

INF2171 — Organisation des ordinateurs et assembleur

Examen intra hiver 2024

Jean Privat & Hugo Leblanc — Dimanche 25 février
Durée 3 heures

- *Aucun document n'est autorisé sauf une feuille de note manuscrite au format lettre (recto-verso).*
- *L'usage de la calculatrice ou de tout appareil électronique est interdit.*
- *Répondez et rendez seulement le formulaire (avant dernière feuille).*
- *L'annexe est détachable (dernière feuille).*
- *Les questions faisant apparaître le symbole ♣ peuvent présenter plusieurs bonnes réponses. Les autres ont une unique bonne réponse.*
- *Chacune des 20 questions vaut 5 points. Des points négatifs pourront être affectés à de très mauvaises réponses.*

1 Structure et fonctionnement d'un ordinateur

Question 1 Quelle est l'alternative classique à l'architecture de type RISC ?

- | | |
|--|---|
| <input type="checkbox"/> A Engine Instruction Set Computer | <input type="checkbox"/> F Hard Instruction Set Computer |
| <input type="checkbox"/> B Spaced Instruction Set Computer | <input type="checkbox"/> G Dedicated Instruction Set Computer |
| <input type="checkbox"/> C General Instruction Set Computer | <input type="checkbox"/> H Elaborate Instruction Set Computer |
| <input type="checkbox"/> D Complex Instruction Set Computer | <input type="checkbox"/> I Friendly Instruction Set Computer |
| <input type="checkbox"/> E Compiled Instruction Set Computer | <input type="checkbox"/> J eXtrem Instruction Set Computer |

2 Arithmétique des ordinateurs

Question 2 ♣ Parmi les représentations suivantes, quelles sont celles du nombre $33_{(16)}$ (33 hexadécimal) ? Indiquez toutes les bonnes réponses.

- | | | | |
|---|---|--|--|
| <input type="checkbox"/> A $52_{(10)}$ | <input type="checkbox"/> C $110011_{(2)}$ | <input type="checkbox"/> E $41_{(8)}$ | <input type="checkbox"/> G $41_{(16)}$ |
| <input type="checkbox"/> B $011011_{(2)}$ | <input type="checkbox"/> D $21_{(16)}$ | <input type="checkbox"/> F $51_{(10)}$ | <input type="checkbox"/> H $63_{(8)}$ |

Question 3 ♣ Parmi les séquences d'instructions RISC-V suivantes, quelles sont celles qui positionnent le registre `s0` à la valeur `0x123` ? Indiquez toutes les bonnes réponses.

- | | | |
|---|---|--|
| <input type="checkbox"/> A <code>li s0, 0x100</code>
<code>addi s0, s0, 23</code> | <input type="checkbox"/> C <code>li s0, 0x1FE</code>
<code>xori s0, s0, 0xDD</code> | <input type="checkbox"/> E <code>li s0, 0x1230</code>
<code>srli s0, s0, 4</code> |
| <input type="checkbox"/> B <code>li s0, 0x146</code>
<code>addi s0, s0, -35</code> | <input type="checkbox"/> D <code>li s0, 0x678</code>
<code>andi s0, s0, 0x123</code> | <input type="checkbox"/> F <code>li s0, 0x1230</code>
<code>slli s0, s0, 1</code> |

Question 4 ♣ Quelles opérations causent un débordement en considérant la taille maximale des résultats ? Indiquez toutes les bonnes réponses.

- | | |
|--|--|
| <input type="checkbox"/> A Résultat sur 8 bits signé, $70 + 70$ | <input type="checkbox"/> E Résultat sur 16 bits signé, $-15000 - 15000$ |
| <input type="checkbox"/> B Résultat sur 8 bits non signé, $70 + 70$ | <input type="checkbox"/> F Résultat sur 8 bits signé, $-1 - 1$ |
| <input type="checkbox"/> C Résultat sur 8 bits signé, $-1 + -128$ | <input type="checkbox"/> G Résultat sur 16 bits non-signé, $0x1234 * 0x10$ |
| <input type="checkbox"/> D Résultat sur 16 bits non-signé, $0xFEDC + 0xBA98$ | <input type="checkbox"/> H Résultat sur 32 bits non-signé, $0x1234 * 0x5678$ |

brouillon

3 Instructions et code machine RISC-V

Question 5 Que fait la directive RISC-V « `.word 0x10` » ?

