modes réel et virtuel décrits en 3.1 ainsi que la pagination décrite en 3.6. Cette dernière est augmentée d'une forme permettant d'utiliser des pages de 4Mo :



#### **6.4 Prédiction d'instructions**

Le problème de la prédiction d'instructions est celui posé lors d'une instruction de branchement conditionnelle. En effet, le dispositif de recherche d'instructions lit à l'avance les instructions en séquence. Mais, lors d'un branchement, il est impossible de savoir à l'avance s'il aura lieu ou pas.

Les dispositifs utilisés jusque là se contentaient de faire comme si le branchement ne devait pas avoir lieu puis, le cas échéant, d'arrêter le pipe-line d'exécution et de vider la file d'attente des instructions, pour relancer la recherche à partir de la nouvelle adresse.

La méthode utilisée sur le Pentium est la suivante :

L'unité de recherche des instructions lit les instructions en séquence jusqu'à rencontrer un branchement conditionnel. Lorsque cela se produit elle fait appel à une unité de prédiction de branchement qui lui indique si elle doit continuer la recherche en séquence ou à partir de l'adresse indiquée dans l'instruction de branchement.

L'unité de prédiction des branchements fait appel à une mémoire associative contenant des informations sur les 256 branchements précédemment rencontrés. A l'aide de cet historique elle produit une prévision qui se révèle correcte dans 80 à 85% des cas.

#### **6.5 Les réels**

L'unité de calcul en réels est 3,5 fois plus rapide que celle qui équipait le 80486.

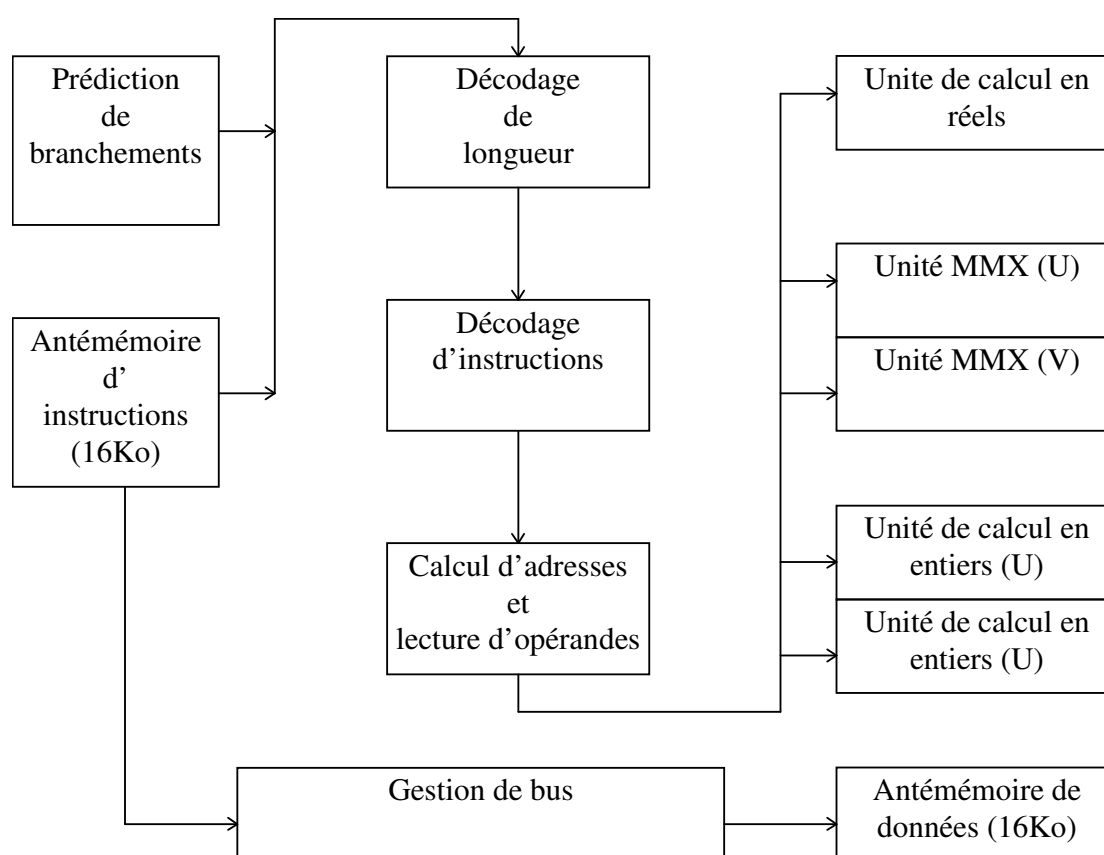
L'addition et la multiplication s'effectuent en 3 cycles d'horloge tandis que la division prend de 18 à 38 cycles. Toutefois en raison de sa réalisation en pipe-line, elle peut produire un résultat par cycle si elle est alimentée en continu.

Elle est constituée de 3 pipe-lines (addition, multiplication et division) qui échangent les opérandes avec l'antémémoire de données au travers d'un bus de 64 bits.

## 7 Le Pentium MMX (P55C)

Afin de répondre à la demande de puissance de calcul issue des applications multimédia, INTEL a introduit un jeu de registres et d'instructions permettant de traiter simultanément plusieurs opérations. Ces instructions s'avèrent particulièrement utiles pour les traitements d'image (calculs sur des groupes de pixels) tant en 2D qu'en 3D. Elles sont inspirées de celles que l'on trouve sur les processeurs de signal (DSP) et correspondent donc aux types de calculs utilisés en traitement de signal (son et image). Toutefois ces instructions utilisent des ressources de l'unité de calcul en réels ce qui fait qu'elles ne peuvent se faire en même temps. Il est en outre nécessaire de basculer par une instruction spéciale le processeur entre le mode MMX et le mode calcul en réels. Ce qui abaisse sensiblement les performances globales du processeur.

### 7.1 Architecture interne



**Remarque :** Les deux unités MMX reproduisent le modèle de fonctionnement des unités U et V de calcul en entiers du Pentium. Toutefois elles fonctionnent en concurrence avec l'unité de calcul en réels ce qui oblige le programmeur à basculer explicitement entre ces deux modes.

### 7.2 Les registres et les instructions

Le Pentium MMX est enrichi de 8 registres de 64 bits réservés aux instructions MMX, toutefois ces registres portent les mêmes noms que ceux de l'unité de calcul en réels et ne peuvent donc être désignés simultanément par le programmeur.

57 nouvelles instructions viennent s'ajouter à celle du Pentium elle permettent d'accélérer les calculs d'image ou de signal en permettant le traitement simultané de plusieurs opérandes (groupes de pixels ou

calcul matriciel). les opérandes de 64 bits de ces instructions peuvent être interprétés comme une valeur unique de 64 bits ou 2 valeurs de 32 bits ou 4 valeurs de 16 bits ou, enfin, 8 valeurs de 8 bits.

On trouve 7 types d'instructions MMX :

\* Transfert de données sur 32 ou 64 bits entre les registres MMX et la mémoire

\* Conversions de 16 vers 32 bits, de 32 vers 16 bits, de 8 vers 16 bits, de 16 vers 32 bits et de 32 vers 64 bits. Ces conversions peuvent tenir compte ou pas du signe.

\* Arithmétiques :

**addition** : 1 addition entre opérandes de 64 bits  
ou 2 additions simultanées entre opérandes de 32 bits  
ou 4 additions simultanées entre opérandes de 16 bits  
ou 8 additions simultanées entre opérandes de 8 bits

Les opérandes peuvent être considérés comme des entiers naturels ou des entiers relatifs. De plus il est possible d'utiliser un mode d'addition qui ne produit jamais de débordement c'est à dire qui remplace tout résultat trop grand par la valeur maximale représentable. Ce fonctionnement correspond au comportement normalement attendu dans les calculs de traitement de signal.

**soustraction** : même type de fonctionnement que l'addition

**multiplication** : 4 multiplications simultanées entre opérandes de 16 bits fournissent 4 résultats sur 32 bits chacun.

**multiplication avec cumul** : cette opération, très utilisées en traitement de signal, consiste en 4 multiplications simultanées entre opérandes de 16 bits fournissent 4 résultats sur 32 bits chacun puis addition deux à deux de ces résultats pour obtenir deux résultats finals sur 32 bits.

\* Comparaison : 1 comparaison entre opérandes de 64 bits  
ou 2 comparaisons simultanées entre opérandes de 32 bits  
ou 4 comparaisons simultanées entre opérandes de 16 bits  
ou 8 comparaisons simultanées entre opérandes de 8 bits

Seuls les test d'égalité et de supériorité peuvent être effectués.

\* Logiques : Opération ET OU OU exclusif et NAND sur 64 bits

\* Décalages : 1 décalage logique (DLG ou DLD) entre opérandes de 64 bits  
ou 2 décalages logiques (DLG ou DLD) simultanés entre opérandes de 32 bits  
ou 4 décalages logiques (DLG ou DLD) simultanés entre opérandes de 16 bits  
ou 2 décalages arithmétiques à droite simultanés entre opérandes de 32 bits  
ou 4 décalages arithmétiques à droite simultanés entre opérandes de 16 bits

\* Basculement MMX / FPU : Cette instruction permet de basculer entre les opérations MMX et les opérations réelles. Elle doit être appelée quand on veut passer d'un type à l'autre. Ce problème devrait être résolu dans les versions suivantes de processeurs MMX.

### 7.3 Mémoire

Outre les 2 antémémoires de 16Ko intégrées, le pentium MMX peut gérer une antémémire de second niveau de 512 Ko et une mémoire de 64 Go.

### 7.4 Horloge

L'horloge est de 233 ou 266 ou 300 Mhz

## 8 Le Pentium Pro ou P6

---

Le P6 présente la particularité d'être constitué de deux puces réunies dans un même boîtier.

La première a une surface de 300mm<sup>2</sup> et comporte 5,5 millions de transistors, elle constitue le processeur proprement dit.

La seconde a une surface de 200mm<sup>2</sup> et comporte 15,5 millions de transistors, elle constitue une antémémoire de second niveau c'est à dire se plaçant entre les antémémoires placées dans le processeur et la mémoire. Cette antémémoire est de 256Ko et est constituée de RAM statique. Elle accepte des transferts à la vitesse du processeur et permet des transferts vers la mémoire à 1/2, 1/3 ou 1/4 de cette vitesse (le Pentium ne permettait qu'un rapport de 2/3).

Remarque : Il existe une version du Pentium Pro appelée P6L qui ne possède pas cette antémémoire de deuxième niveau.

Les principales améliorations par rapport au Pentium sont les suivantes :

Le P6 peut traiter jusqu'à 5 instructions simultanément grâce à la multiplication du nombre d'unités d'exécutions implantées dans le processeur

Les antémémoires de données et d'instructions de 8Ko chacune se voient secondées par une antémémoire générale de 256Ko ou 512Ko.

La méthode de prédiction de branchement a été améliorée et permet de traiter de 10 à 15 branchements imbriqués avec un taux de succès de la prédiction de 90%.

La profondeur de prévision d'instructions (nombre d'instructions lues d'avance) atteint 20 à 30

Des instructions de déplacement de données (MOV) conditionnelles ont été ajoutées.

Grâce à son tampon d'accès à la mémoire, le P6 peut lancer un nouveau transfert de données avec la mémoire pendant qu'un autre est en cours. Ce tampon permet d'avoir jusqu'à 8 accès en cours simultanément.

Toutefois le P6 a été optimisé pour exécuter du code en 32 bits et s'avère plus lent que le Pentium lorsqu'il traite du code en 16 bits.

### **8.1 Architecture interne et Fonctionnement**

Du point de vue du fonctionnement interne, le P6 peut être considéré comme un processeur RISC. Toutefois, dans la mesure où INTEL voulait qu'il reste compatible avec la famille 80x86, il a été doté d'un décodeur d'instructions constitué d'un pipe-line de 14 niveaux (au lieu de 5 dans le Pentium).

Ce décodeur d'instructions procède de la façon suivante :

Le premier niveau calcule la nouvelle valeur du compteur ordinal en utilisant éventuellement la table de prédiction des branchements

Les trois niveaux suivants vont rechercher 64 octets dans l'antémémoire d'instructions qui est organisé en mots de 32 octets. Ils découpent ces 64 octets en instructions et les transmettent au niveau suivant.

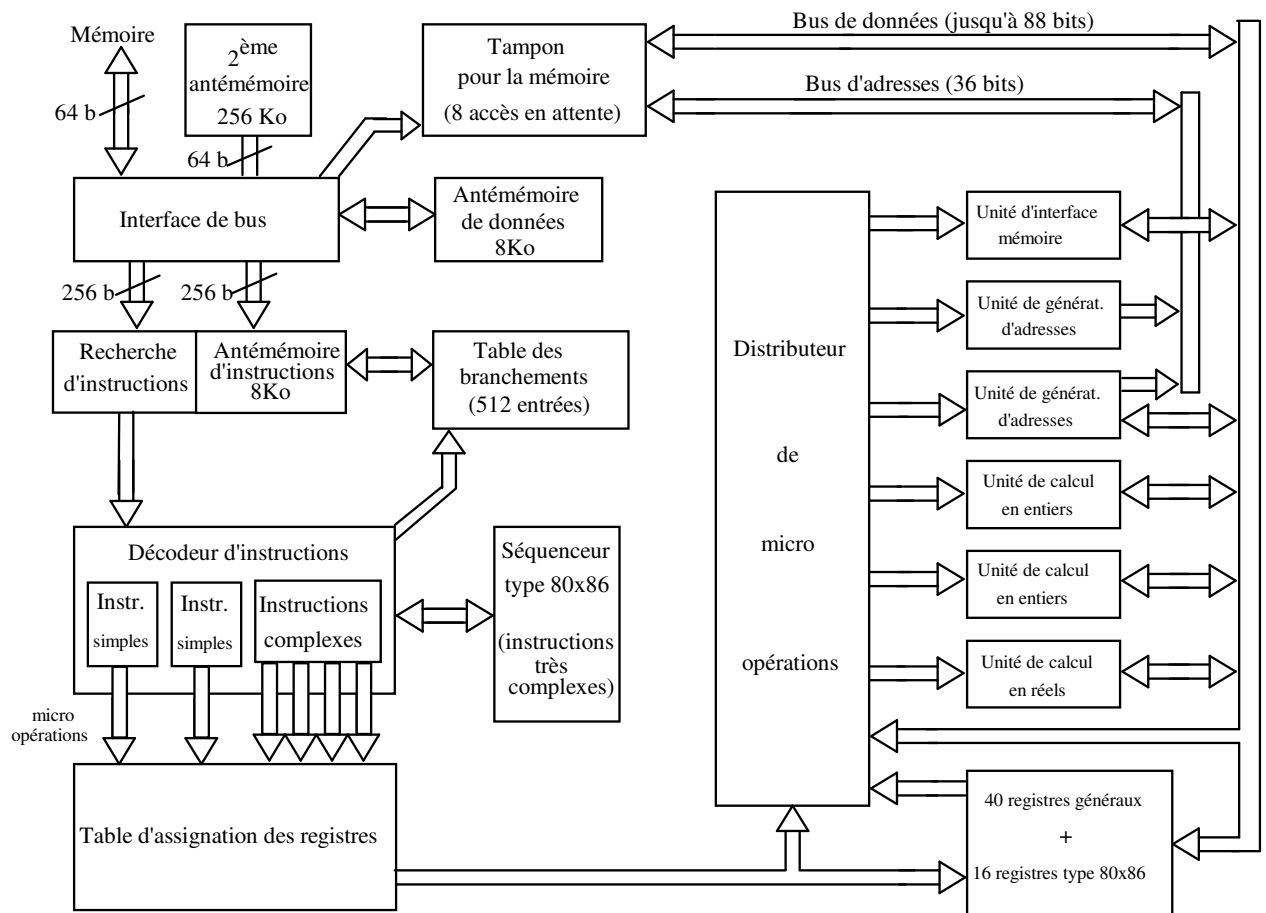
Les 2 niveaux suivants est en fait constitué de 3 unités fonctionnant en parallèle. Deux d'entre-elles se chargent des instructions simples, tandis que la troisième traite les instructions plus complexes. Enfin, les instructions très complexes (comportant des préfixes d'itération par exemple) sont transmises à un séquenceur comparable à celui qui équipait les 80x86.

A l'issue de ce traitement, les instructions ont été transformées en micro-opérations qui ressemblent à des instructions RISC à 3 opérandes. Les instructions 80x86 simples génèrent 1 micro-opération, les autres 4, tandis que celles traitées par le séquenceur (instructions répétitives ou concernant des chaînes de caractères) peuvent en créer jusqu'à 204.

Le 7<sup>ème</sup> niveau assigne les registres du 80x86 à des registres du P6 par le biais d'une table spéciale. Le P6 possède 40 registres généraux en plus des 8 pour les entiers et des 8 pour les réels.

Le 8<sup>ème</sup> niveau ajoute les informations d'état et d'utilisation des registres aux micro-opérations qui ont alors toutes 118 bits de long.

Les niveaux suivants sont constitués d'un distributeur de micro-opérations vers des unités de génération d'adresse, de traitement des entiers ou des réels fonctionnant en parallèle.



## 8.2 Horloge

Le P6 fonctionne avec une horloge à 133, 150 ou 166 MHz mais le bus externe peut fonctionner à 1/4, 1/3 ou 1/2 fois cette vitesse permettant d'utiliser des mémoires RAM de temps d'accès courant (80ns)



## **9 Le Pentium II ou Klamath**

---

Il s'agit d'une version du pentium Pro dotée des registres et des instructions MMX (voir le Pentium MMX ci-dessus). Le problème de lenteur lors de l'exécution de code en 16 bits rencontré par le Pentium Pro a été résolu sur le Pentium II en le dotant du même dispositif d'antémémoire de descripteurs de segments que le Pentium.

Les antémémoires d'instructions et de données sont passées à 16Ko tandis que l'antémémoire de second niveau est de 512Ko à 2Mo selon les modèles.

Le pentium II ne se présente pas sous la forme d'un circuit intégré mais d'un module multi-CI sur un circuit imprimé doté d'un connecteur.

INTEL propose plusieurs modèles qui se différencient par des caractéristiques différentes de ce connecteur (slot1 ou slot 2) et des vitesses d'horloge différentes.

On trouve les vitesses d'horloges suivantes : 200, 233, 266, 350 ou 400 MHz.

Certaines versions du pentium II se voient baptisés de noms comme Xeon, Celeron, Katmai, Dixon, Willamette ... En général les différences ne portent que sur les vitesses d'horloge, les tailles d'antémémoire et le type de connecteur.

## **10 Le P7 ou Merced**

---

C'est un processeur de type VLIW (Very Long Instruction Word) c'est à dire possédant des instructions très longues dont chacune contient plusieurs opérations indépendantes. L'intérêt majeur de ce type d'instructions est qu'elles sont très puissantes tout en pouvant être exécutées très rapidement dans la mesure où les différentes opérations qui les constituent peuvent être traitées en parallèle (elles ne sont constituées que d'opérations indépendantes).

Pour ce genre de processeurs il est indispensable de disposer de compilateurs capables de regrouper plusieurs instructions indépendantes en une seule instruction longue (le compilateur détecte lui même les instructions pouvant être regroupées).

Ce principe a été adoptée par Intel et HP pour l'architecture 64 bits IA-64 du Merced. Il est compatible avec la famille 80x86 et les PA-RISC de HP.

Le Merced constitue le premier élément de la gamme baptisée EPIC (Explicit Parallel Instruction Computer).

### **10.1 Principes**

\* Mot d'instruction de taille fixe (128 bits) contenant 3 instructions simples. Des bits dans le mot d'instruction permettent d'indiquer au CPU ce qui peut être exécuté en parallèle (il faut donc que les compilateurs soient capables de détecter le parallélisme).

\* Exécution en parallèle des deux branches d'une alternative puis abandon des résultats obtenus dans la mauvaise branche.

### **10.2 Instructions**

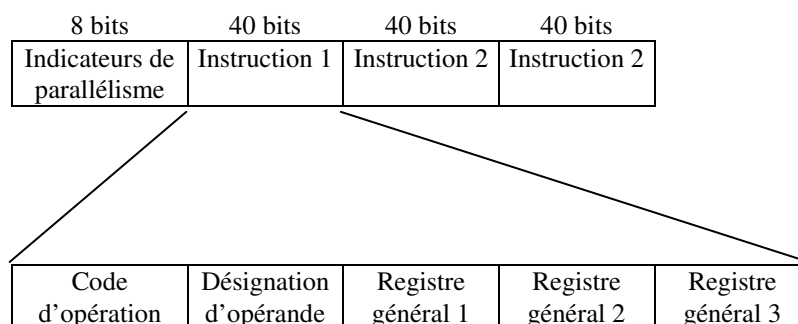
Les instructions sont regroupées 3 par 3 dans un mot de 128 bits. Chaque instruction contient :

Un code d'opération

La désignation de 3 registres généraux (parmi 128 pour les entiers et 128 pour les réels)

La désignation d'un opérande ne mémoire.

