# 2022 年度 「コンピュータアーキテクチャ」定期試験 問題
(担当: 石浦菜岐佐)

## 試験開始までこの面を上にして待つこと

- 試験時間は 70 分で, 持ち込みは一切不可である.
- 問題は全部で 4 問あり 65 点満点である.
- 解答用紙の所定の欄に解答せよ.

**採点結果の閲覧** 本試験は, 採点が終り次第, 結果を WWW で閲覧できるようにします.

- 閲覧を希望する人は, 解答用紙の「(1) 好きな数字 4 桁」と「(2) 勉強時間」の欄に, それぞれ 4 桁の数字を記入し, その下にマークして下さい.
  - (2) はこの定期試験のための勉強時間で, 例えば, **8 時間 25 分であれば 0825** のように, 時間を 2 桁, 分を 2 桁で表現して下さい. 統計をホームページで公表する予定です.
  - (1)(2) の数字は閲覧時のキーとして必要になるので, 記憶に自信のない人は下に控えを取っておいて下さい.
  - なお, 次の場合には閲覧ができなくなるので注意して下さい.
    - (2) の数字が勉強時間を表していないと判断される場合
    - マークがない/薄い場合, マークにミスがある場合, 数字とマークが一致しない場合

控え ⇒

| (1) 好きな数字 4 桁 | | | | (2) 勉強時間 | | | |
|---|---|---|---|---|---|---|---|
| | | | | 時間 | | 分 | |
| | | | | | | | |

- 閲覧ページは講義ページ (`http://ist.ksc.kwansei.ac.jp/~ishiura/arc/`) からリンクします.
  - ページの認証は, 中間試験と同じです.
  - 表示されるフォームに (1)(2) の数字を入力して下さい. 数字を忘れた場合は閲覧できません. また, メールなどによる照会には一切応じません.
  - 閲覧の期限は 2/10(月) です.
- 閲覧を希望しない人は, (1)(2) を空欄にしておいて下さい.

付録: 演算の記法

| | | |
|---|---|---|
| `*` … 乗算 | `/` … 整数除算 | `%` … 整数剰余 |
| `<<`$_{\mathtt{L}}$ … 論理左シフト | `<<`$_{\mathtt{A}}$ … 算術左シフト | `>>`$_{\mathtt{L}}$ … 論理右シフト |
| `>>`$_{\mathtt{A}}$ … 算術右シフト | `&` … ビット毎の論理積 | `\|` … ビット毎の論理和 |
| `~` … ビット毎の論理否定 | $\oplus$ … ビット毎の排他的論理和 | `\|\|` … 連結 |
| $c$`?`$a$`:`$b$ … $c$ が真なら $a$, 偽なら $b$ | *sx(f)* … $f$ の 32bit への符号拡張 | *zx(f)* … $f$ の 32bit へのゼロ拡張 |

`Mem[`$a$`,`$b$`]` … 主記憶の $a$ 番地から始まる $b$ バイトのデータ
`RF[`$k$`]` … レジスタファイルの $k$ 番目のレジスタ
$R_{x:y}$ … レジスタ $R$ の $x \sim y$ ビット目

