

学籍番号 _____ 氏名 _____

1 CASL II の命令の種類

(1) 以下の文章の () に当てはまる適当な語句を①～⑳の中から選びなさい。(各1点)

CASL II の命令は、(ア)と(イ)、(ウ)に分けることができる。(ア)は、アセンブラという変換プログラムに対して、指示を行う命令である。この命令は1と0の組み合わせの(エ)に変換されない。

(イ)は、COMET II の(オ)の動作の指示を行う。(イ)は、アセンブラにより特定のビットパターンを(エ)に変換され、実行時には、そのビットパターンが(カ)に格納される。この命令は、(キ)語または(ク)語から構成される。その1語は(ケ)ビットである。

(ウ)とは、特定の機能を果たすためのいくつかの命令の集まりに名前を付けたものである。

- ① COMET II ② CPU ③ ハードディスク
- ④ マクロ命令 ⑤ 機械語 ⑥ アセンブラ命令
- ⑦ 主記憶装置 ⑧ コンパイラ ⑨ アセンブラ
- ⑩ 機械語命令 ⑪ アドレス ⑫ データ
- ⑬ 0 ⑭ 1 ⑮ 2 ⑯ 4 ⑰ 8 ⑱ 16 ⑳ 32

(ア) _____ (イ) _____ (ウ) _____ (エ) _____ (オ) _____
 (カ) _____ (キ) _____ (ク) _____ (ケ) _____

(2) 以下の命令は、(1)の問いの(ア)(イ)(ウ)のどの命令に属するか?。①～⑩を分類せよ。(各1点)

- ① ST ② DS ③ END ④ RET ⑤ DC
- ⑥ SUBA ⑦ LD ⑧ ADDA ⑨ LAD ⑩ START

(ア)に属する命令 _____
 (イ)に属する命令 _____
 (ウ)に属する命令 _____

2. ハンドアセンブル

(1) ソースプログラムと、それに対応するメインメモリーの内容を(ア)～(オ)を16進数で書け。ただし、メインメモリーは実行後の内容を示すものとする。(各1点)

ソースプログラム(CASL II) メインメモリー
 アドレス 内容(16進数)

| | | | | |
|-----|-------|-------|------|------|
| PGM | START | | 0020 | 1010 |
| | LD | GR1,A | 0021 | (ア) |
| | SUBA | GR1,B | 0022 | (イ) |
| | ST | GR1,C | 0023 | (ウ) |
| | RET | | 0024 | (エ) |
| A | DC | 16 | 0025 | (オ) |
| B | DC | 18 | 0026 | (カ) |
| C | DS | 1 | 0027 | (キ) |
| | END | | 0028 | 0012 |
| | | | 0029 | (ク) |

(ア) (イ) (ウ) (エ)
 (オ) (カ) (キ) (ク)

(2) このプログラム実行開始時のプログラムレジスタ(PR)の値はいくらか?。16進数で答えよ。(1点)

(3) このプログラムの実行完了時のフラグレジスタの値を答えよ。(1点)

3. 命令とアドレス

(1) 以下に対応するCASL IIの命令コードとオペランドを1行で書け。ラベルは書かなくても良い。(各2点)

- a. 実行開始番地をNEXTと指示する。
- b. プログラムの終了を示す。
- c. メモリーの領域を確保して、その値を(FFAB)₁₆と(0012)₁₆、(5F7B)₁₆に設定する。
- d. プログラム実行に必要なメモリー領域として、5語(ワード)確保する。
- e. 主記憶装置の内容をレジスタR1にコピーする。コピー元は、ラベルXが示すアドレスに汎用レジスタR3の内容を加算したアドレスとする。
- f. 汎用レジスタR3の内容をラベルYが示す主記憶装置のアドレスにコピーする。
- g. ラベルZが示すアドレスを、汎用レジスタR4に格納する。
- h. 汎用レジスタR1とR2の内容を算術加算する。加算結果は、R2に残す。
- i. 汎用レジスタR1からラベルAが示す内容を算術減算する。加算結果は、R1に残す。

