

Examen – R1.03 : architecture Codage/décodage et assembleur

Durée : 2 h
Ecrivez au stylo

Documents interdits
Machine à calculer interdite

Le sujet comporte 6 exercices. Un barème indicatif est mentionné pour chaque exercice.

1. Programmation en assembleur 68000 - tantque (3 points)

Les indications sur les instructions de comparaison et de branchement sont fournies en annexe.

L'algorithme suivant est défini pour A,B,C entiers naturels sur 16 bits.

```
tantque   (40 < C)
          ou (B < A)
faire
    {B := B + 1 ;
    C := C - 5 ; }
fintantque
```

1.1. Recopiez l'algorithme sur votre copie et, pour chaque inégalité, faites apparaître la rupture de séquence associée (c'est-à-dire la condition de la rupture de séquence et sa cible).

1.2. Sur la base de votre analyse précédente, traduisez l'algorithme en assembleur 68000.

Indications :

- Traduisez uniquement la partie instructions (ne déclarez pas les données).
- Vous devez bien sûr conserver l'ordre des instructions de l'algorithme.
- **Il est impératif de commenter vos instructions assembleur**, en particulier pour les comparaisons et les branchements.

2. Programmation en assembleur 68000 – si alors sinon (5 points)

Les indications sur les instructions de comparaison et de branchement sont fournies en annexe.

L'algorithme suivant est défini pour A,B,C,X,Y entiers relatifs sur 16 bits.

```
si      ( B < A )
        et ( C > 40 )
ou
        ( X > C )
ou
        ( 20 ≥ Y+C )
alors
    { C := C - A ; }
sinon
    { Y := Y + X ; }
```

Comme l'instruction CMP.B/W/L doit avoir un registre de données en destination, **vous utiliserez le registre D0** : pour recopier la variable A dans l'évaluation de la 1^{ère} condition, puis recopier C dans l'évaluation de la 3^{ème} condition, puis calculer Y+C dans l'évaluation de la 4^{ème} condition.

Mêmes questions et recommandations que pour l'algorithme de l'exercice 1 :

2.1. Recopiez l'algorithme et, pour chaque inégalité, faites apparaître la rupture de séquence associée. NB. Pour plus de lisibilité, comme cela a été vu en cours, vous pouvez numéroter chaque condition.

2.2. Sur la base de votre analyse précédente, traduisez l'algorithme en assembleur 68000.

3. Désassemblage d'un programme (2 points)

Les indications sur les instructions de comparaison et de branchement sont fournies en annexe.

L1	MOVE.L Var2,D3
L2	ADD.L #5,D3
L3	CMP.L Var1,D3
L4	BCS L7
L5	SUBI.L #3,Var2
L6	BRA L1
L7	NOP

3.1. Ecrivez le commentaire associé à chaque instruction.

3.2. Donnez l'algorithme (dont la structure de contrôle associée) correspondant à ce programme.

4. Entiers naturels - codage en base deux sur 8 bits (3 points)

- Quelle est la dynamique de ce codage (intervalle des valeurs qui peuvent être codées) ?
- Décoder (en base dix) $\langle \$B9 \rangle_{\text{deux}}$ et $\langle \$85 \rangle_{\text{deux}}$
- Codez en base deux $\langle 103 \rangle_{\text{dix}}$ et $\langle 56 \rangle_{\text{dix}}$
- Lors d'une addition ou d'une soustraction, quel indicateur est utile dans le registre d'état ? Quel problème peut se produire ? Comment est-il détecté ? Que vaut alors l'indicateur ?
- Effectuez l'opération suivante (en notant clairement les retenues et en précisant la valeur de l'indicateur utile) : $\langle \$B9 \rangle_{\text{deux}} + \langle \$85 \rangle_{\text{deux}}$
Après l'avoir effectuée, analysez le calcul et donnez son équivalence en base dix.

5. Entiers relatifs - codage en complément à deux sur 8 bits (3 points)

- Quelle est la dynamique de ce codage ? Quels sont les codages des deux bornes de cet intervalle ?
- Décoder (en base dix) $\langle \$19 \rangle_{\text{cpt2}}$; $\langle \$B9 \rangle_{\text{cpt2}}$ et $\langle \$F4 \rangle_{\text{cpt2}}$
- Codez en complément à deux $\langle 120 \rangle_{\text{dix}}$; $\langle -102 \rangle_{\text{dix}}$ et $\langle -1 \rangle_{\text{dix}}$
- Lors d'une addition ou d'une soustraction, quel indicateur est utile dans le registre d'état ? Quel problème peut se produire ? Comment est-il détecté lors d'une addition ? Que vaut alors l'indicateur ?
- Effectuez l'opération suivante (en notant clairement les emprunts et en précisant la valeur de l'indicateur utile) : $\langle \$B9 \rangle_{\text{deux}} - \langle \$F4 \rangle_{\text{deux}}$
Après l'avoir effectuée, analysez le calcul et donnez son équivalence en base dix.

