

GIF-1001 Ordinateurs: Structure et Applications
Automne 2016
Examen mi-session
26 octobre 2016
Durée: 170 minutes

Cet examen comporte 8 questions sur 10 pages (incluant celle-ci), comptabilisées sur un total de 100 points. L'examen compte pour 40% de la note totale pour la session. Assurez-vous d'avoir toutes les pages. Les règles suivantes s'appliquent:

- Vous avez droit à une feuille aide-mémoire 8.5×11 recto-verso écrite à la main;
- Écrivez vos réponses dans le cahier de réponses qui vous a été remis;
- L'annexe A contient une liste d'instructions ARM;
- L'annexe B contient la table ASCII.

La table ci-dessous indique la distribution des points pour chaque question.

| | | | | | | | | | |
|-----------|---|----|----|----|----|----|----|----|-------|
| Question: | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Total |
| Points: | 6 | 23 | 11 | 12 | 12 | 21 | 15 | 10 | 110 |

Bonne chance!

1. Répondez aux questions suivantes sur les ordinateurs.
 - (a) (1 point) Quelle est la composante électronique qui a remplacé le tube à vide et qui a ouvert la porte aux microprocesseurs?
 - (b) (2 points) Selon l'architecture de von Neumann, quelles sont les quatre (4) composantes principales d'un processeur?
 - (c) (1 point) Quelles sont les trois (3) grandes opérations du cycle d'instructions?
 - (d) (2 points) Identifiez un périphérique qui est à la fois une entrée et une sortie pour l'ordinateur et expliquez pourquoi il s'agit d'une entrée et pourquoi il s'agit d'une sortie.

2. Répondez aux questions suivantes sur la représentation des données dans un ordinateur.
 - (a) (1 point) Combien de bits sont nécessaires pour stocker le jour du mois?
 - (b) (3 points) Convertissez les nombres décimaux suivants en hexadécimal sur 20 bits: 4096, 512 et 170.
 - (c) (2 points) Additionnez les deux nombres suivants qui sont donnés en représentation complément-2 sur 12 bits et donnez la réponse en hexadécimal et en décimal: 0xFF5 et 0xFE3.
 - (d) (1 point) Indiquez si l'addition de la question précédente a provoqué une retenue (*carry*) et indiquez si elle a provoqué un débordement (*overflow*).
 - (e) (1 point) Donnez deux nombres en hexadécimal sur 4 bits qui vont provoquer une retenue et un débordement lorsqu'on les additionne ensemble.
 - (f) (2 points) Lors d'une addition de deux nombres en complément-2, comment peut-on faire pour détecter un débordement? Donnez un exemple sur 5 bits.
 - (g) (2 points) Donnez le nombre de bits à 1, le nombre de bits à 0 et le nombre total de bits dans le nombre suivant: 0xF0E1D2C3B4A59687.
 - (h) (2 points) Quel nombre est représenté par 0x42280000 encodé sur 32 bits avec IEEE 754? Rappel: la norme IEEE 754 emploie la formule

$$(\text{signe})1, \text{mantisse} \times 2^{(\text{exposant}-127)},$$

et les bits sont stockés selon la figure 1.



Figure 1: Convention IEEE-754 sur 32 bits.

- (i) (2 points) Quel nombre est représenté par 0x7FC42000 encodée sur 32 bits avec IEEE 754?
- (j) (2 points) Quel nombre est représenté par 0xff800000 encodé sur 32 bits avec IEEE 754?
- (k) (3 points) Donnez la représentation IEEE 754 32 bits en hexadécimal de la fraction $\frac{7}{8}$.
- (l) (2 points) Comment pouvez-vous représenter en mémoire sur 16 bits le nombre de pizzas que vous aller manger durant vos études universitaires avec une précision d'exactly $\frac{1}{32}$ de pizza? Dites quel est le nombre maximal de pizzas que vous pourrez représenter avec votre système.