La forme générale est donc la suivante :



### **10.3 Branchements**

En général, le CPU n'exécute pas de branchement. Les compilateurs organisent les 2 branches d'une alternative en utilisant le parallélisme entre instructions placées dans le même mot d'instruction. Ainsi le CPU exécute les deux cas comme s'ils étaient non conditionnels.

Le compilateur a marqué chaque instruction par un prédicat (sur 6 bits). Lorsqu'une condition est évaluée, un prédicat devient VRAI et l'autre FAUX. Tous les calculs faits par des instructions associées au prédicat FAUX sont oubliés et ceux associés au prédicat VRAI sont conservés.

Il existe 64 registres (P0 à P63) pour stocker ces prédicats.

Les instructions de branchement traditionnelles restent pour pallier les déficiences des compilateurs qui ne sauraient pas organiser les instructions avec prédicats. Dans ce cas, le principe utilisé est celui de la prédiction habituelle (cf Pentium).

### **10.4 Erreurs de mémoire**

les compilateurs tentent de placer les instructions de recherche d'un opérande en mémoire le plus tôt possible de façon à ce que le CPU ne soit pas bloqué en attente d'un opérande non encore obtenu de la mémoire.

Cela pose toutefois le problème des exceptions levées par le CPU lors de la non disponibilité d'un opérande en mémoire. Il faut, en effet, éviter que le CPU ne lève une exception lors de la recherche d'un opérande non disponible en mémoire alors que cet opérande ne devait de toute façon pas être pris car il était utilisé dans une branche d'alternative qui sera abandonnée ensuite. Dans une telle situation, il faut que le CPU mette de côté l'exception et ne la lève que si elle s'était produite dans un branche d'alternative qui doit être conservée.

### **10.5 Horloge**

La technologie à 0,18 microns devrait permettre d'atteindre des vitesses d'horloge de 1GHz.

## **11 Le Mc Kinley**

C'est le successeur du Merced, il s'agit d'un processeur 64 bits cadencé à 1GHz.

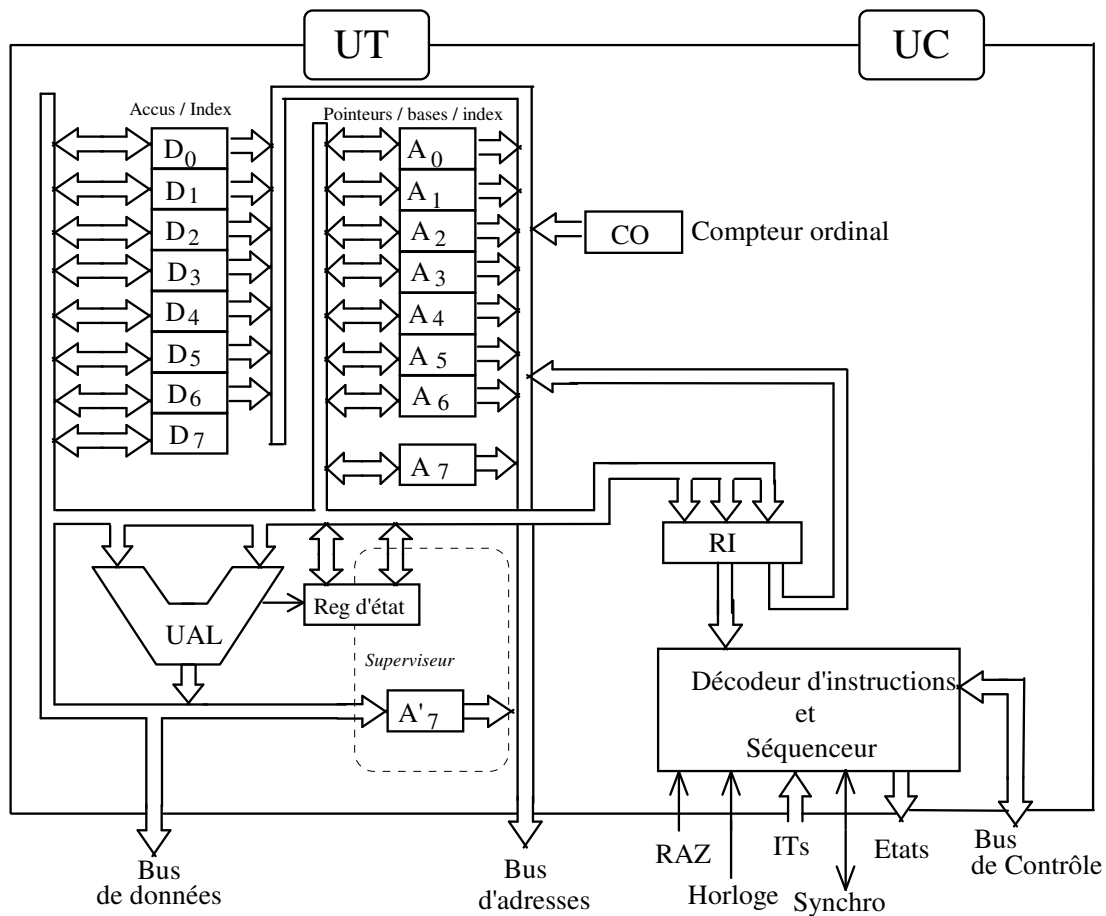
## Microphotographie de la puce du 68000

# LES MICROPROCESSEURS MOTOROLA

## 1 Le 68000 et le 68008

Le 68008 est exactement identique au 68000 sauf que son bus de données est sur 8 bits et, par conséquent, que les accès à la mémoire seront faits en plusieurs fois. Ceci mis à part, tout ce sera dit par la suite sur le 68000 est tout aussi valable pour le 68008.

### 1.1 Architecture interne



### Erreur ! Les arguments du commutateur ne sont pas spécifiés.

Dans la conception du 68000, l'effort n'a pas porté, comme c'était le cas pour le 8086, sur le parallélisme entre l'exécution et les bus mais sur le nombre et l'universalité des registres.

Bien qu'étant un microprocesseur 16 bits, le 68000 est doté de registres de 32 bits et l'architecture interne est celle d'un 32 bits.

Le 68000 peut adresser 8 Méga ( $2^{23}$ ) mots de 16 bits en mémoire. Bien qu'organisée en mots de 16 bits, la mémoire est accessible octet par octet ainsi qu'en mots longs de 32 bits. Les adresses internes sont sur 24 bits dont les 23 bits de fort poids servent à désigner le mot de mémoire tandis que le bit de faible poids est utilisé, lorsque l'opérande est sur un octet, pour désigner cet octet dans le mot (octet de fort ou de faible poids).

On peut remarquer, qu'une fois encore, la démarche de MOTOROLA a été guidée par des critères de facilité de mise en place de langages évolués et de systèmes d'exploitation modernes. Ainsi on trouve des registres banalisés : les uns à la fois accumulateurs et index, les autres à la fois base, pointeurs et index.

De plus, le 68000 est doté de 2 niveaux d'utilisation : utilisateur et superviseur, ce dernier mode permettant l'emploi d'instructions privilégiées.

## 1.2 Les Registres

Le 68000 possède 17 registres de 32 bits :

**D0** à **D7** : accessibles en 8, 16 ou 32 bits, ils servent d'accumulateur et d'index.

**A0** à **A6** : accessibles en 16 ou 32 bits, ils servent de pointeurs, de base ou d'index.

**A7** : registre pointeur de pile de l'utilisateur.

**A'7** : registre pointeur de pile du superviseur (accessible seulement en mode superviseur)

Il comporte de plus :

**CO** : Compteur ordinal sur 32 bits dont 24 seulement sont utilisés.

**SR** : Registre d'état sur 16 bits. Il est divisé en 2 octets dont l'un n'est accessible qu'en mode superviseur. La partie de l'utilisateur contient les indicateurs de l'UAL (signe, résultat nul, débordement, retenue). La partie du superviseur contient les masques d'interruption et les bits de mode trace et superviseur.

## 1.3 Les opérandes

Le 68000 traite les opérandes suivants :

- Entiers naturels sur 8, 16 ou 32 bits
- Entiers relatifs sur 8, 16 ou 32 bits en complément à 2
- Décimaux codés en DCB sur 2 digits
- Booléens sur 1 bit
- Pointeurs sur 32 bits dont 24 seulement sont utilisés

## 1.4 Les instructions

Le 68000 possède le jeu d'instruction suivant:

### **Arithmétiques**

#### Addition :

Il existe 6 types d'addition dont les mnémoniques sont : **ADD ADDA ADDI ADDQ ADDX** et **ABCD**

Ils permettent l'addition binaire sur les registres Dj (ADD) ou Aj (ADDA), avec des valeurs immédiates (ADDI, ADDQ), avec un bit d'extension (ADDX) sur 8, 16 ou 32 bits. Ainsi que l'addition décimale en DCB sur 8 bits (ABCD).

#### Soustraction :

On retrouve les 6 types correspondants pour la soustraction dont les mnémoniques sont : **SUB SUBA SUBI SUBQ SUBX** et **SBCD**.

#### Multiplication :

Elle est effectuée entre 2 opérandes sur 16 bits et le résultat est obtenu sur 32 bits. Deux types sont proposés selon que les opérandes sont signés (**MULS**) ou non signés (**MULU**).

#### Division :

Elle est effectuée entre un dividende sur 32 bits et un diviseur sur 16 bits pour produire un quotient et un reste chacun sur 16 bits. Deux types sont proposés selon que les opérandes sont signés (**DIVS**) ou non signés (**DIVU**).

#### Mise à zéro :

**CLR** fait une mise à zéro d'un opérande sur 8 16 ou 32 bits.

#### Extension de signe :

**EXT** étend la représentation d'un nombre signé de 8 à 16 bits ou de 16 à 32 bits.

#### Changement de signe :

Il en existe 3 types dont les mnémoniques sont : **NEG NEGX** et **NBCD**. Ils permettent de changer le signe d'un opérande binaire sur 8, 16 ou 32 bits (**NEG**) avec un bit d'extension (**NEGX**). Ainsi que pour un décimal codé en DCB sur 8 bits (**NBCD**).

### **Logiques**

#### Comparaison :

Il existe 4 types de comparaison dont les mnémoniques sont : **CMP CMPA CMPI** et **CMPM** selon que l'opérande est un registre Dj (**CMP**) ou Aj (**CMPA**), une variable en mémoire (**CMPM**) ou une valeur immédiate (**CMPI**). La comparaison porte sur 8, 16 ou 32 bits.

#### ET logique :

Deux types **AND** et **ANDI** selon que l'opérande est un registre (**AND**) ou une valeur immédiate (**ANDI**). L'opération porte sur 8, 16 ou 32 bits.

#### OU logique :

Deux types **OR** et **ORI** selon que l'opérande est un registre (**OR**) ou une valeur immédiate (**ORI**). L'opération porte sur 8, 16 ou 32 bits.

#### OU exclusif :

Deux types **EOR** et **EORI** selon que l'opérande est un registre (**EOR**) ou une valeur immédiate (**EORI**). L'opération porte sur 8, 16 ou 32 bits.

#### Complément :

**NOT** complément logique sur 8, 16 ou 32 bits

#### Test :

**TST** comparaison à zéro d'un opérande sur 8, 16 ou 32 bits.

### **Opérations sur les bits**

**BCHG** : test puis inversion de la valeur d'un bit dans un mot de 8 ou 32 bits

**BCLR** : test puis mise à zéro d'un bit dans un mot de 8 ou 32 bits.

**BSET** : test puis mise à 1 d'un bit dans un mot de 8 ou 32 bits.

**BTST** : test d'un bit dans un mot de 8 ou 32 bits.

### **Transferts**

#### Transfert simple :

Il en existe 5 types dont les mnémoniques sont : **MOVE MOVEA MOVEM MOVEP** et **MOVEQ** selon que l'opérande est un registre Dj (**MOVE**) ou Aj (**MOVEA**), une variable en mémoire (**MOVE**) ou une valeur immédiate (**MOVEI**).

**MOVEM** permet de transférer plusieurs registres et **MOVEP** permet des transferts d'octets plus particulièrement destinés au pilotage de périphériques

#### Echange :

**EXG** échange le contenu de deux registres.

#### Chargement d'adresse :

**LEA** permet de mettre dans un registre Aj l'adresse de l'opérande.

#### Empilement d'adresse :

**PEA** permet d'empiler l'adresse de l'opérande.

### **Décalages**

Tous les décalages peuvent être faits sur 8, 16 ou 32 bits.

#### Arithmétiques :

**ASL / ASR** à gauche / à droite.

#### Logiques :

**LSL / LSR** à gauche / à droite.

Cycliques :

**ROL / ROR** et avec un bit d'extension **ROXL** et **ROXR**.

Echange entre fort poids et faible poids d'un mot de 32 bits :

**SWAP**.

## Ruptures de séquence

Branchement inconditionnel :

**BRA** ou **JMP** selon que l'opérande est un déplacement (sur 8 ou 16 bits) ou une adresse

Branchements conditionnels :

si égal	<b>BEQ</b>
si différent	<b>BNE</b>
si <	<b>BCS</b> entre entiers naturels <b>BLT</b> entre relatifs
si < ou =	<b>BLS</b> entre entiers naturels <b>BLE</b> entre relatifs
si >	<b>BHI</b> entre entiers naturels <b>BGT</b> entre relatifs
si > ou =	<b>BCC</b> entre entiers naturels <b>BGE</b> entre relatifs
si débordement	<b>BVS</b>
si non débordement	<b>BVC</b>
si positif	<b>BPL</b>
si négatif	<b>BMI</b>

Répétitions :

Ce sont des instructions constituées sur le modèle des branchements conditionnels (les tests effectués sont les mêmes) mais, lorsque la condition est fausse, un registre Dj est décrémenté et le branchement est effectué si la valeur de ce registre est différente de -1.

Lorsque la condition est vraie on continue en séquence.

Remarque : 2 conditions sont ajoutées à la liste ci-dessus : T toujours vrai et F toujours faux.

Les mnémoniques sont ceux du branchement conditionnel correspondant précédés de la lettre D.

Appel de sous programme :

**BSR** ou **JSR** selon que l'opérande est un déplacement (sur 8 ou 16 bits) ou une adresse

Retour de sous programme :

Il existe 2 modes: **RTR** et **RTS** selon que l'on dépile le CO seul (**RTS**) ou avec le registre condition (**RTR**).

## Affectation conditionnelle

L'opérande sur 8 bits reçoit 11111111 si la condition est vraie et 00000000 sinon. Les conditions sont les mêmes que pour les répétitions. Les mnémoniques sont ceux du branchement conditionnel correspondant précédés de la lettre S.

## Instructions orientées vers les langages évolués

Ce sont des instructions permettant le passage de paramètres aux procédures.

**LINK** empile le registre Aj spécifié, copie la valeur du pointeur de pile dans ce registre puis ajoute la valeur contenue dans l'instruction au pointeur de pile. **LINK** réserve donc une zone dans la pile pour les paramètres et prépare (après avoir sauvegardé son ancienne valeur) le registre Aj pour devenir un pointeur sur cette zone.

**UNLK** transfère le registre Aj spécifié dans le pointeur de pile puis dépile le registre Ai.

**CHK** génère une interruption si le registre désigné est inférieur à 0 ou supérieur à la valeur de l'opérande sur 16 bits. Cette instruction permet de vérifier les limites d'un tableau.

**TRAP** provoque l'exécution de l'interruption dont le numéro est calculé à partir de la valeur (0 à 15) placée dans l'instruction.

**TRAPV** provoque l'exécution de l'interruption de numéro 7 si l'indicateur de débordement est positionné.

Remarque : Les interruptions seront décrites plus en détail en 1.9.

## Sémaphores

**TAS** réalise l'opération de "Test And Set" permettant la réalisation de sémaphores. L'opérande est un octet dont seul le bit 7 est testé puis mis à 1.

Toutes les instructions décrites jusqu'à présent sont accessibles en mode utilisateur du moins sous les conditions décrites ci-après. Nous allons maintenant répertorier les instructions privilégiées qui sont réservées au mode superviseur.

### **Instructions privilégiées**

**ANDI ORI** et **EORI** modifiant le registre d'état.

**MOVE** portant sur le registre d'état ou le registre pointeur de pile.

**RESET** active la ligne physique de réinitialisation normalement connectée aux éléments externes (contrôleurs de mémoire, de périphériques etc).

**RTE** retour de sous programme fonctionnant comme RTR (avec restitution du registre SR) mais à partir de la pile du superviseur.

**STOP** copie la valeur précisée dans l'instruction dans le registre d'état puis arrête le 68000. Le redémarrage aura lieu lors d'une réinitialisation (reset) ou de la réception d'une interruption de priorité supérieure à celle indiquée dans les bits de masque d'interruption du registre d'état.

## **1.5 Mémoire et adressage**

La mémoire est organisée en 8 Méga ( $2^{23}$ ) mots de 16 bits. Le bus d'adresses du 68000 comporte 23 lignes et celui de données 16.

Toutefois le 68000 permet, dans la plupart de ses instructions, d'utiliser des opérandes sur 8 bits en ne lisant que la moitié du mot de mémoire. Il utilise pour cela 2 lignes supplémentaires UDS et LDS. UDS permet de n'accéder qu'à l'octet de fort poids (bits 8 à 15) et LDS à celui de faible poids (bits 0 à 7).

Une lecture ou une écriture en mémoire se déroule sur 4 périodes d'horloge :

- durant la 1<sup>ère</sup> période, le signal indiquant s'il s'agit d'une lecture ou d'une écriture (R/W) est positionné en même temps que l'adresse.
- durant la 2<sup>ème</sup> période, l'adresse est validée par le signal AS et les signaux d'accès aux octets (LDS , UDS) sont émis.
- durant la 3<sup>ème</sup> période, le transfert sur le bus de données est effectué.
- durant la 4<sup>ème</sup> période, tous les signaux retrouvent leur état de repos.

La possibilité d'insérer des cycles d'attente entre la 3<sup>ème</sup> et la 4<sup>ème</sup> période permet d'accéder à des mémoires ou des contrôleurs de périphériques lents.

Le 68000 possède les modes d'adressage suivants :

#### **inhérent :**

l'instruction n'a pas d'opérande

#### **registre :**

l'opérande est l'un des registres Dj ou Aj

#### **direct ou étendu :**

l'opérande est désigné par son adresse absolue sur 16 bits (direct) ou sur 32 bits (étendu). Dans le premier cas la valeur est étendue à 32 bits par recopie du bit de fort poids.

#### **indirect :**

l'adresse est obtenue à partir du contenu d'un registre Aj suivant les méthodes suivantes :

#### **indirect simple :**

l'adresse de l'opérande est le contenu du registre A désigné.



avec **post-incrémentation** :

comme l'indirect simple mais le registre Aj utilisé est augmenté de 1, 2 ou 4 selon la taille de l'opérande.

avec **pré-décrémentation** :

le registre Aj utilisée est tout d'abord diminué de 1, 2 ou 4 selon la taille de l'opérande puis sa valeur est considérée comme adresse.

avec **déplacement** :

l'adresse de l'opérande est obtenue par addition au contenu du registre Aj d'un entier relatif sur 16 bits placé dans l'instruction.

avec **index et déplacement** :

l'adresse de l'opérande est obtenue par addition au contenu du registre Aj du contenu d'un registre index Aj ou Dj et d'un entier relatif sur 8 bits placé dans l'instruction.

**relatif** :

Ce mode est utilisable de deux façons :

. Pour les instructions de rupture de séquence autres que JMP ou JSR : L'adresse où aller est obtenue en ajoutant au CO courant un déplacement constitué d'un entier relatif sur 8 ou 16 bits placé dans l'instruction.

. Pour les autres instructions : L'adresse de l'opérande peut être obtenue par calcul à partir de la valeur courante du CO selon les modes suivants :

**relatif avec déplacement** :

l'adresse de l'opérande est obtenue par addition au CO d'un entier relatif sur 16 bits placé dans l'instruction.

**relatif avec index et déplacement** :

l'adresse de l'opérande est obtenue par addition au CO du contenu d'un registre index Aj ou Dj et d'un entier relatif sur 8 bits placé dans l'instruction.

Remarque : Les instructions JMP et JSR peuvent utiliser les 2 modes relatifs décrits ci-dessus ainsi que les modes indirect simple, avec déplacement, avec index et déplacement, direct et étendu.

**immédiat** :

La valeur de l'opérande est dans l'instruction sur 8, 16 ou 32 bits.

## **1.6 Horloge**

L'horloge de séquençement du 68000 peut être de 8 , 10 , 12.5 ou 16MHz. Le 68008 n'est, quant à lui, disponible qu'en 8 ou 10MHz.

En ce qui concerne les instructions, le 68000 possède une file d'attente de 2 mots en plus du registre instruction. Il procède de la façon suivante :

Quand l'exécution d'une instruction commence, le mot de 16 bits contenant le code opération et le mot suivant ont été lus. Si l'instruction est sur plusieurs mots, chaque fois qu'un mot est utilisé de façon interne un nouveau mot est lu pour le remplacer.

Le 68000 effectue donc une prévision d'instruction sur un seul mot mais, contrairement au 8086, ce n'est pas une unité particulière qui effectue les recherches d'instructions.