- A Réserve 2 octets, le premier initialisé à 0 et le dernier initialisé à 0x10 (hexa)
- B Réserve 10 octets, le dernier initialisé à 0x4 (hexa) et les autres à 0
- C Réserve 16 octets, le premier initialisé à 0x10 (hexa) et les autres à 0
- D Réserve 4 octets, le dernier initialisé à 0x10 (hexa) et les autres à 0
- E Réserve 2 octets, le premier initialisé à 0x10 (hexa) et le second à 0
- F Réserve 16 octets, le dernier initialisé à 0x4 (hexa) et les autres à 0
- G Réserve 10 octets, tous initialisé à 0x10 (hexa)
- H Réserve 16 octets, tous initialisé à 0x10 (hexa)
- I Réserve 4 octets, le premier initialisé à 0x10 (hexa) et les autres à 0
- J Réserve un seul octet, initialisé à 0x10 (hexa)
- K Réserve 10 octets, le dernier initialisé à 0x10 (hexa) et les autres à 0
- L Réserve 16 octets, tous initialisé à 0x04 (hexa)
- M Réserve 10 octets, le premier initialisé à 0x4 (hexa) et les autres à 0
- N Réserve 10 octets, le premier initialisé à 0x10 (hexa) et les autres à 0
- O Réserve 10 octets, tous initialisé à 0x04 (hexa)
- P Réserve 16 octets, le dernier initialisé à 0x10 (hexa) et les autres à 0
- Q Réserve 16 octets, le premier initialisé à 0x4 (hexa) et les autres à 0

Question 6 Désassemblez l'instruction machine RISC-V 64 bits qui est codée en mémoire par les 4 octets suivants : « B3 C4 65 02 ».

- | | | | |
|---|---|---|--|
| <input type="checkbox"/> A <code>div t1, a1, s1</code> | <input type="checkbox"/> D <code>xori a1, t1, 41</code> | <input type="checkbox"/> G <code>xori s1, a1, 38</code> | <input type="checkbox"/> J <code>xori s1, a1, 6</code> |
| <input type="checkbox"/> B <code>xori t1, a1, 41</code> | <input type="checkbox"/> E <code>xor s1, a1, t1</code> | <input type="checkbox"/> H <code>div s1, a1, t1</code> | <input type="checkbox"/> K <code>xori t1, a1, 9</code> |
| <input type="checkbox"/> C <code>divi s1, a1, 6</code> | <input type="checkbox"/> F <code>xor t1, a1, s1</code> | <input type="checkbox"/> I <code>xori a1, s1, 38</code> | <input type="checkbox"/> L <code>divi t1, a1, 9</code> |

Question 7 ♣ Dans une architecture RISC-V 64 bits, quelles sont les directives qui génèrent des données d'exactly la même taille qu'un registre. Indiquez toutes les bonnes réponses.

- | | | |
|--|---|--|
| <input type="checkbox"/> A <code>.byte 1, 2, 3, 4</code> | <input type="checkbox"/> F <code>.word 0,0,0,0</code> | <input type="checkbox"/> K <code>.align 2</code> |
| <input type="checkbox"/> B <code>.space 8</code> | <input type="checkbox"/> G <code>.half 0, 1, 2, 3</code> | <input type="checkbox"/> L <code>.eqv size, 8</code> |
| <input type="checkbox"/> C <code>.eqv size, 64</code> | <input type="checkbox"/> H <code>.half 0xFF, 0xFF</code> | <input type="checkbox"/> M <code>.ascii "64"</code> |
| <input type="checkbox"/> D <code>.dword 64</code> | <input type="checkbox"/> I <code>.string "bonbons"</code> | <input type="checkbox"/> N <code>.dword 0</code> |
| <input type="checkbox"/> E <code>.word 2</code> | <input type="checkbox"/> J <code>.space 4</code> | <input type="checkbox"/> O <code>.space 64</code> |

brouillon

4 Programme mystère

Soit le programme mystère suivant :

```

    li a7, 5
    ecall
    mv s0, a0
    ecall
    mv s1, a0
    li s2, 0
aaaa:
    beqz s0, bbbb
    add s2, s2, s1
    addi s0, s0, -1
    j aaaa
bbbb:
    li a7, 1
    mv a0, s2
    ecall
    li a7, 10
    ecall

```

Question 8 Qu'affiche-t-il lorsque l'on saisit les nombres 0 et 10 ?

- A 0
- B 1
- C 2
- D 3

- E 4
- F 5
- G 6
- H 7

- I 8
- J Il n'affiche rien du tout
- K Il part en boucle infinie
- L Il affiche n'importe quoi

Question 9 Qu'affiche-t-il lorsque l'on saisit les nombres 2 et 3 ?

- A 0
- B 1
- C 2
- D 3

- E 4
- F 5
- G 6
- H 7

- I 8
- J Il n'affiche rien du tout
- K Il part en boucle infinie
- L Il affiche n'importe quoi

Question 10 Quel est le premier caractère affiché lorsque l'on saisit les nombres 2000 et 4000 ?

- A 0
- B 1
- C 2
- D 3

- E 4
- F 5
- G 6
- H 7

- I 8
- J Il n'affiche rien du tout
- K Il part en boucle infinie
- L Il affiche n'importe quoi

brouillon

5 Modes d'adressage

Soit le morceau de programme suivant exécuté sur une architecture RISC-V 64 bits. Sachant que l'étiquette `cafe` est associée à l'adresse `0x10010000`, indiquez quel est le contenu (hexadécimal) final des registres demandés.

```
.data
cafe:
    .word 0xCAFE
    .eqv  CA, 0xFE
    .byte  CA, 0xFE
    .ascii "CAFE"

.text
la s0, cafe
lw s1, cafe
li s2, CA
lb s3, 1(s0)
lwu s4, 4(s0)
```

Question 11 Le registre `s0`.