## 付録: MIPS の命令セット (抜粋)

(1) R-type

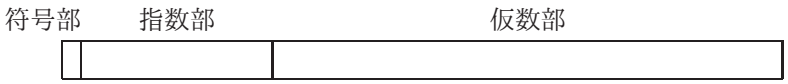
| ニモニック | 機械語 31 26 25 21 20 16 15 11 10 6 5 0 | | | | | | 動作 |
|---|---|---|---|---|---|---|---|
| add $Rd,Rs,Rt$ | 000000 | $Rs$ | $Rt$ | $Rd$ | 00000 | 100000 | $Rd$=$Rs$+$Rt$, PC=PC+4 |
| addu $Rd,Rs,Rt$ | 000000 | $Rs$ | $Rt$ | $Rd$ | 00000 | 100001 | $Rd$=$Rs$+$Rt$, PC=PC+4 |
| sub $Rd,Rs,Rt$ | 000000 | $Rs$ | $Rt$ | $Rd$ | 00000 | 100010 | $Rd$=$Rs$−$Rt$, PC=PC+4 |
| subu $Rd,Rs,Rt$ | 000000 | $Rs$ | $Rt$ | $Rd$ | 00000 | 100011 | $Rd$=$Rs$−$Rt$, PC=PC+4 |
| and $Rd,Rs,Rt$ | 000000 | $Rs$ | $Rt$ | $Rd$ | 00000 | 100100 | $Rd$=$Rs$&$Rt$, PC=PC+4 |
| or $Rd,Rs,Rt$ | 000000 | $Rs$ | $Rt$ | $Rd$ | 00000 | 100101 | $Rd$=$Rs$ \| $Rt$, PC=PC+4 |
| xor $Rd,Rs,Rt$ | 000000 | $Rs$ | $Rt$ | $Rd$ | 00000 | 100110 | $Rd$=$Rs$⊕$Rt$, PC=PC+4 |
| nor $Rd,Rs,Rt$ | 000000 | $Rs$ | $Rt$ | $Rd$ | 00000 | 100111 | $Rd$=~($Rs$ \| $Rt$), PC=PC+4 |
| sll $Rd,Rt,Sa$ | 000000 | 00000 | $Rt$ | $Rd$ | $Sa$ | 000000 | $Rd$=$Rt$<<$_L$ $Sa$, PC=PC+4 |
| srl $Rd,Rt,Sa$ | 000000 | 00000 | $Rt$ | $Rd$ | $Sa$ | 000010 | $Rd$=$Rt$>>$_L$ $Sa$, PC=PC+4 |
| sra $Rd,Rt,Sa$ | 000000 | 00000 | $Rt$ | $Rd$ | $Sa$ | 000011 | $Rd$=$Rt$>>$_A$ $Sa$, PC=PC+4 |
| slt $Rd,Rs,Rt$ | 000000 | $Rs$ | $Rt$ | $Rd$ | 00000 | 101010 | $Rd$=($Rs$<$Rt$)?1:0 (符号付き比較), PC=PC+4 |

(2) I-type

| ニモニック | 機械語 31 26 25 21 20 16 15 0 | | | | 動作 |
|---|---|---|---|---|---|
| addi $Rt,Rs,Imm$ | 001000 | $Rs$ | $Rt$ | $Imm$ | $Rt$=$Rs$+$sx$($Imm$), PC=PC+4 |
| addiu $Rt,Rs,Imm$ | 001001 | $Rs$ | $Rt$ | $Imm$ | $Rt$=$Rs$+$sx$($Imm$), PC=PC+4 |
| andi $Rt,Rs,Imm$ | 001100 | $Rs$ | $Rt$ | $Imm$ | $Rt$=$Rs$&$zx$($Imm$), PC=PC+4 |
| ori $Rt,Rs,Imm$ | 001101 | $Rs$ | $Rt$ | $Imm$ | $Rt$=$Rs$ \| $zx$($Imm$), PC=PC+4 |
| xori $Rt,Rs,Imm$ | 001110 | $Rs$ | $Rt$ | $Imm$ | $Rt$=$Rs$⊕$zx$($Imm$), PC=PC+4 |
| slti $Rt,Rs,Imm$ | 001010 | $Rs$ | $Rt$ | $Imm$ | $Rt$=($Rs$<$sx$($Imm$))?1:0 (符号付き比較), PC=PC+4 |
| lw $Rt,Imm(Rs)$ | 100011 | $Rs$ | $Rt$ | $Imm$ | $Rt$=Mem[$Rs$+$sx$($Imm$),4], PC=PC+4 |
| sw $Rt,Imm(Rs)$ | 101011 | $Rs$ | $Rt$ | $Imm$ | Mem[$Rs$+$sx$($Imm$),4]=$Rt$, PC=PC+4 |
| beq $Rt,Rs,Imm$ | 000100 | $Rs$ | $Rt$ | $Imm$ | PC=($Rt$==$Rs$) ? (PC+4)+$sx$($Imm$)*4 : (PC+4) |
| bne $Rt,Rs,Imm$ | 000101 | $Rs$ | $Rt$ | $Imm$ | PC=($Rt$!=$Rs$) ? (PC+4)+$sx$($Imm$)*4 : (PC+4) |

