

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	①
	LAD	GR2, -10, GR1	②
	SUBA	GR0, B	③
	LD	GR2, A	④
	LD	GR1, A, GR2	⑤
	ADDA	GR1, =33	⑥
	RET		⑦
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	3	-	-	0	0	0
②	3	-1	-	0	1	0
③	3	-1	-11	0	1	0
④	-8	-1	-11	0	1	0
⑤	-8	-1	3	0	0	0
⑥	-8	-33	3	0	1	0
⑦	-8	0	3	0	0	1

(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。

算術演算の結果が-32768~32767に収まらなかったとき、OF=1となる。

b. フラグレジスタ SF=1 になる場合について説明せよ。

算術演算の結果が負の場合(ビット番号15が1)、SF=1となる。

c. フラグレジスタ ZF=1 になる場合について説明せよ。

算術演算の結果がゼロの場合(全てのビットが0)、ZF=1となる。

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。

PGM	START	
	LD	GR0, A
	ADDA	GR0, B
	ST	GR0, C
	RET	
A	DC	128
B	DC	256
C	DS	1
	END	

(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。

PGM	START	
	LD	GR0, DATA
	LD	GR1, A
	ADDA	GR0, DATA, GR1
	LD	GR1, B
	ADDA	GR0, DATA, GR1
	ST	GR0, ANS
	RET	
DATA	DC	2, 4, 6, 8, 10
A	DC	2
B	DC	4
ANS	DS	1
	END	

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
    
```

このプログラムは、AKITA のうちの A と I と A の JIS X0201 コードの値を加算するプログラムになっている。それぞれの値は、

A → (41)₁₆
I → (49)₁₆

である。したがって、その和は、

(41+49+41)₁₆ = (CB)₁₆

となる。したがって、このプログラムを実行すると、

ANS = (00CB)₁₆

となる。

(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
    
```

このプログラムと以下のプログラムは、アセンブルすると同じマシン語になる。従って、問題で与えられたプログラムは、1+1 を計算してその結果をメモリーに格納する動作をすることになる。このプログラムを実行させると、アドレス#0029 の値が#0000 から#0002 に変化する。

```

PGM   START
      LD     GR0, A
      ADDA  GR0, B
      ST     GR0, C
      RET
A     DC    1
B     DC    1
C     DC    0
      END
    
```

[注意]

DC 命令で確保されて初期化した領域にデータを書き込むことは可能である。

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		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