- | | | | | | | | |
|----------------------------|--------------------|----------------------------|--------------------|----------------------------|--------------------|----------------------------|--------------------|
| <input type="checkbox"/> A | 0x0000100100000000 | <input type="checkbox"/> E | 0x0800100100000000 | <input type="checkbox"/> I | 0x0000000001100000 | <input type="checkbox"/> M | 0x0000000001100080 |
| <input type="checkbox"/> B | 0x0000011000000000 | <input type="checkbox"/> F | 0x0110008000000000 | <input type="checkbox"/> J | 0x1001000000000000 | <input type="checkbox"/> N | 0x0000000000000110 |
| <input type="checkbox"/> C | 0x0110000000000000 | <input type="checkbox"/> G | 0x0000000008001001 | <input type="checkbox"/> K | 0x0000000000001001 | <input type="checkbox"/> O | 0x0000000010010000 |
| <input type="checkbox"/> D | 0x0000000008000110 | <input type="checkbox"/> H | 0x0000000010010008 | <input type="checkbox"/> L | 0x8000011000000000 | <input type="checkbox"/> P | 0x1001000800000000 |

Question 12 Le registre `s1`.

- | | | | | | | | |
|----------------------------|--------------------|----------------------------|--------------------|----------------------------|--------------------|----------------------------|--------------------|
| <input type="checkbox"/> A | 0xCAFE000000000000 | <input type="checkbox"/> E | 0x000000000000FEFE | <input type="checkbox"/> I | 0x000000000000CACA | <input type="checkbox"/> M | 0x0000000043414645 |
| <input type="checkbox"/> B | 0x0000000045464143 | <input type="checkbox"/> F | 0x00000000CAFEFEFE | <input type="checkbox"/> J | 0x000000000000FECA | <input type="checkbox"/> N | 0xCAFEFEFE00000000 |
| <input type="checkbox"/> C | 0x00000000000000FE | <input type="checkbox"/> G | 0x00000000000000CA | <input type="checkbox"/> K | 0xFE00000000000000 | <input type="checkbox"/> O | 0xFECA000000000000 |
| <input type="checkbox"/> D | 0x4341464500000000 | <input type="checkbox"/> H | 0xCA00000000000000 | <input type="checkbox"/> L | 0x000000000000CAFE | <input type="checkbox"/> P | 0x4546414300000000 |

Question 13 Le registre `s2`.

- | | | | | | | | |
|----------------------------|--------------------|----------------------------|--------------------|----------------------------|--------------------|----------------------------|--------------------|
| <input type="checkbox"/> A | 0xFE00000000000000 | <input type="checkbox"/> E | 0xCAFE000000000000 | <input type="checkbox"/> I | 0x000000000000CACA | <input type="checkbox"/> M | 0x0000000045464143 |
| <input type="checkbox"/> B | 0x00000000CAFEFEFE | <input type="checkbox"/> F | 0xCA00000000000000 | <input type="checkbox"/> J | 0x00000000000000CA | <input type="checkbox"/> N | 0x000000000000FECA |
| <input type="checkbox"/> C | 0x00000000000000FE | <input type="checkbox"/> G | 0xCAFEFEFE00000000 | <input type="checkbox"/> K | 0xFECA000000000000 | <input type="checkbox"/> O | 0x4341464500000000 |
| <input type="checkbox"/> D | 0x4546414300000000 | <input type="checkbox"/> H | 0x000000000000FEFE | <input type="checkbox"/> L | 0x000000000000CAFE | <input type="checkbox"/> P | 0x0000000043414645 |

Question 14 Le registre `s3`.

- | | | | | | | | |
|----------------------------|--------------------|----------------------------|--------------------|----------------------------|----------------------|----------------------------|--------------------|
| <input type="checkbox"/> A | 0x00000000CAFEFEFE | <input type="checkbox"/> E | 0xCAFEFFFFFFFFFFFF | <input type="checkbox"/> I | 0xFFFFFFFFFFFFFFFF | <input type="checkbox"/> M | 0x00000000000000FE |
| <input type="checkbox"/> B | 0xFEFEFFFFFFFFFFFF | <input type="checkbox"/> F | 0x0000000000000000 | <input type="checkbox"/> J | 0xFEFFFFFFFFFFFFFFFF | <input type="checkbox"/> N | 0x0000000000FEFEFE |
| <input type="checkbox"/> C | 0x0000000000000043 | <input type="checkbox"/> G | 0xFEFE000000000000 | <input type="checkbox"/> K | 0xFFFFFFFFFFFEFEFE | <input type="checkbox"/> O | 0xFFFFFFFFFFFFFFFF |
| <input type="checkbox"/> D | 0x000000000000FEFE | <input type="checkbox"/> H | 0x00000000000000CA | <input type="checkbox"/> L | 0xFFFFFFFFFFFFFCA | <input type="checkbox"/> P | 0xFFFFFFFFCAFEFEFE |

Question 15 Le registre `s4`.

