CHAPITRE N°8 : Jeu d'instructions

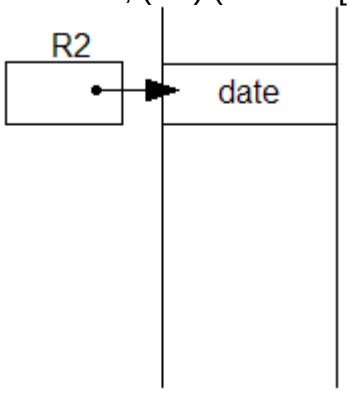
PLAN :

I) CISC (complex instruction set computer)

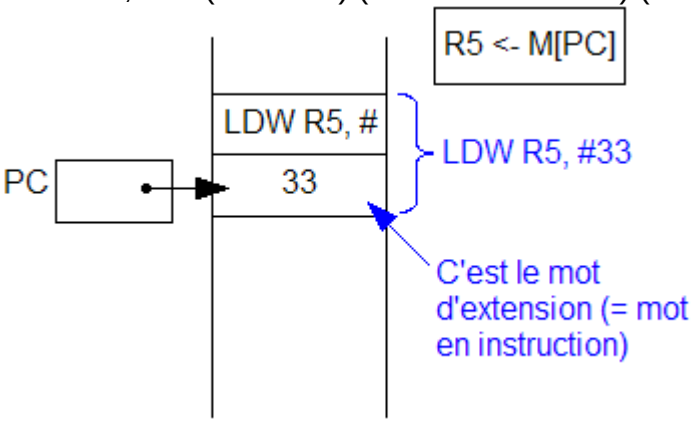
IBM 360 : première machine à utiliser la micro-programmation.

Ex : ADD R1, R3, R5 (mode registre).

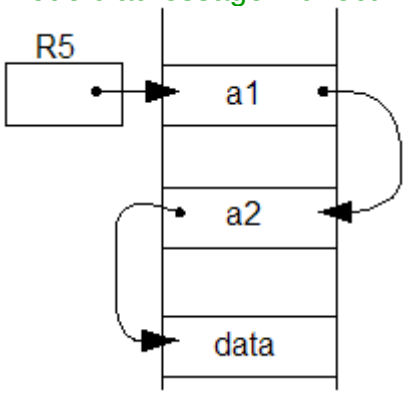
structure très simple
problème pour la mémoire :
LDW R0, R2 (mettre la valeur R2 dans R0) (mode registre)
LDW R0, (R2) (R0 <- M[R2]) (mode basé) (le contenu de R2 pointe dans R0) :



Pour mettre des constantes :
LDW R5, #33 (R5 <- 33) (33 est en décimal) (mode immédiat) :



mode d'adressage indirect :



Ceci est équivalent à :
LDW R4, (R5)
LDW R4, (R4)
LDW R0, (R4)

II) RISC (reduced instruction set computer)

- on a réduit la complexité du jeu d'instructions => moins d'instructions
=> le processeur travaille plus vite
- les instructions sont plus simples
- moins de modes d'adressages
- jeu orthogonal
  transfert : LOAD (de la mémoire au registre) ou STORE (du registre à la mémoire)
  opérations : ADD, SUB, NOT, ... (un seul mode d'adressages : le mode registre)
=> simplification de la machine !!

voir feuille carte de programmation

pour charger en prenant en compte les types :
LDW (pour un mot), LDB (pour un bit), etc

mode direct : l'adresse de l'opérande est dans le mot d'extension (symbole du mode direct : @)

> *Ex : JEA instruction (ex : @0xFF00)*
> *l'adresse : FF00*
> *on va faire le saut sur l'adresse FF00.*

III) Instruction d'opérandes linéaires

1) Logique booléenne

NOT R1, R2 // $\overline{R1}$ -> R2
notation : $\overline{R1}$: /R1

2) Décalages

1001
0100 = 1001 >> 1
O100 = 1001 / 2
on voudrait sauvegarder le bit qui sort : on le stocke dans CF (mémoire)
il faut prévoir 2 décalages (un pour les entiers naturels, et un autre pour les entiers relatifs)
(cette modification portera sur la division par 2)

modification : 1001 -> 1100 et CF=1 //on reporte le premier chiffre aussi en première position

en résumé :
décalage à droite normal (logique)
SRL shift right logical
1001 -> 0100 et CF=1
décalage à droite arithmétique
SRA shift right arithmetic
1001 -> 1100 et CF=1
autre décalage :
RRC : rotate right through carry
on fait un décalage à droite, et on met la valeur de CF au début du mot.

décalages à gauche :
pas de problème avec les entiers relatifs et naturels
opération : on multiplie par 2.
commandes :
SHL
SLL
SLA
RLC (le contraire de tout à l'heure).

---

3) Arithmétique

NEC R1, R2 :
R1 # 1 -> R2.

comparaison : CMP R1,R2 :
R1 – R2 (on ne s'occupe pas de stocker le résultat)
elle modifie des indicateurs MF, ZF, VF, CF
et après, avec une fonction teste, on pourra voir le résultat de cette fonction utilisée par exemple avec : BEQ +32.

SWB (la permutation)

ex : AB EF <- EF AB.

input et output : pour les entrées/sorties. On ne les utilise plus.

ex : OUT R1, @32
renvoie la valeur de R1 dans le bit d'adresse 32.

Retour

CHARDON Marion
Webmestre