## 付録: IEEE 754 の単精度浮動小数点表現

- フォーマット




| | |
|---|---|
| 符号部 | 1 ビット |
| 指数部 | 8 ビットで, バイアス値 127 のバイアス表現; 00000000 と 11111111 は特殊用途に使用 |
| 仮数部 | 23 ビットで, 隠しビットを使用 |

- 正規化数以外の表現

| | |
|---|---|
| ゼロ | 指数部が 00000000 で仮数部が 0000···0 のとき, 符号部が正なら $+0$ を, 負なら $-0$ を表す |
| 非正規化数 | 指数部が 00000000 で仮数部が 0000···0 以外のとき, 仮数部を $m$ とすると $0.m \times 2^{-126}$ を表す |
| 無限大 | 指数部が 11111111 で仮数部が 0000···0 のとき, 符号部が正なら $+\infty$ を, 負なら $-\infty$ を表す |
| NaN | 指数部が 11111111 で仮数部が 0000···0 以外のとき |

1 次の各問に答えよ. なお, (7)(8) 以外には原則として部分点は与えない.

(1) MIPS の機械語 `3325FFF1` (16 進表現) に対するアセンブリを示せ. 答を導出する過程も示すこと. なお, `addi`, `lw`, `sw`, `beq`, `bne` 命令の $Imm$ フィールドの値は必ず符号付き整数として解釈すること. [4 点]

(2) 左下の C プログラムの断片をコンパイルした結果, 右下の MIPS アセンブリプログラムが得られるとする. 空欄に入る命令を示せ. ただし, `int` 型は 32 ビットの符号付き整数, `unsigned int` 型は 32 ビットの符号なし整数として扱うものとする. また, 変数 `a`, `y` の番地はレジスタ `$30` に格納された基底番地にそれぞれ 0, 4 を加算して得られるものとする. [4 点]

```
unsigned int a, y;
  ...
y = a % 128;
```

⇒

```
lw   $11,0($30)
     空欄
sw   $12,4($30)
```

(3) 左下の C プログラムの断片をコンパイルした結果, 右下の MIPS アセンブリプログラムが得られるとする. 空欄に入る命令を示せ. ただし, `int` 型は 32 ビットの符号付き整数, `unsigned int` 型は 32 ビットの符号なし整数として扱うものとする. また, 変数 `a`, `y` の番地はレジスタ `$30` に格納された基底番地にそれぞれ 0, 4 を加算して得られるものとする. [4 点]

```
int a, y;
  ...
y = a * 40;
```

⇒

```
lw   $11,0($30)
sll  $12,$11,2
add  $12,$12,$11
     空欄
sw   $12,4($30)
```

(4) 10 進数

$$-0.4140625$$

の IEEE754 単精度 2 進浮動小数点表現を求めよ. 答は 16 進表現で示せ. 必ず答を導出する過程も示すこと. [4 点]

(5) $a = 1.010 \times 2^{-4}$, $b = 1.111 \times 2^{-6}$ とする. ただし, 両数の仮数部は 2 進表現である. このとき, $a - b$ の値を求めよ. 結果は正規化して小数点以下 3 桁に丸めよ. 丸めには偶数丸めを用いよ. 必ず答を導出する過程も示すこと. [4 点]

