

3. Chemin des données, cycle machine, microcommandes; parallélisme des opérations. Architecture et fonctions des unités (UAL, module de contrôle, registres, etc.) (suite)

3.2. Architecture et fonctions des unités (UAL, module de contrôle, registres, etc.) (suite)

3.2.4.3. Compteur de programme

Le plus souvent c'est un registre de 16 bits, qui contient l'adresse de l'instruction suivante que l'exécution est à la veille.

3.2.4.4. Registre de code conditionnel (CCR-Code Condition Register)

Le plus souvent c'est un registre de 8 bits, dans lequel chaque bit montre une condition déterminée du résultat de l'instruction qui juste était exécutée. Ces 8 bits peuvent être vérifiés individuellement par le logiciel (le programme) et de différentes actions en résultat de ces vérifications peuvent être entreprises.

3.2.4.5. Décodeur d'instructions

Ce registre décode le COP et définit quelle est l'instruction, déclarée à être exécutée. Cela représente le traitement de l'instruction courante.

4. Couche microprogrammée-machine hypothétique; architecture et analyse. Modèle à programme. Structure des instructions. Méthodes d'adressage

4.1. Couche microprogrammée - machine hypothétique; architecture et analyse(*Résumé***)**

Vue de la couche microprogrammée, un processeur central d'ordinateur comprend deux composantes principales: le chemin des données et l'unité de commande. Le chemin des données est constitué d'une Unité Arithmétique et Logique et d'un ensemble de registres généraux ou mémoires bloc-notes, le tout relié aux entrées/sorties de l'Unité Arithmétique et Logique par des bus internes. Le cycle fonctionnel de la micromachine consiste à extraire les opérandes des registres, à effectuer dans l'Unité Arithmétique et Logique un traitement sur ces opérandes et à ranger éventuellement le résultat du traitement dans un registre.

Le module de commande comprend une mémoire de commande qui contient les microinstructions du microprogramme à exécuter. Chaque microinstruction est source d'une génération de signaux, les microcommandes, nécessaires à la commande des composants du chemin des données pendant l'exécution d'un cycle fonctionnel de la micromachine (un microcycle). Chaque microcycle est divisé en plusieurs sous-cycles pilotés par une horloge.

Le séquenceur des microinstructions est propre à chaque machine. Certaines machines disposent d'un compteur ordinal dans la micromachine (le micro compteur ordinal). Quelquefois la micro-instruction contient elle-même l'adresse de base de la micro-instruction suivante.

Les micro-instructions peuvent être à encodage horizontal correspondant le plus souvent à un bit par microcommande, ou à encodage vertical c'est-à-dire plusieurs champs de bits, nécessitant souvent un décodage complexe pour fournir les microcommandes ou encore à une solution intermédiaire. L'organisation horizontale conduit à des micro-instructions très larges en nombre de bits et à des micromachines à haut degré de parallélisme et performantes en vitesse d'exécution. L'organisation verticale concerne plutôt des petites machines, lentes, moins complexes et disposant de micro-instructions de moindre largeur.

Les performances de la micromachine peuvent être améliorées en utilisant diverses techniques. L'une d'entre elles, la nano programmation, consiste à définir des microinstructions qui sont en fait des pointeurs (d'un faible nombre de bits) qui réfèrent des nano instructions (d'un nombre de bits important) dont l'objet est de commander le chemin des données. La nano programmation a pour but de réduire le nombre de bits de la mémoire de commande au prix d'une augmentation relative du temps d'exécution.

4.2. Modèle à programme

Le modèle à programme de tout ordinateur (où configuration d'ordinateur) est construit des registres de son processeur central (CPU). Autrement dit- le modèle à programme unifie les registres, accessibles à programme, de la structure de l'ordinateur. Leur contenu et leur destination sont

différents pour les différents modèles de processeurs. Leur longueur (capacité) dépend de la longueur du mot de machine (W) pour l'ordinateur concret- Figure 13 :

