Leçon assembleur

Langage assembleur 🡪 langage binaire (le plus faible langage)

Syntaxes : Intel ou AT&T

Utilité : Micro-programmes performants

Instructions possibles :

Manipulations en mémoire

Calculs arithmétiques

Comparaisons

Sauts, appels

|  |  |  |
| --- | --- | --- |
| **Instruction** | **Opération** | **Type d’opérandes** |
| **------** | **Déclarations** | **------** |
| **DS** | Avec initialisation |  |
| **DSW** | Sans initialisation | Nom facultatif (réservation de place) |
| **------** | **Manipulation mémoire** | **------** |
| **LD** dest, source | dest <- source |  |
| **ST** source, var | var <- source |  |
| **LEA** dest, var | dest <- adresse de var |  |
| **SWAP** oper | Echange les 16 bits forts et les 16 bits faibles de oper |  |
| **------** | **Arithmétique** | **------** |
| **ADD** *op1, op2* | op1 <- op1 + op2 |  |
| **SUB** *op1, op2* | op1 <- op1 - op2 |  |
| **MUL** *op1, op2* | op1 <- op1 \* op2 | Entiers relatifs |
| **MULU** *op1, op2* | op1 <- op1 \* op2 | Entiers naturels |
| **DIV** *op1, op2* | op1 <- op1 / op2 | Entiers relatifs |
| **DIVU** *op1, op2* | op1 <- op1 / op2 | Entiers naturels |
| **INC** *oper* | op1 <- oper + 1 (incrémente) |  |
| **DEC** *oper* | op1 <- oper – 1 (décrémente) |  |
| **NEG** *oper* | op1 <- -(oper) (oppose) |  |
| **------** | **Logique** | **------** |
| **OR** *op1, op2* | op1 <- op1 OU op2 |  |
| **AND** *op1, op2* | op1 <- op1 ET op2 |  |
| **NOT** *oper* | ?? |  |
| **------** | **Comparaison** | **------** |
| **CMP** *op1, op2* | Positionne le registre d’état en fonction des op |  |
| **BEQ** *op1, op2* | op1 = op2 |  |
| **BNE** *op1, op2* | op1 != op2 |  |
| **BLTU / BLT …** | op1 < op2 |  |
| **BLEU / BLE …** | op1 <= op2 |  |
| **BGTU / BGT …** | op1 > op2 |  |
| **BGEU / BGE …** | op1 >= op2 |  |
| **BDEBU / BDEB …** | op1 ?????? op2 |  |
| **------** | **Décalage** | **------** |
| **SHR** *oper* | logique à droite |  |
| **SHL** *oper* | Logique à gauche |  |
| **SAR** *oper* | arithmétique à droite |  |
| **SAL** *oper* | Arithmétique à gauche |  |
| **ROR** *oper* | Cyclique à droite |  |
| **ROL** *oper* | Cyclique à gauche |  |
| **------** | **Branchements** | **------** |
| **JMP** *etiq* | Inconditionnel |  |
| **Bxx** *etiq* | Conditionnel |  |
| **------** | **Pour les piles** | **------** |
|  |  |  |
|  |  |  |
| **------** |  | **------** |
|  |  |  |
|  |  |  |
|  |  |  |
| **------** |  | **------** |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
| **------** |  | **------** |
|  |  |  |