GIF-1001, 28 février 2012 Nom/Matricule : _____Solution_____

Examen 1

Cet examen vaut 40% de la note totale du cours. Les questions seront corrigées sur un total de 40 points. La valeur de chaque question est indiquée avec la question. Une calculatrice scientifique peut être utilisée. Cependant, aucune documentation, autre que l'annexe, n'est permise. Vous pouvez répondre aux questions directement sur ce questionnaire et/ou dans le cahier bleu mis à votre disposition.

Q1 (2 points) : Quel mot ou concept relié aux ordinateurs correspond à la définition suivante :

Définition	Mot(s)
Programme permettant de charger en mémoire et d'exécuter d'autres programmes	*Système d'exploitation*
Programme en mémoire non-volatile exécuté au démarrage de l'ordinateur	*BIOS*
Instruction habituellement utilisée pour accéder aux périphériques gérés par le système d'exploitation	*INT*
Petite mémoire gérée par le BIOS contenant les informations de base sur les périphériques de votre ordinateur	*RAM CMOS*

Q2 (2 points) : Effectuez les opérations mathématiques suivantes et indiquez les drapeaux qui vaudront "1" après l'opération [Zéro, Signe, Overflow (Débordement) et Carry/Borrow (Retenue/Emprunt)]. Tous les nombres sont sur 8 bits.

A	B	C	D
0x69 + 0x92	0x64 + 0x21	0x01 - 0xFF	0x80 - 0x01
Résultat : 0xFB	Résultat : 0x85	Résultat : 0	Résultat : 0x7F
Drapeaux : S	Drapeaux : S, V	Drapeaux : Z, C	Drapeaux : V

Q3 (4 points) : Expliquez pourquoi un programme qui ne respecte pas le principe de localité sera plus long à exécuter sur votre ordinateur qu'un programme avec des instructions et un but comparables, mais qui respecte le principe de localité.

Parce que le programme fera plus souvent des accès longs au disque dur ou à la mémoire. Le programme trouvera moins souvent l'information requise (instruction ou données) dans les caches très rapides de l'ordinateur.

Q4 (2 points) : L'unité de contrôle central de votre microprocesseur produit des mots de contrôle. Expliquez ce qu'est un mot de contrôle dans un contexte de microprocesseur.

Un mot de contrôle est un ensemble de bits/signaux qui contrôle les divers composants du microprocesseur afin d'exécuter une partie d'une instruction.

Q5 (4 points) : Dites combien d'accès à la mémoire (lecture ou écriture de la mémoire) au minimum sont nécessaires pour lire et exécuter les instructions suivantes :

A) MOV AX, BX
B) ADD [BX], AX
C) MOV AX, [0x0102]
D) INT 21H

A) 1 accès pour lire l'instruction
B) 1 accès pour lire l'instruction + 1 accès pour lire la mémoire à l'adresse BX
C) 2 accès pour lire l'instruction + 1 accès pour lire la mémoire à l'adresse 0x102
D) 1 accès pour lire l'instruction + 2 accès pour lire la table des vecteurs d'interruption + 3 accès pour sauvegarder l'adresse de retour et les drapeaux sur la pile

Q6 (2 points) : Parmi les énoncés suivants reliés aux accès des périphériques, encerclez uniquement les énoncés vrais :

- Le microprocesseur interroge automatiquement certains périphériques pour vérifier s'ils ont des évènements à signaler lorsque des registres du microprocesseur l'exigent. *FAUX*
- Lors d'une interruption, le microprocesseur exécute une routine dans sa ROM interne afin de traiter l'interruption. *Habituellement vrai, mais pas toujours dans la ROM*
- Le microprocesseur lira ou écrira un périphérique uniquement lorsqu'une instruction provenant de la mémoire lui dira de le faire. *VRAI sauf si le microprocesseur lit ses instructions d'un périphérique, ce qui n'arrive jamais.*
- Il y a un bus dédié aux échanges entre le microprocesseur et les périphériques pour accélérer les accès aux périphériques. La mémoire est reliée au microprocesseur avec un autre bus. *FAUX*
- Dans certains ordinateurs, un périphérique aura l'adresse 0 et un mot de mémoire aura aussi l'adresse 0. Dans ces systèmes, une instruction permettra de lire la mémoire et une autre instruction permettra de lire un périphérique. *VRAI*
- Dans la plupart des ordinateurs cependant, les périphériques et la mémoire partageront un espace d'adresse commun : les périphériques sont traités comme la mémoire. *VRAI*