3. Pour les questions suivantes, considérez une mémoire pouvant stocker 128 octets (1 octet par adresse) dont une partie du contenu est illustré ci-dessous.

| Adresse | Valeur |
|-----------------|-----------------|
| ... | ... |
| 50 _h | 00 _h |
| 51 _h | 61 _h |
| 52 _h | 42 _h |
| 53 _h | 63 _h |
| 54 _h | 21 _h |
| 55 _h | 00 _h |
| 56 _h | 10 _h |
| 57 _h | 12 _h |
| 58 _h | 14 _h |
| 59 _h | 15 _h |
| ... | ... |
| 70 _h | 23 _h |
| 71 _h | 67 _h |
| 72 _h | 69 _h |
| 73 _h | 66 _h |
| ... | ... |

- (a) (1 point) Quelle est la largeur minimale requise du bus d'adresse pour accéder à toutes les adresses de cette mémoire?
- (b) (1 point) Quelle est la largeur du bus de données utilisé par cette mémoire?
- (c) (1 point) Donnez le nombre total de bits pouvant être stockés dans cette mémoire.
- (d) (2 points) Après avoir analysé un programme, vous avez déterminé qu'il met le mot de passe de l'utilisateur à l'adresse 81, et que le mot de passe a quatre (4) caractères. Quel est-il?
- (e) (2 points) En faisant l'hypothèse d'un système petit boutiste (*little endian*) et qu'un nombre non-signé de 32 bits est en mémoire à l'adresse 86, quel est ce nombre en hexadécimal?
- (f) (2 points) En faisant l'hypothèse d'un système gros boutiste (*big endian*) et qu'un nombre non-signé de 32 bits est en mémoire à l'adresse 84, quel est ce nombre en hexadécimal?
- (g) (2 points) Quel type d'information vu dans le cadre du cours (entier, rationnel, caractère, chaîne de caractères, ...) est stocké en mémoire aux adresses 70_h à 73_h inclusivement?

4. Un système de type “memory-mapped I/O” possède les caractéristiques suivantes:

- un bus d’adresse de 21 bits, avec les 2 bits les plus significatifs (MSB) utilisés pour le décodeur d’adresse;
- un bus de données de 16 bits;
- une mémoire RAM où chaque octet possède une adresse différente;
- trois autres périphériques sont branchés sur les bus;
- il stocke les données en mémoire avec la convention “big endian”;
- si on nomme les bits les plus significatifs (MSB) du bus d’adresse b_{20} et b_{19} , le décodeur sélectionne les périphériques de la façon suivante:

| b_{20} | b_{19} | Périphérique activé |
|----------|----------|---------------------|
| 0 | 0 | RAM |
| 0 | 1 | Périphérique 1 |
| 1 | 0 | Périphérique 2 |
| 1 | 1 | Périphérique 3 |

- (a) (2 points) Quelle est la taille maximale de la mémoire RAM? Écrivez votre réponse en kilo-octets (Ko).
- (b) (2 points) Quelle est la carte de la mémoire (*memory map*) de ce système?

Nous modifions le système afin qu’il puisse utiliser un bus de données de 32 bits au lieu de seulement 16. Le bus d’adresse ne change pas.

- (c) (2 points) Décrivez l’impact de ce changement sur la taille maximale de la mémoire RAM. Écrivez votre réponse en kilo-octets (Ko).
- (d) (2 points) Quelle est la carte de la mémoire (*memory map*) de ce système?

Finalement, nous remplaçons la mémoire RAM du système pour une mémoire qui attribue une adresse différente à chaque mot de 16 bits au lieu de seulement 8. Le système possède maintenant un bus de données de 32 bits, et cette nouvelle mémoire.

- (e) (2 points) Décrivez l’impact de ce changement sur la taille maximale de la mémoire RAM. Écrivez votre réponse en kilo-octets (Ko).
- (f) (2 points) Quelle est la carte de la mémoire (*memory map*) de ce système?

5. Répondez aux questions suivantes portant sur le micro-processeur du simulateur du travail pratique 1. Un rappel du jeu d'instructions est fourni:

Toutes les instructions du microprocesseur sont sur 16 bits et se décomposent comme suit:

- Bits 15 à 12: Opcode de l'instruction
- Bits 11 à 8: Registre utilisé comme premier paramètre.
- Bits 7 à 0: Registre ou constante utilisés comme deuxième paramètre

Le jeu d'instruction supporte les instructions suivantes où Rd est le registre destination, Rs le registre source et Rc le registre de condition:

| Mnémonique | Opcode | Description |
|--------------|--------|---|
| MOV Rd Rs | 0000 | Écriture de la valeur du registre Rs dans le registre Rd |
| MOV Rd Const | 0100 | Écriture d'une constante dans le registre Rd |
| ADD Rd Rs | 0001 | Addition des valeurs des registres Rd et Rs et insertion du résultat dans le registre Rd |
| ADD Rd Const | 0101 | Addition de la valeur du registre Rd avec une constante et insertion du résultat dans Rd |
| SUB Rd Rs | 0010 | Soustraction de la valeur Rs à l'intérieur de registre Rd. |
| SUB Rd Const | 0110 | Soustraction d'une constante à l'intérieur du registre Rd |
| LDR Rd [Rs] | 1000 | Chargement d'une valeur se trouvant à l'adresse Rs de l'ordinateur dans un registre. |
| STR Rd [Rs] | 1001 | Écriture de la valeur d'un registre à l'adresse Rs de l'ordinateur. |
| JZE Rc Const | 1111 | Saut à l'instruction située à l'adresse identifiée par la constante, mais seulement si Rc = 0 (sinon, cette instruction n'a aucun effet). |
| JZE Rc Rs | 1011 | Saut à l'instruction située à l'adresse Rs seulement si Rc = 0 (sinon, cette instruction n'a aucun effet). |

Table 1: Jeu d'instructions du microprocesseur

- (a) (5 points) Traduisez le programme suivant en binaire, et écrivez votre réponse en hexadécimal. Les numéros de ligne sont indiqués à gauche.

```

1 MOV R3 #32
2 LDR R2 [R3]
3 ADD R2 R2
4 ADD R2 #1
5 STR R2 [R3]
```

- (b) (7 points) Soit le programme suivant. Pour chaque ligne, on indique l'adresse (qui commence à 0x0), suivie de l'instruction en format binaire. Les numéros de ligne sont indiqués à gauche. Expliquez ce que fait ce programme en une phrase.

```

1 0x0 0x4000
2 0x1 0x8000
3 0x2 0x0100
4 0x3 0x4200
5 0x4 0x6101
6 0x5 0x1200
7 0x6 0xF109
8 0x7 0x4300
9 0x8 0xF304
10 0x9 0x9201
```

6. Répondez aux questions suivantes, portant sur l'assembleur ARM.

- (a) (1 point) Quel est le programme qui permet de créer un programme qui contient à la fois de l'assembleur et du code d'un autre langage?
- (b) (1 point) Quelle est la particularité des instructions de type SIMD?
- (c) (1 point) Auquel des deux grands types de jeu d'instructions le jeu d'instructions ARM appartient-il?
- (d) (3 points) Quels sont les registres spéciaux utilisés dans l'architecture ARM et quelles sont leurs fonctions?
- (e) (5 points) Considérez le code suivant (les numéros de ligne sont indiqués à gauche):

```
1  MOV SP, #0x3
2  MOV LR, PC
3  test
4  BX LR
5  BL test
6  CMP LR, PC
7  BLT test
8  MOV R0, #0x1
9  B test
10 fin
11 B fin
```

Indiquez l'ordre des instructions exécutées par le microprocesseur en utilisant leur numéro de ligne correspondant.

- (f) (5 points) Écrivez du code assembleur qui place dans R0 la valeur 1 si R1 est pair ou -1 si R1 est impair.
- (g) (5 points) Écrivez du code assembleur qui calcule la factorielle d'un nombre placé dans R0. En d'autres mots, implémentez le pseudo-code suivant:

```
R1 ← 1 ;
while R0 ≠ 0 do
  R1 ← R1 × R0 ;
  R0 ← R0 - 1 ;
end
```

7. Répondez aux questions portant sur le code assembleur ARM suivant (les numéros de ligne sont indiqués à gauche):

```
1   B        main
2
3   tableau DC32 0x10, 0x42, 0xA4, 0xA0, 0x32, 0x05, 0x45, 0x02, 0x00
4
5   main
6   LDR SP, =maPile
7   ADD SP, SP, #64
8
9   LDR R0, =tableau
10  BL fonctionMystere
11  MOV R5, R0
12
13  B main
14
15  fonctionMystere
16  PUSH {R1, R2, LR}
17  LDR R1, [R0], #4
18  MOV R2, R1
19
20  debut
21  CMP R1, #0x00
22  BEQ fin
23
24  CMP R2, R1
25  MOVLT R2, R1
26  LDR R1, [R0], #4
27  B debut
28
29  fin
30  MOV R0, R2
31  POP {R1, R2, LR}
32  BX LR
33
34  maPile DS32 16
```

- (1 point) Pourquoi utilise-t-on l'instruction BL et non B à la ligne 10?
- (2 points) Pourquoi utilise-t-on les instructions PUSH et POP aux lignes 16 et 31?
- (3 points) Que fait l'instruction LDR R1, [R0], #4 à la ligne 17?
- (4 points) Quelle est la valeur de R5 après l'exécution de l'instruction MOV R5, R0 à la ligne 11?
- (5 points) Décrivez succinctement ce que fait la fonction `fonctionMystere`. De plus, indiquez ses arguments, la façon dont elle les obtient du programme principal, et comment elle retourne sa valeur de sortie.

