

# 学年末試験(3E 電子計算機)

2005年3月2日

## 1 プログラミングの基本

### 1.1 機械語命令基本

以下のプログラムの文を書け。プログラム全体を書く必要はなく、答案用紙には1行のみ書く。

- [問1] レジスタ GR1 の内容をレジスタ GR2 へ代入(格納)する。
- [問2] レジスタ GR2 の内容とラベル DATA が示す内容を加算して、その結果をレジスタ GR2 に格納する。
- [問3] レジスタ GR3 の内容とラベル HOGE が示す内容のビット毎の論理和を計算して、その結果をレジスタ GR3 に格納する。
- [問4] レジスタ GR4 とレジスタ GR5 の内容を符号付き整数と考え、その比較を行う。
- [問5] レジスタ GR6 の全てのビットを右へ2ビットシフトさせる。あいたビットにはゼロが入る。
- [問6] フラグレジスタの ZF の値が1のとき、ラベル LOOP に処理が移動する。
- [問7] スタック領域に、ラベル FUGA のアドレスをプッシュする。
- [問8] サブルーチン KASAN を呼び出す。ただし、KASAN はサブルーチンの先頭を表すラベルとする。
- [問4] カウンター GR1 を+1したい(値を1増加させる)。そのためのプログラム文を書け。
- [問5] 汎用レジスタ GR0 の値を-1倍したい。その部分のプログラム文を書け。
- [問6] 次のプログラムは、レジスタ GR1 の符号付き整数の値を9倍する。  に入る適当な文を書け。

```
LD GR2,GR1

ADDA GR1,GR2
```

## 2 プログラムの動作

- [問1] 以下のシフト命令を実行した後、汎用レジスタ GR1 の内容はどうなるか?。16ビットの内容を2進数で書け。(ア)~(エ)の実行前の GR1 の内容は、全て#F0F0とする。
- (ア) SLA GR1,3  
(イ) SRA GR1,3  
(ウ) SLL GR1,3  
(エ) SRL GR1,3
- [問2] 以下のプログラムの各行が実行された直後のフラグレジスタの値を示せ。

```
PGM START ;1行目
LD GR1,AA ;2行目
CPA GR1,BB ;3行目
RET ;4行目
AA DC 2 ;5行目
BB DS 3 ;6行目
END
```

### 1.2 基本テクニック

- [問1] 機械語命令の「算術加算」と「論理加算」の違いを説明せよ。
- [問2] アセンブラ命令を全て書け。
- [問3] GR1 をカウンターとして用いる。GR1 を1に初期化するプログラム文を書け。

### 3 プログラムの構造

右のプログラムについて、問いに答えよ。ただし、このプログラムは、ラベル PGM から実行されると仮定する。

- [問 1] 答案用紙に、このプログラムのメインルーチンとサブルーチン、そしてデータの領域を示せ。サブルーチンが複数ある場合は、サブルーチン 1、サブルーチン 2、… のように示すこと。
- [問 2] サブルーチンの処理の始まりに書かれている PUSH 0,GR1 の役割を述べよ。
- [問 3] サブルーチンの処理の終わりに書かれている POP GR1 の役割を述べよ。
- [問 4] このプログラムの実行後のラベル ODD と MIN が示す内容はどうか？。

```
PGM START
LAD GR1,DATA ;DATA の先頭アドレス
LAD GR2,LAST ;DATA の最終アドレス+1
LAD GR2,-1,GR2
CALL SBODD
ST GR3,ODD
CALL SBMIN
ST GR3,MIN
RET
DATA DC 9,5,6,7,2,1,4,6,3
LAST DS 0
ODD DS 1
MIN DS 1
END
SBODD START
PUSH 0,GR1
PUSH 0,GR4
LAD GR3,0
LOOP1 LD GR4,GR1
AND GR4,MASK
JZE NEXT
LAD GR3,1,GR3
NEXT CPA GR2,GR1
LAD GR1,1,GR1
JPL LOOP1
POP GR4
POP GR1
RET
MASK DC #0001
END
SBMIN START
PUSH 0,GR1
LD GR3,0,GR1 ; 暫定最小
LOOP3 LAD GR1,1,GR1
CPA GR3,0,GR1
JPL SMALL
JUMP NEXT
SMALL LD GR3,0,GR1
NEXT CPA GR2,GR1
JPL LOOP3
POP GR1
RET
END
```