Q7 (7 points) : Le programme suivant calcule la factorielle d'un nombre avec une fonction récursive (Calcule N!: retourne 1 si N <= 1, sinon retourne N*(N-1)!).

```
Facto PROC
        PUSH BP
        PUSH AX
        MOV BP, SP
        MOV AX, [BP + 6]        ;Pour la partie b de la question, la valeur 6 est correcte!!!
        CMP AX, 1
        JG      PlusGrandQueUn
        MOV [BP+6], 1
        POP AX
        POP BP
        RET
```

```
PlusGrandQueUn:
        DEC AX
        PUSH AX
        CALL    Facto
        POP AX
        IMUL [BP + 6]           ;AX = AX * [BP + 6]
        MOV [BP + 6], AX
        POP AX
        POP BP
        RET
Facto ENDP
```

a) Écrivez les instructions nécessaires pour appeler cette fonction
   a. *PUSH N*
   b. *CALL Facto*
   c. *POP Resultat*
b) Cette fonction fonctionne-t-elle? Si elle ne fonctionne pas, comment la corriger?
   a. *La fonction est correcte*
c) À l'aide d'un dessin, illustrez l'évolution de la pile lorsqu'on calcule 3! En utilisant cette fonction.
   a. *La fonction s'appelle elle-même et empile des adresses de retour à toutes les fois, ainsi que BP, AX et N-1!. Ensuite tout est dépilé dans l'ordre inverse d'empilement.*

Q8 (5 points) : Un compilateur traduit le programme C suivant en instructions/directives assembleur 8086 :

Programme en C	**Instruction correspondante en assembleur 8086**
short Var1;	Var1 DW 0
short Var2;	Var2 DW 0
short* Var3;	Var3 DW 0

Un peu plus loin, dans le même programme, le compilateur veut traduire une autre instruction/directive C en une ou plusieurs instructions/directives assembleur 8086. Quelles seront les instructions générées par l'assembleur et l'éditeur de liens:

Programme en C	**Instruction(s) correspondante(s) en assembleur 8086**
Var2 = Var1 – (*Var3) + 4;	*MOV AX, Var1*
	MOV BX, Var3
	MOV CX, [BX]
	SUB AX, CX
	ADD AX, #4
	MOV Var2, AX

Considérez les éléments suivants afin de répondre:

- Short est un entier sur deux octets. La directive DW réserve deux octets de mémoire.
- Les adresses initiales de Var1, Var2 et Var3 sont 0x2000, 0x2002 et 0x2004 respectivement.
- Vous pouvez utiliser les registres que vous voulez.
- Var3 est un pointeur. *Var3 signifie ce qu'il y a à l'adresse indiquée par la valeur de Var3.
- Le 8086 ne permet pas de référer à deux adresses de mémoire différentes à l'intérieur d'une instruction.

Q9 (4 points) : Le contrôleur d'un disque dur utilise un algorithme pour prédire quel sera le prochain bloc de données du disque dur qui sera lu : dans 90% du temps, la tête de lecture du disque est placée juste avant le bloc de donnée à lire et l'accès disque est beaucoup plus court (il suffit de lire le bloc seulement). Lorsque l'algorithme de prédiction se trompe (10% des lectures de bloc), il faut, en moyenne 7ms pour déplacer la tête de lecture sur la piste contenant le bloc de donnée à lire.