(2) LAD命令に関して、以下の動作を説明せよ。(各2点)

- a. LAD GR1,3
- b. LAD GR1,1,GR1
- c. LAD GR1,0,GR2

(3) COMET II 主記憶装置に関して、以下の問いに答えよ。

- a. 番地当たりの情報量は、何ビットか?。(各2点)
- b. 主記憶装置のアドレスは何ビットで表現されるか?。
- c. 主記憶装置のアドレスの範囲を答えよ。

4. レジスタの値

(1) 以下のプログラムの場合、それぞれの実行段階でのレジスタの値(10進数)を答えよ。(汎用レジスタ3個で1点。フラグレジスタ3個で1点)

```

PGM   START
LD     GR0, A
LD     GR1, =-1
LAD    GR2, -10, GR1
SUBA   GR0, B
LD     GR2, A
LD     GR1, A, GR2
ADDA   GR1, =33
RET
A      DC      3
B      DC      11, 22, -33, 44
END
  
```

解答は下表に書け。`-`がある欄は書かなくて良い

| 実行段階 | 汎用レジスタ | | | フラグレジスタ | | |
|------|--------|-----|-----|---------|----|----|
| | GR0 | GR1 | GR2 | OF | SF | ZF |
| ① | | - | - | | | |
| ② | | | - | | | |
| ③ | | | | | | |
| ④ | | | | | | |
| ⑤ | | | | | | |
| ⑥ | | | | | | |
| ⑦ | | | | | | |

(2) 算術加算(ADDA)または算術減算(SUBA)の命令を実行した場合のフラグレジスタの設定についての以下の問いに答えよ。(各2点)

a. フラグレジスタ OF=1 になる場合について説明せよ。

b. フラグレジスタ SF=1 になる場合について説明せよ。

c. フラグレジスタ ZF=1 になる場合について説明せよ。

5. プログラムの作成

(1) 和を計算するプログラムを作成せよ。条件は以下の通り。(10点)

- 128+256 を計算する。
- 128 と 256 は、実行前に予め主記憶装置に格納しておく。
- 計算結果の 384 も主記憶装置に格納する。

(2) 指標レジスタを使うプログラムを作成せよ。条件は、以下の通り。(5点)

- 計算に使うデータの並び(2, 4, 6, 8, 10)は、以下の命令を用いて、データを格納する。ラベル名は、DATA です。

```
DATA DC 2, 4, 6, 8, 10
```

- この DATA の 1 番目と 3 番目と 5 番目のデータ、を加算するプログラムを作成する。即ち、2+6+8 を計算する。
- 計算結果は、主記憶装置に格納すること。
- 指標レジスタを用いて、先の加算を計算するプログラムを作成せよ。

6. 応用問題

この問題は難しいが、配点は低いので最後に考えること。

(1) 以下のプログラムを実行させると、アドレス ANS の値は
どうなるか。16 進数で答えよ。(ヒント: 文字は 1 ワー
ドの下位 8 ビットに格納される) (3 点)

```
PGM   START
LD     GR0, AKITA
LD     GR1, =2
ADDA   GR0, AKITA, GR1
LD     GR1, =4
ADDA   GR0, AKITA, GR1
ST     GR0, ANS
RET
AKITA DC   'AKITA'
ANS   DS   1
END
```

(2) 以下のプログラムの動作を説明せよ。実行するとどうなるか?。ただし、このプログラムを実行する場合、NEXT=#0020 とする。(3 点)

```
PGM   START   NEXT
NEXT  DC       #1000
      DC       #0027
      DC       #2000
      DC       #0028
      DC       #1100
      DC       #0029
      DC       #8100
      DC       #0001
      DC       #0001
      DC       #0000
      END
```

