

1 CASL II の命令の種類

(1) CASL II の命令は、アセンブラ命令と機械語命令、マクロ命令の 3 種類に分けることが出来る。以下の特徴を持つ命令は、どの種類に属するか?(各 1 点)

- ① 機械語命令を寄せ集めて、ひとつの命令としている。
- ② 1 語または 2 語の特定のビットパターンに変換される。
- ③ 3 語以上の特定のビットパターンに変換される。
- ④ アセンブラーに対して、指示を行う命令。
- ⑤ ビットパターンに変換されない。
- ⑥ CPU の動作を記述し、その命令に対応した回路が用意されている。
- ⑦ 対応する機械語が無い。
- ⑧ 処理の対象であるデータをメモリーに設定するために使われることが多い。

(2) 以下の命令は、先の 3 種類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
- ⑪ SUBL ⑫ ADDL ⑬ XOR ⑭ AND ⑮ OR

2. コンピューターの動作の仕組み

以下について、簡潔に 1～2 行程度で説明せよ。

- [ア] CPU の役割
- [イ] メモリーの役割
- [ウ] レジスターとは何か?
- [エ] メモリーのアドレスとは何か

3. ハンドアセンブル

(1) 以下のソースプログラムに対応するメインメモリーの内容を、解答用紙に 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。さらに、ラベル BEGIN のアドレスは、#0020 とする。

PGM	START	BEGIN
BEGIN	LD	GR0, A
	AND	GR0, B
	ST	GR0, C
	RET	
A	DC	7
B	DC	4
C	DC	1
	END	

4. 命令

(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)

- [ア] 実行開始番地を NEXT と指示する。
- [イ] 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。
- [ウ] 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。
- [エ] 主記憶装置の内容をレジスタ GR1 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 GR3 の内容を加算したアドレスとする。
- [オ] 汎用レジスタ GR3 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。
- [カ] ラベル Z が示すアドレスを、汎用レジスタ GR4 に格納する。
- [キ] 汎用レジスタ GR1 と GR2 の内容を符号付整数として加算する。加算結果は、GR2 に残す。
- [ク] 汎用レジスタ GR1 からラベル A が示す内容を符号付整数として減算する。加算結果は、GR1 に残す。
- [ケ] 汎用レジスタ GR1 からラベル A が示す内容を符号無整数として減算する。加算結果は、GR1 に残す。
- [コ] 汎用レジスタ GR1 とラベル A が示す内容の各ビット毎の論理和を計算する。結果は、GR1 に残す。

5. 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。

PGM	START	
	LD	GR0, A ← ①
	LAD	GR1, 5 ← ②
	LAD	GR2, -10, GR1 ← ③
	SUBA	GR0, A ← ④
	LD	GR2, B ← ⑤
	LD	GR1, A, GR2 ← ⑥
	ADDA	GR1, B ← ⑦
	RET	
A	DC	-3
B	DC	1, -3, 5, -7
	END	

6. 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。

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

- 計算に使うデータの並び(2, 4, 6, 8, 10, 12)は、以下の命令を用いて、メインメモリーに格納すること。ラベル名は、DATA です。

```
DATA DC 2, 4, 6, 8, 10, 12
```

- この DATA の 2 番目と 4 番目と 6 番目のデータを加算する。即ち、4+8+12 を計算する。
- 計算結果は、主記憶装置に格納すること。
- データへのアクセスには、指標レジスタを用いること。

7. 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。
時間が余った人向け

(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。(2 点)

```
PGM      START
LD       GR0, IIJIMA
LAD      GR1, 1
AND      GR0, IIJIMA, GR1
LAD      GR1, 3
ADDA     GR0, IIJIMA, GR1
ST       GR0, ANS
RET
IIJIMA   DC 'IIJIMA'
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。

```
PGM      START      NEXT
LD       GR1, NEXT
LAD      GR2, 1
ADDA     GR1, NEXT, GR2
LAD      GR3, 2
ST       GR1, NEXT, GR3
RET
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		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
7			-	1	SUBL r1,r2 subtract 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

