Jeu d'instructions

Le **jeu d'instructions** est l'ensemble des opérations qu'un processeur d'ordinateur peut exécuter, c'est-à-dire l'ensemble des circuits logiques qui y sont câblés. Ces circuits permettent d'effectuer des opérations élémentaires (addition, ET logique…) ou plus complexes (division, passage en mode basse consommation…).

Chaque instruction machine est désignée par un code, dit « code-opération » et abrégé *opcode*, auquel peuvent être associées des opérandes. Une suite d'instructions et des opérandes, stockées dans la mémoire, forme un programme informatique. Chaque processeur, à son démarrage, commence à exécuter ses instructions à partir d'une adresse fixe spécifiée par le constructeur.

On distingue les microprocesseurs à jeu d'instructions complexe (CISC) et réduit (RISC) à la taille de leur jeu d'instruction. Il existe plusieurs famille de jeu d'instructions.

Sommaire

Classes des jeux d'instructions

Les jeux d'instructions des processeurs diffèrent surtout dans la façon dont sont spécifiés les opérandes des instructions. Voici schématiquement représentées différentes classes.

Légende :

- Les registres, l'accumulateur ou la pile sont représentés dans les tons jaunes.
- La mémoire centrale est représentée dans les tons mauves.
- Le chemin de données (mode *lecture*) entre registres (ou l'accumulateur ou la pile) est en vert,
- Le chemin de données (mode *écriture*) entre la sortie de l'Unité Arithmétique et Logique et les registres (ou l'accumulateur ou la pile) est en bleu ;
- Le chemin de données (mode *lecture* ou *écriture*) avec la mémoire est en violet.

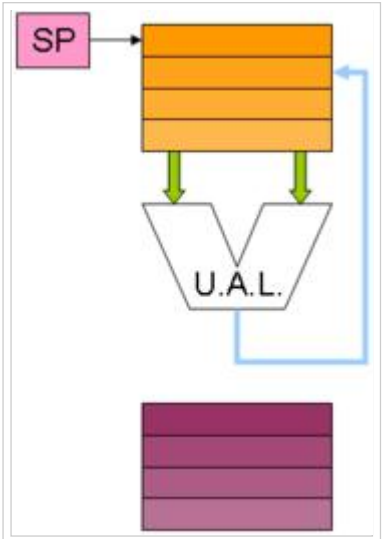
« 0 adresse »

Dans cette architecture, les instructions vont directement agir sur la pile. Les opérandes sont automatiquement chargés depuis le pointeur de pile (SP, Stack Pointer), et le résultat est à son tour empilé.

L'opération A = B + C sera traduite par la séquence suivante :

```
PUSH B  ; Empile B
PUSH C  ; Empile C
ADD     ; Additionne B et C
POP A   ; Stocke le haut de la pile à l'adresse A et dépile
```

Ce type d'architecture est populaire chez les utilisateurs de calculatrices HP fonctionnant en notation polonaise inversée (post-fixée), en tout cas plus que ceux qui ont pu utiliser les machines Burroughs de la gamme B 5000 ou des miniordinateurs Hewlett-Packard de la gamme HP 3000. Ce type d'architecture est aussi utilisé pour le FPU des processeurs x86.



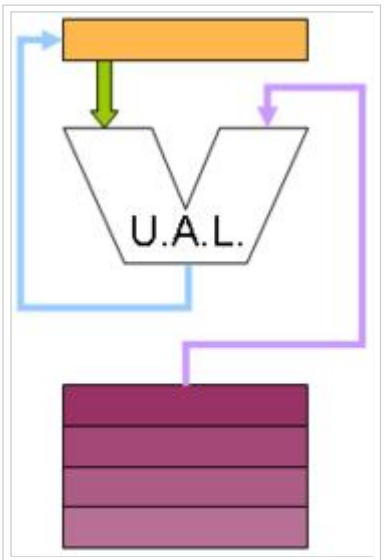
Architecture d'un processeur « 0 adresse », dite « à pile ».

« à accumulateur »

Sur une machine de ce type, historiquement ne disposant que d'un seul registre, appelé Accumulateur, tous les calculs se font implicitement sur celui-ci.

