

Gestion de la mémoire

ARO

2 - Instructions

Definition

TBD

Table des matières

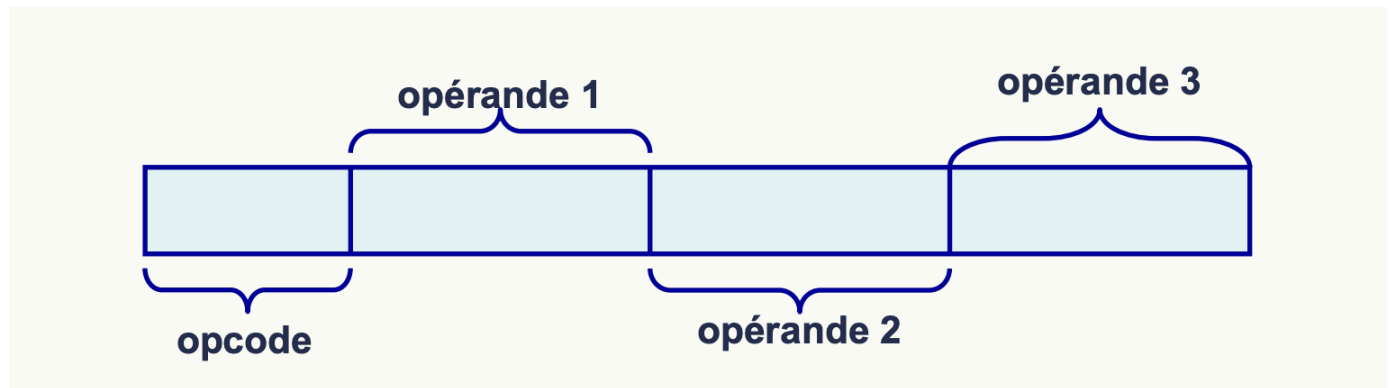
- 1. Types d'instruction 2
- 2. Format d'une instruction 2
 - 2.1. Opcode 2
 - 2.2. Opérande(s) 2
 - 2.3. Exemple(s) 2
 - 2.3.1. 1 opérande 2
 - 2.3.2. 2 opérandes 2
 - 2.3.3. 3 opérandes 2
- 3. Execution d'une instruction 3
- 4. Modes d'adressage 3
 - 4.1. Adressage immédiat 3

1. Types d'instruction

Il existe 3 types d'instructions que nous allons voir.

1. instructions de calcul/traitement → (arithmétique et logique)
2. instructions de transfert de donnée
 - interne → interne
 - interne → externe
3. instructions de branchement (saut)

2. Format d'une instruction



2.1. Opcode

Le champ «opcode» est l'identificateur de l'instruction et permet donc de définir ce que doit celle-ci va devoir faire.

2.2. Opérande(s)

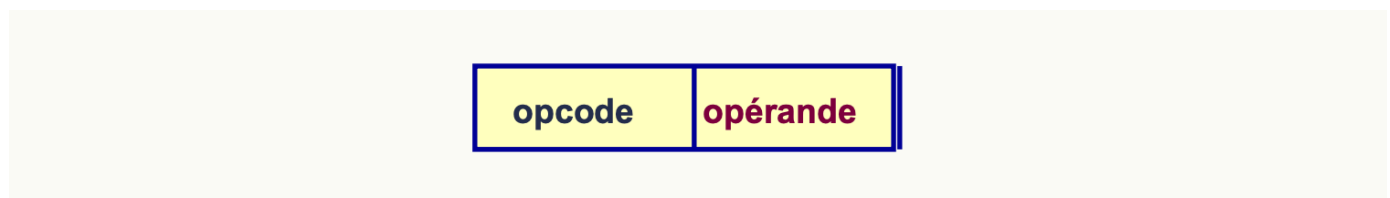
- Les champs opérandes spécifient la destination et les deux opérandes pour le traitement
 - Le nombre d'opérandes peut varier de 1 à 3.
 - Ci-dessous exemple avec le cas général

2.3. Exemple(s)

2.3.1. 1 opérande

INC r1

Incrément de r1 de 1.



2.3.2. 2 opérandes

ADD r1, r2

Addition de r1 et r2 puis stockage dans r1.



2.3.3. 3 opérandes

ADD r1, r2, r3

opcode	op��r��nde	op��r��nde	op��r��nde
--------	------------	------------	------------

3. Execution d'une instruction

Un processeur effectue sans arr  t une boucle compos  e de trois phases:

1. **FETCH** → recherche de l'instruction : dans un premier temps le processeur va chercher l'instruction    ex  cuter en r  cup  rant l'instruction point  e par le PC (**Program Counter**). Cette instruction sera stock  e dans un autre registre du processeur, le IR (**Instruction Register**).
2. **DECODE** → d  codage de l'instruction : dans un deuxi  me temps, chaque instruction est identifi  e, gr  ce    un code (**opcode**). En fonction de ce code, le processeur choisit la t  che    ex  cuter, c'est-  dire la s  quence de micro-instructions    ex  cuter.
3. **EXECUTE** → ex  cution de l'instruction : finalement on ex  cute l'instruction puis revenons    la premi  re.

Deux   tapes suppl  mentaires seront vue dans la suite du cours qui permettront de compl  ter ce que fait un processeur, soit **MEMORY** et **WRITE BACK**.

4. Modes d'adressage

Le mode d'adressage correspond    la m  thode d'acc  s aux donn  es. En fonction de l'instruction    ex  cuter, la donn  e    utiliser peut  tre diff  rente. Nous pouvons imaginer quelques cas:

- Valeur imm  diat: op  r  nde = donn  e
- Valeur dans un registre: op  r  nde = no registre
- Valeur en m  moire: op  r  nde = adresse
 - nombreuses variantes pour ce dernier cas

4.1. Adressage imm  diat