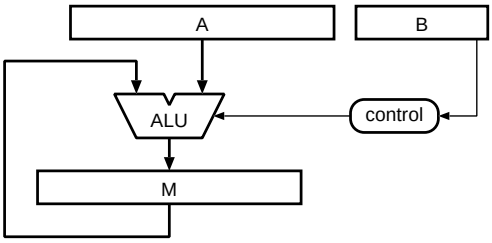
(6) 論理アドレス空間 64KB のコンピュータに, 容量 16KB, ブロックサイズ 32B の2-wayセットアソシアティブ方式のキャッシュが搭載されているとする. 次の 1~8 の順に主記憶の番地 (16 進数) にアクセスがあったとき, キャッシュミスは何番目のアクセスで発生するか (「1, 2, 5, 7 番目」のように解答せよ). ただし, ブロック置き換えアルゴリズムにはLRUを用いるものとする. また, メモリーはバイト単位でアクセスされるものとする. 必ず答を導出する過程も示すこと. [4 点]

`1: 7014,  2: 5008,  3: 601C,  4: 7018,`
`5: 400C,  6: 6004,  7: 9018,  8: 7004`

(7) スーパスケーラに比べて VLIW の方が消費電力が小さい理由を簡潔に述べよ. [4 点]

(8) コンピュータシステムや組込みシステムをマルチプロセッサ/マルチコアにより構成することによってシステムの信頼性を向上させる手法を簡潔に説明せよ.
[4 点]

(9) 左下の図はシフトと加算の繰り返しにより符号無し整数の乗算を行う回路の例である. A は $2n$ ビット, B は $n$ ビット, M は $2n$ ビットのレジスタであり, 右下のアルゴリズムにより, $n$ ビットの符号無し整数 $a$ と $b$ の乗算結果を求めることができる. 空欄 (a) と (b) に入る式を示せ. [4 点]




```
A=a, B=b, M=0;    /* 初期設定 */
for (i=0; i<n; i++) {
   if ((B&1)==1)) { M=M+A; }
   A = [ (a) ];
   B = [ (b) ];
}
/* M に a と b の乗算結果が得られる */
```

(10) 左下の C プログラムの断片をコンパイルした結果, 右下の MIPS アセンブリプログラムが得られるとする. 空欄に入る命令を示せ. ただし, int 型は 32 ビットの符号付き整数, unsigned int 型は 32 ビットの符号なし整数として扱うものとする. また, 変数 i, s, n の番地はレジスタ $30 に格納された基底番地にそれぞれ 0, 4, 8 を加算して得られるものとする. [4 点]

```
int i, s, n;
  ...
s = 0;
i = 1;
while (i!=n) {
  s = s + i;
  i = i + 1;
}
  ...
```

⇒

```
  ...
addi   $3,$0,0
addi   $2,$0,1
lw     $4,8($30)
[ 空欄 ]
sll    $0,$0,0   ; nop
add    $3,$3,$2
addi   $2,$2,1
beq    $0,$0,-5
sll    $0,$0,0   ; nop
sw     $2,0($30)
sw     $3,4($30)
  ...
```

2 下表は, 5 ステージの命令実行制御を行う MIPS の各ステージの動作を示したものである. 空欄に入るべき適切な式を示せ. [8 点] (各 1 点)

| 命令 | IF | ID | EX | MEM | WB |
|---|---|---|---|---|---|
| add $Rd, Rs, Rt$ | | | Y = A+B | PC = NPC | RF[ ⑦ ] = Y |
| addi $Rt, Rs, Imm$ | IR = Mem[ ① ,4] | A = RF[ ③ ] | Y = A+I | PC = NPC | RF[ ⑧ ] = Y |
| lw $Rt, Imm(Rs)$ | NPC = ② | B = RF[ ④ ] | Y = A+I | PC = NPC, M = Mem[Y,4] | RF[$IR_{20:16}$] = M |
| sw $Rt, Imm(Rs)$ | | I = $sx(IR_{15:0})$ | Y = A+I | PC = NPC, Mem[Y,4] = ⑥ | |
| beq $Rt, Rs, Imm$ | | | Y = ⑤ | PC = Y ? NPC+I<<$_L$2 : NPC | |