8. Répondez aux questions suivantes.

- (a) (1 point) Donnez un exemple d'un nombre hexadécimal qu'on ne peut pas représenter en décimal.
- (b) (1 point) Que désigne l'acronyme RISC? Donnez un exemple de famille de microprocesseurs lui correspondant.
- (c) (1 point) Que désigne l'acronyme CISC? Donnez un exemple de famille de microprocesseurs lui correspondant.
- (d) (1 point) Quel sous-système détermine la première micro-instruction à exécuter pour une certaine instruction?
- (e) (1 point) Quel est le nombre maximal d'instructions dans un jeu d'instructions?
- (f) (1 point) Donnez deux exemples de mnémoniques d'instructions de gestion de la séquence d'instructions.
- (g) (1 point) En assembleur ARM, donnez l'instruction qui permet de placer l'adresse de `maVariable` dans le registre R5.
- (h) (1 point) Dans une architecture de type "memory-mapped I/O", quelle composante est responsable d'activer la broche d'activation (*enable*) des périphériques?
- (i) (1 point) Quelle composante est responsable de configurer l'ALU dans le bon mode?
- (j) (1 point) Expliquez la différence entre les mémoires volatiles et non-volatiles?

A Annexe: Instructions ARM et codes de conditions

| Instruction | Description |
|--------------------|--|
| ADD Rd, Rs, Op1 | $Rd \leftarrow Rs + Op1$ |
| AND Rd, Rs, Op1 | $Rd \leftarrow Rs \text{ AND } Op1$ |
| ASR Rd, Rs, #imm | $Rd \leftarrow Rs / 2^{imm}$ |
| Bcc Offset | PC \leftarrow PC + Offset, si cc est rencontré |
| BLcc Offset | Comme B, LR \leftarrow adresse de l'instruction suivante |
| CMP Rs, Op1 | Change les drapeaux comme Rs - Op1 |
| LDR Rd, [Rs, Op2] | $Rd \leftarrow Mem[Rs + Op2]$ |
| LDR Rd, [Rs], Op2 | $Rd \leftarrow Mem[Rs], Rs \leftarrow Rs + Op2$ |
| LDR Rd, [Rs, Op2]! | $Rs \leftarrow Rs + Op2, Rd \leftarrow Mem[Rs]$ |
| LSL Rd, Rs, #imm | $Rd \leftarrow Rs \times 2^{imm}$ |
| MUL Rd, Rs, Op1 | $Rd \leftarrow Rs \times Op1$ |
| MVN Rd, Op1 | $Rd \leftarrow !Op1$ (inverse les bits) |
| POP {Reg List} | Récupère la liste de registres de la pile |
| PUSH {Reg List} | Met la liste de registres sur la pile |
| STR Rd, [Rs, Op2] | $Mem[Rs + Op2] \leftarrow Rd$ |
| STR Rd, [Rs], Op2 | $Mem[Rs] \leftarrow Rd, Rs \leftarrow Rs + Op2$ |
| STR Rd, [Rs, Op2]! | $Rs \leftarrow Rs + Op2, Mem[Rs] \leftarrow Rd$ |
| SUB Rd, Rs, Op1 | $Rd \leftarrow Rs - Op1$ |

Table 2: Instructions ARM. Op1 dénote une opérande de type 1, et Op2 une opérande de type 2.

| Code | Condition | Code | Condition |
|------|--------------------|------|--------------------|
| CS | Retenue (carry) | CC | Pas de retenue |
| EQ | Égalité | NE | Inégalité |
| VS | Débordement | VC | Pas de débordement |
| GT | Plus grand | LT | Plus petit |
| GE | Plus grand ou égal | LE | Plus petit ou égal |
| PL | Positif | MI | Négatif |

Table 3: Codes de condition.

