R1.03 - Architecture des ordinateurs

TP3 – Instruction Machine

**4) c)**

**A : 6000e8**

**B : 6000e9**

**C : 6000ea**

**D : 6000eb**

**E : 6000ec**

**F : 6000f0**

**e)**

**Avant Execution Programme : RIP : 0x4000b0**

**Rax : 0x2 RIP : 0x4000b7**

**RBX : 0x2 RIP : 0x4000be**

**Rax : 0x4 RIP : 0x4000c0**

**Rax : 0xe RIP : 0x4000c7**

**Aucune Var RIP : 0x4000ce**

**Rax : 0x7 RIP : 0x4000d0**

**Rax : 0x6 / eFalgs 0x206 RIP : 0x4000d7**

**Aucune Var RIP : 0x4000de**

**Rax : 0x1 RIP : 0x40000e3**

**5)**

**x/7bx 0x4000b0 = 8a 04 25 e8 00 60 00**

**x/7bx 0x4000b7 = 8a 1c 25 e9 00 60 00**

**x/2bx 0x4000be = f6 e3**

**x/7bx 0x4000c0 = 02 04 25 ea 00 60 00**

**x/7bx 0x4000c7 = 8a 1c 25 eb 00 60 00**

**x/2bx 0x4000ce = f6 f3**

**x/7bx 0x4000d0 = 2a 04 25 ec 00 60 00**

**x/7bx 0x4000d7 = 88 04 25 f0 00 60 00**

**x/5bx 0x4000de = b8 01 00 00 00 00**

**6)**

**1er Etape : A (6000e8) -> AL**

**2eme Etape : B (6000e9) -> BL**

**3eme Etape : f6 f3 = AL\*BL -> AL donc AxB -> AL**

**4ème Etape : C (6000ea) + AL (B\*A) -> AL**

**5ème Etape : D (6000eb) -> BL**

**6ème Etape : AL/BL -> AL donc (C+B\*A)/D -> AL**

**7ème Etape : AL-E (6000ec) -> AL donc ((c+b\*a)/d)-e)**

**8ème Etape : AL -> F donc ((c+b\*a)/d)-e) -> F**

**F contient cette formule : ((c+b\*a)/d)-e)**

**7)**

**Le R/M byte (Register/Memory Byte) et le SIB byte (Scaled Index Base) sont des composants importants des instructions x86/x86-64. Ils sont utilisés pour indiquer quels registres ou adresses mémoire doivent être utilisés. Voici ce que signifient ces termes :**

* **R/M byte (Register/Memory Byte) : Le R/M byte est un champ d'octets dans une instruction x86/x86-64 qui indique comment l'opérande source ou l'opérande de destination doit être interprété.**
* **SIB byte (Scaled Index Base) : Le SIB byte est un champ d'octets supplémentaire qui est utilisé en conjonction avec le R/M byte pour calculer l'adresse effective d'une opération mémoire.**

**Ces champs R/M et SIB sont utilisés pour déterminer comment les opérations doivent être effectuées et quelles adresses mémoire sont utilisées, ce qui est essentiel pour comprendre le fonctionnement des instructions x86/x86-64.**

**8)**

**Les architectures RISC (Reduced Instruction Set Computer) et CISC (Complex Instruction Set Computer) sont types de processeurs. Voici les principales différences entre elles :**

**Architecture RISC (Reduced Instruction Set Computer) :**

1. **Nombre d'instructions : Les processeurs RISC ont un ensemble d'instructions relativement réduit**
2. **Longueur fixe des instructions : Les instructions RISC ont généralement une longueur fixe**
3. **Chargement et stockage : Les instructions RISC favorisent le chargement et le stockage des données entre la mémoire et les registres.**
4. **Simplicité : Les instuctions RISC sont conçues pour être simples et s'exécuter rapidement. Cela signifie que les opérations complexes nécessitent souvent plusieurs instructions.**

**Architecture CISC (Complex Instruction Set Computer) :**

1. **Nombre d'instructions : Une seule instruction CISC peut effectuer des opérations complexes, telles que la multiplication, la division, et même des opérations de chargement/stocage en mémoire.**
2. **Longueur variable des instructions : Les instructions CISC ont souvent des longueurs variables, ce qui rend le décodage plus complexe.**
3. **Chargement et stockage : Les instructions CISC permettent souvent d'accéder à la mémoire directement pour effectuer des opérations sur les données.**

**La plupart des processeurs Intel sont généralement considérés comme des processeurs CISC. Les processeurs AMD sont aussi basées sur l’arcitecture CISC Haut du formulaire**