## 4 CASL IIのプログラム例

プログラム中の  に入る適当な文を書け．

### 4.1 加算と条件分岐

- ラベル A と B が示す値を算術加算して，その結果をラベル WA の領域に格納する．
- 計算がオーバーフローした場合は 'OVER' と表示する．

```

PRG  START
    LD  GR1,A      ;A の値を GR1 に入れる
    ADDA GR1,B     ;GR1=GR1+B
     ;OF が 1 ならば L1 へ
     ; 無条件で L2 へ
L1    ; "OVER" を表示
L2   ST  GR1,WA   ;GR1 の値を WA へ入れる
    RET
A    DC  20000
B    DC  30000
WA   DS  1
BUFF DC  'OVER'   表示する文字列
LEN  DC  4        表示する文字列の長さ
    END
    
```

### 4.2 論理演算とアドレス修飾

- ラベル A と B の示す値の論理積 (A AND B)，論理和 (A OR B)，排他的論理和 (A XOR B) を 3 語確保されたラベル ANS に格納する．

```

PRG  START
    LAD GR2,0      ;GR2 に 0 を入れる
    LD  GR1,A      ;A の値を GR1 に入れる
    AND GR1,B      ;GR1 と B の論理積
     ;ANS+0 番地に入れる
     ; インクリメント
     ;A の値を GR1 に入れる
     ;GR1 と B の論理積
     ;ANS+1 番地に入れる
    LAD GR2,1,GR2 ; インクリメント
    LD  GR1,A      ;A の値を GR1 に入れる
    XOR GR1,B      ;GR1 と B の排他的論理和
    ST  GR1,ANS,GR2 ;ANS+2 番地に入れる
    RET
A    DC  #0030
B    DC  #00F9
ANS  DS  3        ;3 語確保
    END
    
```

### 4.3 シフト演算

- ラベル A が示す値の 0.75 倍をラベル KOTAE が示す場所に格納する．
- ヒント  $0.75 = 1 - 1/4$

```

PRG  START
    LD  GR1,A      ;A の値を GR1 に入れる
    LD  GR2,A      ;A の値を GR2 に入れる
     ;GR2 の値を 1/4 倍する
     ;A-1/4A の計算
    ST  GR1,KOTAE ;GR1 を KOTAE に入れる
    RET
A    DC  50
KOTAE DS  1
    END
    
```

### 4.4 繰り返し処理

- ラベル DATA (先頭アドレス) に入っている値の中から最大値を探すプログラムである．
- データの個数は，ラベル KOSUU が示している．
- 探した最大値は，ラベル MAX が示す場所に格納する．

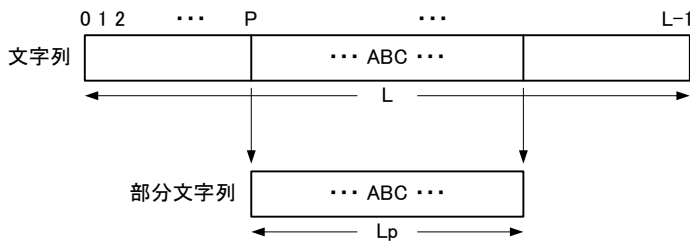
```

PRG  START
    LAD GR2,0      ;カウンタを 0 にする
    LD  GR1,KOSUU  ;GR1 にデータを入れる
     ; データ数をデクリメント
     ; 先頭のデータの読み込み
    ST  GR0,MAX    ; 暫定最大値の保存
LOOP LAD GR2,1,GR2 ; カウンタをインクリメント
     ; データの読み込み
    CPA GR0,MAX    ; 最大値と比較
     ;GR0 が小さいとき SKIP へ
    ST  GR0,MAX    ; 最大値の保存
SKIP CPA GR1,GR2  ; データ数とカウンタの比較
     ; 数の方が多いとき LOOP へ
    RET
DATA DC  10,15,8,20,7
KOSUU DC  5
MAX   DS  1
    END
    
```

