

LE MICROPROCESSEUR

GENERALITE

Définition 1 : Circuit intégré numérique complexe capable d'interpréter et d'exécuter les instructions d'un programme.

Description : Intégration sur une puce de fonctions logiques combinatoires (logiques et/ou arithmétiques) et séquentielles (registres, compteur, etc.)

Concept : Concept de microprocesseur créé par la Société Intel en 1971.

1er microprocesseur = le 4004 :

- unité de calcul 4 bits
- fonctionnant à 108 kHz
- intégration d'environ 2300 transistors.

Rôle :

1. Il organise l'enchaînement des tâches précisées dans la mémoire (programme)
2. Il rythme et synchronise l'exécution de ces tâches
3. Il gère les informations extérieures au système

Principe de von Neuman :

- ✓ Programme et données dans la même mémoire
- ✓ Exécution séquentielle des tâches

ARCHITECTURE DE BASE

Les 2 éléments principaux d'un microprocesseur :

1. Unité de commande
2. Unité de traitement

Unité de commande

- ✓ **Permet de séquencer le déroulement des instructions.**
- ✓ **Effectue la recherche en mémoire de l'instruction.**
- ✓ **Décode les instructions à exécuter.**

L'unité de commande est composée par :

1. **Le compteur de programme** : contient l'adresse de l'instruction à exécuter.
2. **Le registre d'instruction & décodeur d'instruction** : range les instructions à exécuter dans le registre d'instruction **puis décodées** par le décodeur d'instruction.
3. **Bloc logique de commande (ou séquenceur)** : **organise** l'exécution des instructions au rythme d'une horloge.

Unité de traitement (le cœur du microprocesseur)

- ✓ **Regroupe les circuits qui assurent les traitements nécessaires à l'exécution des instructions.**

1. **UAL (Unité arithmétique et logique) :**
un circuit complexe qui **assure les fonctions logiques** (ou, et, comparaison ...) **ou arithmétiques** (addition, soustraction).
2. **Registre d'état :** Glé/ment composé de 8bits dont chacun de ces bits est un indicateur et son état dépend du résultat de la dernière opération de l'UAL.
3. **Les accumulateurs :** ce sont des registres de travail qui **stocke** un opérande au début d'une opération arithm. Et le résultat à la fin de l'opération.

EXECUTION D'UNE INSTRUCTION

- ✓ **Le microprocesseur ne comprend que informations**

binaires

- ✓ **Chaque instruction est représentée par un code**

Différent

Un cycle d'exécution s'effectue en 3 phases :

Phase 1. Recherche de l'instruction :

- Adresse de l'instruction est placée sur le bus d'adresse par l'unité de commande
- Conteneur de la casse mémoire sélectionnée est disposé sur le bus de donnée
- Stockage de l'instruction dans le registre d'instruction

Phase 2. Décodage de l'instruction :

- L'unité de commande transforme l'instruction en une suite de commande nécessaire au traitement de l'instruction
- L'unité de commande récupère une valeur dans le bus de donnée si l'instruction en a besoin
- L'opérande est stocké dans un registre

Phase 3. Exécution de l'instruction :

- Le microprogramme réalisant l'instruction est exécuté

- Les drapeaux sont positionnés

N.B : les drapeaux (ou flag) sont des indicateurs d'état

- L'unité de commande positionne le PC pour l'instruction suivante

JEU D'INSTRUCTION

- ✓ Ensemble des opérations élémentaires que le microprocesseur pourra exécuter.