La durée d'une instruction dépend :

- . du mode d'adressage : 0 à 12 cycles supplémentaires.
- . de la taille de l'opérande : 4 cycles supplémentaires pour un opérande sur 32 bits.
- . de l'opération proprement dite : 4 à 16 cycles pour un transfert, 38 à 70 cycles pour la multiplication et 158 pour la division signée, 10 cycles pour un branchement conditionnel, 18 pour un appel de sous-programme et 16 à 20 pour un retour de sous-programme.
- . de l'éventuelle utilisation de cycles d'attente pour les accès à la mémoire.

## 1.7 Les Entrées/Sorties

Comme c'est l'habitude chez MOTOROLA, le 68000 ne possède aucune instruction d'entrée/sortie et les contrôleurs de périphériques sont adressés comme des zones mémoire.

Le 68000 génère des signaux compatibles avec ceux des microprocesseurs 8 bits (6800 et 6809) de façon à pouvoir travailler avec les contrôleurs de périphériques développés pour ces derniers.

## 1.8 Privilèges

Le processeur connaît 2 niveaux de privilège : utilisateur et superviseur. Il les distingue par la valeur du bit S de la partie superviseur de registre d'état.

Le passage du niveau superviseur au niveau utilisateur est obtenu par toute instruction ayant pour effet de modifier le bit S du registre d'état. Le passage du niveau utilisateur au niveau superviseur ne peut avoir lieu que lors de la prise en compte d'une interruption.

On peut remarquer que le 68000 ne possédant pas de gestion de mémoire, l'existence de ces 2 modes doit être considérée comme une facilité pour la mise en place de protection et non comme une vraie sécurité.

Toutefois le 68000 génère 3 signaux FC<sub>0</sub> à FC<sub>2</sub> indiquant quel est le type de transfert en cours sur le bus (code ou données de l'utilisateur, code ou données du superviseur, acceptation d'interruption).

On peut, en utilisant ces signaux, mettre en place un dispositif physique efficace de protection de la mémoire.

## 1.9 Les Interruptions

La prise en compte d'une interruption se fait de la façon suivante :

- 1°) Sauvegarde du registre d'état et passage en mode superviseur
- 2°) Détermination du numéro de l'interruption
- 3°) Sauvegarde du contexte
- 4°) Passage au nouveau contexte

Il existe une table (placée en mémoire à partir de l'adresse physique 0) contenant 255 vecteurs (un vecteur est un mot de 32 bits constituant une adresse à l'exception du vecteur associé à la réinitialisation qui est sur 64 bits et contient, outre l'adresse, la valeur du pointeur de pile du superviseur) associée aux interruptions.

Une interruption peut être provoquée de façon matérielle ou logicielle.

### - **logicielle :**

Ce peut être l'instruction TRAP suivie d'un numéro (0 à 15) correspondant aux interruptions 32 à 47.

Ce peut être l'instruction TRAPV qui provoque l'interruption de numéro 7 si le bit de débordement est positionné.

Ce peut être aussi l'instruction CHK qui provoque l'interruption de numéro 6 si les limites sont dépassées.

Ce peuvent être enfin les instructions DIVS ou DIVU si le diviseur est nul (interruption de numéro 5).

### - **matérielle :**

Ce peut être l'activation de la ligne RESET qui provoque l'interruption de numéro 0.

Ce peut être l'activation des 3 lignes d'interruption dont les valeurs permettent d'indiquer un niveau de priorité de 1 à 7 (le niveau 7 correspond à l'interruption non masquable). Si l'interruption est prise en compte (priorité suffisante), le 68000 en demande le numéro qui doit lui être envoyé par l'auteur de l'interruption sur le bus de données. Si aucune valeur n'est envoyée, les interruptions 25 à 31 sont prises par défaut en fonction du numéro de priorité. L'absence de réponse provoque la prise en compte de l'interruption de n° 24, tandis qu'un contrôleur de périphérique non initialisé provoque l'interruption de n° 15.

Ce peut être la découverte d'un code opération inconnu (Interruption de n° 4, 10 ou 11 selon la valeur du code opération).

Ce peut être une tentative de violation de privilège (instruction interdite en mode utilisateur) qui provoque l'interruption de n° 8.

Ce peut être le fonctionnement en mode trace qui exécute une interruption de n° 9 après chaque instruction.

Ce peut être aussi l'activation de la ligne BERR qui indique une erreur de bus et qui provoque l'interruption de n° 2.

Ce peut être enfin une détection d'erreur d'adressage (interruption de n° 2) qui correspond à la désignation d'un opérande de type mot ou d'une instruction comme étant constitué de 2 octets pris dans 2 mots différents. Ceci correspond au problème d'adresses paires ou impaires qu'INTEL résout par des accès multiples et que MOTOROLA considère comme une erreur puisque le 68000 a été voulu comme un vrai 16 bits.

### **1.10 Environnement multiprocesseur**

D'un point de vue matériel, 2 signaux permettent au 68000 de partager ses bus avec d'autres processeurs :

**BG** qui permet au 68000 d'indiquer qu'il va libérer les bus dès la fin du cycle en cours.

**BGACK** qui indique au 68000 qu'un autre processeur a pris possession de ses bus.

**BR** qui permet à un autre processeur de demander l'accès aux bus du 68000.

D'un point de vue logiciel, l'instruction TAS (Test And Set) permet la mise en place de sémaphores dans la mesure où elle est ininterrompible et où les bus sont verrouillés pendant toute la durée de cette instruction.

## 2 Le 68010 et le 68012

---

La seule différence entre le 68010 et le 68012 concerne l'espace mémoire adressable :

Le 68010 peut accéder (comme le 68000) à 8M ( $2^{23}$ ) mots de 16 bits alors que le 68012 accède à 1 Giga ( $2^{30}$ ) mots de 16 bits en deux blocs disjoints de 512 Méga mots ( $2^{29}$ ) de 16 bits (son bus d'adresses comporte les lignes A1 à A29 et la ligne A31). Ceci mis à part, tout ce qui sera dit par la suite concernant le 68010 est tout aussi valable pour le 68012.

### 2.1 Architecture interne

Le 68010 présente la même architecture que le 68000 à laquelle viennent s'ajouter quelques registres (voir 2.2) ainsi que le dispositif physique permettant de suspendre l'exécution d'une instruction lors d'une erreur de bus (défaut de page mémoire virtuelle) et de la reprendre dès que le sous programme lié à l'interruption de numéro 2 a été exécuté (chargement de la page depuis une mémoire de masse).

Contrairement aux 80286 et 80386, le 68010 ne gère pas lui-même la mémoire virtuelle mais possède la faculté de suspendre puis de poursuivre des cycles d'accès à la mémoire. Ceci lui permet de fonctionner avec un gestionnaire externe de mémoire virtuelle.

### 2.2 Les registres

Le 68010 possède les 17 registres de 32 bits D0 à D7, A0 à A7 et A7 décrits pour le 68000. Il comporte de plus :

- CO** compteur ordinal sur 32 bits dont 24 seulement sont utilisés par le 68010 et 31 par le 68012
- SR** registre d'état identique à celui du 68000.

Les 3 registres suivants ne sont accessibles qu'en mode superviseur :

- VBR** sur 32 bits, il constitue le pointeur sur la table de vecteurs d'interruptions. Cette table qui était impérativement à l'adresse 0 sur le 68000 peut, sur le 68010 être placée n'importe où en mémoire. Il est surtout possible grâce à cette méthode de gérer plusieurs tables.

- SFC** sur 3 bits, il contient l'état que l'on imposera au 68010 de présenter sur les lignes FC<sub>0</sub> à FC<sub>2</sub> lors de la lecture provoquée par l'instruction MOVES.

- DFC** comme SFC lors de l'écriture provoquée par le MOVES.

Ces 2 registres et l'instruction **MOVES** (utilisable en mode superviseur seulement) permettent de transgresser les niveaux de privilèges même lorsque la mémoire de l'utilisateur et la mémoire du superviseur sont physiquement séparées et distinguées grâce aux lignes FC<sub>0</sub> à FC<sub>2</sub> par un gestionnaire de mémoire externe.

### 2.3 Les opérandes

Le 68010 traite les mêmes opérandes que le 68000.

### 2.4 Les instructions

Le 68010 possède toutes les instructions du 68000 parmi lesquelles les deux suivantes ont été modifiées:

**MOVE** mettant en jeu la partie de l'utilisateur du registre d'état : Le 68000 ne permettait de transférer que vers ce registre alors que le 68010 permet aussi d'en ranger le contenu en mémoire ou dans un autre registre.

**MOVE** mettant en jeu le registre d'état devient une instruction privilégiée quel que soit le sens du transfert (le 68000 ne privilégiait que le transfert vers ce registre).

**RTE** permet de restituer des contextes courts (4 mots comme le 68000) ou longs (29 mots) pour les traitements d'erreurs

Le 68010 possède en outre 4 instructions supplémentaires :

**MOVEC** instruction privilégiée qui effectue un transfert entre un registre Dj ou Aj et l'un des 4 registres de contrôle SFC , DFC , A7 et VBR.

**MOVES** instruction privilégiée qui effectue un transfert entre un registre Dj ou Aj et un mot de la mémoire en transgressant les privilèges grâce à l'utilisation des registres SFC et DFC (voir 2.2).

**RTD** effectue un retour de sous-programme avec désallocation de la zone de la pile réservée aux paramètres.

**BKPT** permet la mise en place de points d'arrêt lors de la mise au point de programmes.

## **2.5 Mémoire et Adressage**

L'organisation de la mémoire et les types d'adressage sont les mêmes que sur le 68000 (voir 1.5).

Lorsqu'une erreur de bus est signalée au 68010 par l'activation de la ligne BERR, le cycle en cours est suspendu jusqu'à ce que BERR redevienne inactive.

A ce moment, le 68010 entame la procédure de traitement de l'erreur d'accès à la mémoire en empilant:

- le registre d'état (1 mot)
- le compteur ordinal (2 mots)
- le vecteur d'interruption (1 mot)
- l'état interne au moment de l'erreur (25 mots dont 22 utilisés)

puis en exécutant l'interruption de numéro 2.

Le traitement associé à cette interruption peut résoudre le problème ayant causé l'erreur (chargement d'une page de mémoire virtuelle par exemple) puis relancer le 68010 par une instruction RTE. Le cycle suspendu sera alors terminé sauf si l'erreur était survenue lors d'une instruction TAS qui sera réexécutée.

Le traitement associé à cette interruption peut aussi terminer par logiciel l'accès suspendu en utilisant les tampons prévus à cet effet dans la pile (3 des 22 mots de l'état interne) puis relancer le 68010 par un RTE après avoir modifié un indicateur de l'état interne pour indiquer au microprocesseur qu'il ne doit pas poursuivre le cycle suspendu mais le considérer comme terminé.

## **2.6 Horloge**

L'horloge de séquençement du 68010 peut être de 8 , 10 ou 12.5 MHz.

En ce qui concerne la durée d'exécution des instructions peu de choses ont changé par rapport au 68000. On peut toutefois citer 3 améliorations sensibles :

- . Le 68010 n'effectue pas la lecture systématique (en aveugle) de l'opérande comme le faisait le 68000 ce qui permet de gagner 2 cycles lorsque les instructions n'utilisent pas cet opérande.
- . Les opérations de multiplication et division ont été accélérées : 40 à 42 cycles pour la multiplication et 122 pour la division signée.
- . Lors de l'exécution d'instructions de répétition (DBxx), le 68010 détecte les situations dans lesquelles la boucle tout entière peut contenir sur 3 mots. Il place alors 2 de ces 3 mots dans la file d'attente d'instructions et l'autre dans le registre instruction puis n'effectue plus de recherche d'instruction en mémoire tant que la condition de boucle reste vraie.

## **2.7 Les Entrées/Sorties**

Le 68010 présente les mêmes signaux que le 68000 de façon à pouvoir travailler avec les contrôleurs de périphériques de la famille 8 bits.

## **2.8 Privilèges**

Le 68010 reprend le principe des 2 niveaux de privilège définis sur le 68000. Les seules améliorations apportées à ce fonctionnement sont les instructions devenues privilégiées, les instructions supplémentaires (voir 2.4) et les 3 registres permettant de transgresser les niveaux de privilège et de gérer plusieurs tables de vecteurs d'interruptions (voir 2.2).

## **2.9 Les Interruptions**

On ne décrira dans ce chapitre que ce qui a changé vis à vis du 68000 (voir 1.9).

Le 68000 ne sauvegardait, lors de la prise en compte d'une interruption, que le registre d'état et le compteur ordinal. Le 68010 possède 2 types de sauvegardes selon l'interruption :

**format court** (4 mots) correspond aux interruptions normales  
registre d'état (1 mot)  
compteur ordinal (2 mots)  
format/déplacement (1 mot)

Dans le dernier mot le format est 0 sur 4 bits pour désigner le format court, le déplacement est celui qu'il faut effectuer depuis le début de la table des vecteurs d'interruptions pour accéder à la bonne entrée.

**format long** (29 mots) correspond aux interruptions liées à des erreurs (de bus ou d'adresse) et à la réinitialisation (RESET).

registre d'état (1 mot)  
compteur ordinal (2 mots)  
format/déplacement (1 mot)  
mot d'état (1 mot)  
adresse où a eu lieu l'erreur (2 mots)  
réservé extensions futures (1 mot)  
tampon de sortie (1 mot)  
réservé extensions futures (1 mot)  
tampon d'entrée (1 mot)  
réservé extensions futures (1 mot)  
tampon d'instruction (1 mot)  
informations internes (16 mots)

Les informations supplémentaires permettent le traitement des erreurs (voir 2.5 par exemple).

Le format contenu dans le 4<sup>ème</sup> mot est 1000.

L'instruction de retour de traitement d'interruption (RTE) utilise les 4 bits de format pour déterminer si elle doit dépiler en format long ou court. L'interruption de numéro 14 est déclenchée lorsque les 4 bits de format du 4<sup>ème</sup> mot ne sont pas 0000 ou 1000.

## **2.10 Environnement multiprocesseur**

Le 68010 possède les lignes BR, BG et BGACK permettant le partage des bus (voir 1.A). L'instruction TAS (Test And Set) permet la réalisation de sémaphores dans la mesure où elle utilise un type particulier d'accès à la mémoire (Read-ModifyWrite) reconnu par le 68010 comme ne pouvant pas être repris après une suspension causée par une interruption d'erreur.

Lorsqu'une erreur se produit au cours d'une instruction TAS, la terminaison du traitement d'interruption n'entraîne pas une poursuite du cycle en cours mais une reprise du cycle complet (Lecture-Modification-Ecriture).

Le 68010 possède une ligne **RMC** qui est activée pendant toute la durée d'un cycle de type (Read-Modify-Write) et qui peut être utilisée pour faire un verrouillage physique du bus (équivalent de la ligne LOCK des processeurs INTEL).

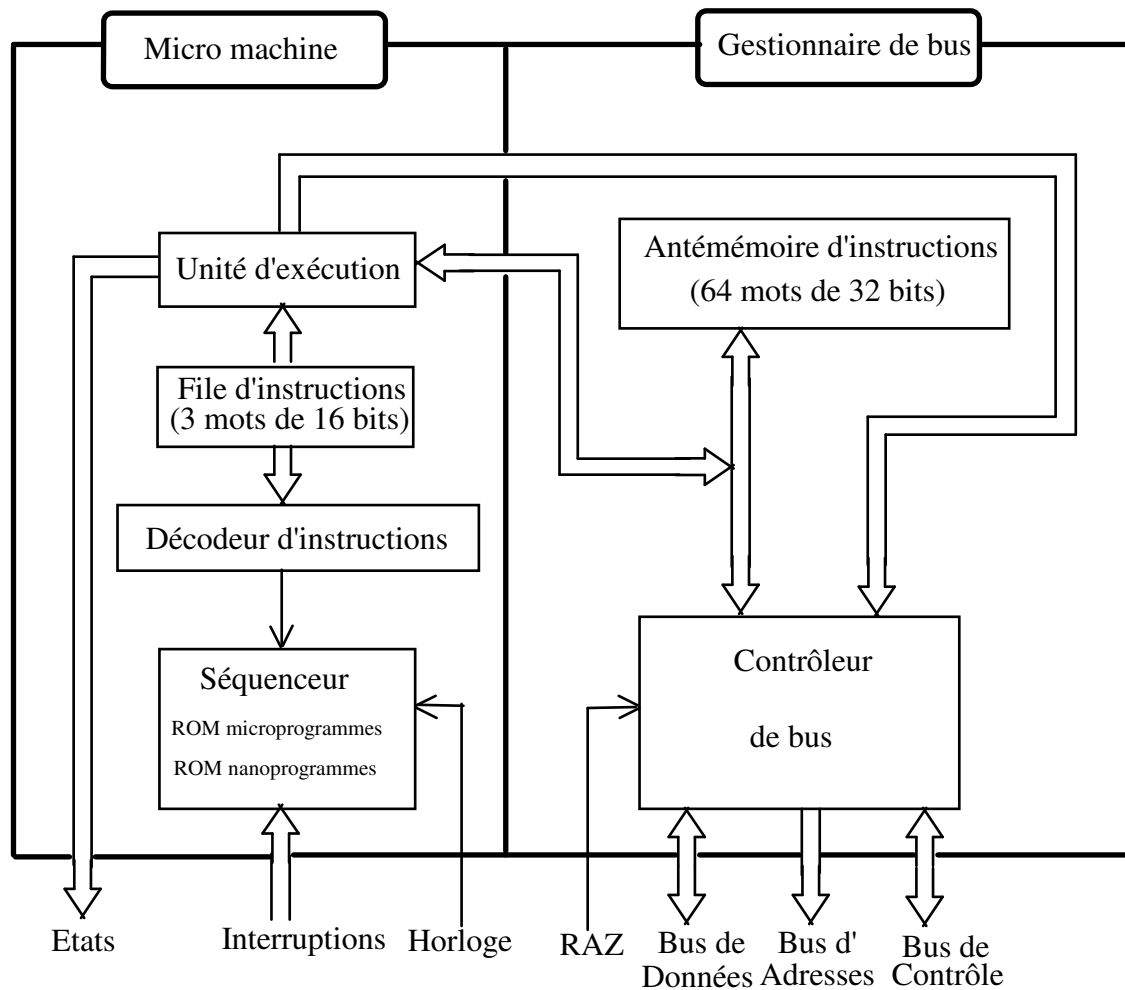
### 3 Le 68020

#### 3.1 Architecture interne

Le 68020 est constitué de 2 unités :

- L'unité de gestion de bus
- La micro machine

Il peut être représenté par le schéma suivant :



L'unité de gestion de bus se charge :

- du contrôle physique des bus
- de l'accès aux opérandes
- de la recherche des instructions et de leur mise dans l'anté-mémoire

La **micro machine** exécute les instructions de l'antémémoire de façon microprogrammée à 2 niveaux (microcode, nanocode) en raison de la complexité du jeu d'instructions.

Ces deux unités fonctionnent en parallèle de façon à optimiser l'utilisation des bus.

L'**unité d'exécution** est fabriquée selon une architecture en pipe-line permettant à plusieurs instructions de s'exécuter en chevauchement.



Le bus de données est de 32 bits mais l'accès à des opérandes sur 8 , 16 ou 32 bits est possible. L'espace mémoire adressable est de 1 Giga ( $2^{30}$ ) mots de 32 bits.

### 3.2 Registres

Le 68020 possède tous les registres du 68010.

. Le registre d'état **SR** utilise 2 bits supplémentaires permettant d'introduire un nouveau mode trace et un indicateur de traitement d'interruption.

Le 68020 possède 3 registres supplémentaires :

Le registre A7' (SSP) des 68000 et 68010 est remplacé par 2 registres :

A7' (**ISP**) pointeur sur la pile utilisée lors du traitement des interruptions

A7" (**MSP**) pointeur de pile superviseur.

Deux registres liés à l'antémémoire d'instructions sont ajoutés :

**CACR** registre de contrôle de l'antémémoire

**CAAR** registre adresse de l'antémémoire

### 3.3 Les opérandes

Le 68020 traite les opérandes suivants :

- Entiers naturels sur 8 16 32 ou 64 bits
- Entiers relatifs sur 8 16 32 ou 64 bits en complément à 2
- Décimaux codés en DCB sur 1 ou 2 digits
- Booléens sur 1 bit
- Suites de bits de 1 à 32 bits
- Pointeurs sur 32 bits

### 3.4 Les instructions

Outre les instructions décrites pour le 68000, le 68020 possède les extensions suivantes :

#### **Arithmétiques**

Multiplication : Les opérandes peuvent être sur 16 ou 32 bits et le résultat sur 32 ou 64 bits.

#### Division :

Le dividende peut être sur 32 ou 64 bits et le diviseur sur 16 ou 32 bits. Le reste et le quotient sont toujours de la taille du diviseur.

Deux instructions **DIVSL** et **DIVSU** permettent de traiter un dividende et un diviseur sur 32 bits pour obtenir un quotient et un reste chacun sur 32 bits avec signe (**DIVSL**) ou sans signe (**DIVSU**).