- | | | | | | | | |
|----------------------------|--------------------|----------------------------|--------------------|----------------------------|--------------------|----------------------------|--------------------|
| <input type="checkbox"/> A | 0x00000000FEFE4143 | <input type="checkbox"/> E | 0x00000000CAFECAFE | <input type="checkbox"/> I | 0x00000000FE434146 | <input type="checkbox"/> M | 0x000000000000FECA |
| <input type="checkbox"/> B | 0x4341FE4143FE4143 | <input type="checkbox"/> F | 0xFFFFFFFFFFFFFFFF | <input type="checkbox"/> J | 0x00000000CAFECAFE | <input type="checkbox"/> N | 0x00000000CAFE4341 |
| <input type="checkbox"/> C | 0x4143FE4341464500 | <input type="checkbox"/> G | 0x0000000000004645 | <input type="checkbox"/> K | 0x0000000043414645 | <input type="checkbox"/> O | 0x00000000000000FE |
| <input type="checkbox"/> D | 0x000000000000CAFE | <input type="checkbox"/> H | 0x00000000CAFEFEFE | <input type="checkbox"/> L | 0x0000000045464143 | <input type="checkbox"/> P | 0x000000004143FEFE |

6 Programmation

Soit le programme incomplet suivant dont l'objectif est d'afficher la somme et le produit des éléments d'un tableau.

Note : pour simplifier l'exercice, le tableau est donné directement dans le programme (et non lu de l'entrée). Ainsi ce programme avec le tableau initial tel que donné devrait afficher « 33 336000 ». Si l'on remplaçait le premier mot de `tab` par `6` le programme afficherait « 38 2016000 ».

```
# Somme et produits de nombres dans un tableau
.data
# Tableau de 10 valeurs
tab: .word 1, 5, 8, -7, 2, 3, 10, 2, 10, -1
      .eqv tablen, 10 # Nombre d'éléments du tableau

# Appels système RARS utilisés
      .eqv PrintInt, 1
      .eqv PrintChar, 11
      .eqv Exit, 10

.text
# Initialisations # TODO1
la s0, tab      # s0 Adresse de l'élément courant
li s1, tablen   # s1 Nombre d'éléments restants à traiter
l?????        # s2 Somme actuelle
l?????        # s3 Produit actuel

loop:
# Boucler sur chaque élément # TODO2
b?????

# Récupérer dans s4 la valeur de l'élément courant # TODO3
l?????

# Mettre à jour la somme et le produit actuels # TODO4
a?????
m?????

# Passer à l'élément suivant # TODO5
a?????
a?????
j loop

end:
# Afficher somme et produit
li a7, PrintInt
mv a0, s2
ecall
li a7, PrintChar
li a0, ' '
ecall
li a7, PrintInt
mv a0, s3
ecall
li a7, Exit
ecall
```


Question 16 Que faut-il pour initialiser le registre `s3` dans le groupe d'instructions `TOD01` ?

- | | | | |
|--|--|---|---|
| <input type="checkbox"/> A <code>lw s3, 0(tablen)</code> | <input type="checkbox"/> D <code>la s3, tab</code> | <input type="checkbox"/> G <code>lw s3, 0(tab)</code> | <input type="checkbox"/> J <code>lw s3, -1(loop)</code> |
| <input type="checkbox"/> B <code>lw s3, 1(loop)</code> | <input type="checkbox"/> E <code>li s3, 1</code> | <input type="checkbox"/> H <code>li s3, tablen</code> | <input type="checkbox"/> K <code>li s3, -1</code> |
| <input type="checkbox"/> C <code>li s3, tab</code> | <input type="checkbox"/> F <code>lw s3, -1(tab)</code> | <input type="checkbox"/> I <code>li s3, 0</code> | <input type="checkbox"/> L <code>la s3, tablen</code> |

Question 17 Que faut-il mettre à l'instruction manquante du `TOD02` ?

- | | | | |
|--|---|---|--|
| <input type="checkbox"/> A <code>bgt s1, s2, end</code> | <input type="checkbox"/> E <code>bgtz s1, loop</code> | <input type="checkbox"/> I <code>blez s1, end</code> | <input type="checkbox"/> M <code>ble s1, s2, loop</code> |
| <input type="checkbox"/> B <code>bgtz s1, end</code> | <input type="checkbox"/> F <code>bgez s2, loop</code> | <input type="checkbox"/> J <code>bgtz s2, loop</code> | <input type="checkbox"/> N <code>ble s1, s2, end</code> |
| <input type="checkbox"/> C <code>bgt s1, s2, loop</code> | <input type="checkbox"/> G <code>bgez s1, loop</code> | <input type="checkbox"/> K <code>blez s2, loop</code> | <input type="checkbox"/> O <code>bge s1, s2, loop</code> |
| <input type="checkbox"/> D <code>bgtz s2, end</code> | <input type="checkbox"/> H <code>bgez s2, end</code> | <input type="checkbox"/> L <code>blez s2, end</code> | <input type="checkbox"/> P <code>blez s1, loop</code> |

Question 18 Que faut-il mettre à l'instruction manquante du `TOD03` ?

