

INF2171 — Organisation des ordinateurs et assembleur

Examen Intra — Automne 2023

Jean Privat — Dimanche 29 octobre — 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 Comment le processeur détermine l'instruction machine à exécuter?

- A** Le registre compteur ordinal (pc) contient l'adresse en mémoire de l'instruction à exécuter.
- B** Le processeur connaît la ligne active dans l'éditeur de code assembleur.
- C** Les instructions sont stockés dans l'unité de contrôle et dépilées au fur et à mesure de leur exécution.
- D** Le registre d'index (x0) contient la position de l'instruction dans l'unité de contrôle.
- E** Le registre compteur ordinal (pc) contient la position de l'instruction dans l'unité de contrôle.
- F** Le registre d'index (x0) contient l'adresse en mémoire de l'instruction à exécuter.
- G** L'instruction est en mémoire à l'adresse 0

Question 2 Que signifie le « R » de RISC ?

- | | | |
|---|---|---|
| <input type="checkbox"/> A Read / Lecture | <input type="checkbox"/> D Rules / Règles | <input type="checkbox"/> G Risk / Risque |
| <input type="checkbox"/> B Return / Retour | <input type="checkbox"/> E Register / Registre | <input type="checkbox"/> H Random / Aléatoire |
| <input type="checkbox"/> C Recursion / Récursivité | <input type="checkbox"/> F Relative / Relatif | <input checked="" type="checkbox"/> I Reduced / Réduit |

brouillon

2 Instructions et code machine RISC-V

Question 3 Que fait la directive RISC-V « `.byte 0x39` » ?

- | | |
|--|--|
| <input type="checkbox"/> A Réserve deux octets, initialisés à 63 00 (hexa) | <input type="checkbox"/> F Réserve un seul octet, initialisé à 39 (hexa) |
| <input type="checkbox"/> B Réserve un seul octet, initialisé à 63 (hexa) | <input type="checkbox"/> G Réserve un seul octet, initialisé à 09 (hexa) |
| <input type="checkbox"/> C Réserve 39 octets, tous initialisés à 00 (hexa) | <input type="checkbox"/> H Réserve deux octets, initialisés à 90 00 (hexa) |
| <input type="checkbox"/> D Réserve deux octets, initialisés à 39 00 (hexa) | <input type="checkbox"/> I Réserve 9 octets, tous initialisés à 00 (hexa) |
| <input type="checkbox"/> E Réserve 63 octets, tous initialisés à 00 (hexa) | <input type="checkbox"/> J Réserve deux octets, initialisés à 09 00 (hexa) |

Question 4 Désassemblez l'instruction machine RISC-V codée mémoire par les 4 octets suivants : 93 05 A0 02

- | | | | |
|--|--|--|--|
| <input type="checkbox"/> A <code>sub a1, a2, a2</code> | <input type="checkbox"/> E <code>li a1, -42</code> | <input type="checkbox"/> I <code>li s1, '*'</code> | <input type="checkbox"/> M <code>sub a1, a1, a2</code> |
| <input type="checkbox"/> B <code>sll s1, a1, s1</code> | <input type="checkbox"/> F <code>slli s1, x0, 1</code> | <input type="checkbox"/> J <code>sll a1, a1, s1</code> | <input type="checkbox"/> N <code>li s1, -42</code> |
| <input type="checkbox"/> C <code>addi a1, a1, '*'</code> | <input type="checkbox"/> G <code>add a1, a1, a2</code> | <input type="checkbox"/> K <code>slli s1, s1, 1</code> | <input type="checkbox"/> O <code>add a1, a2, a2</code> |
| <input type="checkbox"/> D <code>li a1, '*'</code> | <input type="checkbox"/> H <code>addi a1, a1, -42</code> | <input type="checkbox"/> L <code>addi s1, s1, -42</code> | <input type="checkbox"/> P <code>addi s1, s1, '*'</code> |

3 Arithmétique des ordinateurs

Question 5 ♣ Parmi les représentations suivantes, quelles sont celles du nombre $41_{(16)}$ (41 hexadécimal) ? Indiquez toutes les bonnes réponses.