#### Extension de signe :

A l'instruction **EXT** qui étend la représentation d'un nombre signé de 8 à 16 bits ou de 16 à 32 bits a été ajoutée l'instruction **EXTB** qui étend la représentation d'un nombre signé de 8 à 32 bits

#### Compactage/Décompactage des décimaux :

l'instruction **PACK** effectue le compactage d'un décimal de 2 chiffres représenté sous le format d'un chiffre par octet en une représentation sur un seul octet (1 chiffre sur 4 bits).

L'instruction **UNPK** effectue l'opération inverse.

#### **Logiques**

### Comparaison :

Aux 4 modes de comparaison CMP, CMPA, CMPI et CPM vient s'ajouter **CMP2** qui compare le contenu d'un registre à 2 valeurs consécutives en mémoire sur 8, 16 ou 32 bits et positionne l'indicateur de retenue lorsque le registre est hors des limites définies par ces 2 opérandes.

### **Instructions orientées vers les langages évolués**

Le 68020 possède deux instructions permettant la programmation par modules :

**CALLM** qui empile le contexte du module courant, prépare la pile pour recevoir les paramètres dont le nombre est désigné par l'instruction et prend un nouveau contexte dont l'adresse est désignée par l'instruction.

**RTM** qui termine l'exécution d'un module en dépilant le contexte du module appelant et en déplaçant le pointeur de pile au-delà de la zone utilisée pour le passage des paramètres.

Et deux instructions de contrôle :

**CHK2** qui génère une interruption si le registre désigné est strictement hors des limites définies par l'opérande (voir CMP2 ci-dessus).

**TRAPxx** qui génère une interruption si la condition exprimée par xx est vraie. xx peut être toute condition reconnue par les branchements conditionnels Bxx (voir 1.4).

### **Traitement des suites de bits**

**BFCHG** complémente une suite de bits

**BFCLR** met à 0 une suite de bits

**BFSET** met à 1 une suite de bits

**BFEXTS** extrait une suite de bits et en étend la représentation à un entier relatif sur 32 bits

**BFEXTU** extrait une suite de bits et en étend la représentation à un entier naturel sur 32 bits

**BFFFO** indique le rang du premier bit à 1 dans une suite de bits

**BFINS** recopie la suite de bits de faible poids d'un registre dans la suite de bits de même taille de l'opérande

**BFTST** compare à 0 une suite de bits

### **Instructions pour multi-processeurs ou coprocesseur**

#### Instructions orientées vers les multi-processeurs

**TAS** : Test And Set déjà existant sur le 68000

**CAS** : C'est une extension du Test And Set. L'opérande est comparé au contenu d'un registre Dj (sur 8, 16 ou 32 bits). S'il y a égalité, le contenu d'un 2<sup>ème</sup> registre Dj (valeur de mise à jour) est écrit à la place de l'opérande, sinon la valeur de l'opérande vient remplacer le contenu du registre de comparaison.

On peut donc décrire CAS par l'algorithme suivant :

comparer l'opérande et le registre de comparaison

si égalité alors

    opérande <- registre de mise à jour

sinon

    registre de comparaison <- opérande

fsi

**CAS2** : étend à 2 opérandes les fonctions de CAS :

comparer l'opérande 1 et le registre de comparaison 1

si égalité alors

```

comparer l'opérande 2 et le registre de comparaison 2
si égalité alors
    opérande 1 <- registre de mise à jour 1
    opérande 2 <- registre de mise à jour 2
fsi
sinon
    registre de comparaison 1 <- opérande 1
    registre de comparaison 2 <- opérande 2
fsi

```

CAS2 met donc en jeu 6 registres :

- 2 registres Dj de comparaison
- 2 registres Dj de mise à jour
- 2 registres Aj ou Dj contenant les adresses des opérandes

### Instructions orientées vers les coprocesseurs

MOTOROLA a doté le 68020 d'instructions destinées à le faire fonctionner avec des coprocesseurs de la famille 68xxx.

Les instructions liées aux coprocesseurs possèdent un champ de 3 bits permettant de désigner 8 coprocesseurs.

A l'heure actuelle MOTOROLA réserve 6 codes dont 2 seulement sont disponibles :

- 68851 gestionnaire de pagination de mémoire
- 68881 coprocesseur arithmétique

Les autres codes sont laissés à la disposition des utilisateurs pour des coprocesseurs particuliers.

Ces instructions possèdent un préfixe permettant de désigner le co-processeur, dans la suite ce préfixe sera représenté par cp :

**cpBxx** : effectue un branchement si la condition exprimée par xx et transmise pour évaluation au coprocesseur est vraie.

**cpDBxx** : teste une condition comme cpBxx mais, lorsque la condition est vraie, on continue en séquence alors que lorsqu'elle est fausse le registre Dj désigné par l'instruction est décrémenté et un branchement a lieu si sa valeur est différente de -1.

**cpGEN** : permet de transmettre une commande et un opérande à un conditions xx désignées ci-dessus dépendent du coprocesseur. Elles sont codées sur 6 bits dans l'instruction et transmises ainsi au co-processeur.

### 3.5 Mémoire et adressage

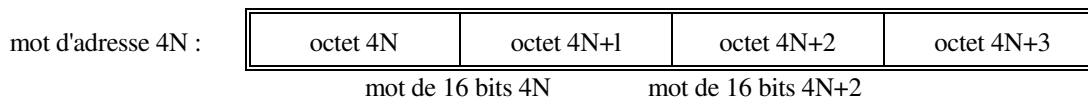
La taille du bus de données peut être dynamiquement modifiée en indiquant au 68020 par 2 signaux **DSACK<sub>1</sub>** et **DSACK<sub>0</sub>** la taille choisie :

DSACK1	DSACK0	signification
0	0	insertion de cycles d'attente
0	1	bus sur 8 bits
1	0	bus sur 16 bits
1	1	bus sur 32 bits

Parallèlement, le 68020 indique lors de chaque accès la taille de l'opérande à l'aide des 2 lignes  $SIZ_1$  et  $SIZ_0$  :

$SIZ_1$	$SIZ_0$	taille du bus
0	0	8 bits
0	1	16 bits
1	0	24 bits
1	1	32 bits

Le bus d'adresses est donc prévu pour désigner des octets, les adresses paires désignent des mots de 16 bits et les adresses multiples de 4 des mots de 32 bits. La mémoire est organisée en mots de 32 bits. Une adresse de mot est donc de type  $4N$  et, à l'intérieur du mot, l'organisation est la suivante :



Le 68020, contrairement aux 68000 et 68010, n'exige aucun alignement d'adresse pour les opérandes toutefois, selon l'adresse, le nombre d'accès ne sera pas le même :

	adresse $4N$	adresse $4N+1$	adresse $4N+2$	adresse $4N+3$
octet	1 cycle	1 cycle	1 cycle	1 cycle
mot de 16 bits	1 cycle	1 cycle	1 cycle	2 cycles
mot de 32 bits	1 cycle	2 cycles	2 cycles	2 cycles

Les instructions doivent impérativement être à des adresses paires. La taille d'une instruction est toujours un multiple de mots de 16 bits de sorte que l'alignement soit conservé dès lors que le programme est implanté à partir d'une adresse paire.

Cette contrainte assure la compatibilité au niveau du code avec les 68000 et 68010 et permet d'accélérer la recherche des instructions.

En fonction de la taille de l'opérande ( $SIZ_1$ ,  $SIZ_0$ ) et du bus ( $DSACK_1$  et  $DSACK_0$ ), le 68020 positionne correctement, à chaque accès, l'opérande sur les 32 lignes constituant le bus de données.

Un cycle d'accès à la mémoire se déroule en 3 périodes d'horloge :

- 1<sup>ère</sup> période : le signal indiquant s'il s'agit d'une lecture ou d'une écriture (R/W), l'adresse et les signaux  $SIZ_1$   $SIZ_0$  sont émis.
- 2<sup>ème</sup> période : l'adresse est validée par le signal AS et le transfert a lieu.
- 3<sup>ème</sup> période : le transfert se termine et les signaux retrouvent leur état de repos.

Des cycles d'attente peuvent être insérés, lorsque c'est nécessaire, entre la 2<sup>ème</sup> et la 3<sup>ème</sup> période. Ainsi, selon l'alignement et la taille de l'opérande, un accès complet occupera 3 ou 6 périodes d'horloge.

Le 68020 gère à la façon du 68010 les erreurs de bus permettant la mise en place d'une mémoire virtuelle (voir 2.5).

Le 68020 possède les modes d'adressage du 68000 (voir 1.5) modifiés et augmentés de la façon suivante :

**relatif :**

Les déplacements peuvent être exprimés sur 8, 16 ou 32 bits (8 ou 16 sur les 68000 et 68010).

**indirect avec index et déplacement :**

Le fonctionnement reste le même que sur le 68000 mais l'instruction contient en outre un indicateur d'échelle qui permet d'utiliser dans le calcul de l'adresse le contenu de l'index multiplié par 1, 2, 4 ou 8. Le déplacement peut être sur 8, 16 ou 32 bits (8 seulement sur le 68000).

**indirect mémoire :**

L'opérande est désigné par le biais d'un pointeur contenu dans un mot de mémoire selon l'une des 2 méthodes suivantes :

**indirect mémoire avec post indexation :**

L'adresse du pointeur en mémoire est obtenue par addition au contenu d'un registre Aj d'un entier relatif sur 16 ou 32 bits. L'adresse de l'opérande est obtenue par addition au contenu de ce pointeur en mémoire d'un registre index Aj ou Dj préalablement multiplié par 1, 2, 4 ou 8 (facteur d'échelle) puis d'un déplacement final constitué d'un entier relatif sur 16 ou 32 bits.

**indirect mémoire avec pré indexation :**

L'adresse du pointeur en mémoire est obtenue par addition au contenu d'un registre Aj d'un entier relatif sur 16 ou 32 bits et d'un registre index Aj ou Dj préalablement multiplié par 1, 2, 4 ou 8 (facteur d'échelle). L'adresse de l'opérande est obtenue par addition au contenu de ce pointeur en mémoire d'un déplacement final constitué d'un entier relatif sur 16 ou 32 bits.

### **3.6 Horloge**

L'horloge de séquençement du 68020 peut-être de 12.5 , 16.67 , 20 , 25 ou 33 MHz.

Grâce à son antémémoire de 64 mots de 32 bits et à la nécessité d'alignement en adresses paires des instructions (voir 3.5), le 68020 effectue les lectures d'instructions sous forme de mots de 32 bits.

La présence d'un contrôleur de bus permet au 68020 de remplir l'antémémoire d'instructions pendant que la micro-machine fonctionne et de faire les accès à la mémoire relatifs à l'exécution d'une instruction pendant qu'une autre instruction est lancée (overlap).

Le temps d'exécution d'une instruction variera donc en fonction des 2 paramètres suivants :

- l'instruction est ou non dans l'antémémoire
- l'instruction s'effectue ou pas en recouvrement avec une autre.

### **3.7 Les Entrées/Sorties**

Le 68020 ne possède aucune instruction d'entrée/sortie. Il ne possède pas non plus les signaux compatibles avec les contrôleurs de périphériques de la famille 8 bits dont étaient dotés les 68000 et 68010. Il est donc nécessaire soit d'utiliser les nouveaux contrôleurs de MOTOROLA (famille 16 et 32 bits) soit d'adapter les signaux existants.

On peut toutefois remarquer que le bus de données du 68020 étant dynamique (voir 3.5), l'accès à des registres 8 bits ne pose pas de problème.

### **3.8 Privilèges**

Le 68020 possède, comme le 68000 et le 68010, les 2 niveaux de privilège utilisateur et superviseur.

### **3.9 Les interruptions**

Le 68000 ne possède qu'un seul format de pile d'interruption, le 68010 en possède 2 et le 68020 en possède 6 :

formats 0 et 1 (4 mots) : identiques au format court du 68010 (voir 2.9). Le format 1 est réservé à la sauvegarde de contextes transitoires causés par la prise en compte d'une interruption pendant un traitement d'erreur.

format 2 (6 mots) : les 2 mots additionnels sont l'adresse de l'instruction qui a causé l'interruption (CHK, CHK2, cpTRAPxx, TRAPxx, TRAPV, mode trace et division par 0).

format 9 (10 mots) : ce format de pile est utilisé lors d'interruptions liées au coprocesseur. Les 4 mots additionnels contiennent les valeurs des registres internes.

format 10 (16 mots) : ce format de pile est utilisé lorsqu'une erreur de bus survient alors que la dernière instruction lue l'a été en entier (pas de reprise à faire).

format 11 (46 mots) : ce format de pile est utilisé lorsqu'une erreur de bus survient alors que la dernière instruction lue ne l'a pas été en entier. Ce format contient, outre les informations du format 10, le contexte permettant de faire la reprise d'instruction lorsque la cause de l'erreur aura été corrigée.

L'instruction de retour d'interruption (RTE) reconnaît ces différents formats. L'interruption de n° 14 est déclenchée lorsque le format n'est pas reconnu.

Le 68020 connaît 2 modes de trace (génération d'une interruption de n° 9) :

- génération de l'interruption après chaque instruction
- génération de l'interruption après chaque instruction de rupture de séquence

La liste des interruptions reconnues est étendue par rapport à celle des 68000 et 68010 (voir 1.9) :

- Interruption de n° 13 : violation de protocole coprocesseur
- Interruptions de n°48 à 54 : erreurs détectées par le coprocesseur arithmétique
- Interruptions de n° 56 à 58 : erreurs détectées par le coprocesseur de gestion de mémoire.

### **3.10 Environnement multiprocesseur et coprocesseur**

Du point de vue des signaux de partage de bus, on retrouve ceux décrits pour le 68010 en 2.A.

Le jeu d'instructions relatif au fonctionnement en multi-processeur a été étendu par l'adjonction de CAS et CAS2 (voir 3.4).

En ce qui concerne la communication avec des coprocesseurs, elle n'est plus émulée par logiciel mais intégrée dans le microcode. Ainsi un ensemble d'instructions propres aux coprocesseurs a été défini et intégré dans la micro-machine (voir 3.4).

Les coprocesseurs apparaissent dans l'espace d'adressage du 68020 sous la forme de 8 mots de 32 bits contenant les registres suivants :

**Réponse** (16 bits) : reçoit les réponses faites par le coprocesseur.

**Contrôle** (16 bits) : permet au 68020 de signaler au coprocesseur la prise en compte d'une réponse de type erreur ou d'arrêter un traitement en cours dans le coprocesseur.

**Sauvegarde** (16 bits) : reçoit le mot d'état du coprocesseur pendant une instruction de type cpSAVE.

**Restitution** (16 bits) : reçoit le mot d'état du coprocesseur pendant une instruction de type cpRESTORE.

**Opération** (16 bits) : reçoit le code de l'opération demandée au coprocesseur.

**Commande** (16 bits) : le 68020 écrit dans ce registre pour lancer le coprocesseur.

**Condition** (6 bits) : condition à tester par le coprocesseur lors d'instructions conditionnelles.

**Opérande** (32 bits) : permet d'échanger des opérandes entre le 68020 et le coprocesseur. Si l'opérande occupe plus de 32 bits il est lu ou écrit dans ce registre en plusieurs fois.

**Sélection** (16 bits) : utilisé par le coprocesseur, lorsqu'il demande au 68020 le transfert de l'un de ses registres, pour désigner ce registre.

**Adresse instruction** (32 bits) : utilisé par le 68020, à la demande du coprocesseur, pour indiquer l'adresse de l'instruction en cours.

**Adresse opérande** (32 bits) : utilisé par le 68020, à la demande du coprocesseur, pour indiquer l'adresse d'un opérande.

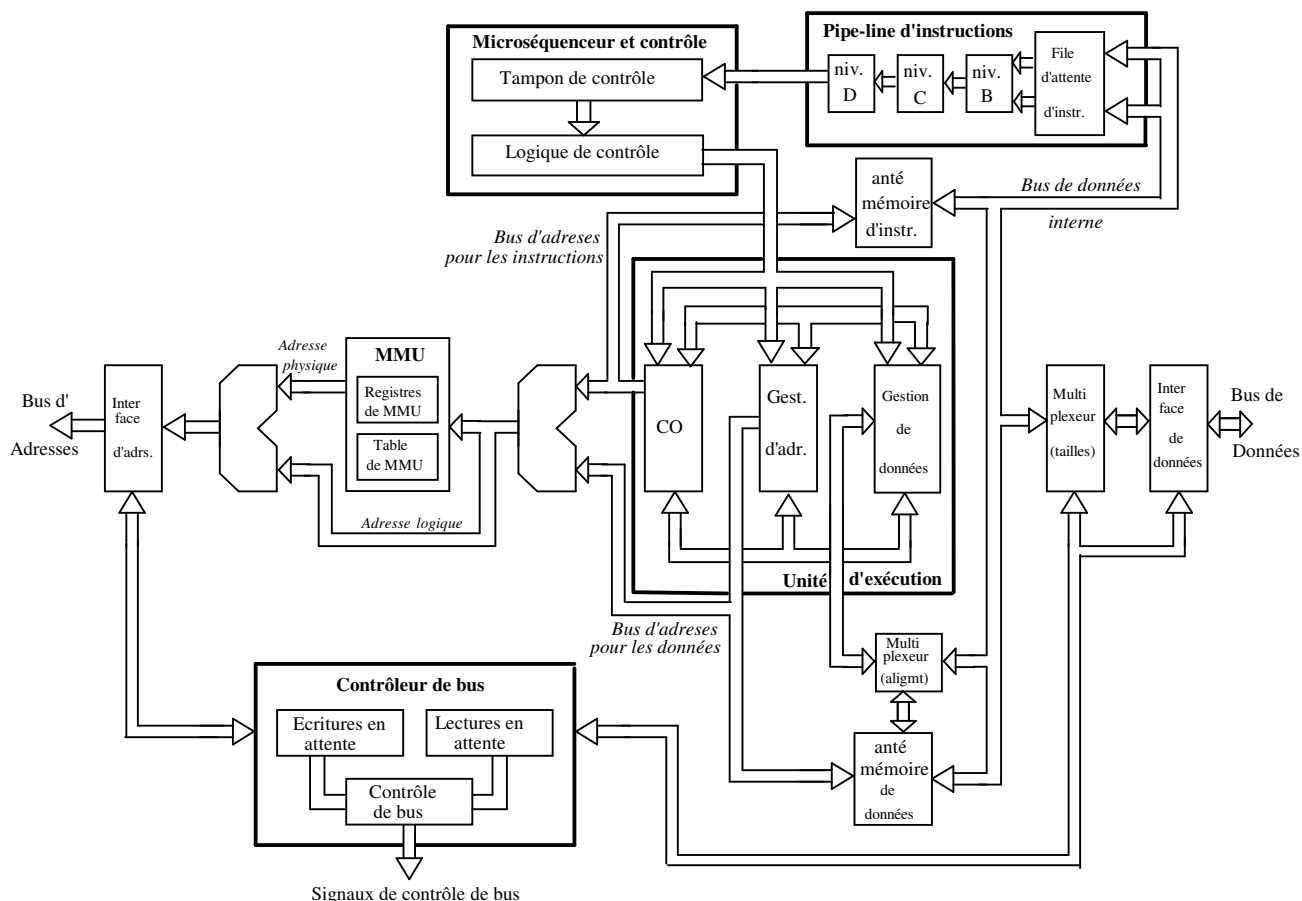
Les coprocesseurs peuvent provoquer la prise en compte d'une interruption d'erreur par le 68020 en positionnant la réponse appropriée dans le registre de réponse. Trois types d'interruption sont possibles

- erreur pré-instruction : après traitement de l'interruption le 68020 relancera l'instruction.
- erreur mi-instruction : après traitement de l'interruption le 68020 continuera l'instruction (reprise d'instruction).
- erreur post-instruction : après traitement de l'interruption le 68020 passera à l'instruction suivante. Ces 3 types correspondent respectivement aux formats 0 , 9 et 2 de piled'interruption (voir 3.9).

## 4 Le 68030

Le 68030 est une version améliorée du 68020 qui intègre une pagination de mémoire et une antémémoire pour les données. L'architecture interne a été redéfinie pour obtenir de meilleures performances.

### 4.1 Architecture interne



Le contrôleur de bus comporte les amplificateurs de données et d'adresses (data pads, address pads), les multiplexeurs permettant la gestion de la taille du bus de données (size multiplexer) et le micro-contrôleur qui enchaîne les accès selon les priorités (micro bus controller).

. La **micromachine** contient l'unité de traitement (execution unit) et toute la logique de contrôle. Comme le 68020, le 68030 est microprogrammé à deux niveaux (microcode et nanocode).

. Le **pipe-line d'instructions** (instruction pipe) assure le décodage des instructions et génère les signaux qui permettent le décodage et l'interprétation du microcode et du nanocode.

. Les **antémémoires** d'instructions et de données fonctionnent indépendamment du reste de la machine. Elles stockent les informations lues par le contrôleur de bus. Leur taille est de 64 mots de 32 bits.