1. **Type d'instructions** : les 4 groupes d'instructions que l'on retrouve dans chaque microprocesseur sont :

- Transfert de données
- Operation arithmetiques
- Operation logiques
- Contrôle de sequence

2. **Codage des instructions** : chaque instruction est toujours codée sur un nombre entier d'octets afin de faciliter son decodage par le processeur, une instruction est composée de deux champs :
 - **Le code instruction** (*indique au proc. quelle instruction realiser*)
 - **Le champ opérande** (*qui contient la donnée*)
3. **Mode d'adressage** : définit la manière dont le microproc va accéder à l'operande :
 - **Adressage de registre** (*traite la donnée du registre*)
 - **Adressage immédiat** (*definit immediatement la valeur de la donnee*)
 - **Adressage direct** (*traite la donnée en mémoire*)
4. **Temps d'exécution** : Chaque instruction nécessite un cycle d'horloge pour s'effectuer :
 - **La durée de ce cycle** dépend de la fréquence d'horloge du séquenceur

- **Le nombre de ce cycle** dépend de la complexité de l'instruction et aussi du mode d'adressage

LANGAGE DE PROGRAMMATION

- ✓ **Langage machine** : langage bas niveau compris par le Microprocesseur
- ✓ **Langage assembleur** : langage le plus proche du langage machine
- ✓ **Langage de programmation** : langage haut niveau permet de faire abstraction du microprocesseur, facilité de programmation

Lng de prg° -> *Compilation* -> **Lng assembleur** -
> *assemblage* -> **Lng machine**

PERFORMANCE D'UN MICROPROCESSEUR

- ✓ On peut caractériser la puissance d'un microproc. par **le nombre d'instructions qu'il est capable de traiter par seconde.**

1. **CPI (Cycle Par Instruction)** : nombre moyen de cycle d'horloge nécessaire pour l'exécution d'une instruction.
2. **MIPS (Millions d'Instruction Par Seconde)** : la puissance de traitement du microproc.

AMELIORATION DU MICROPROCESSEUR

Il existe 2 méthodes pour augmenter les performances d'un microproc. :

~ AMELIORATION 1 ~

- ✓ L'ensemble de l'amélioration des microproc visent à diminuer le temps d'exécution du programme en **augmentant sa fréq d'horloge. Or, cela**

provoque un surcroît de consommation
et de température (*limitation matérielle*)

- ✚ **SOLUTION** : Equiper les proc. de systèmes de refroidissement OU diminuer la tension d'alimentation

~ AMELIORATION 2 ~

- ✓ Augmenter la puissance de traitement du proc. en **diminuant le CPI** . (*choix du jeu d'instruction[CISC/RISC] & l'architecture de base [superscalaire, pipeline, mém. cache]*)

- ✚ **SOLUTION 1** : Choix du jeu d'instruction CISC/RISC

A. **L'architecture CISC/RISC** (*Comparaison*) :

CISC	RISC
------	------

<i>Ex : intel</i>	<i>Ex : android</i>
<ul style="list-style-type: none"> ➤ Instru. complexes prenant plusieurs cycles ➤ Instru. au format variable ➤ Decodeur complexe (<i>micro-code</i>) ➤ Peu de registres ➤ Toutes les instructions sont susceptibles d'accéder à la mém. ➤ Bcp de modes d'adressage ➤ Compilateur simple 	<ul style="list-style-type: none"> ➤ Instru. simples ne prenant qu'un seul cycle ➤ Instru. au format fixe ➤ Decodeur simple (<i>câblé</i>) ➤ Bcp de registres ➤ Seules les instru. LOAD et STORE ont accès à la mém. ➤ Peu de modes d'adressage ➤ Compilateur complexe

B. L'architecture CISC/RISC (*Raison d'utilisation*) :

CISC (<i>passé</i>)	RISC (<i>année 70</i>)
<ul style="list-style-type: none"> ➤ Il semblait préférable d'ajouter au jeu d'instructions du microproc une instruction complexe car la mem. travaillait très lentement /rapport au proc. 	<ul style="list-style-type: none"> ➤ 80% des traitements de langage de haut niveau font appel à seulement 20% du jeu d'instructions d'un microprocesseur. D'où l'idée de réduire le jeu d'instructions et d'en améliorer la vitesse de