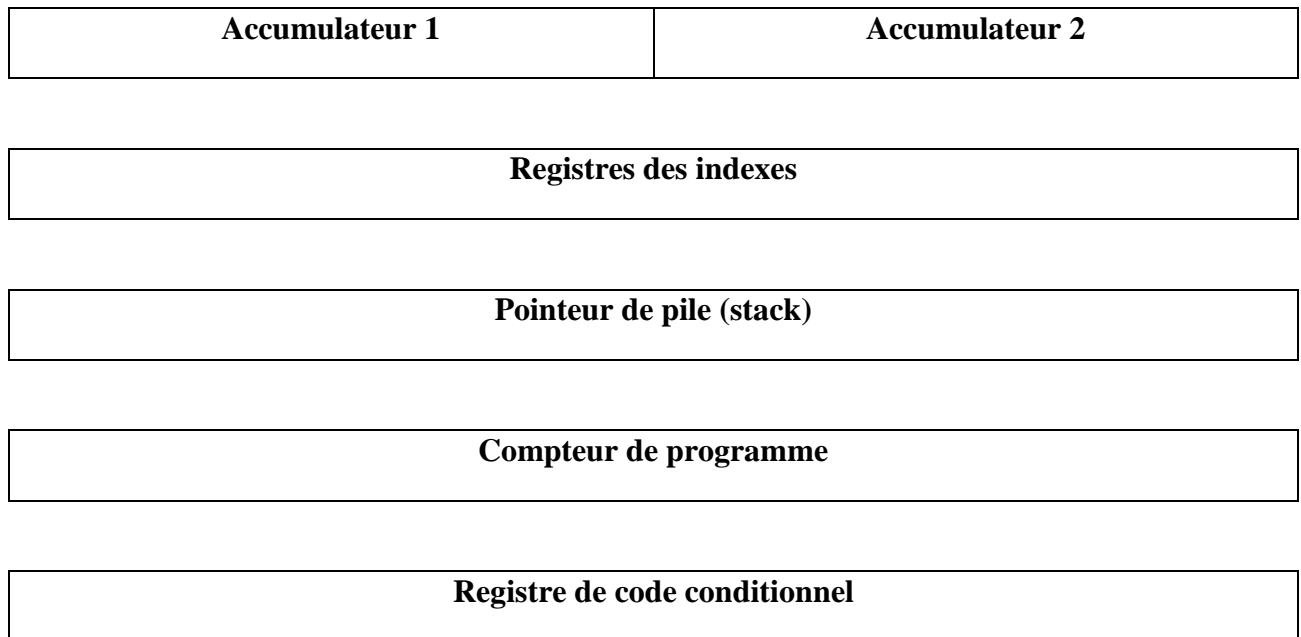


Figure 13. Exemple du modèle à machine

4.3. Structure des instructions

Maintenant nous allons nous arrêter sur l'un des questions de base dans le fonctionnement de l'ordinateur - la structure des instructions. Pour ce but nous allons présenter le cycle de toutes les instructions de base avec un schéma commun – voir Figure 14.

Le cycle de toute instruction est composé des : COP (code de l'instruction) plus A (adresse de l'instruction) – voir Figure 10.

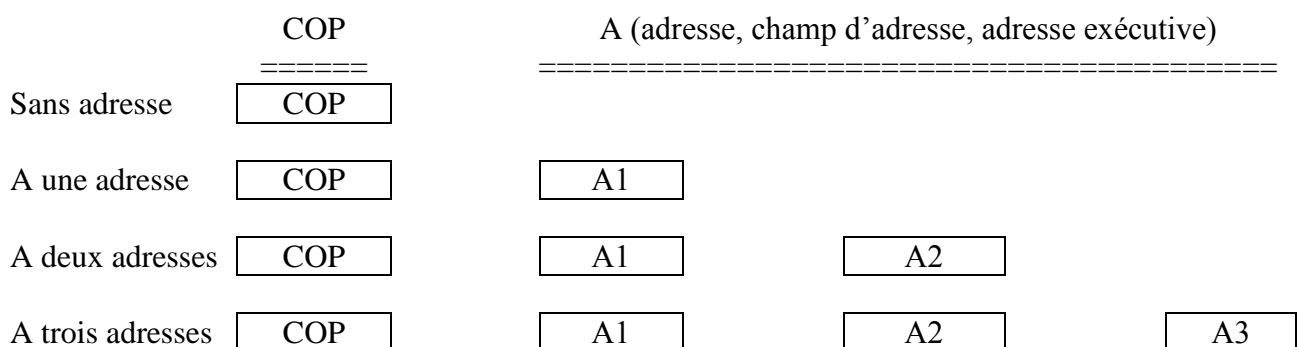


Figure 14. Structure de base des instructions

La structure de l'instruction sans adresse apparait quand il y a la mise à l'état initial d'un registre de service où l'établissement d'une marque donnée dans un état choisi ;

La structure des instructions à une adresse sont utilisées dans les processeurs qui contiennent un registre interne de service, appelé accumulateur (A_{cc}) : $(A_{cc}) + (A1) \rightarrow (A_{cc})$;

La structure des instructions à deux adresses montre les adresses des deux opérands et le résultat est enregistré sur la première adresse: $(A1) + (A2) \rightarrow (A1)$;

La structure des instructions à trois adresses est la plus proche de l'opération traditionnelle. Elle montre les adresses des deux opérands et l'adresse, où le résultat va être enregistré: $(A1) + (A2) + (A2)$.