- | | | | |
|--|--|--|--|
| <input type="checkbox"/> A $101_{(8)}$ | <input type="checkbox"/> C $33_{(16)}$ | <input type="checkbox"/> E $65_{(10)}$ | <input type="checkbox"/> G $76_{(8)}$ |
| <input type="checkbox"/> B $1000101_{(2)}$ | <input type="checkbox"/> D $1000001_{(2)}$ | <input type="checkbox"/> F $63_{(10)}$ | <input type="checkbox"/> H $29_{(16)}$ |

Question 6 ♣ Parmi les séquences d'instructions RISC-V suivantes, quelles sont celles qui positionnent le registre `s0` à la valeur `0x614` ? Indiquez toutes les bonnes réponses.

- | | | |
|--|--|---|
| <input type="checkbox"/> A <code>li s0, 0x855</code>
<code>addi s0, s0, -0x271</code> | <input type="checkbox"/> C <code>li s0, 0x6BF</code>
<code>xori s0, s0, 0xAB</code> | <input type="checkbox"/> E <code>li s0, 0x9EB</code>
<code>not s0, s0</code> |
| <input type="checkbox"/> B <code>li s0, 0xC25</code>
<code>srai s0, s0, 1</code> | <input type="checkbox"/> D <code>li s0, 0x30A</code>
<code>slli s0, s0, 1</code> | <input type="checkbox"/> F <code>li s0, 0x271</code>
<code>addi s0, s0, 0x3A3</code> |

brouillon

4 Programme mystère

Soit le programme mystère suivant:

```

        li s0, 0
        li s1, -1
aaa:    li a7, 5
        ecall
        blez a0, ccc
        blez s1, bbb
        ble a0, s1, bbb
        slli s1, s1, 1
        sub s0, s0, s1
bbb:    mv s1, a0
        add s0, s0, s1
        j aaa
ccc:    li a7, 1
        mv a0, s0
        ecall
        li a7, 10
        ecall

```

Question 7 Qu'affiche-il lorsque l'on saisit le nombre 0 ?

- | | | |
|---|------------------------------|--|
| <input checked="" type="checkbox"/> A 0 | <input type="checkbox"/> E 4 | <input type="checkbox"/> I 8 |
| <input type="checkbox"/> B 1 | <input type="checkbox"/> F 5 | <input type="checkbox"/> J Il n'affiche rien du tout |
| <input type="checkbox"/> C 2 | <input type="checkbox"/> G 6 | <input type="checkbox"/> K Il part en boucle infinie |
| <input type="checkbox"/> D 3 | <input type="checkbox"/> H 7 | <input type="checkbox"/> L Il affiche n'importe quoi |

Question 8 Qu'affiche-il lorsque l'on saisit les nombres 4, 2, 1 et -2 ?

- | | | |
|------------------------------|---|--|
| <input type="checkbox"/> A 0 | <input type="checkbox"/> E 4 | <input type="checkbox"/> I 8 |
| <input type="checkbox"/> B 1 | <input type="checkbox"/> F 5 | <input type="checkbox"/> J Il n'affiche rien du tout |
| <input type="checkbox"/> C 2 | <input type="checkbox"/> G 6 | <input type="checkbox"/> K Il part en boucle infinie |
| <input type="checkbox"/> D 3 | <input checked="" type="checkbox"/> H 7 | <input type="checkbox"/> L Il affiche n'importe quoi |

Question 9 Qu'affiche-il lorsque l'on saisit les nombres 1, 2, 4 et -4 ?

- | | | |
|---|------------------------------|--|
| <input type="checkbox"/> A 0 | <input type="checkbox"/> E 4 | <input type="checkbox"/> I 8 |
| <input checked="" type="checkbox"/> B 1 | <input type="checkbox"/> F 5 | <input type="checkbox"/> J Il n'affiche rien du tout |
| <input type="checkbox"/> C 2 | <input type="checkbox"/> G 6 | <input type="checkbox"/> K Il part en boucle infinie |
| <input type="checkbox"/> D 3 | <input type="checkbox"/> H 7 | <input type="checkbox"/> L Il affiche n'importe quoi |

Question 10 Dans ce programme, par quelle instruction unique pourrait-on remplacer les deux instructions « blez s1, bbb » et « ble a0, s1, bbb » des lignes 6 et 7 ?