- | | | | |
|--|--|--|--|
| <input type="checkbox"/> A <code>ld s4, 0(s0)</code> | <input type="checkbox"/> E <code>lw s3, 0(s1)</code> | <input type="checkbox"/> I <code>ld s4, 4(s1)</code> | <input type="checkbox"/> M <code>lw s4, 0(s0)</code> |
| <input type="checkbox"/> B <code>ld s4, 0(s1)</code> | <input type="checkbox"/> F <code>ld s3, 4(s1)</code> | <input type="checkbox"/> J <code>ld s3, 0(s1)</code> | <input type="checkbox"/> N <code>ld s3, 0(s0)</code> |
| <input type="checkbox"/> C <code>lw s3, 4(s1)</code> | <input type="checkbox"/> G <code>lw s4, 0(s1)</code> | <input type="checkbox"/> K <code>lw s3, 0(s0)</code> | <input type="checkbox"/> O <code>lw s4, 4(s1)</code> |
| <input type="checkbox"/> D <code>lw s3, 4(s0)</code> | <input type="checkbox"/> H <code>ld s4, 4(s0)</code> | <input type="checkbox"/> L <code>lw s4, 4(s0)</code> | <input type="checkbox"/> P <code>ld s3, 4(s0)</code> |

Question 19 Que faut-il mettre aux instructions manquantes du `TOD04` ?

- | | | | |
|---|---|---|---|
| <input type="checkbox"/> A <code>add s2, s4, s4</code>
<code>mul s3, s4, s4</code> | <input type="checkbox"/> C <code>addi s2, s4, s4</code>
<code>muli s3, s4, s4</code> | <input type="checkbox"/> E <code>addi s3, s3, s4</code>
<code>muli s2, s2, s4</code> | <input type="checkbox"/> G <code>addi s2, s2, s4</code>
<code>muli s3, s3, s4</code> |
| <input type="checkbox"/> B <code>addi s3, s4, s4</code>
<code>muli s2, s4, s4</code> | <input type="checkbox"/> D <code>add s3, s4, s4</code>
<code>mul s2, s4, s4</code> | <input type="checkbox"/> F <code>add s2, s2, s4</code>
<code>mul s3, s3, s4</code> | <input type="checkbox"/> H <code>add s3, s3, s4</code>
<code>mul s2, s2, s4</code> |

Question 20 Que faut-il mettre aux instructions manquantes du `TOD05` ?

- | | | | |
|---|---|--|---|
| <input type="checkbox"/> A <code>addi s0, s0, 1</code>
<code>addi s1, s1, 4</code> | <input type="checkbox"/> E <code>addi s0, s0, -1</code>
<code>addi s1, s1, -1</code> | <input type="checkbox"/> I <code>addi s0, s0, 4</code>
<code>addi s1, s1, -1</code> | <input type="checkbox"/> M <code>addi s0, s0, 4</code>
<code>addi s1, s1, -4</code> |
| <input type="checkbox"/> B <code>addi s0, s0, -1</code>
<code>addi s1, s1, -4</code> | <input type="checkbox"/> F <code>addi s0, s0, 1</code>
<code>addi s1, s1, -1</code> | <input type="checkbox"/> J <code>addi s0, s0, 1</code>
<code>addi s1, s1, -4</code> | <input type="checkbox"/> N <code>addi s0, s0, 4</code>
<code>addi s1, s1, 1</code> |
| <input type="checkbox"/> C <code>addi s0, s0, -4</code>
<code>addi s1, s1, 1</code> | <input type="checkbox"/> G <code>addi s0, s0, -4</code>
<code>addi s1, s1, -1</code> | <input type="checkbox"/> K <code>addi s0, s0, -4</code>
<code>addi s1, s1, 4</code> | <input type="checkbox"/> O <code>addi s0, s0, 4</code>
<code>addi s1, s1, 4</code> |
| <input type="checkbox"/> D <code>addi s0, s0, -1</code>
<code>addi s1, s1, 4</code> | <input type="checkbox"/> H <code>addi s0, s0, -1</code>
<code>addi s1, s1, 1</code> | <input type="checkbox"/> L <code>addi s0, s0, 1</code>
<code>addi s1, s1, 1</code> | <input type="checkbox"/> P <code>addi s0, s0, -4</code>
<code>addi s1, s1, -4</code> |

brouillon

brouillon



Feuille de réponses — Examen intra hiver 2024
INF2171 — Organisation des ordinateurs et assembleur
Jean Privat & Hugo Leblanc — Dimanche 25 février

0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8
9	9	9	9	9	9	9	9

← codez les 8 chiffres de votre code permanent ci-contre, et inscrivez-le à nouveau ci-dessous avec votre nom et prénom. 5 points de pénalité en cas d'oubli ou d'erreur.