Et à la fin de ce point il faut définir la dimension classique du mot (W) dans les ordinateurs : elle est longue de 8 bits (chiffres binaires), qui représentent 1 byte (octet). Actuellement le mot dans les ordinateurs, comme suite des nouvelles possibilités des ordinateurs et la capacité des bus de données et d'adresses des processeurs actuels est de 2 bytes.

Ainsi dans les ordinateurs actuels sont définies :

- **W** – mot à dimension de 2 bytes ;
- **2W** –double mot à dimension de 4 bytes ;
- **4W**- quadruple mot à dimension de 8 bytes.

4.4. Méthodes d'adressage

Sous la notion ADRESSAGE on comprend l'adressage dans la mémoire opérationnelle du tout ordinateur, indépendamment de sa structure et de sa composition. On adresse toute cellule cherchée de cette mémoire opérationnelle, après quoi on lit son contenu. L'adresse de toute cellule de cette mémoire est composée de deux parties intégrantes :

- *Code de l'adresse* où encore nommé *code de l'opération* – **COP** ;
- *Adresse exécutive, champ d'adresse, adresse* - A_i (ChA), où $i=1, 2, \dots, k$, représente l'index du nombre des chiffres dans le ChA de la concrète mémoire opérationnelle de l'ordinateur. Par exemple dans un ordinateur moderne et ordinaire le ChA est de 20 ou au plus de 36 chiffres, ou le bus (le chemin) d'adresses de cet ordinateur est de 20 (36) chiffres.

L'adressage dans la mémoire opérationnelle de l'ordinateur est exécuté sur la base de plusieurs méthodes principales. Ces méthodes peuvent être combinées entre elles. Dans tous les processeurs centraux modernes (CPU) on soutient une diversité assez nombreuse de méthodes d'adressage, qui sont générées du champ spécial, nommé ChA. Il est composé de « k »-bits et complète le champ pour le COP :