✚ SOLUTION 2 : Amélioration de l'architecture de base (pipeline, superscalaire, mem. cache)

PIPELINE	
Principe	
Problème(s)	<ul style="list-style-type: none"> ✓ Plus le pipeline est long, plus le nbr de cas où il est impossible d'atteindre la perf max est élevée. ✓ Les 3 cas de dégradations de perf d'un proc. pipeline (appelée « aléas ») sont : <ul style="list-style-type: none"> • Aléa structurel : au cas où 2 instructions ont besoin d'utiliser la même ressource du proc (<i>conflit de dépendance</i>) • Aléa de données : au cas où une instruction produit un résultat et que l'instruction suivante utilise ce résultat avant d'être écrit dans un

registre

- **Aléa de contrôle** : au cas où une instruction de branchement est exécutée, il faut d'abord connaître l'adresse de destination du branchement afin de charger l'instruction suivante. **Or**, certaines instructions suivent le saut et d'autres en train d'être traitées dans les étages inférieurs qui sont ignorés.

SOLUTION :

1. Vider le pipeline
2. Spécifier après le branchement des instructions qui seront tjrs exécutées
3. Fait appel à la prédiction de branchement

Remarque (s)

- ✓ Le temps de traitement dans chaque unité doit être à peu près égal sinon les unités rapides doivent attendre les unités lentes.
- ✓ **Bulles de pipeline** (*NOP = No Operation*) : ce sont les étages vacants (qui ne sont pas occupés) du

pipeline.

- ✓ **Prédiction de branchement (aléa de contrôle) :** a pour but de recenser (*denombrer*) lors de branchement le comportement le plus probable. Ces mécanismes de prédiction permettent d'atteindre une fiabilité de 90 à 95%

SUPERSCALAIRE

Principe

- ✓ Consiste à doter le microproc de plusieurs unités de traitement travaillant en parallèle afin d'exécuter plusieurs instructions en même temps.

Problème(s)

- ✓ Il faut pouvoir soutenir un flot important d'instructions et pour cela **disposer d'un cache performant.**

Remarque(s)

- ✓ C'est le type d'architecture mise en œuvre dans les **premiers**

CACHE MEMOIRE

Principe

- ✓ Un mémoire très rapide intermédiaire entre le microproc et la mémoire principale.
- ✓ Il existe toujours un écart de performance entre microproc & mém. principale, **i-e** que la mém. principale n'est pas en mesure de délivrer les info° aussi rapidement que le proc est capable de les traiter. La **fonction du mém. cache** est alors de stocker les infos les plus récentes ou les plus souvent utilisées par le microproc. **Ainsi** permettant au microproc d'acquérir les données à sa vitesse propre.
- ❖ Le microproc n'a pas conscience de la présence du cache mém. et lui envoie des requêtes comme s'il s'agit de la mém. principale. Puis il y a ce qu'on appelle :

	<ul style="list-style-type: none"> • Succès de cache : lorsque la donnée/instru° requise est déjà présente dans la cache est alors envoyée directement au proc. • Défaut de cache : lorsque la donnée/instru° n'est pas dans le cache et le « <i>contrôleur de cache</i> » envoie une requête à la mém. principale et stocke l'information récupérée puis la renvoie au proc.
Problème(s)	<ul style="list-style-type: none"> ✓ On la réalise à partir de cellule SRAM de taille réduite à cause du coût. Sa mém. est donc très inférieure à celle de la mém. principale
Remarque(s)	<ul style="list-style-type: none"> ✓ C'est le type d'architecture mise en œuvre dans les premiers Pentium d'Intel (<i>en 1993</i>) ✓ Contrôleur de cache : permet à la mém. cache de mieux gérer les données stockés (<i>Ce qu'il faut remplacer ou pas</i>)