6. Réels - virgule fixe – complément à deux – sur 8 bits (4 points)

On considère un codage en virgule fixe en complément à deux sur 8 bits : 5 bits pour la partie entière et 3 bits pour la partie fractionnaire.

Les codages en notation hexadécimale sont donc notés sous la forme : $\langle \$hh \rangle_{\text{virg.fixe}(5,3)\text{-cpt2}}$

- Combien de nombres réels distincts peuvent être codés ?

b. Quel est le plus petit nombre positif non nul qui peut être codé (donnez son codage et sa valeur en base dix) ? A quoi correspond-il pour ce codage virg.fixe(5,3)-cpt2 ?

c. Quel est le plus grand nombre positif qui peut être codé : quel est son codage et quelle est sa valeur en base dix ?

d. Quel est le nombre négatif, de plus grande valeur absolue, qui peut être codé et quel est son codage ?

e. Décodez (en base dix) :

$\langle \$4C \rangle_{\text{virg.fixe}(5,3)\text{-cpt2}}$; $\langle \$E6 \rangle_{\text{virg.fixe}(5,3)\text{-cpt2}}$; $\langle \$8D \rangle_{\text{virg.fixe}(5,3)\text{-cpt2}}$

f. Codez en virg.fixe(5,3)-cpt.2 : $\langle -9,625 \rangle_{\text{dix}}$; $\langle 13,42 \rangle_{\text{dix}}$ et ; $\langle -13,42 \rangle_{\text{dix}}$

Finalisez ces codages en notation hexadécimale.

Indiquez, le cas échéant, s'il y a une erreur de troncature et quelle est sa valeur.

g. Quel indicateur est significatif de la validité des opérations d'addition ou de soustraction ?

h. Effectuez la soustraction (en notant clairement les emprunts et en précisant la valeur de

l'indicateur utile) : $\langle \$1C \rangle_{\text{virg.fixe}(5,3)\text{-cpt2}}$ - $\langle \$ED \rangle_{\text{virg.fixe}(5,3)\text{-cpt2}}$

Analysez le calcul et donnez son équivalent en base dix.

FIN

Annexe – comparaisons et branchements – assembleur 68000

Instructions de comparaison

CMP/CMPI.B/W/L source,destination ; compare (destination) ? (source)

Deux instructions possibles : **CMP**, **CMPI**.

♦ **CMP.taille** (comparaison) *Taille: .B, .W, .L*

CMP.B/W/L <AE>,Dn ; (destination) - (source)

La destination est inchangée.

♦ **CMPI.taille** (comparaison immédiate) *Taille: .B, .W, .L*

CMPI.B/W/L #<donnée>,<AE> ; (destination) - (source)

La destination est inchangée.

Conditions de branchement

b _{11..b₈}	B..	appellation	interprétation (dest) ? (source)	contexte d'utilisation	test sur les indicateurs
0111	BEQ	EQual	égal	quelconque	Z = 1
0110	BNE	Not Equal	différent	quelconque	Z = 0
0100	BCC	Carry Clear	≥ ; non dépassement	base deux	C = 0
0101	BCS	Carry Set	< ; dépassement	base deux	C = 1
0010	BHI	High	>	base deux	C + Z = 0
0011	BLS	Less or Same	≤	base deux	C + Z = 1
1100	BGE	Greater or Equal	≥	complément à 2	N ⊗ V = 1
1101	BLT	Less Than	<	complément à 2	N ⊕ V = 1
1110	BGT	Greater Than	>	complément à 2	Z + (N ⊕ V) = 0
1111	BLE	Less or Equal	≤	complément à 2	Z + (N ⊕ V) = 1
1011	BMI	MInus	négatif	cpt à 2 ; ±	N = 1
1010	BPL	PLus	positif	cpt à 2 ; ±	N = 0
1000	BVC	oVerflow Clear	pas de débordement	complément à 2	V = 0
1001	BVS	oVerflow Set	débordement	complément à 2	V = 1
0000	BRA	True	vrai	quelconque	1
0001	F	False	faux	-	0

• Utilisez les listes ci-dessous pour choisir la condition, sur la base de la comparaison préalable entre les deux opérandes par une instruction **CMP** :

CMP.B/W/L source,destination ; compare (destination) ? (source)

B.. ad_branchement ; brcht pris si cond. vraie

AAAAA ; instruction exécutée si cond. fausse

Pour des entiers naturels (codage en base deux)

BCC branchement si (destination) ≥ (source) *Carry Clear*
BCS branchement si (destination) < (source) *Carry Set*
BHI branchement si (destination) > (source) *High*
BLS branchement si (destination) ≤ (source) *Less or Same*

Pour des entiers relatifs (codage en complément à deux)

BGE branchement si (destination) ≥ (source) *Greater or Equal*
BLT branchement si (destination) < (source) *Less Than*
BGT branchement si (destination) > (source) *Greater Than*
BLE branchement si (destination) ≤ (source) *Less or Equal*