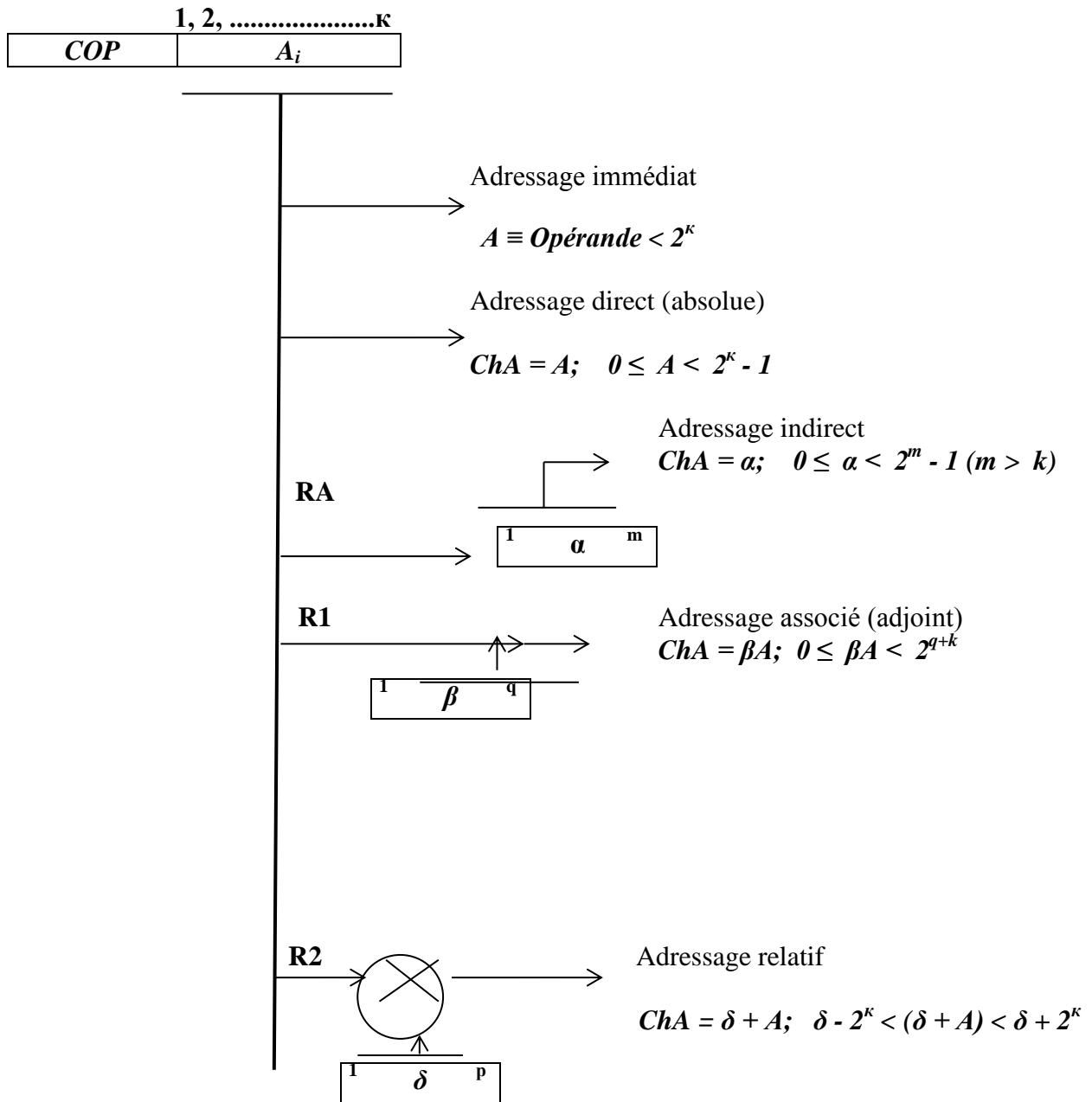


Figure 15. Méthodes de base pour adressage

➤ **Adressage immédiat :**

Chez-il l'opérande elle-même se contient dans l'instruction. Cette méthode est la plus rapide, mais limite la valeur maximale de l'opérande jusqu'à $2^{\kappa} - 1$;

➤ **Adressage direct (absolue) :**

Cet adressage traite le ChA comme l'adresse exécutive, ce qui permet un retour direct vers la mémoire opérationnelle. Ici on a aussi, comme plus haut remarqué, la limitation du ChA, par lequel on puit opérer avec pas plus de 2^{κ} adresses. Il est utilisé le plus souvent pendant le retour vers la mémoire de registre (les registres internes du processeur central), qui en général est avec un volume limité, mais avec une rapidité maximale. Dans ce cas, l'adressage des registres, la longueur de « k »-bits du ChA sera suffisante pour opérer avec le ChA de l'un des registres du processeur central ;

➤ **Adressage indirect :**

Dans cet adressage on utilise un intermédiaire **RA** (registre de service où cellule de la mémoire opérationnelle), dont le contenu α est traité comme ChA. Ainsi on élargit les dimensions du ChA - $2^m > 2^k$ ($m > k$). Avec cela on augmente le temps d'adressage, parce qu'on a besoin d'un adressage intermédiaire ;

➤ **Adressage associé (adjoind) :**

Cet adressage permet la composition du ChA de deux parties : la partie la plus significative – β , est prise du registre de service **RI** (par exemple le registre d'adresse initiale d'une page de la mémoire opérationnelle, et la partie la moins significative – de l'instruction courante. Ainsi le ChA va être complété de q bits de β plus k bits du ChA. Cela permettra un adressage dans un très large diapason par comparaison avec les précédentes méthodes : du 0 à 2^{q+k}

➤ **Adressage relatif :**

Le ChA de ce type d'adressage se calcule par l'addition de la valeur A_i du ChA de l'instruction plus le contenu δ d'un registre de service **R2**. On l'utilise pour la transmission du control (interruption) dans le logiciel (programme), aussi comme pour ainsi nommé adressage de base, où δ est nommée adresse de base et elle est mémorisée dans le registre de service **R2**. Ce registre contient l'adresse initiale d'un segment de la mémoire opérationnelle. Pour cette cause, dans le champ de l'instruction doit être pointé ce même registre de segment. Cet adressage est nommée encore **Adressage avec déplacement (Offset)** et chez lui $ChA = \delta \pm 2^k$.

On a déjà remarqué que dans les différents processeurs centraux modernes (CPU) sont disposées une diversité nombreuse de méthodes d'adressage. Le choix d'une multitude concrète dépend aussi de l'organisation de la mémoire opérationnelle. Il existe deux moyens d'organisation de la mémoire opérationnelle :

- a) Partition pageable ;
- b) Partition segmentée.