. L'**unité de gestion de mémoire** (MMU) contrôle le découpage de la mémoire en pages de 256 octets à 32K octets. Les informations de pagination sont placées dans des tables en mémoire dont les entrées les plus récemment utilisées sont dans une mémoire associative (ATC : Address Translation Cache) de 22 mots.



De plus, le 68030 possède deux registres permettant de définir deux blocs de 16M à 2G octets chacun non assujettis à la pagination (TT registers : Transparent Translation registers).

## 4.2 Les Registres

Le 68030 contient les registres décrits pour le 68020 (voir 3.2).

. Le registre **CACR** (registre de contrôle de l'antémémoire) utilise des bits supplémentaires relatifs au contrôle de l'antémémoire de données qui n'existait pas sur le 68020 (voir 4.5).

. On trouve de plus 6 registres associés à la gestion la mémoire intégrée :

**CRP** (CPU Root Pointer) sur 64 bits, pointe sur la racine de l'arbre de pagination de la tâche en cours d'exécution. Cet arbre contient les informations de pagination de l'espace d'adressage alloué à cette tâche.

**SRP** (Supervisor Root Pointer) sur 64 bits, c'est le CRP associé aux tâches en mode superviseur.

**TC** (Tranlation Control) sur 32 bits, ce registre permet de valider ou d'interdire la pagination ainsi que l'utilisation du registre SRP. Il définit la taille des pages de mémoire et la structure de l'arbre de pagination.

**MMUSR** (Memory Management Unit Status Register) sur 16 bits, il contient les résultats des recherches dans la mémoire associative ou dans l'arbre de pagination.

**TT<sub>0</sub>** et **TT<sub>1</sub>** (Transparent Translation) 2 registres de 32 bits chacun qui permettent de définir 2 blocs de mémoire comme étant exclus de la pagination (l'adresse logique est l'adresse physique). Ces blocs sont utilisés pour faire des déplacements rapides de données (pages graphiques par exemple) sans subir les délais du gestionnaire de pagination.

## 4.3 Les instructions.

Les deux instructions permettant la programmation par modules CALLM et RETM dont était doté le 68020 ont été supprimées.

. 5 instructions privilégiées relatives à la gestion de mémoire ont été introduites :

**PMOVE** : instruction de transfert entre un opérande et un registre du gestionnaire de mémoire.

**PLOAD** : instruction qui place un nouveau descripteur dans la mémoire associative du gestionnaire de mémoire.

**PTEST** : instruction qui teste une adresse logique, le résultat, dans MMUSR, contient entre autres les informations suivantes :

- erreur de bus pendant la recherche
- adresse protégée en écriture
- adresse non répertoriée dans la pagination
- adresse dans un bloc défini par les registres TT<sub>0</sub> ou TT<sub>1</sub>
- page correspondante modifiée (par une écriture)

**PFLUSHA** : instruction qui invalide tous les descripteurs de la mémoire associative.

**PFLUSH** : instruction qui invalide tous les descripteurs de la mémoire associative correspondant à la description fournie.

## 4.4 Mémoire et Adressage

Les modes d'adressage du 68030 sont ceux du 68020 et l'espace mémoire réel est aussi de 1 Giga ( $2^{30}$ ) mots de 32 bits. Les différences portent sur l'existence de l'antémémoire de données et du gestionnaire de mémoire.

### Fonctionnement des antémémoires

Les antémémoires sont organisées en 16 blocs de 4 mots de 32 bits.

Lorsque c'est nécessaire, le contrôleur de bus remplace les informations contenues dans l'un des blocs.

Il est possible, grâce à la ligne CDIS ou par programmation, d'interdire au 68030 d'utiliser ses antémémoires.

L'antémémoire de données utilise la méthode "d'écriture totale" (write through) c'est à dire que, lorsqu'une écriture a lieu, elle est effectuée à la fois dans l'antémémoire et dans la mémoire. La donnée écrite est disponible dans l'antémémoire pour les instructions suivantes même si l'écriture effective en mémoire n'a pas encore eu lieu.

Le remplissage des antémémoires peut se faire de 2 façons :

mot par mot : Le contrôleur de bus charge un seul mot de 32 bits à la fois parmi les 4 qui constituent le bloc de l'antémémoire.

par bloc (burst mode) : cette méthode n'est possible que si la mémoire permet ce type d'accès. Les 4 mots du bloc sont chargés en commençant par celui qui a provoqué le transfert.

### Contrôle des antémémoires

2 registres sont associés aux antémémoires :

**CACR** (Cache Control Register) sur 32 bits, il utilise les 11 bits suivants :

**WA** (Write Allocate) : permet de choisir de faire ou de ne pas faire la mise à jour de l'antémémoire à chaque écriture dans une adresse mémoire dont l'image est dans l'antémémoire. Ceci permet de résoudre les problèmes posés par l'accès à une adresse physique par le biais de 2 adresses logiques différentes permis par le gestionnaire de mémoire. En effet il est, dans ce cas, possible que l'antémémoire contienne 2 images de cette adresse physique vues au travers des 2 adresses logiques. L'écriture par l'une des adresses logiques doit provoquer la mise à jour des 2 images dans l'antémémoire de l'adresse physique correspondante.

**DBE** (Data Burst Enable) : autorise ou interdit l'accès à la mémoire en mode bloc (burst mode).

**CD** (Clear Data cache) : permet de libérer tous les blocs de l'antémémoire de données.

**CED** (Clear Entry in Data cache) : permet de libérer le bloc de l'antémémoire de données désigné par le registre CAAR.

**FD** (Freeze Data cache) : permet de verrouiller le contenu de l'antémémoire (plus de chargement de blocs).

**ED** (Enable Data cache) : permet d'autoriser ou d'interdire l'utilisation de l'antémémoire.

**IBE** (Instruction Burst Enable) : comme DBE pour l'antémémoire d'instructions

**CI** (Clear Instruction cache) : comme CD pour l'antémémoire d'instructions

**CEI** (Clear Entry in Instruction cache) : comme CED pour l'antémémoire d'instructions.

**FI** (Freeze Instruction cache) : comme FD pour l'antémémoire d'instructions.

**EI** (Enable Instruction cache) : comme ED pour l'antémémoire d'instructions.

**CAAR** (CAche Address Register) sur 32 bits, il contient sur 5 bits le numéro de bloc (voir CED et CEI ci-dessus).

### Modes d'accès à la mémoire

Le 68030 possède 3 modes d'accès à la mémoire :

Mode asynchrone : c'est l'accès en 3 périodes décrit en 3.5 pour le 68020.

Mode synchrone : c'est un accès en 32 bits sur 2 périodes d'horloge :

1<sup>ère</sup> période : émission des signaux R/W, SIZ<sub>1</sub>, SIZ<sub>0</sub>, de l'adresse et de son signal de validation AS.

2<sup>ème</sup> période : le transfert a lieu et se termine dès réception par le 68030 du signal STERM. Si le signal STERM n'est pas reçu dès le début de la 2<sup>ème</sup> période, des cycles d'attente sont introduits.

Mode bloc (burst mode) : c'est le mode de transfert utilisé pour les anté-mémoires. Il se déroule en 5 périodes d'horloge :

1<sup>ère</sup> période : émission des signaux R/W, SIZ<sub>1</sub>, SIZ<sub>0</sub>, de l'adresse et de son signal de validation AS.

2<sup>ème</sup> période : le transfert sur 32 bits a lieu et le signal CBACK est activé pour indiquer au 68030 que le mode bloc est possible.

3<sup>ème</sup> période : transfert du 2<sup>ème</sup> mot

4<sup>ème</sup> période : transfert du 3<sup>ème</sup> mot, le signal CBACK retrouve son état de repos pour indiquer que le transfert suivant sera le dernier.

5<sup>ème</sup> période : transfert du 4<sup>ème</sup> mot, les signaux activés en 1<sup>ère</sup> période retrouvent leur état de repos.

Au cours des 2<sup>ème</sup>, 3<sup>ème</sup>, 4<sup>ème</sup> et 5<sup>ème</sup> périodes le signal STERM peut être utilisé pour insérer des cycles d'attente.

Remarque : L'adresse émise en 1<sup>ère</sup> période est celle relative au 1<sup>er</sup> mot elle ne change pas pendant les périodes suivantes. Une logique externe associée à la mémoire doit se charger de faire évoluer ces adresses. Il faut noter que l'adresse émise par le 68030 n'est pas celle du 1<sup>er</sup> mot du bloc de l'anté-mémoire mais celle du mot qui a provoqué la demande de chargement de ce bloc. Un bloc d'antémémoire contient toujours dans l'ordre des mots mémoire d'adresse 4N, 4N+1, 4N+2, 4N+3 (il s'agit ici d'adresses de mots de 32 bits et non d'adresses d'octets (voir 3.5)).

## **4.5 La pagination**

La mémoire est paginée c'est à dire découpée en pages d'égale longueur. Les adresses logiques sont, elles aussi, divisées en pages de même longueur. Le système d'exploitation se chargera d'assigner des pages physiques aux pages logiques (mémoire virtuelle) à la demande des tâches. Le gestionnaire de mémoire prend en compte les adresses logiques et interroge la mémoire associative (ATC voir 4.1).

### La mémoire associative

La mémoire associative contient les informations suivantes :

. **Adresse logique** sur 24 bits pour des pages de taille 256 octets (taille minimale), le nombre de bits de cette adresse diminue lorsque la taille des pages augmente.

- . **FC** sur 3 bits, correspond au code de fonction (lignes FC<sub>0</sub> à FC<sub>2</sub> du 68030 (voir 1.8)).
- . **V** sur 1 bit, indique si cette entrée est valide ou pas.
- . **Adresse physique** sur 24 bits pour des pages de taille 256 octets, désigne l'adresse de la page en mémoire physique.
- . **B** sur 1 bit, indique si une erreur de bus, un descripteur non validé, une violation de protection ou de limite a été associé à cette entrée de la mémoire associative.
- . **CP** sur 1 bit, permet d'autoriser ou d'interdire l'usage de l'antémémoire sur cette page.
- . **WP** sur 1 bit, permet de protéger la page en écriture.
- . **M** sur 1 bit, indique que la page a été modifiée par une écriture.

Lorsque la mémoire associative ne contient pas l'adresse logique demandée il sera nécessaire d'accéder à l'arbre de pagination en mémoire. La mise à jour de la mémoire associative utilise un pseudo algorithme de LRU (Least Recently Used) qui fournit une espérance de probabilité de présence moyenne de 98% .

#### Arbre de pagination

Il est pointé par le registre CRP (CPU Root Pointer) ou par le registre SRP (Supervisor Root Pointer) selon le niveau de privilège de la tâche en cours.

Il possède une structure arborescente à 2 niveaux contenant des tables de pointeurs constituant les noeuds et des tables de pages en guise de feuilles.

Seule la partie de l'arbre utilisable par la tâche en cours doit nécessairement être résidente en mémoire.

Les tables de pages contiennent des descripteurs de page comportant :

- l'adresse de la page en mémoire physique
- des informations de protection
- des informations relatives à l'histoire de la page

Les tables de pointeurs contiennent soit des descripteurs de pages comme ci-dessus soit des descripteurs de table comportant :

- un pointeur sur la table suivante
- la taille de la table suivante
- des informations de protection valables pour toutes les pages sur lesquelles existe un pointeur depuis cette table
- des informations relatives à l'histoire du descripteur

Chaque table du chemin dans l'arbre est indexée par un champ extrait de l'adresse logique (2 index au maximum dont la valeur ne peut excéder 1023).

Remarque : Les tables de page peuvent aussi contenir des descripteurs indirects qui sont des pointeurs vers des descripteurs de page.

## **4.6 Horloge**

L'horloge de séquençement du 68030 peut être de 16.67 , 20 , 25 ou 33.33 MHz.

Comparé au 68020, le 68030 bénéficie des apports de l'antémémoire de données mais surtout des 2 modes supplémentaires d'accès rapide à la mémoire (voir 4.5).

L'utilisation de la pagination n'introduit pratiquement pas de délais dans la mesure où le gestionnaire de mémoire fonctionne en parallèle avec les autres organes du 68030. De même, les antémémoires sont accessibles simultanément et l'exécution d'une instruction sans accès mémoire peut s'effectuer pendant que le contrôleur de bus assure une écriture en mémoire demandée par une précédente instruction. Le décodage des instructions est effectué par un pipe-line de 4 registres de 16 bits (voir 4.1).

#### **4.7 Les autres caractéristiques**

Les entrées/sorties, les privilèges, les interruptions et l'environnement multi-processeur sont traités dans le 68030 comme ils l'avaient été dans le 68020.

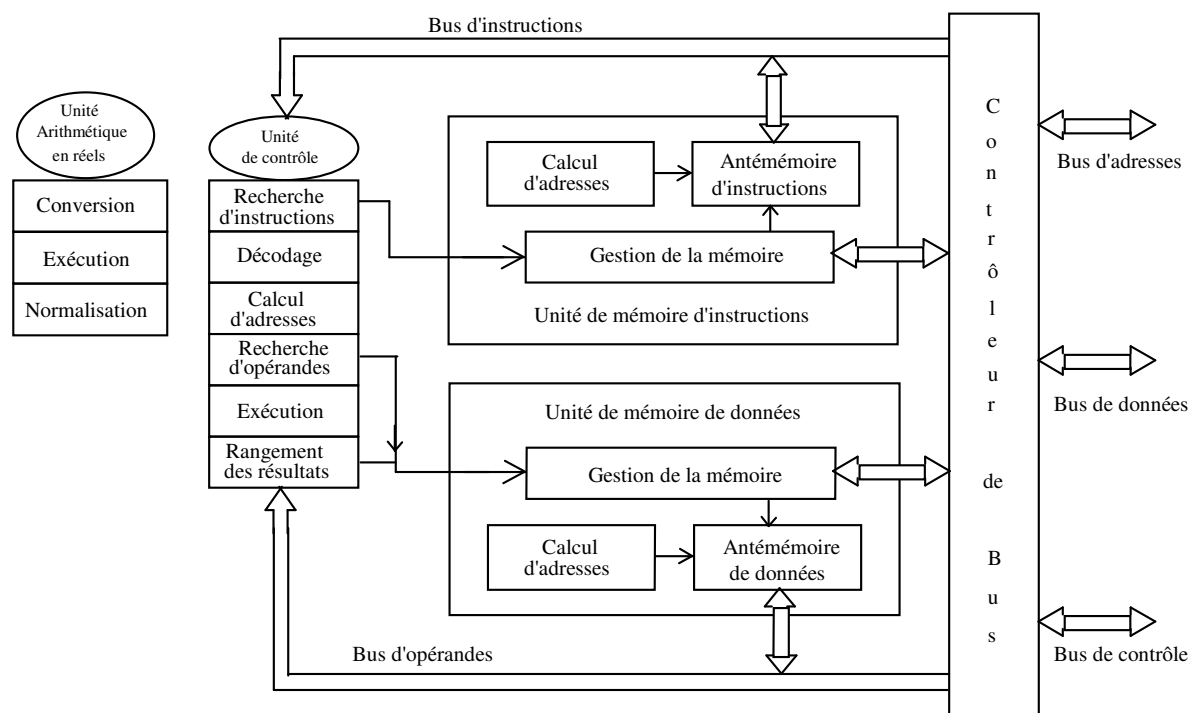
## 5 Le 68040

Le 68040 est un microprocesseur 32 bits intégrant 1,2 millions de transistors.

Il peut exécuter toutes les instructions du 68030.

Il peut exécuter toutes les opérations arithmétiques du coprocesseur 68882 environ 10 fois plus rapidement.

### 5.1 Architecture interne



### 5.2 Fonctionnement de l'unité centrale

La plupart des instructions sont exécutées en un cycle d'horloge en raison de la structure en pipe-line de l'unité centrale. Ce pipe-line est constitué des éléments suivants :

**L'unité de recherche d'instructions** qui contient une file d'attente d'instructions de 8 mots de 16 bits.

**L'unité de décodage** qui décode les instructions contenues dans la file d'attente par blocs de 3 mots de 16 bits.

**L'unité de calcul d'adresses** qui prépare les adresses des opérandes référencés dans les instructions décodées. Cette unité contient les registres d'adresse.

**L'unité d'exécution** qui contrôle les registres de données, l'UAL et la communication avec les autres unités lors de situations exceptionnelles (branchements conditionnels, recherche dans les tables de gestion de mémoire...).

**L'unité de recherche d'opérande** reçoit de l'unité de calcul d'adresses une adresse d'opérande et alimente une file d'attente d'opérandes prêts.

L'**unité de rangement des résultats** reçoit de l'unité de calcul d'adresses une adresse d'opérande et alimente une file d'attente à partir des valeurs que lui transmet l'unité d'exécution.

La recherche des opérandes est prioritaire sur le rangement des résultats pour ce qui concerne les accès à l'antémémoire de données. Toutefois l'unité de rangement des résultats devient prioritaire quand une adresse de recherche correspond à une adresse de rangement (écrire avant de lire pour éviter de lire une valeur non à jour).

### **5.3 Calcul des branchements**

Le déplacement des branchements est ajouté au compteur ordinal par un dispositif spécial de l'unité de décodage. Cette addition est faite sans savoir si le branchement aura lieu ou pas.

Le dispositif de recherche d'instructions poursuit son travail en supposant que le branchement aura lieu toutefois la valeur du compteur ordinal ainsi que les instructions décodées présentes dans l'unité de contrôle sont sauvegardées dans des registres spéciaux de façon à pouvoir, si le branchement n'a pas lieu, retrouver le contexte antérieur.

### **5.4 Exceptions**

La seule différence avec le 68030 concerne les erreurs d'accès à la mémoire.

Les 68020 et 68030 suspendaient l'instruction en cours et exécutaient l'exception de numéro 2 au terme de laquelle l'instruction suspendue était reprise.

Dans le 68040, l'instruction n'est pas suspendue mais considérée comme non exécutée, elle sera donc entièrement reprise après le traitement lié à l'exception. Cette méthode permet d'aller plus vite car le contexte à sauvegarder est plus simple.

Le 68040 empilera :

- Le registre d'état

- Le compteur ordinal

- Les attributs de l'accès qui a créé l'erreur

- Les adresses et les données pour au plus 3 écritures en mémoire en attente.

### **5.5 Gestion des antémémoires et de la mémoire**

Le 68040 intègre 2 antémémoires de 4Ko chacune l'une est utilisée pour les instructions alors que l'autre est réservée aux données.

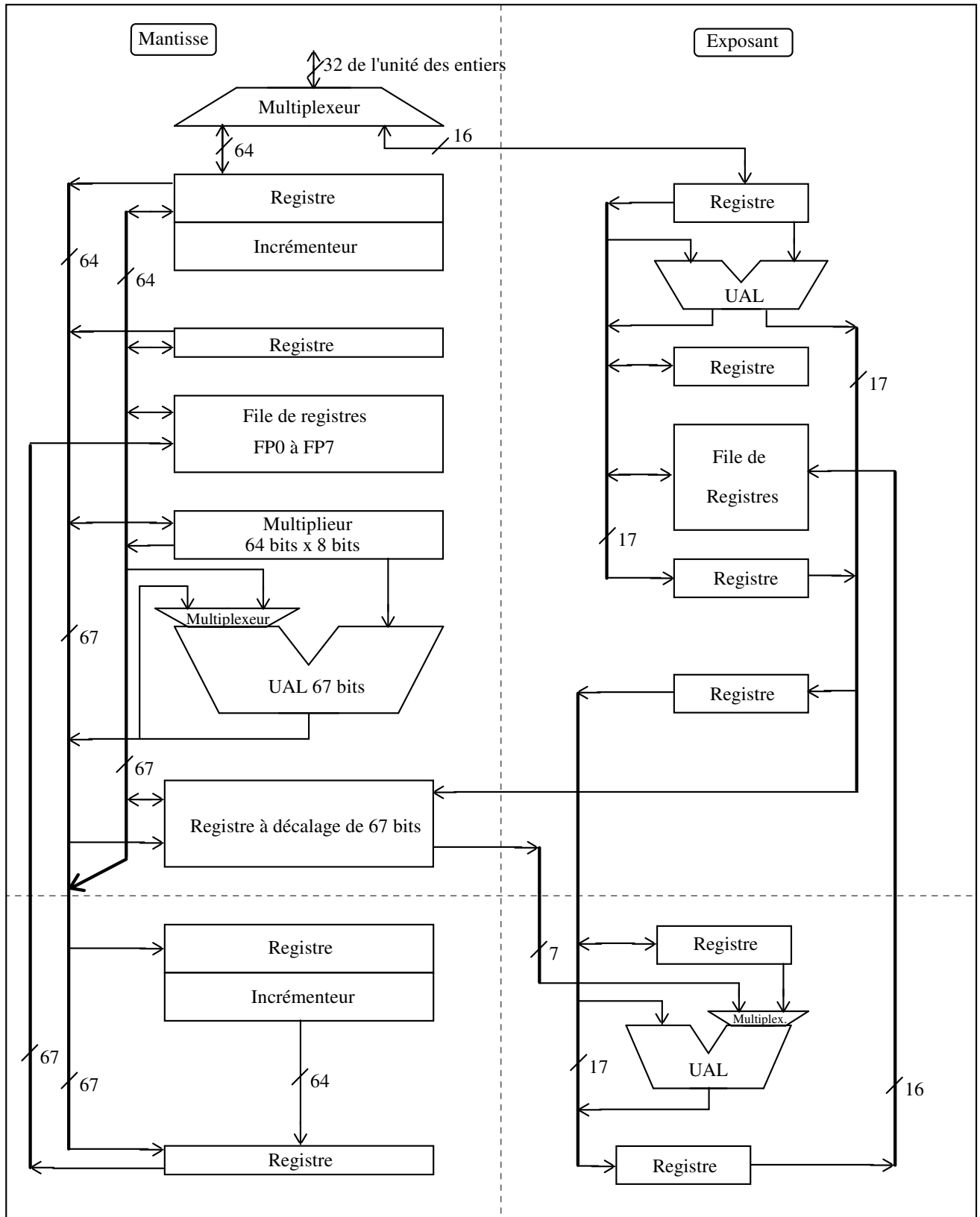
Ces 2 antémémoires peuvent être accédées en même temps selon le modèle d'architecture Harvard qui sépare les chemins d'instructions et de données.