Sachant que le disque dur tourne à 6000 tours par minutes (RPM) et qu'il a 20 secteurs par surface, quel est le temps moyen pour lire un bloc de donnée sur ce disque?

TTBP = Temps d'accès moyen quand la tête est bien placée = temps de lecture du bloc.
TTBP = Temps pour faire 1/20 de tour = 1/20 de 60/6000 s = 0.5ms

TTMP = Temps d'accès moyen quand la tête est mal placée = temps moyen de déplacement de la tête + temps moyen de rotation pour atteindre le bloc de données + temps de lecture du bloc.
TTMP = 7ms + temps de ½ tour + 0.5ms = 7ms + 5ms + 0.5ms

*Temps d'accès moyen = 0.9*TTBP + 0.1TTMP*

Q10 (6 points) : Tous les énoncés qui suivent sont faux pour au moins deux raisons. Dites pourquoi ces énoncés sont faux et dites comment les rendre vrais.

A) Lors d'un transfert par DMA, le contrôleur de DMA prend le contrôle du bus système et désactive le microprocesseur. Ainsi, il peut transférer des données efficacement entre un périphérique et la mémoire ou entre deux mémoires.
   a. *DMA ne transfert pas de données entre deux mémoires*
   b. *DMA ne désactive pas le microprocesseur*

B) Lors de la compilation, le compilateur transforme du texte en instructions binaires. Il associe également chaque variable du programme à un registre. Ensuite, l'éditeur de lien place les instructions et les registres en mémoire
   a. *Les variables sont associées à des adresses en mémoire, la plupart du temps, plutôt qu'à des registres*
   b. *Les registres sont dans le microprocesseur, pas placés en mémoire.*

C) Selon la norme IEEE754, les fractions sont représentées sur 32 bits ou 64 bits. Sur 32 bits, la plus petite fraction qu'il est possible de représenter est -2^{31} et un bit détermine le signe de la fraction. Sur 64 bits, il est possible de représenter deux fois plus de fractions que sur 32 bits.
   a. *-2^{31} : la plus petite fraction est plus grande que cette valeur*
   b. *Sur 64 bits, on peut représenter 2^{32} fois plus de fractions*

D) On évite de brancher plusieurs mémoires à un bus parce que toutes les mémoires lisent ou écrivent le bus de données : une lecture simultanée des mémoires pourrait provoquer une collision. Il faut un bus par mémoire : dans l'architecture Harvard, il y a un bus pour la mémoire d'instruction et un bus pour la mémoire de données.
   a. *On peut brancher plusieurs mémoires à un bus : une seule est activée à la fois (décodeur d'adresse).*
   b. *Il n'y a pas un bus par mémoire.*

E) Une interruption survient uniquement lorsqu'un périphérique active une broche d'interruption du microprocesseur. Chaque périphérique possède sa broche d'interruption respective sur le microprocesseur.
   a. *Il y a plusieurs causes d'interruption (reset, NMI, exceptions, int logicielles).*
   b. *Les broches d'interruption des périphériques sont branchées sur le contrôleur d'interruption.*

F) Pour savoir si un énoncé conditionnel (if) est vrai ou faux, le 8086 évalue si un registre vaut 0. Si la condition évaluée est vraie, un saut est effectué vers l'endroit désigné par des bits contenus dans l'instruction. Si la condition évaluée est fausse, un saut est effectué à l'adresse indiquée par DI, le registre de Destination.
   a. *Le 8086 se sert des drapeaux de l'ALU pour évaluer les conditions.*
   b. *DI n'indique pas l'adresse de branchement.*

Q11 (2 points) : Expliquez ce qu'est une interruption vectorisée. Pourquoi parle-t-on de vecteur d'interruption?
Parce que l'adresse de la routine traitant l'interruption se retrouve dans une table. Au moment de l'interruption, le microprocesseur lit la table pour trouver le vecteur d'interruption, c'est-à-dire un pointeur sur l'ISR à exécuter.