PROCESSEUR SPECIAUX

Microcontrôleur

- ✓ Sys. Minimum sur une seule puce. (CPU + RAM + ROM + ports E/S + compteurs programmables)
- ✓ Adapté pour répondre au mieux aux besoins des applications embarquées (*électroménagers, lecteur carte à puce etc...*)
- ✓ Moins puissant au niveau de la :
 1. Rapidité
 2. Taille de données traitables
 3. Taille de mém. adressable

Processeur DSP (*Digital Signal Processor ou Processeur de signal*)

- ✓ Le processeur DSP est optimisé pour effectuer du traitement numérique du signal (calcul de FFT, convolution, filtrage numérique, etc...)
- ✓ Adapté pour répondre au mieux aux besoins des applications multimédia (*lecteur CD, MP3, VIDEO, etc...*)
- ✓ L'électronique grand public (télévision numérique, téléphone portable, etc...)
- ✓ L'automatique, l'instrumentation, l'électronique automobile, etc...

COMPARAISON PROC AMD & intel

AMD Athlon

- **9 unités de traitement**
 - **1 ALU (traitement entier)**
 - ✓ 3 unités pour traitement des données
 - ✓ 3 unités pour l'adressage des données
 - **1 FPU (traitement réel)**
 - ✓ 1 FPU store
 - ✓ 1 Fadd/MMX/3Dnow
 - ✓ 1 Fmul/MMX/3Dnow
- **Pipeline (*entier*) : 10 étages**
- **Pipeline (*flottant*) : 15 étages**
- **Prédiction dynamique**
- **Exécution du traitement (*out-of-order*)**

Intel Pentium III

- **5 instructions exécutées en même temps sur 5 ports :**
 - **Port0** : ALU, FPU, AGU MMX et SSE
 - **Port1** : ALU, SSE, MMX
 - **Port2** : AGU (*load*)
 - **Port3** : AGU (*store*)
 - **Port4** : Store Data Unit
- **Pipeline (*entier*) : 12 étages**
- **Pipeline (*flottant*) : 25 étages**
- **Prédiction dynamique**
- **Exécution du traitement (*out-of-order*)**
- **3 unités de décodage parallèles (*1 micro-programmées, 2***

- **6 unités de décodage parallèles** (*3 micro-programmées, 3 câblés mais seules 3 peuvent fonctionner en même temps*)
- **Cache L1 : 128Ko**
 - o 64 Ko pour les données
 - o 64 Ko pour les instructions
- **Cache L2 : 512Ko à 8Mo** avec vitesse programmable (*1/2 ou 1/3 de la CPU*)
- **22 M de transistors**

câblés

- **5 pipelines de 10 étages**
 - **Cache L1 : 32Ko**
 - o 16 pour données
 - o 16 pour instructions
- 1. Cache L2 : 512Ko**
(*vitesse 1/2 de la CPU*)
- 9.5 M de transistors**

LES ECHANGES DE DONNEES

GENERALITE

- L'interface d'E/S permet au microprocesseur de communiquer avec le monde extérieur (*Clavier, souris, écran, imprimante, disque dur, lecteur C, etc...*)
- Les techniques d'E/S sont très important pour les performances du système

E/S

Rôle :

1. Connecter le périphérique au bus de données
2. Gérer les échanges entre le proc et le périphérique

Constituée par :

1. Un **registre de commande** dans lequel le proc décrit le travail à effectuer (sens de transfert, mode de transfert)
2. Un ou plusieurs **registres de données** qui contiennent les mots à changer entre le périphérique et la mémoire
3. Un **registre d'état** qui indique si l'unité d'échange est prête, si l'échange s'est bien déroulé, etc...