L'opération A = B + C sera traduite par la séquence suivante :

```
LOAD  B  ; copie le contenu de l'adresse B dans l'accumulateur
ADD   C  ; ajoute le contenu de l'adresse C avec le contenu de l'acc
           l'accumulateur
STORE A  ; stocke la valeur de l'accumulateur à l'adresse A
```

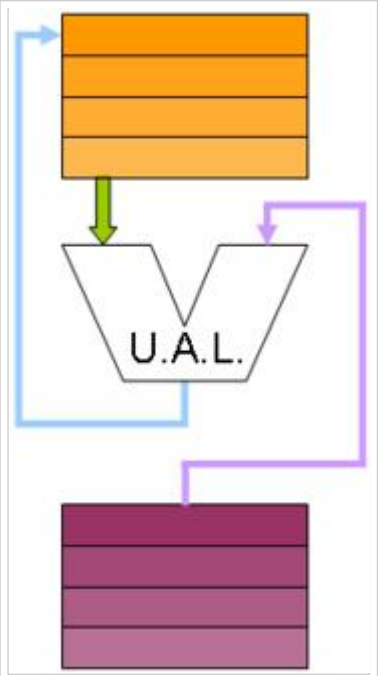


Architecture d'un processeur avec un accumulateur ».

« une adresse, registre - mémoire »

Ici une instruction peut avoir comme opérande un ou plusieurs registres (typiquement un ou deux) et une adresse mémoire. L'exemple A = B + C peut donc être traduit par la séquence :

```
LOAD  R0, B      ; copie le contenu de l'adresse B dans le registre
ADD   R1, R0, C  ; R1 = R0 + C
STORE R1, A      ; stocke la valeur de R1 à l'adresse A
```



Architecture d'un processeur « registre vers mémoire ».

« registre - registre »

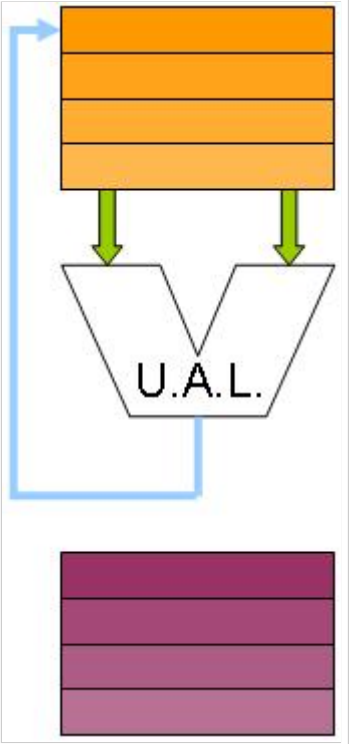
Si les instructions ne peuvent avoir que des registres comme opérandes, il faut deux instructions, LOAD et STORE par exemple, pour respectivement charger un registre depuis une location mémoire et stocker le contenu d'un registre à une adresse donnée.

Le nombre de registres est un facteur important.

Les processeurs RISC actuels sont tous de ce type.

La séquence A = B + C sera traduite en :

```
LOAD  R0, B       ; charge B dans le registre R0
LOAD  R1, C       ; charge C dans le registre R1
ADD   R2, R0, R1  ; R2 ← R0 + R1
STORE R2, A       ; stocke R2 à l'adresse A
```



Architecture d'un processeur « registre vers registre ».

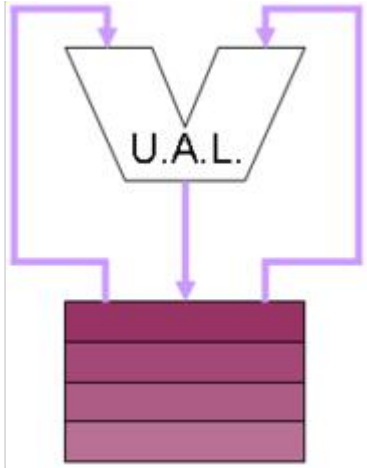
« mémoire - mémoire »

Tous les opérandes d'une instruction sont des adresses mémoire. C'est par exemple le cas pour le superordinateur vectoriel CDC Cyber 205. Cette machine était le concurrent du Cray 1 qui lui *devait* charger les vecteurs dans des registres *préalablement* à chaque calcul.

Le VAX de DEC peut aussi être programmé de cette façon.

L'expression A = B + C :

```
ADD A, B, C  ; Stocke a l'adresse A la somme B + C
```



Architecture d'un processeur « mémoire vers mémoire ».

Grandes Familles

RISC et CISC

Les processeurs CISC embarquent un maximum d'instructions souvent très complexes mais prenant plusieurs cycles d'horloge. À l'opposé, les processeurs RISC ont un jeu d'instructions plus réduit mais chaque instruction n'utilise que quelques cycles d'horloge.

Tous les instructions des processeurs RISC sont de la classe « registre-registre ».

Familles de processeurs

Le jeu d'instruction x86 (CISC) équipe tous les processeurs compatibles avec l'architecture intel (qu'ils soient construit par Intel ou AMD).

La famille des processeurs PowerPC utilise un jeu d'instruction RISC.

Voir aussi

- MMX, 3DNow!, SSE, AltiVec
- Registre
- Microprocesseur
- Assembleur