表1 JIS X0201の一部

| 下位 4ビット | 上位4ビット | | | | | |
|------------|--------|---|---|---|---|---|
| | 2 | 3 | 4 | 5 | 6 | 7 |
| 0 | 間隔 | 0 | @ | P | ` | p |
| 1 | ! | 1 | A | Q | a | q |
| 2 | ” | 2 | B | R | b | r |
| 3 | # | 3 | C | S | c | s |
| 4 | \$ | 4 | D | T | d | t |
| 5 | % | 5 | E | U | e | u |
| 6 | & | 6 | F | V | f | v |
| 7 | ' | 7 | G | W | g | w |
| 8 | (| 8 | H | X | h | x |
| 9 |) | 9 | I | Y | i | y |
| 10 | * | : | J | Z | j | z |
| 11 | + | ; | K | [| k | { |
| 12 | , | < | L | \ | l | |
| 13 | - | = | M |] | m | } |
| 14 | . | > | N | ^ | n | ~ |
| 15 | / | ? | O | _ | o | |

表2 命令語の構成。

命令語の構成は定義しないが、次のような構成を想定する。ここで、OPの数値は16進表示で示す。

15 11 7 3 0 15 0 ←ビット番号

| 第1語 | | 第2語 | | 命令語長 | 命令語とアセンブラとの対応 | | |
|-------------|-----|------|------|------|---------------|--------------|------------------------|
| 主OP | 副OP | r/r1 | x/r2 | | adr | 機械語命令 | 意味 |
| 0 | 0 | - | - | - | 1 | NOP | no operation |
| 1 | 0 | | | | 2 | LD r,adr,x | load |
| | 1 | | | | 2 | ST r,adr,x | store |
| | 2 | | | | 2 | LAD r,adr,x | load address |
| | 4 | | | - | 1 | LD r1,r2 | load |
| 2 | 0 | | | | 2 | ADDA r,adr,x | add arithmetic |
| | 1 | | | | 2 | SUBA r,adr,x | subtract arithmetic |
| | 2 | | | | 2 | ADDL r,adr,x | add logical |
| | 3 | | | | 2 | SUBL r,adr,x | subtract logical |
| | 4 | | | - | 1 | ADDA r1,r2 | add arithmetic |
| | 5 | | | - | 1 | SUBA r1,r2 | subtract arithmetic |
| | 6 | | | - | 1 | ADDL r1,r2 | add logical |
| 3 | 0 | | | | 2 | AND r,adr,x | and |
| | 1 | | | | 2 | OR r,adr,x | or |
| | 2 | | | | 2 | XOR r,adr,x | exclusive or |
| | 4 | | | - | 1 | AND r1,r2 | and |
| | 5 | | | - | 1 | OR r1,r2 | or |
| | 6 | | | - | 1 | XOR r1,r2 | exclusive or |
| 4 | 0 | | | | 2 | CPA r,adr,x | compare arithmetic |
| | 1 | | | | 2 | CPL r,adr,x | compare logical |
| | 4 | | | - | 1 | CPA r1,r2 | compare arithmetic |
| | 5 | | | - | 1 | CPL r1,r2 | compare logical |
| 5 | 0 | | | | 2 | SLA r,adr,x | shift left arithmetic |
| | 1 | | | | 2 | SRA r,adr,x | shift right arithmetic |
| | 2 | | | | 2 | SLL r,adr,x | shift left logical |
| | 3 | | | | 2 | SRL r,adr,x | shift right logical |
| 6 | 1 | - | | | 2 | JMI adr,x | jump on minus |
| | 2 | - | | | 2 | JNZ adr,x | jump on non zero |
| | 3 | - | | | 2 | JZE adr,x | jump on zero |
| | 4 | - | | | 2 | JUMP adr,x | unconditional jump |
| | 5 | - | | | 2 | JPL adr,x | jump on plus |
| | 6 | - | | | 2 | JOV adr,x | jump on overflow |
| 7 | 0 | - | | | 2 | PUSH adr,x | push |
| | 1 | | - | - | 1 | POP r | pop |
| 8 | 0 | - | | | 2 | CALL adr,x | call subroutine |
| | 1 | - | | - | 1 | RET | return from subroutine |
| 9 ~ E | | | | | | その他の命令 | |
| F | 0 | - | | | 2 | SVC adr,x | supervisor call |