3 下表に示すような IF, ID, EX, MEM, WB の 5 ステージパイプラインの命令実行制御を行う MIPS において, 次のプログラムを実行するものとする. 下記の (1) ~ (3) に答えよ. 必ず答を導出する過程を示すこと. 特に, 各命令の各ステージが実行されるタイミングを図示する際に, **ステージ間の順序関係の制約全てを矢印で明示すること**.

```
1:  lw    $5,24($30)
2:  addi  $12,$5,0
3:  lw    $6,0($12)
4:  addi  $30,$30,-12
5:  addi  $22,$22,16
6:  addi  $12,$12,4
```

| 命令 | IF | ID | EX | | MEM | | WB |
|---|---|---|---|---|---|---|---|
| add $Rd, Rs, Rt$ | $IR^1$ = Mem[ ? ,4] | $IR^2 = IR^1$ | $IR^3 = IR^2$ | $Y^3 = A^2 + B^2$ | $IR^4 = IR^3$ | | RF[ ? ] = $Y^4$ |
| addi $Rt, Rs, Imm$ | $NPC^1$ = ? | $NPC^2 = NPC^1$ | $NPC^3 = NPC^2$ | $Y^3 = A^2 + I^2$ | | | RF[ ? ] = $Y^4$ |
| lw $Rt, Imm(Rs)$ | PC = ? | | | $Y^3 = A^2 + I^2$ | $Y^4 = Y^3$ | $M^4$ = Mem[$Y^3$,4] | $RF[IR^4{}_{20:16}] = M^4$ |
| sw $Rt, Imm(Rs)$ | | $A^2$ = RF[ ? ] | | $Y^3 = A^2 + I^2$ | | Mem[$Y^3$,4] = ? | |
| beq $Rt, Rs, Imm$ | | $B^2$ = RF[ ? ]<br>$I^2 = sx(IR^1{}_{15:0})$ | $B^3 = B^2$<br>$I^3 = I^2$ | $Y^3$ = ? | | if ($Y^3$)<br>PC = $NPC^3 + I^3$<< 2 | |

- 各ステージは 1 クロックで実行されるものとする.
- レジスタファイルは, 1 つの書き込みと 2 つの読み出しが 1 クロックで同時に行えるものとする. ただし, 書き込んだ値は次のクロックではじめて読み出せるものとする.
- メモリーの読み書きは 1 クロックで行えるが, 同じメモリーに対して複数のアクセスを同一のクロックでは行えないものとする.

(1) 次のような条件の場合, プログラムの実行に何クロックを要するか. [4 点]

- 命令とデータが同じメモリー Mem に格納されていて, 命令フェッチとデータの読み書きが同じクロックでは行えなない.
- フォワーディング回路 (バイパス回路) はない.
- 命令の順序変更は行わない.

(2) 次のような条件の場合, プログラムの実行に何クロックを要するか. [4 点]

- 命令とデータが別々のメモリーに格納されており, 命令フェッチとデータの読み書きが同じクロックサイクルで行える.
- 可能な限りのフォワーディング回路 (バイパス回路) が実装されている.
- 命令の順序変更は行わない.

(3) (2) の実装において, 命令の順序をどのように変更すれば, プログラムの実行に要するクロック数を最小にできるか. 例えば「1, 3, 2, 4, 5, 6」のように, 命令に付した番号を用いて命令の順序を示せ. [4 点]

4 この授業を通じて勉強したことが, 今後の自分の研究, 仕事, やりたいこと等において, どういう点で役にたちそうかについて記述せよ.

(意味があって自明ではない記述で回答欄の 90%以上が埋まっていて 2 点とする. これを基準に内容に応じて加点/減点する.) [5 点]

Nagisa ISHIURA