Code permanent :
.....
Nom :
.....
Prénom :
.....

Les réponses aux questions sont à donner exclusivement sur cette feuille : les réponses données sur les feuilles précédentes ne seront pas prises en compte. Ne pas utiliser un formulaire d'une autre copie au risque d'avoir toutes les réponses fausses.
Important : noircissez complètement l'intérieur de chaque case (pas de croix, pas de cercle).

- Question 1 :

A	B	C	D	E	F	G	H	I	J
---	---	---	---	---	---	---	---	---	---
- Question 2 :

A	B	C	D	E	F	G	H
---	---	---	---	---	---	---	---
- Question 3 :

A	B	C	D	E	F
---	---	---	---	---	---
- Question 4 :

A	B	C	D	E	F	G	H
---	---	---	---	---	---	---	---
- Question 5 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 6 :

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 7 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 8 :

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 9 :

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 10 :

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 11 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 12 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 13 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 14 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 15 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 16 :

A	B	C	D	E	F	G	H	I	J	K	L
---	---	---	---	---	---	---	---	---	---	---	---
- Question 17 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 18 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---
- Question 19 :

A	B	C	D	E	F	G	H
---	---	---	---	---	---	---	---
- Question 20 :

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Jeu d'instruction RISC-V

Inst	Description	FMT	opcode	fn3	fn7 (ou imm)	Description	Note
add	Add	R	0x33	0x0	0x00	rd = rs1 + rs2	
sub	Subtract	R	0x33	0x0	0x20	rd = rs1 - rs2	
xor	Exclusive Or	R	0x33	0x4	0x00	rd = rs1 ^ rs2	
or	Or	R	0x33	0x6	0x00	rd = rs1 rs2	
and	And	R	0x33	0x7	0x00	rd = rs1 & rs2	
sll	Shift Left Logical	R	0x33	0x1	0x00	rd = rs1 << rs2	
srl	Shift Right Logical	R	0x33	0x5	0x00	rd = rs1 >> rs2	
sra	Shift Right Arithmetic	R	0x33	0x5	0x20	rd = rs1 >> rs2	
slt	Set Less Than	R	0x33	0x2	0x00	rd = (rs1 < rs2)?1:0	
sltu	Set Less Than (U)	R	0x33	0x3	0x00	rd = (rs1 < rs2)?1:0	
addi	Add Immediate	I	0x13	0x0		rd = rs1 + imm	
xori	Xor Immediate	I	0x13	0x4		rd = rs1 ^ imm	
ori	Or Immediate	I	0x13	0x6		rd = rs1 imm	
andi	And Immediate	I	0x13	0x7		rd = rs1 & imm	
slli	Shift Left Logical Imm	I	0x13	0x1	imm[11:6]=0x00	rd = rs1 << imm[5:0]	
srli	Shift Right Logical Imm	I	0x13	0x5	imm[11:6]=0x00	rd = rs1 >> imm[5:0]	
srai	Shift Right Arith Imm	I	0x13	0x5	imm[11:6]=0x10	rd = rs1 >> imm[5:0]	
slti	Set Less Than Imm	I	0x13	0x2		rd = (rs1 < imm)?1:0	
sltiu	Set Less Than Imm (U)	I	0x13	0x3		rd = (rs1 < imm)?1:0	
lb	Load Byte	I	0x03	0x0		rd = M[rs1+imm][7:0]	
lh	Load Half	I	0x03	0x1		rd = M[rs1+imm][15:0]	
lw	Load Word	I	0x03	0x2		rd = M[rs1+imm][31:0]	
ld	Load Double Word	I	0x03	0x3		rd = M[rs1+imm][63:0]	RV64I
lbu	Load Byte (U)	I	0x03	0x4		rd = M[rs1+imm][7:0]	
lhu	Load Half (U)	I	0x03	0x5		rd = M[rs1+imm][15:0]	
lwu	Load Word (U)	I	0x03	0x6		rd = M[rs1+imm][63:0]	RV64I
sb	Store Byte	S	0x23	0x0		M[rs1+imm][7:0] = rs2[7:0]	
sh	Store Half	S	0x23	0x1		M[rs1+imm][15:0] = rs2[15:0]	
sw	Store Word	S	0x23	0x2		M[rs1+imm][31:0] = rs2[31:0]	
sd	Store Double Word	S	0x23	0x3		M[rs1+imm][63:0] = rs2[63:0]	RV64I
beq	Branch ==	B	0x63	0x0		if(rs1 == rs2) PC += imm	
bne	Branch !=	B	0x63	0x1		if(rs1 != rs2) PC += imm	
blt	Branch <	B	0x63	0x4		if(rs1 < rs2) PC += imm	
bge	Branch ≥	B	0x63	0x5		if(rs1 ≥ rs2) PC += imm	
bltu	Branch < (U)	B	0x63	0x6		if(rs1 < rs2) PC += imm	
bgeu	Branch ≥ (U)	B	0x63	0x7		if(rs1 ≥ rs2) PC += imm	
jal	Jump And Link	J	0x6F			rd = PC+4; PC += imm	
jalr	Jump And Link Register	I	0x67	0x0		rd = PC+4; PC = rs1 + imm	
lui	Load Upper Imm	U	0x37			rd = imm << 12	
auipc	Add Upper Imm to PC	U	0x17			rd = PC + (imm << 12)	
ecall	Environment Call	I	0x73	0x0	imm=0x0	Appel système	
ebreak	Environment Break	I	0x73	0x0	imm=0x1	Appel au débogueur	
mul	Multiply	R	0x33	0x0	0x01	rd = (rs1 * rs2)[31:0]	RV32M
mulh	Multiply High	R	0x33	0x1	0x01	rd = (rs1 * rs2)[63:32]	RV32M
mulsu	Multiply High (S) (U)	R	0x33	0x2	0x01	rd = (rs1 * rs2)[63:32]	RV32M
mulu	Multiply High (U)	R	0x33	0x3	0x01	rd = (rs1 * rs2)[63:32]	RV32M
div	Divide	R	0x33	0x4	0x01	rd = rs1 / rs2	RV32M
divu	Divide (U)	R	0x33	0x5	0x01	rd = rs1 / rs2	RV32M
rem	Remainder	R	0x33	0x6	0x01	rd = rs1 % rs2	RV32M
remu	Remainder (U)	R	0x33	0x7	0x01	rd = rs1 % rs2	RV32M