- | | |
|--|---|
| <input type="checkbox"/> A bgt a0, s1, ccc | <input type="checkbox"/> E blez a0, s1, bbb |
| <input checked="" type="checkbox"/> B bleu a0, s1, bbb | <input type="checkbox"/> F bgt a0, s1, aaa |
| <input type="checkbox"/> C ble a0, s1, x0, bbb | <input type="checkbox"/> G bgtz a0, s1, aaa |
| <input type="checkbox"/> D bge a0, s1, aaa | <input type="checkbox"/> H Aucune de ces réponses |

brouillon

5 Modes d'adressage

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

```
.data
.equiv bar, 0x80
foo:  .byte bar, 0x00, 0x10, 0x01
baz:  .dword foo

.text
la s0, foo
lw s1, 0(s0)
li s2, bar
lb s3, 0(s0)
ld s4, 8(s0)
```

Question 11 Le registre `s0`.

- | | | | |
|---|--|---|---|
| <input type="checkbox"/> A 0x0000000000000110 | <input type="checkbox"/> E 0x0000000008000110 | <input type="checkbox"/> I 0x0000011000000000 | <input type="checkbox"/> M 0x0800100100000000 |
| <input type="checkbox"/> B 0x0000000000001001 | <input type="checkbox"/> F 0x0000000008001001 | <input type="checkbox"/> J 0x0000100100000000 | <input type="checkbox"/> N 0x1001000000000000 |
| <input type="checkbox"/> C 0x000000001100000 | <input checked="" type="checkbox"/> G 0x0000000010010000 | <input type="checkbox"/> K 0x0110000000000000 | <input type="checkbox"/> O 0x1001000800000000 |
| <input type="checkbox"/> D 0x000000001100080 | <input type="checkbox"/> H 0x0000000010010008 | <input type="checkbox"/> L 0x0110008000000000 | <input type="checkbox"/> P 0x8000011000000000 |

Question 12 Le registre `s1`.

- | | | | |
|---|---|---|---|
| <input type="checkbox"/> A 0x0000000000000110 | <input type="checkbox"/> E 0x0000000008000110 | <input type="checkbox"/> I 0x0000011000000000 | <input type="checkbox"/> M 0x0800100100000000 |
| <input type="checkbox"/> B 0x0000000000001001 | <input type="checkbox"/> F 0x0000000008001001 | <input type="checkbox"/> J 0x0000100100000000 | <input type="checkbox"/> N 0x1001000000000000 |
| <input type="checkbox"/> C 0x000000001100000 | <input type="checkbox"/> G 0x0000000010010000 | <input type="checkbox"/> K 0x0110000000000000 | <input type="checkbox"/> O 0x1001000800000000 |
| <input checked="" type="checkbox"/> D 0x000000001100080 | <input type="checkbox"/> H 0x0000000010010008 | <input type="checkbox"/> L 0x0110008000000000 | <input type="checkbox"/> P 0x8000011000000000 |

Question 13 Le registre `s2`.

- | | | | |
|--|--|---|---|
| <input type="checkbox"/> A 0x0000000000000001 | <input type="checkbox"/> E 0x0100000000000000 | <input type="checkbox"/> I 0x1000000000000000 | <input type="checkbox"/> M 0xFFFFFFFFFFFFFF10 |
| <input type="checkbox"/> B 0x0000000000000008 | <input type="checkbox"/> F 0x01FFFFFFFFFFFFFFF | <input type="checkbox"/> J 0x1FFFFFFFFFFFFFFF | <input type="checkbox"/> N 0xFFFFFFFFFFFFFF80 |
| <input type="checkbox"/> C 0x0000000000000010 | <input type="checkbox"/> G 0x0800000000000000 | <input type="checkbox"/> K 0x8000000000000000 | <input type="checkbox"/> O 0xFFFFFFFFFFFFFFF1 |
| <input checked="" type="checkbox"/> D 0x0000000000000008 | <input type="checkbox"/> H 0x08FFFFFFFFFFFFFFF | <input type="checkbox"/> L 0x8FFFFFFFFFFFFFFF | <input type="checkbox"/> P 0xFFFFFFFFFFFFFFF8 |

Question 14 Le registre `s3`.