## 5 応用問題

以下の問題は難しいが、配点は低い。時間が余った人はトライせよ。これは、平成 15 年秋の基本情報処理技術者試験に出題された問題である。

副プログラム SUBS は、次に示すように指定された文字列の一部分(部分文字列と呼ぶ)を、指定された領域に複写するプログラムである。



- GR1 ~ GR5 に、次の値が設定されて、主プログラムから渡される。

GR1	文字列が格納されている領域の先頭アドレス
GR2	文字列の長さ L
GR3	部分文字列の開始位置 P ( $0 \leq P < L$ )
GR4	部分文字列の長さ Lp
GR5	部分文字列を複写する領域の先頭アドレス

ただし、部分文字列の領域と複写先の文字列の領域は重ならない。

- 次に示すどちらかの処理結果を GR0 に設定して主プログラムに戻る。

0	正常終了
1	長さ指定に誤り ( $P+Lp > L$ )

- 副プログラムから戻るとき、汎用レジスタ GR1 ~ GR7 の内容は元に戻す。

[問 1] プログラム中の [ ] に入れる正しい答えを、解答群の中から選べ。

ア	LAD GR2,1,GR2	イ	LAD GR2,-1,GR2
ウ	LAD GR4,1,GR4	エ	LAD GR4,-1,GR4
オ	LD GR0,0,GR3	カ	LD GR0,0,GR4
キ	LD GR1,0,GR3	ク	LD GR1,0,GR4

[問 2] 次の記述中の [ ] に入れる正しい答えを、解答群の中から選べ。

SUBS の仕様では、 $P+Lp > L$  の場合、長さ指定に誤りがあるとしている。これを誤りとしないうで次のとおりに変更する。

- 文字列の終わりまで複写する。
- 複写できた部分文字列の長さを GR0 に設定する。

そのために、SUBS の処理を次のように変更する。

(1) に対応する修正として、行番号 7 にラベル GETPS を追加し、行番号 16,17 の 2 行を次のように変更する。

```

SETER LD GR4,GR2
      [ ]
      JUMP GETPS
    
```

(2) に対する修正として、行番号 7 の直後 [ ] d の値を一時保存し、行番号 18 でこの値を GR0 に設定する。

[c に関する解答群]

ア	SUBA GR3,GR2	イ	SUBA GR4,GR2
ウ	SUBA GR4,GR3	エ	SUBA GR5,GR4

[d に関する解答群]

ア	GR1	イ	GR2	ウ	GR3
エ	GR4	オ	GR5		

```

1 SUBS START
2   RPUSH
3   LD GR7,GR3
4   ADDA GR7,GR4
5   CPA GR2,GR7           ; 長さのチェック
6   JMI SETER
7   ADDL GR3,GR1
8 LOOP LD GR4,GR4
9   JZE SETS
10  [ ] a
11  ST GR0,0,GR5         ; 文字を格納する
12  LAD GR3,1,GR3
13  [ ] b
14  LAD GR5,1,GR5
15  JUMP LOOP
16 SETERLAD GR0,-1       ; エラーを設定する
17  JUMP FIN
18 SETS LAD GR0,0        ; 正常終了を設定する
19 FIN RPOP
20  RET
21  END
    
```

## 6 参考資料

### 6.1 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進数表示で示す．

第1語		第2語		命令語長	命令語とアセンブラとの対応		
OP		r/r1	x/r2		adr	機械語命令	意味
主OP	副OP						
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	7			-	1	SUBL r1,r2	subtract logical
	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

## 6.2 文字の符号表

JIS X 0201 ラテン文字・片仮名用 8ビット符号で規定する文字の符号表を使用する。

右に符号表の一部を示す。1文字は8ビットからなり、上位4ビットを列で、下位4ビットを行で示す。例えば、間隔、4、H、\のビット構成は、16進数表示で、それぞれ 20、34、48、5Cである。16進数表示で、ビット構成が 21~7E(及び表では省略している A1~DF) に対応する文字を図形文字という。図形文字は、表示(印刷)装置で、文字として表示(印字)できる。

この表にない文字とそのビット構成が必要な場合は、問題中で与える。

列 行	02	03	04	05	06	07
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	