TECHNIQUES D'ÉCHANGE DE DONNÉES

- Avant d'envoyer ou de recevoir des infos, le microprocesseur doit savoir si un périphérique est prêt à recevoir ou à transmettre une info pour que **l'échange de données** se fasse correctement
- Il existe 2 modes d'échange d'infos :

ECHANGE PROGRAMME

❖ **Par scrutation** (*type d'échange lent*)

- o **Principe** : le microprocesseur interroge l'interface (*périphérique*) pour savoir si des transferts sont prêts, tant que ces derniers ne sont pas prêts, le microproc attend
- o **Avantages** : Simplicité

o Inconvénients :

- ✓ Le microprocesseur se retrouve souvent en phase d'attente
- ✓ L'initiative de l'échange de données est dépendante du programme exécuté

❖ **Par interruption** (*type d'échange lent*)

o Principe :

- ✓ Suspendre un programme pour en exécuter un autre
- ✓ Pouvoir le faire n'importe quand
- ✓ Pouvoir revenir au programme interrompu et continuer à faire ce que l'on était en train de faire

o Les interruptions sont classées par ordre de priorité au cas où plusieurs interviendraient en même temps

o Le programme principal et le programme d'interruption vont utiliser les mêmes ressources microprocesseur (*registre, cache, etc...*)

- ✓ Nécessité de sauvegarder le contexte avant d'exécuter le sous-programme d'interruption
- ✓ Nécessité de restituer le contexte à la fin du sous-programme d'interruption.

ECHANGE DIRECT AVEC LA MEMOIRE (DMA)

- o Permet le transfert de données entre l'interface E/S et la mémoire sans passer par le microprocesseur**
- o Nécessité d'un circuit supplémentaire = Contrôleur de DMA**
- o Avantages :**
 - ✓ Transfert de données sans passer le microprocesseur
 - ✓ Le microprocesseur peut exécuter une autre tâche
- o Inconvénients :**
 - ✓ Le contrôleur DMA prend possession du bus = le

microprocesseur n'a plus accès à la mémoire

TYPES DE LIAISONS

Liaison parallèle	Liaison série
<ul style="list-style-type: none">• Tous les bits d'un mot sont transmis en même temps sur autant de fils que de bits à transmettre	<ul style="list-style-type: none">• Tous les bits d'un mot sont transmis les uns à la suite des autres sur un seul fil <p>➤ La transmission en série peut se concevoir 2 façons différentes :</p> <ol style="list-style-type: none">1. Mode synchrone : <i>l'émetteur et le récepteur possède une horloge synchronisée qui cadence la transmission</i>2. Mode asynchrone : la

	<i>transmission s'effectue au rythme de la présence des données. Les caractères envoyés sont encadrés par un signal start / stop</i>
❖ On caractérise une liaison par sa vitesse de transmission ou débit <i>(en bit/s)</i>	
<ul style="list-style-type: none"> • Faible distance • Vitesse élevée • Coût au m élevé 	<ul style="list-style-type: none"> • Longue distance • Vitesse peu importante • Coût plus faible

❖ Paramètres d'un protocole de liaison série
<ul style="list-style-type: none"> • Longueur du mot transmis : 7 ou 8 bits • Vitesse de transmission : détermine la fréquence d'horloge: <i>110 bit/s à 128000 bit/s</i> • Bit de start : inverse état repos • Bit de stop : identique à l'état repos (<i>parfois 1,5 ou 2 bits stop</i>)

- **Bit de parité** : parfois le mot est suivi d'un bit de parité servant à la détection d'erreur

Parité paire : nbre total de bit est pair

Parité impaire : nbre total de bit est impair

- **La durée de transmission d'une donnée dépendra des paramètres du protocole choisi**

❖ **Contrôle de flux d'une liaison série**

- **Permet d'envoyer des informations seulement lorsque le récepteur est prêt**
- **Contrôle logiciel :**
 - ✓ Lorsque le récepteur ne peut plus recevoir de données, il émet une information sur la ligne série
 - ✓ L'émetteur doit toujours être à l'écoute avant d'émettre une donnée
- **Contrôle matériel :**
 - ✓ Il faut rajouter des lignes de contrôle supplémentaire en plus de la ligne de transmission.