B Annexe: Table ASCII

| Dec | Hx | Oct | Char | Dec | Hx | Oct | Char | Dec | Hx | Oct | Char |
|-----|----|-----|-------|-----|----|-----|------|-----|----|-----|------|
| 0 | 0 | 000 | NUL | 43 | 2B | 053 | + | 86 | 56 | 126 | V |
| 1 | 1 | 001 | SOH | 44 | 2C | 054 | , | 87 | 57 | 127 | W |
| 2 | 2 | 002 | STX | 45 | 2D | 055 | - | 88 | 58 | 130 | X |
| 3 | 3 | 003 | ETX | 46 | 2E | 056 | . | 89 | 59 | 131 | Y |
| 4 | 4 | 004 | EOT | 47 | 2F | 057 | / | 90 | 5A | 132 | Z |
| 5 | 5 | 005 | ENQ | 48 | 30 | 060 | 0 | 91 | 5B | 133 | [|
| 6 | 6 | 006 | ACK | 49 | 31 | 061 | 1 | 92 | 5C | 134 | \ |
| 7 | 7 | 007 | BEL | 50 | 32 | 062 | 2 | 93 | 5D | 135 |] |
| 8 | 8 | 010 | BS | 51 | 33 | 063 | 3 | 94 | 5E | 136 | ^ |
| 9 | 9 | 011 | TAB | 52 | 34 | 064 | 4 | 95 | 5F | 137 | _ |
| 10 | A | 012 | LF | 53 | 35 | 065 | 5 | 96 | 60 | 140 | ' |
| 11 | B | 013 | VT | 54 | 36 | 066 | 6 | 97 | 61 | 141 | a |
| 12 | C | 014 | FF | 55 | 37 | 067 | 7 | 98 | 62 | 142 | b |
| 13 | D | 015 | CR | 56 | 38 | 070 | 8 | 99 | 63 | 143 | c |
| 14 | E | 016 | SO | 57 | 39 | 071 | 9 | 100 | 64 | 144 | d |
| 15 | F | 017 | SI | 58 | 3A | 072 | : | 101 | 65 | 145 | e |
| 16 | 10 | 020 | DLE | 59 | 3B | 073 | ; | 102 | 66 | 146 | f |
| 17 | 11 | 021 | DC1 | 60 | 3C | 074 | i | 103 | 67 | 147 | g |
| 18 | 12 | 022 | DC2 | 61 | 3D | 075 | = | 104 | 68 | 150 | h |
| 19 | 13 | 023 | DC3 | 62 | 3E | 076 | ¿ | 105 | 69 | 151 | i |
| 20 | 14 | 024 | DC4 | 63 | 3F | 077 | ? | 106 | 6A | 152 | j |
| 21 | 15 | 025 | NAK | 64 | 40 | 100 | @ | 107 | 6B | 153 | k |
| 22 | 16 | 026 | SYN | 65 | 41 | 101 | A | 108 | 6C | 154 | l |
| 23 | 17 | 027 | ETB | 66 | 42 | 102 | B | 109 | 6D | 155 | m |
| 24 | 18 | 030 | CAN | 67 | 43 | 103 | C | 110 | 6E | 156 | n |
| 25 | 19 | 031 | EM | 68 | 44 | 104 | D | 111 | 6F | 157 | o |
| 26 | 1A | 032 | SUB | 69 | 45 | 105 | E | 112 | 70 | 160 | p |
| 27 | 1B | 033 | ESC | 70 | 46 | 106 | F | 113 | 71 | 161 | q |
| 28 | 1C | 034 | FS | 71 | 47 | 107 | G | 114 | 72 | 162 | r |
| 29 | 1D | 035 | GS | 72 | 48 | 110 | H | 115 | 73 | 163 | s |
| 30 | 1E | 036 | RS | 73 | 49 | 111 | I | 116 | 74 | 164 | t |
| 31 | 1F | 037 | US | 74 | 4A | 112 | J | 117 | 75 | 165 | u |
| 32 | 20 | 040 | Space | 75 | 4B | 113 | K | 118 | 76 | 166 | v |
| 33 | 21 | 041 | ! | 76 | 4C | 114 | L | 119 | 77 | 167 | w |
| 34 | 22 | 042 | " | 77 | 4D | 115 | M | 120 | 78 | 170 | x |
| 35 | 23 | 043 | # | 78 | 4E | 116 | N | 121 | 79 | 171 | y |
| 36 | 24 | 044 | \$ | 79 | 4F | 117 | O | 122 | 7A | 172 | z |
| 37 | 25 | 045 | % | 80 | 50 | 120 | P | 123 | 7B | 173 | { |
| 38 | 26 | 046 | & | 81 | 51 | 121 | Q | 124 | 7C | 174 | } |
| 39 | 27 | 047 | ' | 82 | 52 | 122 | R | 125 | 7D | 175 | ~ |
| 40 | 28 | 050 | (| 83 | 53 | 123 | S | 126 | 7E | 176 | ~ |
| 41 | 29 | 051 |) | 84 | 54 | 124 | T | 127 | 7F | 177 | DEL |
| 42 | 2A | 052 | * | 85 | 55 | 125 | U | | | | |