QBonus (2 points) : Omer Saint-Amand a créé la fonction suivante qui calcule une moyenne :

```
Moyenne :
        PUSH AX
        PUSH BX
        ADD SP, 4
        POP AdrRet
        POP AX
        POP BX
        ADD     AX, BX
        SHR     AX, 1
        PUSH    AX
        PUSH AdrRet
        SUB     SP,6
        POP     BX
        POP AX
```

```
            ADD SP,2
            RET
AdrRet      DW 0
```

Bien que monsieur OSA soit très fier de sa fonction, cette fonction contient deux erreurs qui peuvent conduire à un échec d'exécution du programme. Quelles sont ces erreurs?

Annexe A : Liste non exhaustive des instructions du 8086

Instruction	Description
ADD a,b	Effectue a = a+b.
AND a,b	Effectue a = a ET b, où ET est un ET logique.
CALL proc	Appelle la procédure proc et empile l'emplacement de retour.
CLC	Met à 0 le drapeau de retenue (carry).
CMP a,b	Effectue a-b, a et b sont inchangés.
DEC a	Décrémente a.
DIV mot	Effectue AX = DXAX/mot, non signé, le résultat est tronqué (arrondi inférieur).
IDIV mot	Effectue AX = DXAX/mot, signé, le résultat est tronqué (arrondi inférieur).
IMUL mot	Effectue DXAX = AX*mot, signé.
IN dst, port	Met la valeur lue sur le port de I/O port dans dst.
INC a	Incrémente a.
INT a	Appelle la routine de service d'interruption a. Empile les drapeaux et l'emplacement de retour.
IRET	Retourne d'une int. en dépilant l'emplacement de retour et les drapeaux.
JC label	Saute à l'instruction désignée par label si le drapeau Carry est 1.
JMP label	Saute à label. La prochaine instruction exécutée est désignée par label.
JNC label	Saute à l'instruction désignée par label si le drapeau Carry est 0.
JNZ label	Saute à l'instruction désignée par label si le drapeau Zéro est 0.
JNS label	Saute à l'instruction désignée par label si le drapeau Signe est 0.
JZ label	Saute à l'instruction désignée par label si le drapeau Zéro est 1.
JS label	Saute à l'instruction désignée par label si le drapeau Signe est 1.
LEA dst,var	Met l'adresse de la variable var dans dst.
MOV dst,src	Met le contenu de src dans dst. Ne change pas les drapeaux.
MUL octet	Effectue AX = AL*octet, non signé.
NEG a	Inverse tous les bits de a, puis ajoute 1.
NOT a	Inverse tous les bits de a.
OR a,b	Effectue a = a OU b, où OU est un OU logique.
OUT port, src	Met la valeur de src sur le port de I/O port.
POP mot	Dépile un mot. Ne change pas les drapeaux.
POPF	Dépile les drapeaux.
PUSH mot	Empile un mot. Ne change pas les drapeaux.
PUSHF	Empile les drapeaux.
RET et RETF	Retourne d'une procédure en dépilant l'emplacement de retour. RET dépile IP. RETF dépile IP puis CS.
RCL a,b	Fait une rotation de b bits vers la gauche. La rotation inclut le bit de Carry.
RCR a,b	Fait une rotation de b bits vers la droite. La rotation inclut le bit de Carry.
SAL a,b	Décale tous les bits de a vers la gauche d'un nombre de bits égal à b. Des zéros sont mis à droite. Carry prend la valeur du bit disparu.
SAR a,b	Décale tous les bits de a vers la droite d'un nombre de bits égal à b. Le bit le plus significatif de a est mis à gauche. Carry prend la valeur du bit disparu.
STC	Met à 1 le drapeau de Carry
SUB a,b	Effectue a = a-b.
TEST a,b	Effectue a ET b, où ET est un ET logique.
XOR a,b	Effectue a = a XOR b, où XOR est un OU eXclusif.