Format des instructions

	31	25	24	20	19	15	14	12	11	7	6	0
R	fn7			rs2	rs1	fn3	rd			opcode		
I	imm[11:0]			rs1	fn3	rd			opcode			
S	imm[11:5]			rs2	rs1	fn3	imm[4:0]			opcode		
B	imm[12:10:5]			rs2	rs1	fn3	imm[4:1:11]			opcode		
U	imm[31:12]						rd			opcode		
J	imm[20:10:1:11:19:12]						rd			opcode		

Note: petit-boutiste et aligné sur 4 octets

Registres

Registre	Nom d'ABI	Description	Qui sauve ?
x0	zero	Constante zero	—
x1	ra	Return address	Appellant
x2	sp	Stack pointer	Appelé
x3	gp	Global pointer	—
x4	tp	Thread pointer	—
x5-x7, x28-x31	t0-t6	Temporaires	Appellant
x8, x9, x18-x27	s0-s11	Sauvegardés	Appelé
x10-x17	a0-a7	Arguments (et retours)	Appellant

Pseudoinstructions

Pseudoinstruction	Instruction(s)	Description
la rd, lab	auipc rd, H addi rd, rd, L]	Load Address
l[bhwd] rd, lab	auipc rd, H l[bhwd] rd, L(rd)	Load
s[bhwd] rd, lab, rt	auipc rt, H s[bhwd] rd, L(rt)	Store
nop	addi x0, x0, 0	No Operation
li rd, imm	addi rd, x0, imm	Load Immediate
li rd, imm	Myriade	Load Immediate (v2)
mv rd, rs	addi rd, rs, 0	Move (copie)
not rd, rs	xori rd, rs, -1	Not (bit à bit)
neg rd, rs	sub rd, x0, rs	Negate (opposé)
seqz rd, rs	sltiu rd, rs, 1	Set if = zero
snez rd, rs	sltu rd, x0, rs	Set if ≠ zero
sltz rd, rs	slt rd, rs, x0	Set if < zero
sgtz rd, rs	slt rd, x0, rs	Set if > zero
beqz rs, lab	beq rs, x0, lab	Branch if = zero
bnez rs, lab	bne rs, x0, lab	Branch if ≠ zero
blez rs, lab	bge x0, rs, lab	Branch if ≤ zero
bgez rs, lab	bge rs, x0, lab	Branch if ≥ zero
bltz rs, lab	blt rs, x0, lab	Branch if < zero
bgtz rs, lab	blt x0, rs, lab	Branch if > zero
bgt rs, rt, lab	blt rt, rs, lab	Branch if >
ble rs, rt, lab	bge rt, rs, lab	Branch if <
bgtu rs, rt, lab	bltu rt, rs, lab	Branch if > (U)
bleu rs, rt, lab	bgeu rt, rs, lab	Branch if < (U)
j lab	jal x0, lab	Jump
jal lab	jal ra, lab	Jump And Link
jr rs	jalr x0, rs, 0	Jump Register
jalr rs	jalr ra, rs, 0	JAL Register
ret	jalr x0, ra, 0	Return
call lab	auipc x1, H jalr x1, x1, L	Call (loin)
tail lab	auipc x6, H jalr x0, x6, L	Tail call (loin)
fence	fence iorw, iorw	Fence (totale)