Il y a en outre 2 gestionnaires de mémoire chacun étant doté d'une table de 64 entrées de conversion d'adresses qui joue le rôle d'antémémoire vis à vis des tables de translation qui sont en mémoire centrale.

La taille des pages de la mémoire centrale peut être choisie de 4 ou 8Ko.

Les modes utilisateur et superviseur sont connus et gérés séparément.

## Unité Arithmétique en Flottants du 68040



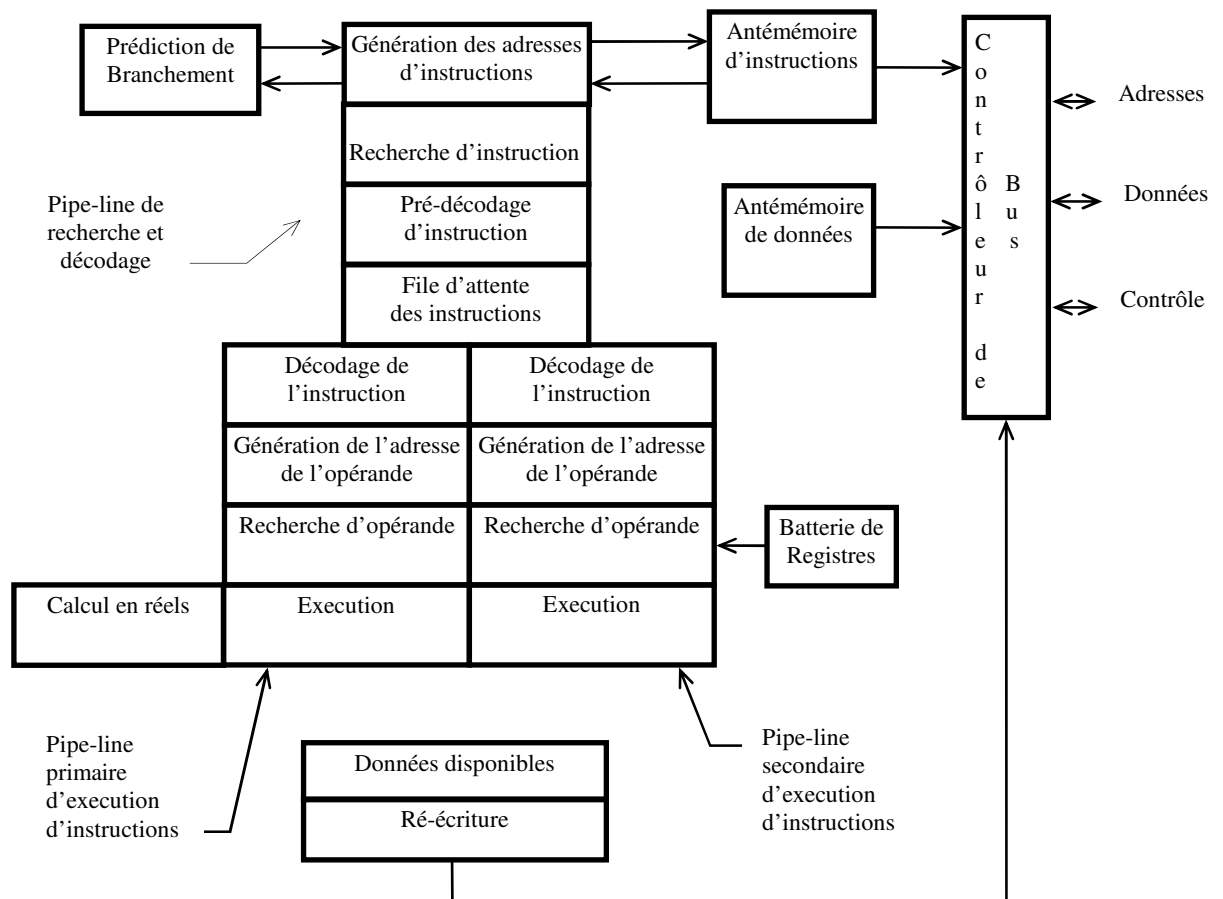


## 6 Le 68060

Le 68060 est un microprocesseur offrant une compatibilité totale avec les 68x00. Avec son horloge à 50 ou 66mhz, il est jusqu'à 2,5 fois plus rapide qu'un 68040 à 40Mhz.

Le 68060 intègre 2,5 millions de transistors pour une consommation de 4,5W sous 3,3V.

### 6.1 Architecture interne



### 6.2 Fonctionnement du 68060

L'exécution des instructions fait appel à un premier pipe-line à 4 niveaux prenant en charge la recherche et le pré-décodage des instructions suivi de deux pipe-lines à 4 niveaux permettant l'exécution en parallèle de deux instructions.

Le fonctionnement du **pipe-line de recherche et décodage** est le suivant :

Le premier niveau du pipe-line calcule l'adresse de la prochaine instruction en relation avec l'unité de prédiction de branchement.

Le niveau suivant va chercher l'instruction dans l'anté-mémoire des instructions.

Le niveau suivant effectue un premier décodage et détermine les informations nécessaires aux pipe-lines d'exécution. Ce niveau crée un nouveau codage de l'instruction de taille fixe (32 bits).

Le niveau suivant est une file d'attente à 16 entrées auprès de laquelle s'alimentent les pipe-lines d'exécution.

**Le pipe-line primaire d'exécution** traite les opérations entières et réelles. Il peut exécuter toutes les instructions des processeurs de la famille 68x00. Il est constitué de quatre unités pouvant travailler en parallèle:

- Une UAL (opération arithmétiques, logiques, déplacements, contrôle de limites ...)
- Un registre à décalage (tous les décalages se font en 1 cycle)
- Un multiplieur (toutes les multiplications se font en 2 cycles)
- Un dispositif pour les autres opérations (sur les bits, en décimal ...)

**L'unité de calcul en réels** est compatible avec celles de la famille 68x00 (norme IEEE 754). Elle traite des opérands sur 80 bits et est elle même constituée de 4 modules :

- Un additionneur effectuant les calculs en 3 cycles (la comparaison ne prend qu'un seul cycle).
- Un multiplieur effectuant les calculs en 3 cycles.
- Un diviseur effectuant les calculs en 40 à 70 cycles.
- Un dispositif assurant les autres opérations.

**Le pipe-line secondaire d'exécution** ne traite que les entiers et ne peut pas exécuter toutes les instructions des processeurs de la famille 68x00. Il n'est constitué que de :

- Une UAL simplifiée
- Un registre à décalage

Le fonctionnement des **pipe-lines d'exécution** est le suivant :

Le premier niveau termine le décodage fait dans les niveaux précédents et va rechercher dans la batterie de registres ceux utiles au calcul de l'adresse de l'opérande.

Le niveau suivant calcule l'adresse de l'opérande. Ce calcul s'effectue la plupart du temps en 1 cycle d'horloge.

Le niveau suivant recherche les opérands dans les registres ou en mémoire via l'anté-mémoire.

Le dernier niveau exécute l'opération. La majorité des calculs sur les entiers s'effectuent en un cycle d'horloge tandis que la plupart de ceux sur les réels s'effectuent en 3 cycles.

Grace à cette architecture, le 68060 peut, lors de chaque cycle, effectuer jusqu'à 3 opérations en parallèle :

- 2 calculs sur des entiers et une prédiction de branchement
- ou un calcul d'entiers, un calcul de réels et une prédiction de branchement

**La batterie de registres** est un ensemble de registres banalisés qui sont renommés pour correspondre à ceux des 68x00 selon les besoins de chaque instruction.

### **6.3 Prédiction de branchements**

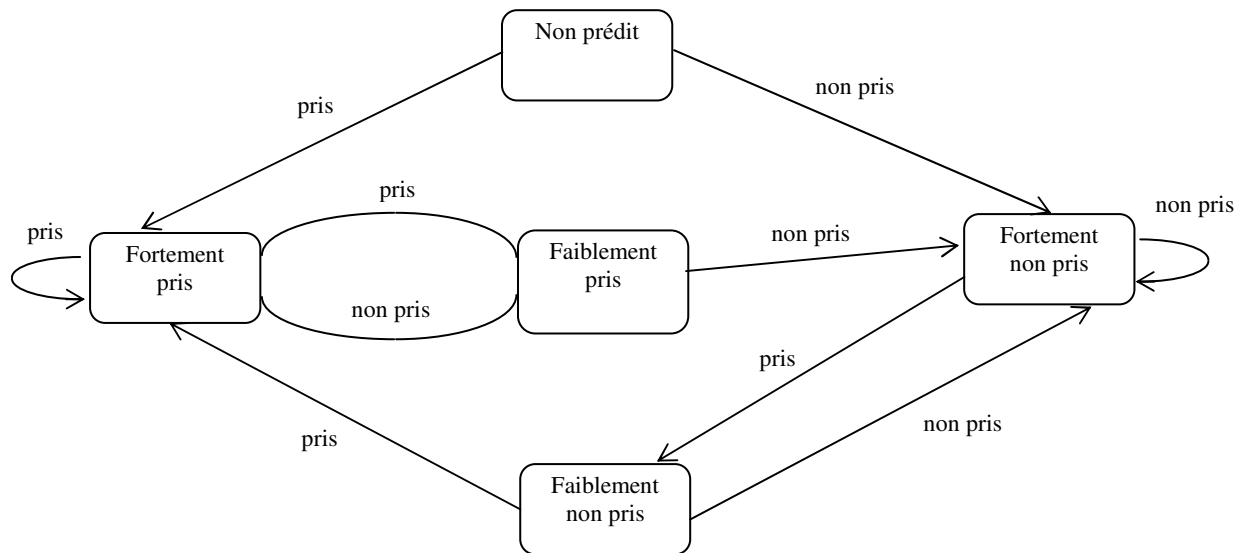
Le problème de la prédiction d'instructions est celui posé lors d'une instruction de branchement conditionnelle. En effet, le dispositif de recherche d'instructions lit à l'avance les instructions en séquence. Mais, lors d'un branchement, il est impossible de savoir à l'avance s'il aura lieu ou pas.

Les dispositifs utilisés jusque là se contentaient de faire comme si le branchement ne devait pas avoir lieu puis, le cas échéant, d'arrêter le pipe-line d'exécution et de vider la file d'attente des instructions, pour relancer la recherche à partir de la nouvelle adresse. Sur le 68060 une telle opération nécessite 7 cycles d'horloge.

Afin d'éviter qu'une telle situation se produise trop souvent, la méthode utilisée sur le 68060 est la suivante :

L'unité de recherche des instructions lit les instructions en séquence jusqu'à rencontrer un branchement conditionnel. Lorsque cela se produit elle fait appel à une unité de prédiction de branchement qui lui indique si elle doit continuer la recherche en séquence ou à partir de l'adresse indiquée dans l'instruction de branchement.

Le modèle de prédiction est un automate à 5 états dont le diagramme est le suivant :



Chaque fois qu'une instruction de branchement est exécutée, on met à jour son état dans l'historique en fonction du fait que le branchement à eu lieu ou non.

Quand l'unité de génération d'adresse d'instruction rencontre un branchement, elle fait appel à une mémoire associative contenant des informations sur les 256 derniers branchements rencontrés pour y trouver l'historique de ce branchement. Si le branchement n'a pas encore d'historique, on considère a priori que s'il s'agit d'un branchements en avant il n'aura pas lieu tandis que s'il s'agit d'un branchement en arrière il aura lieu. A l'aide de cette méthode elle produit une prévision qui se révèle correcte dans 90% des cas.

#### **6.4 Les antémémoires**

Le 68060 possède 2 antémémoires de 8Ko dotées chacune d'un contrôleur et d'une table de calcul d'adresse. Ces antémémoires sont reliées à la mémoire centrale par un bus de données de 32 bits et un bus d'adresses de 32 bits.

**L'antémémoire de données** est constituée de 4 banques permettant la lecture d'un opérande dans l'antémémoire en simultanéité totale avec une écriture vers la mémoire. Les accès par le 68060 se font sur 64 bits.

L'antémémoire d'instructions n'est, quant à elle, accédée par le 68060 que sur 32 bits.

## 7 Les coprocesseurs

---

Les coprocesseurs de la famille MOTOROLA sont les suivants :

68880 associé au 68000  
68881 associé au 68020  
68882 associé au 68030

Les coprocesseurs 68881 et 68882 sont totalement compatibles avec l'unité de calcul arithmétique du 68040 (pour plus de détails sur cette unité se référer au chapitre 5).

Le format des informations traitées est le même que celui utilisé par les coprocesseurs d'INTEL (voir 5.2)

Le coprocesseur est constitué d'un pipe-line à 3 étages :

Unité de conversion  
Unité de calcul  
Unité de normalisation

L'addition et la soustraction sont réalisées en 2 cycles d'horloge (80ns), la multiplication en 5 cycles (200ns), la division en 38 cycles (1,52µs).

Une opération plus complexe comme la racine carrée demande 103 cycles (4,12µs).

## 8 La famille Power PC

---

La création de cette famille de microprocesseurs a été lancée en 1991 par MOTOROLA, IBM et APPLE pour concurrencer INTEL.

L'architecture adoptée pour ces processeurs est de type RISC (Reduced Instruction Set Computer) largement inspirée de l'architecture POWER développée dans les années 80 à partir des concepts de l'IBM 801.

Elle a fait l'objet de 3 livres la définissant :

Livre 1 : jeu d'instructions en mode utilisateur, registres, adresses logiques

Livre 2 : mode virtuel, environnement multi-processeur

Livre 3 : vision superviseur, exceptions, interruptions

Il est dès lors possible de conserver l'architecture et la compatibilité vue du programmeur tout en adaptant un processeur à un système d'exploitation spécifique par redéfinition du livre 3.

Il s'agit de créer une famille de processeurs 32 puis 64 bits qui est, pour l'instant, constituée des générations suivantes :

### Première génération :

Apparu en 1993, le **MPC601** est un processeur 32 bits utilisant un bus de données de 64 bits. Les adresses sont sur 32 bits (4 Go) mais les adresses virtuelles sont de 52 bits (4To).

Il se présente sous la forme d'un boîtier carré de 304 broches contenant 2,8 millions de transistors. Sa consommation de 6,5W sous 3,6V à 80MHz.

### Deuxième génération :

Apparu fin 1993, le **MPC603** est une version à faible consommation (2,5W sous 3,3V à 80 Mhz) du 601. Il se présente sous la forme d'un boîtier carré de 240 broches. Le **MPC602** est une version simplifiée du 603 permettant d'offrir un processeur meilleur marché.

Apparu en 1994, le **MPC604** est une version améliorée du MPC601 (plus de parallélisme, meilleure gestion de la prévision pour les branchements). Il est constitué de 3,6 millions de transistors dans un boîtier carré de 304 broches et sa consommation reste inférieure à 10W à 100MHz

Apparu en 1996, le **MPC620** qui travaille en 64 bits mais possède un bus de données de 128 bits, le bus d'adresse est de 40 bits mais les adresses logiques sont sur 64 bits ( $1,84 \cdot 10^{19}$  octets) tandis que les adresses virtuelles sont de 80 bits ( $1,2 \cdot 10^{24}$  octets). On peut remarquer que le choix d'adresses virtuelles d'une telle taille (80 bits) peut permettre de constituer une adresse de machine du type Ethernet (sur 48 bits soit  $2,8 \cdot 10^{14}$  machines différentes!) puis une adresse de 32 bits (4Go) pour accéder à une information contenue sur cette machine. Le MPC620 est constitué de 7 millions de transistors dans un boîtier de 482 broches et sa consommation est de 30W à 133 MHz sous 3,3V.

### Troisième génération :

Baptisée G3, prévue pour 1998. C'est une amélioration du 620 pour pouvoir utiliser des horloges de 300 à 400 MHz. cette génération devrait être constituée de processeurs 32 puis 64 bits.

### Quatrième génération :

Baptisée G4, prévue pour 1999, elle est constituée de microprocesseurs 64 bits avec des horloges de 500 MHz.

### Cinquième génération :

Baptisée 2K, prévue pour 2001, elle devrait utiliser des horloges jusqu'à 1GHz

## Principales caractéristiques des processeurs de la famille Power PC (générations 1 et 2)

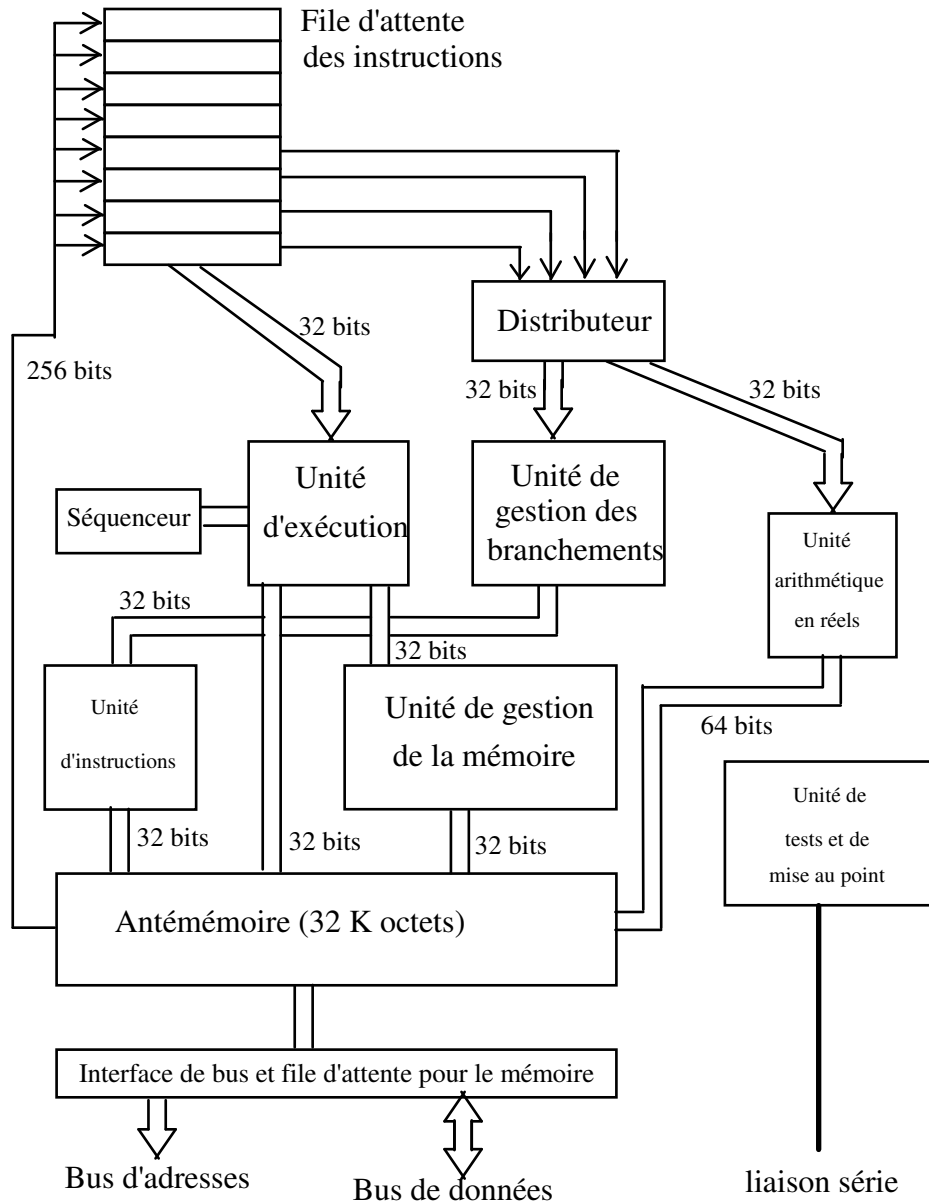
Modèle	MPC 601	MPC 601v	MPC 602	MPC 603	MPC 603e	MPC 604	MPC 604e	MPC 620
Taille des registres (en bits)	32	32	32	32	32	32	32	64
Bus d'adresses (en bits)	32	32	32	32	32	32	32	40
Bus de données (en bits)	64	64	64 <sup>15</sup> (32+32)	32 ou 64	32 ou 64	64	64	128
Antémémoire données (Ko)	32	32	4	8	16	16	32	32
Antémémoire instructions (Ko)			4	8	16	16	32	32
Nombre d'unités en parallèle	3	3	4	5	5	6		6
Fréquence d'horloge (MHz)	50, 66, 80, 100	100	66	66 ou 80	100, 200, 240	100 , 120, 133	166, 180, 200	166, 180, 200
Taille de la puce (mm <sup>2</sup> )	120	74	50	83	98	197		311
Nombre de transistors	2,8M	2,8M	1M	1,6M	2,6M	3,6M		7M
Consommation électrique (W)	10	6	1,2	3	3,5	10	20	30
Tension d'alimentation (V)	3,6	2,5	3,3	3,3	3,3	3,3	3,3	3,3
Nombre de broches du boîtier	304	304	144	240	240	304 ou 256		625

<sup>15</sup> Le bus de données du MPC602 est multiplexé dans le temps pour permettre des transferts de 64 bits en deux fois 32 bits.

