
STRUCTURE DES ORDINATEURS

EXAMEN

2 heures
avec documents de cours

N.B. : - Les réponses aux questions doivent être argumentées et aussi concises que possible.
- Le barème est donné à titre indicatif.

Question 1 (3 points)

Expliquez la différence entre traduction et interprétation.

Question 2 (6 points)

On s'intéresse dans cet exercice au câblage de fonctions logiques.

(2.1) (1 point)

Montrez comment on peut construire une porte ET au moyen de deux portes NON-ET.

(2.2) (3 points)

Dessinez le schéma logique d'un encodeur à deux bits, qui offre 4 entrées et deux sorties. L'une seulement des 4 entrées est à l'état 1 à un instant donné, et toutes les autres sont à 0. Les valeurs prises par les deux bits de sortie précisent de façon binaire quelle est l'entrée qui est à l'état 1.

(2.3) (2 points)

Dessinez le schéma logique d'un circuit filtrant, à mettre en amont de l'encodeur, afin que lorsque plusieurs des 4 entrées sont à 1, l'encodeur fournisse toujours comme numéro binaire de sortie le plus petit des numéros des entrées à 1.

Question 3 (6 points)

On s'intéresse dans cet exercice à comparer les codes machine à 0, 1, 2 ou 3 adresses. Les instructions disponibles sur chaque machine sont :

- 0 adresses : « PUSH M », « POP M », « ADD », « SUB », « MUL », « DIV » ;
- 1 adresse : « LOAD M », « STORE M », « ADD M », « SUB M », « MUL M », « DIV M » ;
- 2 adresses : « MOV X,Y » ($X \leftarrow Y$), « ADD X,Y » ($X \leftarrow X + Y$), « SUB X,Y » ($X \leftarrow X - Y$), « MUL X,Y » ($X \leftarrow X \times Y$), « DIV X,Y » ($X \leftarrow X/Y$) ;
- 3 adresses : « LOAD S,M », « STORE S,M », « MOV S,T » ($S \leftarrow T$), « ADD S,T,U » ($S \leftarrow T + U$), « SUB S,T,U » ($S \leftarrow T - U$), « MUL S,T,U » ($S \leftarrow T \times U$), « DIV S,T,U » ($S \leftarrow T/U$),

où M est une adresse mémoire sur 16 bits, X et Y spécifient indifféremment soit une adresse mémoire sur 16 bits, soit un numéro de registre sur 4 bits, et S , T et U spécifient uniquement un numéro de registre sur 4 bits. Les machines à 2 et 3 adresses disposent chacune de 16 registres nommés de $R0$ à $R15$. On suppose que les codes opération de toutes les instructions sont sur 8 bits, quelles que soient les tailles des opérands qui les suivent, et qui sont multiples de 4 bits.

Écrivez, pour chacune des quatre architectures décrites ci-dessus, le code machine correspondant au fragment de programme :

$$X = (A + B \times C) / (D - E \times F) ,$$

où A , B , C , D , E , F et X sont des données stockées en mémoire. Donnez pour chacun la taille en demi-octets (c'est-à-dire en multiples de 4 bits) du code écrit, et le nombre d'instructions nécessaires.

Question 4

(5 points)

Une constatation courante en informatique est que 20% du code est responsable de 80% du temps d'exécution (règle dite « *des 20-80* »).

(4.1) (1 point)

À quoi ceci est-il dû ?

(4.2) (2 points)

Quels mécanismes de l'ordinateur sont-ils spécifiquement adaptés pour tirer avantage de ce constat ?

(4.3) (2 points)

Considérons un programme tel que 1% du code soit responsable de plus de 50% du temps d'exécution. Sachant que l'écriture en langage C de ce programme prendrait 10 mois-homme, et que le codage en langage assembleur prend dix fois plus de temps que celui en langage C mais donne des routines quatre fois plus rapides, comparez les trois stratégies suivantes en termes de temps de codage et d'exécution :

- i. On écrit entièrement le programme en langage C ;
- ii. On écrit entièrement le programme en assembleur ;
- iii. On écrit entièrement le programme en langage C, puis on recode ensuite 1% du programme en assembleur.