Directives RARS et GNU as

Directive	Signification
.align <i>n</i>	Des octets à 0 pour aligner sur 2^n
.ascii <i>s</i>	Code ascii de chacun des caractères de <i>s</i>
.asciz <i>s</i>	Code ascii de chacun des caractères de <i>s</i> suivi de '\0'
.byte <i>n</i>	Une ou plusieurs valeurs sur un octet
.data	Travaille dans la section Data
.double <i>n</i>	Une ou plusieurs valeurs flottantes double précision
.dword <i>n</i>	Une ou plusieurs valeurs sur 8 octets (double mot)
.eqv <i>s, n</i>	Attribue la valeur <i>n</i> au symbole <i>s</i>
.float <i>n</i>	Une ou plusieurs valeurs flottantes simple précision
.half <i>n</i>	Une ou plusieurs valeurs sur 2 octets (demi mot)
.space <i>n</i>	<i>n</i> octets à 0
.string <i>s</i>	Alias pour .asciz
.text	Travaille dans la section Text
.word <i>n</i>	Une ou plusieurs valeurs sur 4 octets (mot)

Quelques appels système RARS

Nom	a7	Signification
PrintInt	1	Afficher le nombre a0 (décimal)
PrintString	4	Afficher la chaîne pointée par a0
ReadInt	5	Lire nombre dans a0 (décimal)
ReadString	8	Lire chaîne dans le tampon a0 de taille a1
Sbrk	9	Alloue a0 octets dans le tas, retourne l'adresse dans a0
Exit	10	Quitter
PrintChar	11	Afficher le caractère de code ASCII a0
ReadChar	12	Saisir un caractère dans a0 (code ASCII)

Taille des entiers

Bits	Octets	Nom RISC-V	Signé	Min	Max
8	1	byte	non	0	255
			oui	-128	127
16	2	halfword	non	0	65 535
			oui	-32 768	32 767
32	4	word	non	0	$\approx 4.29 \times 10^9$
			oui	$\approx -2.15 \times 10^9$	$\approx 2.15 \times 10^9$
64	8	doubleword	non	0	$\approx 1.84 \times 10^{19}$
			oui	$\approx -9.22 \times 10^{18}$	$\approx 9.22 \times 10^{18}$
128	16	quadword	non	0	$\approx 3.40 \times 10^{38}$
			oui	$\approx -1.70 \times 10^{38}$	$\approx 1.70 \times 10^{38}$

Table ASCII

Dec Hex	Dec Hex	Dec Hex	Dec Hex	Dec Hex	Dec Hex	Dec Hex	Dec Hex
0 00 NUL '\0'	16 10 DLE	32 20 SP ' '	48 30 0	64 40 @	80 50 P	96 60 ' '	112 70 p
1 01 SOH	17 11 DC1	33 21 !	49 31 1	65 41 A	81 51 Q	97 61 a	113 71 q
2 02 STX	18 12 DC2	34 22 "	50 32 2	66 42 B	82 52 R	98 62 b	114 72 r
3 03 ETX	19 13 DC3	35 23 #	51 33 3	67 43 C	83 53 S	99 63 c	115 73 s
4 04 EOT	20 14 DC4	36 24 \$	52 34 4	68 44 D	84 54 T	100 64 d	116 74 t
5 05 ENQ	21 15 NAK	37 25 %	53 35 5	69 45 E	85 55 U	101 65 e	117 75 u
6 06 ACK	22 16 SYN	38 26 &	54 36 6	70 46 F	86 56 V	102 66 f	118 76 v
7 07 BEL '\a'	23 17 ETB	39 27 ' '	55 37 7	71 47 G	87 57 W	103 67 g	119 77 w
8 08 BS '\b'	24 18 CAN	40 28 (56 38 8	72 48 H	88 58 X	104 68 h	120 78 x
9 09 HT '\t'	25 19 EM	41 29)	57 39 9	73 49 I	89 59 Y	105 69 i	121 79 y
10 0A LF '\n'	26 1A SUB	42 2A *	58 3A :	74 4A J	90 5A Z	106 6A j	122 7A z
11 0B VT '\v'	27 1B ESC	43 2B +	59 3B ;	75 4B K	91 5B [107 6B k	123 7B {
12 0C FF '\f'	28 1C FS	44 2C ,	60 3C <	76 4C L	92 5C \	108 6C l	124 7C
13 0D CR '\r'	29 1D GS	45 2D -	61 3D =	77 4D M	93 5D]	109 6D m	125 7D }
14 0E SO	30 1E RS	46 2E .	62 3E >	78 4E N	94 5E ^	110 6E n	126 7E ~
15 0F SI	31 1F US	47 2F /	63 3F ?	79 4F O	95 5F _	111 6F o	127 7F DEL