- | | | | |
|---|--|---|--|
| <input type="checkbox"/> A 0x0000000000000001 | <input type="checkbox"/> E 0x0100000000000000 | <input type="checkbox"/> I 0x1000000000000000 | <input type="checkbox"/> M 0xFFFFFFFFFFFFFF10 |
| <input type="checkbox"/> B 0x0000000000000008 | <input type="checkbox"/> F 0x01FFFFFFFFFFFFFFF | <input type="checkbox"/> J 0x1FFFFFFFFFFFFFFF | <input checked="" type="checkbox"/> N 0xFFFFFFFFFFFFFF80 |
| <input type="checkbox"/> C 0x0000000000000010 | <input type="checkbox"/> G 0x0800000000000000 | <input type="checkbox"/> K 0x8000000000000000 | <input type="checkbox"/> O 0xFFFFFFFFFFFFFFF1 |
| <input type="checkbox"/> D 0x0000000000000008 | <input type="checkbox"/> H 0x08FFFFFFFFFFFFFFF | <input type="checkbox"/> L 0x8FFFFFFFFFFFFFFF | <input type="checkbox"/> P 0xFFFFFFFFFFFFFFF8 |

Question 15 Le registre `s4`.

- | | | | |
|---|--|---|---|
| <input type="checkbox"/> A 0x0000000000000110 | <input type="checkbox"/> E 0x0000000008000110 | <input type="checkbox"/> I 0x0000011000000000 | <input type="checkbox"/> M 0x0800100100000000 |
| <input type="checkbox"/> B 0x0000000000001001 | <input type="checkbox"/> F 0x0000000008001001 | <input type="checkbox"/> J 0x0000100100000000 | <input type="checkbox"/> N 0x1001000000000000 |
| <input type="checkbox"/> C 0x000000001100000 | <input checked="" type="checkbox"/> G 0x0000000010010000 | <input type="checkbox"/> K 0x0110000000000000 | <input type="checkbox"/> O 0x1001000800000000 |
| <input type="checkbox"/> D 0x000000001100080 | <input type="checkbox"/> H 0x0000000010010008 | <input type="checkbox"/> L 0x0110008000000000 | <input type="checkbox"/> P 0x8000011000000000 |

6 Programmation

Soit le programme incomplet suivant dont l'objectif est d'afficher l'élément maximum d'un tableau (max) ainsi que le nombre de fois que ce maximum apparaît (cpt).

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 « 5 2 ». Si l'on remplaçait le premier mot de 'tab' par 6 le programme afficherait « 6 1 »

```

.data
# Tableau à traiter
tab: .word 1, 3, 2, 3, 3, 5, 3, 5
      .eqv tablen, 8 # Nombre d'éléments
      # Appels système RARS utilisés
      .eqv PrintInt, 1
      .eqv PrintChar, 11
      .eqv Exit, 10
.text
la s0, tab
TODO1          # meilleur maximum
li s2, 1       # nombre d'occurrence
# for(s3=1; s3<tablen; s3++) {
li s3, 1      # index courant
loop: li t0, tablen
      TODO2
      # récupérer l'élément courant dans s4
      slli s4, s3, 2
      TODO3
      # comparaisons avec max courant
      blt s4, s1, cont
      beq s4, s1, inc
      # nouveau max trouvé
      TODO4
inc:  addi s2, s2, 1
cont: # élément suivant
      TODO5      # }
fin:  # Affichage des résultats
      li a7, PrintInt
      mv a0, s1
      ecall
      li a7, PrintChar
      li a0, ' '
      ecall
      li a7, PrintInt
      mv a0, s2
      ecall
      li a7, Exit
      ecall

```

Question 16 Que faut-il mettre à la place du TODO1 ?

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

Question 17 Que faut-il mettre à la place du TODO2 ?

- | | | | |
|---|---|--|---|
| <input type="checkbox"/> A blez s3, tablen | <input type="checkbox"/> E bge s3, t0, tablen | <input type="checkbox"/> I bgez s3, tablen | <input type="checkbox"/> M blez s3, tab |
| <input type="checkbox"/> B ble s3, t0, cont | <input type="checkbox"/> F bge s3, t0, cont | <input type="checkbox"/> J ble s3, t0, fin | <input type="checkbox"/> N bgez s3, inc |
| <input type="checkbox"/> C bge s3, t0, inc | <input type="checkbox"/> G bge s3, t0, tab | <input type="checkbox"/> K ble s3, t0, inc | <input type="checkbox"/> O ble s3, t0, tab |
| <input checked="" type="checkbox"/> D bge s3, t0, fin | <input type="checkbox"/> H blez s3, cont | <input type="checkbox"/> L bgez s3, cont | <input type="checkbox"/> P ble s3, t0, tablen |

Question 18 Que faut-il mettre à la place du TODO3 ?