## Microphotographie de la puce du MPC601

## 8.1 Les MPC 601 , 602 et 603

### 8.1.1 Architecture



Le processeur est constitué des éléments suivants :

#### 1°) La file d'attente des instructions et le distributeur :

La file d'attente peut contenir 8 instructions. Elle est alimentée par l'antémémoire au travers d'un bus de 256 bits permettant des transferts rapides.

Le distributeur examine les 4 dernières et les distribue aux diverses unités selon les règles suivantes:

- L'unité de branchement peut recevoir n'importe laquelle des 4 instructions
- L'unité d'exécution ne peut recevoir que la dernière instruction



L'unité arithmétique en réels peut recevoir n'importe laquelle des 4 instructions

Il peut distribuer jusqu'à 3 instructions par cycle d'horloge.

#### 2°) L'unité d'instructions :

Cette unité pilote la recherche d'instructions dans l'antémémoire. La recherche se fait à partir d'adresses provenant soit :

- de l'unité d'instructions elle-même par incrémentation du compteur ordinal
- de l'unité de branchement lors d'une rupture de séquence
- du séquenceur lors d'évènements spéciaux (interruption, exception ...)

Afin d'accélérer le calcul de l'adresse réelle normalement effectué par le gestionnaire de mémoire, l'unité d'instructions conserve les 4 transformations d'adresses virtuelles les plus récemment utilisées et évite le passage par le gestionnaire de mémoire lors d'une nouvelle utilisation de l'une de ces transformations.

Cette table appelée TSA (Translation Shadow Array) constitue en fait une petite antémémoire vis à vis du gestionnaire de mémoire.

#### 3°) L'unité d'exécution :

Elle est constituée d'un pipe-line à 4 niveaux :

- Lecture dans la file des instructions
- Décodage de l'instruction
- Exécution de l'opération (arithmétique, logique ou de décalage)
- Rangement du résultat

Cette unité effectue les opérations en virgule fixe ainsi que les calculs d'adresse pour les instructions de transfert d'information

#### 4°) Le séquenceur :

Il prend en charge les fonctions effectuées à la mise sous tension (initialisations et test de l'antémémoire). Il contient l'horloge temps réel et gère les interruptions, les exceptions et les erreurs.

Le séquenceur contient quelques registres de contrôle assez peu utilisés et les quelques instructions le concernant lui sont transmises par l'unité d'exécution.

#### 5°) L'unité de branchement :

Les conditions de branchement peuvent porter sur les bits du registre d'état (CR) ou la valeur d'un registre compteur (ou une combinaison des 2).

L'unité de branchement peut calculer l'adresse de la prochaine instruction lors d'un branchement sur valeur de registre compteur sans délai.

Par contre, les autres branchements conditionnels ne peuvent être évalués à l'avance que lorsque les instructions non encore exécutées ne modifieront pas la condition (on sait, pour chaque instruction, quels sont les bits du registre d'état qu'elle est susceptible de modifier).

Lorsqu'il n'est pas possible de procéder de la sorte l'unité de branchement fait l'hypothèse qu'un branchement aura lieu s'il revient en arrière et n'aura pas lieu s'il va en avant (cette hypothèse est issue d'études statistiques effectuées sur les branchements).

Toutefois, un bit dans l'instruction de branchement permet de provoquer le choix inverse. Ceci permet aux compilateurs d'intervenir sur l'hypothèse faite par l'unité de branchement en fonction de la connaissance qu'ils ont du programme.

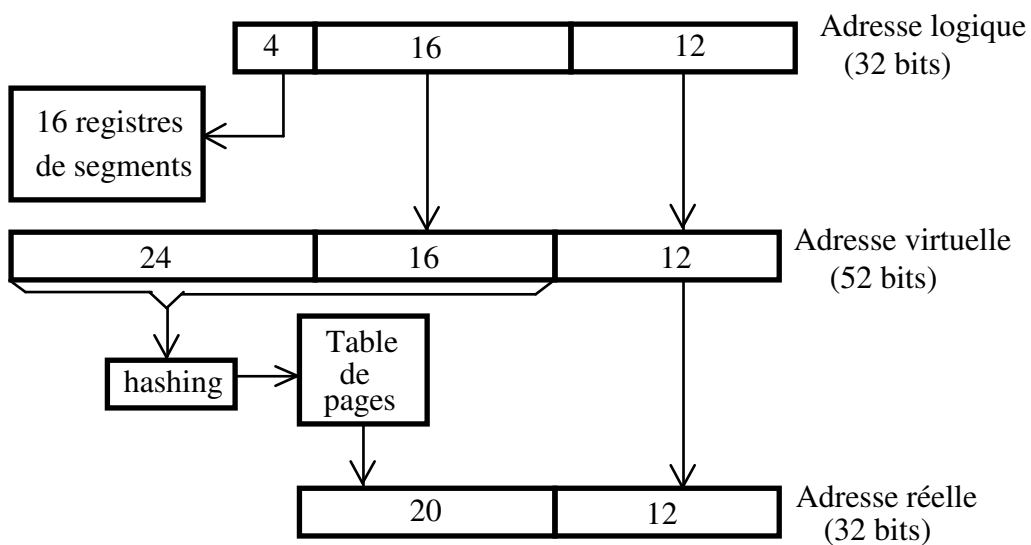
Quand l'unité de branchement a provoqué un choix non certain, les instructions lues à l'issue de ce choix sont marquées de façon à ne pas être exécutées tant que ce choix n'est pas confirmé par l'évaluation exacte de la condition du branchement. Lorsque cette évaluation peut avoir lieu, ces instructions sont validées si le choix avait été bon et retirées de la file dans le cas contraire.

Dès que le choix est certain, l'instruction de branchement est retirée de la file des instructions.

6°) Le gestionnaire de mémoire :

Le gestionnaire de mémoire assure la transformation d'adresses virtuelles en adresses réelles et assure la protection en lecture et/ou écriture ainsi que la prise en compte du mode (superviseur / utilisateur). Il gère des segments et des pages de 4Ko ou des blocs dont la taille peut aller de 128Ko à 8Mo.

Pour gérer les pages, il contient 16 registres de segment de 32 bits dont 24 bits définissent le segment virtuel tandis que les autres déterminent les autorisations d'accès et le type d'adressage utilisé (réel, virtuel ou d'entrée/sortie). Ces registres fournissent une adresse de page qui est recherchée dans une table par hashing. Lorsque la page ne peut pas être trouvée après 16 tentatives, une exception de défaut de page est levée (elle sera utilisée pour mettre à jour la mémoire centrale à partir de la mémoire de masse). Afin d'accélérer la recherche une mémoire associative de 256 mots est utilisée pour conserver les adresses de pages les plus récemment utilisées.



La gestion des blocs fait appel à un mécanisme différent permettant de gérer 4 blocs de taille variable entre 128 Ko et 8 Mo.

7°) L'antémémoire :

Sa taille est de 32Ko. Elle utilise 2 mémoires associatives l'une comportant 8 blocs de 128 mots de 288 bits pour les valeurs, l'autre 8 blocs de 64 mots de 27 bits pour les indicateurs. L'algorithme de remplacement utilisé est le LRU (Least Recently Used). Les instructions **LWARX** et **STWCX** permettent d'accéder à des données en mode verrouillé pour éviter les conflits dans un environnement multiprocesseur (LWARX accède à une donnée et la verrouille dans l'antémémoire tandis que STWCX la déverrouille).

Les transferts entre l'antémémoire et le gestionnaire de mémoire ou l'unité d'instructions se font en 256 bits ce qui permet d'atteindre un débit maximal de 2,1 Go/s à 66Mhz.

Un bit de parité est associé à chaque octet de l'antémémoire.

8°) Interface de bus et file d'attente pour la mémoire

La file d'attente possède 2 entrées en lecture et 3 en écriture. Chaque entrée contient jusqu'à 32 octets. Les informations à destination ou en provenance de la mémoire transitent par cette file d'attente.

Le gestionnaire associé à cette file prend en charge les problèmes de cohérence des données entre le processeur, l'antémémoire et la mémoire (problème du décalage dans le temps des mises à jour des informations en chacun de ces 3 points).

L'unité de gestion de bus pilote le bus d'adresses de 32 bits et le bus de données de 64 bits. Elle effectue des transferts par blocs de 32 octets avec un débit maximum de 422 Mo/s à 66 Mhz.

Un transfert se déroule en 3 étapes :

- \* Etablissement des signaux d'arbitrage de bus permettant la mise en place de partage de bus dans un environnement multi-processeurs (demande et autorisation d'accès séparés pour le bus d'adresse et celui de données).
- \* Etablissement des signaux de transfert (adresse, taille et sens du transfert)
- \* Fin du transfert (retour à l'état de repos des signaux et des bus).

### 9°) Unité arithmétique en réels

Cette unité respecte le standard IEEE 754.

Elle est constituée d'un pipe-line à 5 niveaux :

- Lecture de l'instruction dans la file d'attente des instructions
- Décodage de l'opération à effectuer
- Exécution de l'opération (2 niveaux de pipe-line)
- Rangement des résultats

Elle comporte 32 registres de 32 bits. Le calcul des adresses des opérandes est sous-traité par l'unité d'exécution.

## **8.1.2 Les registres et les opérandes**

### 1°) Registres d'usage général

On dispose de 32 registres d'usage général de 32 bits baptisés R0 à R31.

### 2°) Registres pour les branchements

\* 8 registres d'état (CR0 à CR7) de 4 bits chacun concaténés dans un registre CR de 32 bits :

- bit LT : indique un résultat négatif ou , après une comparaison, inférieur
- bit GT : indique un résultat positif ou , après une comparaison, supérieur
- bit EQ : indique un résultat nul ou , après une comparaison, égal
- bit SO : indique un débordement.

Les instructions sur les entiers positionnent le registre d'état CR0, les instructions sur les réels positionnent CR1 tandis que les comparaisons (quels que soient les opérandes) positionnent le registre d'état choisi dans l'instruction. Les instructions de branchement peuvent tester n'importe quel bit du CR entier.

\* 1 registre d'état pour les opérations entières XER (32 bits dont 11 utilisés) qui n'est mis à jour que par les instructions qui le demandent explicitement. Sa structure est la suivante :

- bit OV : débordement
- bit SO : débordement cumulé (mémorise un débordement même après qu'une nouvelle instruction n'ayant pas provoqué de débordement ait été exécutée)
- bit CA : retenue
- octet de compte : utilisé pour indiquer la taille des chaînes de caractères

\* 1 registre d'état pour les opérations réelles FPSCR mis à jour par toutes les opérations sur des réels. Sa définition est conforme à la norme IEEE 754.

\* Un registre compteur CTR (32 bits) utilisé par les branchements avec comptage (répétitions N fois). Un branchement avec comptage consiste à décrémenter ce registre CTR puis à effectuer le branchement si CTR est nul.

\* Un registre de lien LR (32 bits). Ce registre peut être mis à jour par une instruction de branchement pour contenir l'adresse de retour (ce branchement peut alors se comporter comme un appel de sous programme sans utilisation de la pile). Ce registre peut servir aussi d'adresse de branchement (on réalise ainsi le retour de sous programme).

### 3°) Registres pour les réels :

L'unité arithmétique sur les réels possède 32 registres de 64 bits (FP0 à FP31) plus le registre d'état FPSCR décrit ci-dessus.

### 4°) Opérandes :

Les opérandes utilisables sont les suivants :

- entiers naturels sur 8 bits ou 16 bits
- entiers relatifs sur 16 bits
- pointeurs sur 32 bits
- chaînes de caractères (0 à 127 octets)
- reels simple ou double précision

## **8.1.3 Instructions**

### 1°) Arithmétique sur les entiers

**Addition** : il existe 3 additions. L'addition normale, l'addition avec report de la retenue contenue dans le bit CA du XER et l'addition accompagnée d'un décalage de 16 bits à gauche.

**Soustraction** : avec ou sans retenue reportée.

**Multiplication** : il existe une multiplication renvoyant les 32 bits de faible poids du résultat et une pour les 32 bits de fort poids de plus chacune possède une version sur les entiers naturels et une sur les entiers relatifs.

**Division** : pour les entiers naturels et pour les entiers relatifs.

### **Changement de signe**

**Extension de signe** de 8 vers 16 bits ou de 16 vers 32 bits.

**Comparaison** : positionne les bits du registre d'état dans le champ (CR0 à CR7) spécifié dans l'instruction

### 2°) Logiques :

Les opérations **AND**, **OR**, **XOR**, **NAND** et **NOR** sont disponibles. De plus il existe pour les opérations AND, OR et XOR une version qui effectue un décalage sur 16 bits en plus de l'opération et une version qui travaille avec le complément de l'opérande.

On trouve aussi les classiques opérations de **décalage arithmétique, logique et cyclique**.

Il existe une instruction qui **compte le nombre de bits à 0 à droite de l'opérande**.

### 3°) Transferts :

**Concernant XER** (registre d'état) on peut le transférer dans un registre général et réciproquement.

**Concernant FPSCR** (registre d'état pour les réels) il peut être transféré vers ou depuis CR ainsi que vers ou depuis l'un des registres FP.

**Concernant CR** (registre d'état) on peut le transférer vers un registre général, vers XER ou vers FPSCR. Il existe aussi de transferts internes à ce registre (entre un CR<sub>i</sub> et un CR<sub>j</sub>).

Les transferts (**LOAD et STORE**) peuvent se faire avec un opérande sur 8, 16 ou 32 bits. Lorsque l'opérande est sur 16 bits on peut permuter ou pas les 2 octets. De même, lorsque l'opérande est sur 32 bits on peut permuter ou pas les 2 mots de 16 bits.

Il existe aussi des transferts pour les chaînes de caractères et les suites de mots de 16 bits ainsi que pour les réels en simple et double précision.

#### 4°) Opérations sur les réels :

On trouve les opérations suivantes :

- Changement de signe
- Valeur absolue
- Addition, soustraction, division et multiplication
- Multiplication puis addition, multiplication puis soustraction précédées ou pas d'un changement de signe de l'opérande
- Arrondi
- Conversion en entier ou depuis un entier
- Comparaisons

#### 5°) Ruptures de séquence :

La condition peut porter sur l'un des bits du registre d'état CR (constitué en fait de 8 registres de 4 bits) ou sur le compteur CTR. Dans ce dernier cas le registre compteur est décrémenté avant le test. Ceci permet de mettre facilement en place des boucles de type for.

#### 6°) Opérations sur les registres d'état :

Outre les transferts concernant le registre CR déjà décrits, il est possible d'effectuer sur CR des opérations logiques (ET, OU, NAND, NOR et OU exclusif).

Il existe également des instructions permettant de mettre à 0 ou à 1 un bit du FPSCR

#### 7°) Opérations de synchronisation :

**SYNC** attend la terminaison de toutes les instructions en cours et de toutes les écritures en mémoire en attente. Cette instruction permet la synchronisation de plusieurs processeurs.

**ISYNC** fonctionne comme SYNC mais n'effectue pas les instructions qui sont en attente et qui n'ont pas encore été commencées (elles devront être relues après le ISYNC). Cette instruction peut être utilisée après une modification du code dans la mémoire pour éviter qu'une instruction modifiée ne soit pas correctement exécutée parce qu'elle était déjà en attente.

**EIEIO** impose que les transferts vers la mémoire soient effectués dans l'ordre où ils ont été demandés.

**TLBSYNC** permet de mettre à jour les tables de pages utilisées pour la mémoire virtuelle dans un contexte multi-processeur. Ces tables doivent être mises à jour sur tous les processeurs avant de continuer.

#### 8°) Opération de gestion de l'antémémoire :

**DCBT** et **DCBTST** permettent d'indiquer au gestionnaire de l'antémémoire que l'on va utiliser un bloc immédiatement ce qui lui permet de le placer tout de suite dans l'antémémoire.

**DCBZ** alloue un nouveau bloc et le met à 0

**DCBST** provoque la recopie du bloc de l'antémémoire vers la mémoire. Cette instruction permet de synchroniser le contenu de la mémoire avec celui de l'antémémoire (systèmes multi-processeurs à mémoire commune, mémoire vidéo ...).

**DCBF** fonctionne comme DCBST mais le bloc est considéré comme libéré. Il pourra donc être récupéré sans qu'il soit nécessaire de le recopier vers la mémoire.

**DCBI** permet d'indiquer au gestionnaire d'antémémoire qu'un bloc contenant des données est invalidé. Il faudra donc le recharger depuis la mémoire s'il est encore utilisé. Ceci permet de synchroniser le contenu de l'antémémoire avec celui de la mémoire dans un environnement multi-processeur à mémoire commune.

**ICBI** agit comme DCBI pour un bloc d'instructions.

### **8.1.4 Test et mise au point**

Afin de faciliter la mise au point de cartes utilisant le MPC601, il a été doté d'un dispositif spécial assurant l'autotest du processeur et permettant la connexion en série de dispositifs de mise au point compatibles avec les standards IEEE.

### **8.1.5 Le MPC 603**

Il est constitué de 1,6 millions de transistors et représente une version à faible consommation du MPC601 destiné au marché des portables. Les principales différences sont les suivantes :

3 instructions additionnelles de calcul en réels destinées à accélérer certains calculs graphiques

L'antémémoire est séparée en 2 : une pour les instructions de 8Ko et l'autre pour les données de 8Ko également.

Il existe une version du 603 baptisée MPC603e dont les antémémoires sont de 16Ko chacune (ce processeur intègre 2,6 millions de transistors).

### **8.1.6 Le MPC 602**

Il est dérivé du MPC603 et en constitue une version simplifiée moins chère :

Calculs sur des réels en simple précision seulement

antémémoire de 4Ko

Bus multiplexé de 32 bits pour les adresses et de 64 bits pour les données

### **8.1.7 Horloge**

Le MPC 601 peut fonctionner à 50, 66, 80 ou 100MHz

Le MPC 603 fonctionne à 80Mhz tandis que le MPC603e fonctionne à 100MHz

Le MPC602 fonctionne à 66MHz

## 8.2 Le MPC 604

---

Le MPC604 reste compatible avec le 601 la seule différence étant une amélioration de l'organisation du processeur pour atteindre des performances de 1,5 à 2 fois supérieures en ce qui concerne le traitement des entiers.

### 8.2.1 Architecture

L'architecture générale reste bien entendu la même que sur le 601. Toutefois des différences apparaissent sur les points suivants :

- \* Distributeur plus performant
- \* Nombre d'unités de traitement plus élevé
- \* Chaque unité de traitement possède une zone de mise en attente des instructions non encore prêtes ainsi qu'un dispositif de complétion.
- \* Séquenceur permettant une meilleure prise en compte des interruptions
- \* Prédiction statistique des branchements
- \* Antémémoire dédoublée
- \* Gestion en pipe line des bus

#### 1°) La distribution des instructions

Le MPC604 est capable de répartir jusqu'à 4 instructions en direction des 6 unités de traitement suivantes :

- 2 unités de calculs simples sur les entiers
- 1 unité de calculs complexes sur les entiers
- 1 unité de transfert d'informations
- 1 unité d'arithmétique réelle
- 1 unité de traitement des branchements

Il distribue ces instructions même lorsque les opérandes qu'elles nécessitent ne sont pas encore disponibles. En effet chaque unité de traitement dispose de zones d'attente dans lesquelles sont placées les instructions quasi prêtes qui partiront vers les unités concernées dès que tous leurs opérandes seront disponibles.

Les règles de répartition sont les suivantes :

- \* Une instruction par unité et par cycle
- \* Les ressources de l'instruction sont prêtes
- \* Une instruction qui suit un branchement ne sera pas distribuée en même temps que celui-ci afin de permettre à l'unité de gestion des branchements de faire sa prédiction et, le cas échéant, de ne pas distribuer cette instruction.
- \* Respect de l'ordre des instructions lors de leur distribution (ceci ne garantit bien entendu pas que l'ordre soit encore respecté lors de la terminaison de ces instructions).