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

Question 19 Que faut-il mettre à la place du TODO4 ?

- | | | | |
|---|---|---|---|
| <input type="checkbox"/> A add s4, s4, s1
li s2, 1 | <input checked="" type="checkbox"/> E mv s1, s4
li s2, 0 | <input type="checkbox"/> I mv s2, s4
li s1, 1 | <input type="checkbox"/> M mv s4, s2
li s1, 0 |
| <input type="checkbox"/> B add s2, s2, s4
li s1, 0 | <input type="checkbox"/> F mv s4, s1
li s2, 1 | <input type="checkbox"/> J add s1, s1, s4
li s2, 1 | <input type="checkbox"/> N add s4, s4, s1
li s2, 0 |
| <input type="checkbox"/> C add s4, s4, s2
li s1, 1 | <input type="checkbox"/> G add s1, s1, s4
li s2, 0 | <input type="checkbox"/> K mv s2, s4
li s1, 0 | <input type="checkbox"/> O mv s4, s1
li s2, 0 |
| <input type="checkbox"/> D mv s1, s4
li s2, 1 | <input type="checkbox"/> H add s4, s4, s2
li s1, 0 | <input type="checkbox"/> L add s2, s2, s4
li s1, 1 | <input type="checkbox"/> P mv s4, s2
li s1, 1 |

Question 20 Que faut-il mettre à la place du TODO5 ?

- | | | | |
|--|--|--|--|
| <input type="checkbox"/> A addi s3, s3, 4
j loop | <input type="checkbox"/> E addi s2, s2, 1
j inc | <input type="checkbox"/> I addi s2, s2, -4
j loop | <input checked="" type="checkbox"/> M addi s3, s3, 1
j loop |
| <input type="checkbox"/> B addi s3, s3, -4
j loop | <input type="checkbox"/> F addi s3, s3, -1
j loop | <input type="checkbox"/> J addi s3, s3, 1
j inc | <input type="checkbox"/> N addi s3, s3, 4
j inc |
| <input type="checkbox"/> C addi s2, s2, -1
j loop | <input type="checkbox"/> G addi s2, s2, 4
j inc | <input type="checkbox"/> K addi s3, s3, -4
j inc | <input type="checkbox"/> O addi s2, s2, -1
j inc |
| <input type="checkbox"/> D addi s2, s2, 4
j loop | <input type="checkbox"/> H addi s2, s2, 1
j loop | <input type="checkbox"/> L addi s2, s2, -4
j inc | <input type="checkbox"/> P addi s3, s3, -1
j inc |

brouillon

brouillon



Feuille de réponses intra automne 2023 INF2171 — Organisation des ordinateurs et assembleur

0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1
2	2	2	2	2	2	2	2	2
3	3	3	3	3	3	3	3	3
4	4	4	4	4	4	4	4	4
5	5	5	5	5	5	5	5	5
6	6	6	6	6	6	6	6	6
7	7	7	7	7	7	7	7	7
8	8	8	8	8	8	8	8	8
9	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.
Important : noircissez complètement l'intérieur de chaque case (pas de croix, pas de cercle).*

- Question 1 : A B C D E F G
- Question 2 : A B C D E F G H I
- Question 3 : A B C D E F G H I J
- Question 4 : A B C D E F G H I J K L M N O P
- Question 5 : A B C D E F G H
- Question 6 : A B C D E F
- Question 7 : A B C D E F G H I J K L
- 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
- 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 M N O P
- 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 I J K L M N O P
- 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	Substract	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]	
srlr	Shift Right Logical Imm	I	0x13	0x5	imm[11:6]=0x00	rd = rs1 >> imm[5:0]	
srair	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)

Codes ASCII importants

Hex	Dec	Caractère
0x00	0	NUL (fin de chaîne)
0x0A	10	Saut de ligne '\n'
0x20	32	Espace ' '
0x2A	42	Étoile '*'
0x30	48	Premier chiffre '0'
0x41	65	Première majuscule 'A'
0x61	97	Première minuscule 'a'

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	$\approx -1.70 \times 10^{38}$	$\approx 1.70 \times 10^{38}$