Comme il est difficile d'annuler les effets de certaines instructions (écriture dans la mémoire, modification du registre d'état ...), lorsque celles-ci font partie d'un morceau de programme dont on n'est pas sûr qu'il devra être réellement exécuté, elles sont envoyées aux diverses unités avec une indication permettant de les faire passer avec une priorité inférieure. De plus, les résultats obtenus dans ce cas sont mis en attente soit dans les registres soit dans les files de sortie jusqu'à ce qu'il y ait confirmation du fait que ces instructions devaient bien être exécutées. En cas de non confirmation ces résultats seront ignorés.

#### 2°) Les unités de traitement

Chacune des 6 unités de traitement possède un tampon de 2 entrées permettant de mettre en attente des instructions provenant du distributeur.

Dans les 3 unités de traitement des entiers les instructions peuvent sortir du tampon d'attente dans un ordre différent de celui d'entrée. Les 3 autres unités, par contre, respectent l'ordre.

L'unité de calcul en réels travaille en double précision.

Chaque fois qu'une unité produit un résultat elle indique à quel opérande il correspond de façon à ce que le gestionnaire des instructions en attente dans le tampon puisse voir si cet opérande était nécessaire à l'une des instructions en attente.

Lorsque l'exécution est terminée et le(s) résultat(s) produit(s), l'instruction n'est pas pour autant considérée comme complétée. Il faut, en effet, s'assurer d'une part que cette instruction n'a pas provoqué d'exception et d'autre part que les instructions qui la précédaient dans le programme sont elles aussi terminées.

### 3°) L'unité de branchement

L'unité de traitement des branchements a été fortement améliorée dans le sens où elle essaye de faire une prédiction concernant les instructions de branchements conditionnels.

L'unité de branchement utilise une table de 512 entrées contenant l'historique des branchements précédents (algorithme de Smith).

Elle possède en outre une table en mémoire associative contenant les 64 dernières adresses de branchement calculées. Cette table permet d'éviter de refaire le calcul de l'adresse d'un branchement que l'on a déjà traité.

### 4°) Antémémoires

Le MPC604 possède des antémémoires séparées pour les instructions et pour les données afin de permettre des accès simultanés.

Chacune est de 16 Ko et possède un bit de parité par octet.

L'unité d'instructions accède à son antémémoire par mots de 128 bits tandis que celle des données est accédée en 64 bits.

Il existe des instructions permettant d'invalider les antémémoires (tous les accès se font alors directement vers la mémoire) ou de les verrouiller (on ne peut plus y écrire ce qui peut être utile pour assurer la cohérence des contenus des antémémoires dans un système multi-processeurs).

## **8.2.2 Horloge**

Le MPC604 travaille avec une horloge de 100 , 120 , ou 133 MHz .



## 8.3 Le MPC620

---

### 8.3.1 Architecture

Le MPC 620 reste compatible avec le 601 les différences portent sur les points suivants :

#### 1°) Distribution et exécution des instructions

Tout comme le MPC 604, le MPC620 peut exécuter jusqu'à 4 instructions par cycle.

Il possède de même 6 unités fonctionnelles :

3 unités entières dotées d'une file d'entrée de 2 instructions chacune (Deux pour les opérations courantes effectuées en 1 cycle. Une, constituée d'un pipe line à 3 niveaux, pour les opérations plus longues comme la multiplication et la division)

1 unité de lecture/écriture dotée d'une file d'entrée de 3 instructions

1 unité de branchement dotée d'une file d'entrée de 4 instructions

1 unité de réels, constituée d'un pipe line à 3 niveaux et dotée d'une file d'entrée de 2 instructions

Le MPC620 peut exécuter des instructions de façon spéculative jusqu'à 4 niveau de branchements non encore résolus. Les résultats sont mis en attente jusqu'à confirmation des prédictions.

#### 2°) L'unité de branchement

L'unité de branchement est identique à celle du MPC604 aux différences suivantes près :

La table contenant l'historique des branchements précédents est de 2048 entrées. Ce qui permet d'affiner la prédiction..

La table contenant les adresses déjà calculées de branchements est de 256 entrées.

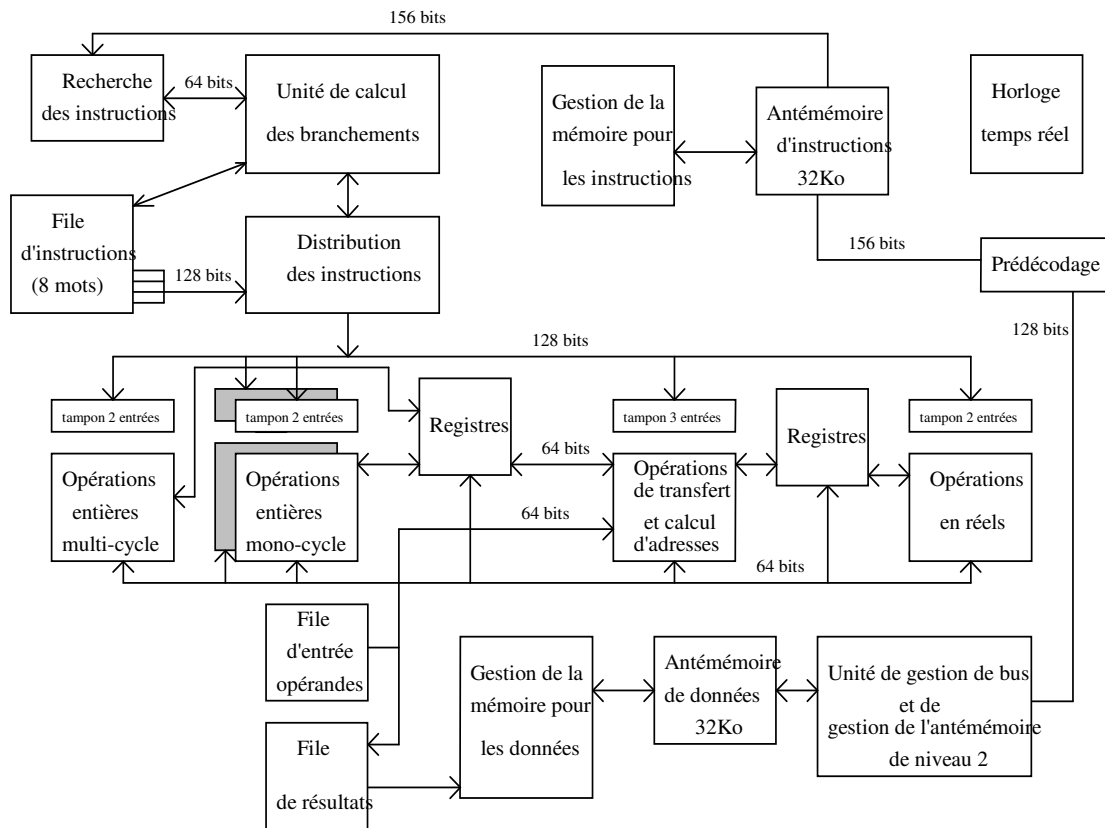
Les branchements relatifs au registre de condition sont prédits dès que la valeur de celui-ci est disponible.

Une pile de 8 entrées est utilisée pour prédire l'adresse de retour d'une procédure.

#### 3°) Les antémémoires

Le MPC620 met en œuvre 2 antémémoires de 16Ko chacune (instructions et données) protégées par un bit de parité au niveau de l'octet.

Un contrôleur d'antémémoire permet d'ajouter un second niveau d'antémémoire externe de 1 à 128 Mo ayant un bus de données de 128 bits et un mécanisme de détection et de correction d'erreurs.

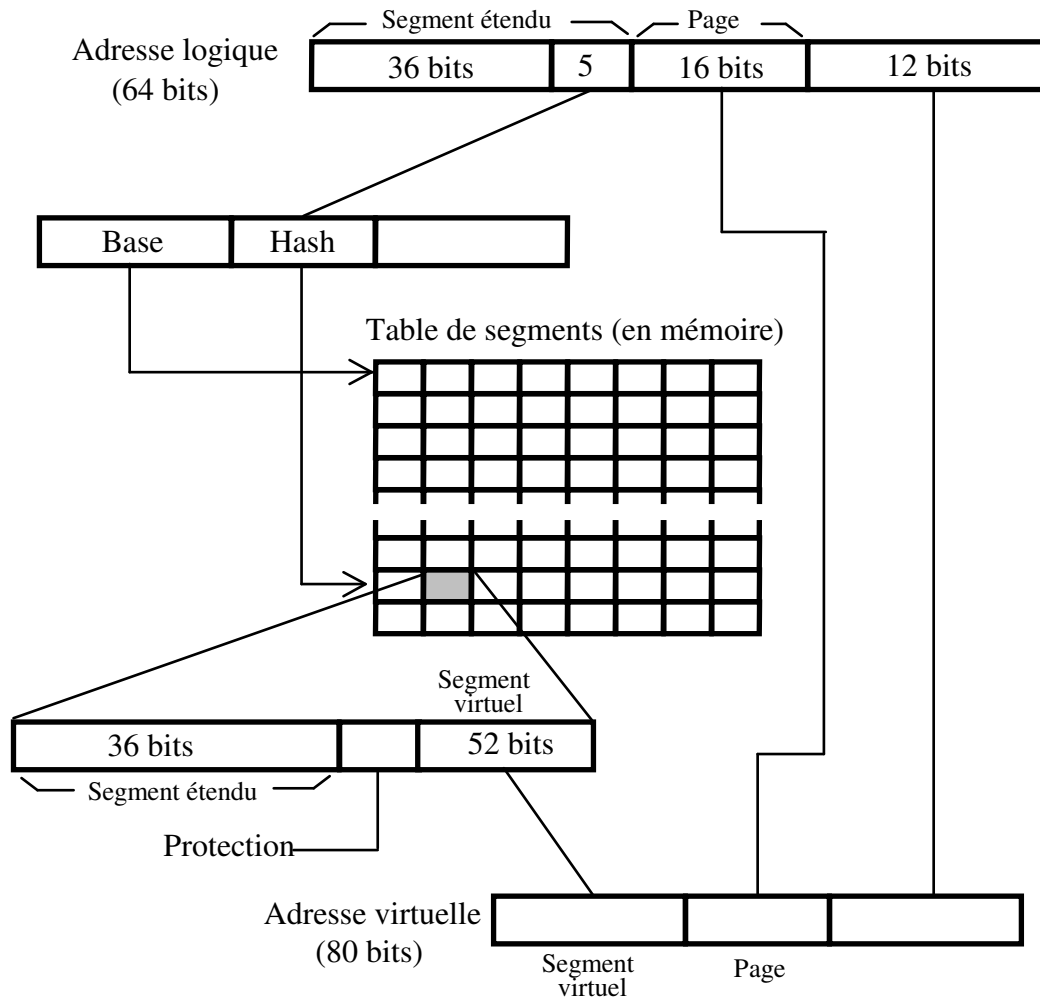


remarque : L'alimentation électrique est de 3,3V en externe de façon à conserver la compatibilité avec les autres processeurs. Toutefois en interne, le microprocesseur fonctionne sous 2,5V.

### 8.3.2 Mémoire

Les adresses logiques sont maintenant de 64 bits, les adresses virtuelles de 80 bits et les adresses physiques de 40 bits. Il a donc été nécessaire de modifier le gestionnaire de mémoire pour qu'il prenne en compte ces nouvelles tailles.

Le calcul des adresses se fait selon le schéma suivant :



A partir de l'adresse logique constituée comme indiqué dans le schéma ci dessus on procède comme sur le MPC601 (voir 8.1.1) pour obtenir l'adresse réelle sur 40 bits.

### **8.3.3 Registres et opérandes**

Tous les registres sont étendus à 64 bits. Afin de continuer à disposer d'instructions de transfert de données sur 32 bits, il a été nécessaire d'ajouter des instructions de transfert qui complètent par des 0 la moitié gauche du registre ou qui font une extension de signe. De la même façon de nouvelles instructions de décalage sur 64 bits sont créées.

### **8.3.4 Horloge**

Le MPC 620 fonctionne avec une horloge de 133 Mhz.

### **8.4 La troisième génération (G3)**

---

Cette génération reprend l'architecture du MPC603 en améliorant l'unité d'exécution en doublant les modules d'opérations sur les entiers. Ceci permet d'exécuter jusqu'à 3 instructions par cycle.

Les antémémoires ont été portées à 32Ko chacune.

Comme pour le MPC 620, l'alimentation électrique est de 3,3V en externe de façon à conserver la compatibilité avec les processeurs de première et deuxième génération. Toutefois en interne, le microprocesseur fonctionne sous 2,5V.

### **8.5 La quatrième génération (G4)**

---

Cette génération de processeurs devrait intégrer jusqu'à 50 millions de transistors et atteindre les vitesses d'horloge de 500 MHz.

### **8.6 La cinquième génération (2K)**

---

Dans cette génération MOTOROLA espère intégrer jusqu'à 100 millions de transistors et atteindre des vitesses d'horloge de 1GHz.

# CONCLUSION

On peut aisément voir à travers cette présentation du marché passé et présent des microprocesseurs que les constructeurs ont, dans un premier temps, proposé des processeurs de plus en plus complexes et de plus en plus performants (CISC : Complex Instruction Set Computers).

Ces progrès ne sont pas dus à des découvertes d'ordre général mais à l'intégration sur la puce de silicium de toujours plus de fonctionnalités. En effet, pas plus la mémoire virtuelle que la pagination, l'utilisation des antémémoires ou l'existence de niveaux de privilèges ne constituent des nouveautés et leur utilisation sur les mini et les gros ordinateurs ne date pas d'hier.

Ces microprocesseurs sont destinés à devenir le coeur de machines mettant en oeuvre des systèmes d'exploitation et des langages de haut niveau. Mais si leurs performances sont extraordinaires on peut s'apercevoir que nombre de leurs instructions sont peu ou pas utilisées par les compilateurs des langages implantés sur ces machines.

Aussi, les constructeurs se sont-ils, par la suite, orientés vers des architectures de processeurs différentes possédant un jeu d'instruction plus réduit (RISC : Reduced Instruction Set Computer) mais permettant l'exécution de la plupart d'entre elles en une seule période d'horloge et favorisant au maximum le recouvrement des instructions (pipe-line).

<b>INTRODUCTION</b>	<b>1</b>
<b>1 Définition d'un microprocesseur</b>	<b>1</b>
<b>2 Historique</b>	<b>1</b>
<b>LES ANCÊTRES</b>	<b>4</b>
<b>1 Les 4 bits</b>	<b>4</b>
<b>2 Les 8 bits</b>	<b>4</b>
<b>3 Un exemple le 6800</b>	<b>5</b>
3.1 Architecture interne	5
3.2 Les registres	6
3.3 Les opérandes	6
3.4 Les instructions	6
3.5 La mémoire et l'adressage	8
3.6 Quelques remarques	8
<b>LES MICROPROCESSEURS ACTUELS</b>	<b>10</b>
<b>1 Présentation générale</b>	<b>10</b>
<b>2 La famille INTEL</b>	<b>11</b>
<b>3 La famille MOTOROLA</b>	<b>13</b>
<b>LES MICROPROCESSEURS INTEL</b>	<b>16</b>
<b>1 Les 8086 / 8088 / 80186 et 80188</b>	<b>16</b>
1.1 Architecture interne	16
1.2 Registres	17
1.3 Les opérandes	18
1.4 Les instructions	18
1.5 La mémoire et l'adressage	21
1.6 L'horloge	22
1.7 Les Entrées/Sorties	22
1.8 Les interruptions	22
1.9 Sémaphores	23
1.10 Compatibilité	23
<b>2 Le 80286</b>	<b>25</b>
2.1 Architecture interne	25
2.2 Les registres	26
2.3 Les opérandes	26
2.4 Les instructions	26
2.5 L'adressage de la mémoire	27
2.6 Les privilèges	28
2.7 La protection	29
2.8 Le changement de tâche	29
2.9 Les interruptions	29
2.10 L'horloge	29

2.11	Compatibilité	30
<b>3</b>	<b>Le 80386</b>	<b>31</b>
3.1	Architecture interne	31
3.2	Les registres	31
3.3	Les opérandes	33
3.4	Les instructions	33
3.5	L'adressage de la mémoire	34
3.6	La pagination	34
3.7	Les privilèges et les changements de tâche	35
3.8	Les interruptions	35
3.9	L'horloge	35
3.10	Compatibilité	36
<b>4</b>	<b>Le 80486</b>	<b>37</b>
4.1	Architecture interne	37
4.2	Les registres et les opérandes	37
4.3	Les instructions	37
4.4	La gestion de la mémoire	38
4.5	L'horloge	38
4.6	Compatibilité	38
<b>5</b>	<b>Les coprocesseurs arithmétiques</b>	<b>40</b>
5.1	Architecture interne des coprocesseurs INTEL 8087 , 80287 et 80387	40
5.2	Représentation des données	41
5.3	Les coprocesseurs arithmétiques de la famille INTEL	41
<b>6</b>	<b>Le Pentium (P5)</b>	<b>42</b>
6.1	Architecture interne	42
6.2	Les registres et les instructions	43
6.3	La gestion de la mémoire	43
6.4	Prédiction d'instructions	44
6.5	Les réels	44
<b>7</b>	<b>Le Pentium MMX (P55C)</b>	<b>45</b>
7.1	Architecture interne	45
7.2	Les registres et les instructions	45
7.3	Mémoire	46
7.4	Horloge	46
<b>8</b>	<b>Le Pentium Pro ou P6</b>	<b>47</b>
8.1	Architecture interne et Fonctionnement	47
8.2	Horloge	48
<b>9</b>	<b>Le Pentium II ou Klamath</b>	<b>49</b>
<b>10</b>	<b>Le P7 ou Merced</b>	<b>49</b>
10.1	Principes	49
10.2	Instructions	49
10.3	Branchements	50
10.4	Erreurs de mémoire	50
10.5	Horloge	50
<b>11</b>	<b>Le P7 ou Merced</b>	<b>50</b>
	<b>LES MICROPROCESSEURS MOTOROLA</b>	<b>52</b>

<b>1 Le 68000 et le 68008</b>	<b>52</b>
1.1 Architecture interne	52
1.2 Les Registres	53
1.3 Les opérandes	53
1.4 Les instructions	53
1.5 Mémoire et adressage	56
1.6 Horloge	57
1.7 Les Entrées/Sorties	58
1.8 Privilèges	58
1.9 Les Interruptions	58
1.10 Environnement multiprocesseur	59
<b>2 Le 68010 et le 68012</b>	<b>60</b>
2.1 Architecture interne	60
2.2 Les registres	60
2.3 Les opérandes	60
2.4 Les instructions	60
2.5 Mémoire et Adressage	61
2.6 Horloge	61
2.7 Les Entrées/Sorties	62
2.8 Privilèges	62
2.9 Les Interruptions	62
2.10 Environnement multiprocesseur	62
<b>3 Le 68020</b>	<b>64</b>
3.1 Architecture interne	64
3.2 Registres	65
3.3 Les opérandes	65
3.4 Les instructions	65
3.5 Mémoire et adressage	67
3.6 Horloge	69
3.7 Les Entrées/Sorties	69
3.8 Privilèges	69
3.9 Les interruptions	69
3.10 Environnement multiprocesseur et coprocesseur	70
<b>4 Le 68030</b>	<b>72</b>
4.1 Architecture interne	72
4.2 Les Registres	73
4.3 Les instructions.	73
4.4 Mémoire et Adressage	74
4.5 La pagination	75
4.6 Horloge	76
4.7 Les autres caractéristiques	77
<b>5 Le 68040</b>	<b>78</b>
5.1 Architecture interne	78
5.2 Fonctionnement de l'unité centrale	78
5.3 Calcul des branchements	79
5.4 Exceptions	79
5.5 Gestion des antémémoires et de la mémoire	79
<b>6 Le 68060</b>	<b>81</b>
6.1 Architecture interne	81
6.2 Fonctionnement du 68060	81
6.3 Prédiction de branchements	82
6.4 Les antémémoires	83



<b>7 Les coprocesseurs</b>	<b>84</b>
<b>8 La famille Power PC</b>	<b>85</b>
<b>8.1 Les MPC 601 , 602 et 603</b>	<b>88</b>
8.1.1 Architecture	88
8.1.2 Les registres et les opérandes	91
8.1.3 Instructions	92
8.1.4 Test et mise au point	94
8.1.5 Le MPC 603	94
8.1.6 Le MPC 602	94
8.1.7 Horloge	94
<b>8.2 Le MPC 604</b>	<b>95</b>
8.2.1 Architecture	95
8.2.2 Horloge	96
<b>8.3 Le MPC620</b>	<b>97</b>
8.3.1 Architecture	97
8.3.2 Mémoire	98
8.3.3 Registres et opérandes	99
8.3.4 Horloge	99
<b>8.4 La troisième génération (G3)</b>	<b>100</b>
<b>8.5 La quatrième génération (G4)</b>	<b>100</b>
<b>8.6 La cinquième génération (2K)</b>	<b>100</b>
<b>CONCLUSION</b>	<b>101</b>