教材開発

新CASLに対応したシミュレータ「Turbo CASLII」

宮城県立塩釜高等技術専門校　情報処理科　新妻　幹也

1．はじめに

昨年６月情報技術者試験センターから平成13年４月以降の新しい試験体系が発表になりました。いくつかの大幅な改正になっていますが，その中で，これまで第２種情報技術者試験の言語で出題されてきたアッセンブリ言語の「CASL」も，かなりの範囲でその言語仕様が変更されることになりました。その変更点を見てみると，命令語の追加や既存の命令語の機能拡張など，昭和60年ごろに「CAP−X」から「CASL」に変更になって以来の改正ではないかと思います。それに伴い，これまで広く使われてきたいくつかの「CASLシミュレータ」も変更なしに使用することは困難になってきました。今回は，ビジュアルタイプのシミュレータの１つ「Turbo CASL」（「技能と技術　2/1999」に掲載）を改正の仕様に合わせて「Turbo CASLII」として再構築しましたので，引き続き多くの教育訓練機関でご利用いただければと思います。

2．変更された主な点

変更された主な点は次のようなものです。

① これまで，語のビット番号は最上位ビット（符号部分）から，0，1，2，・・14，15となっていたが，最下位から0，1，2，・・14，15（最上位ビット）となった。

② これまで命令語の語長は２だったが，命令語の機能拡張により１のものも出てきた。

③ 汎用レジスタはGR0からGR4までだったが，GR7までと，３つ増えた。

④ また，事実上スタックポインタとして使われていた，GR4のレジスタは独立して，SPレジスタとなった。

⑤ フラグレジスタは事実上２ビットで機能していたが，オーバーフロービットが追加になり，３ビットになった。

⑥ 数値的な演算（−32768から＋32767までの符号あり演算）に加えて論理加算，論理減算（０〜65535）が加わった。

⑦ Supervisor Call，No Operation命令が加わった。

⑧ ADDA GR1，GR2のようなレジスタ間の演算機能が取り入れられた。

⑨ オーバーフローフラグが設定されたことに伴い，ジャンプ命令にいくつかの変更があった。

⑩ ラベルがこれまでの６文字から８文字まで指定可能になった。

⑪ アドレスの記述として16進定数とリテラル（＝）による指定が可能となった。

⑫ １つのDC命令で複数の定数を定義可能となった。

⑬ IN，OUT命令において入出力の長さを80文字から256文字まで可能となった。

⑭ EXIT命令が削除された（シミュレータ上は残してある）。

⑮ LAD（旧LEA）で，FRは変化しなくなり，LD

表１　旧COMETと新COMETⅡの命令新旧対照表

ロード，ストア，ロードアドレス命令	旧命令（＊は新設）	新COMETⅡ命令コード	命令語長（Word）	hexコード番号	備　考
LoaD	LD	LD	2	10	
STore	ST	ST	2	11	
Load ADdress	LEA	LAD	2	12	
LoaD	＊	LD	1	14	LD r1, r2（※1）

算術，論理演算命令	旧命令（＊は新設）	新COMETⅡ命令コード	命令語長（Word）	hexコード番号	備　考
ADD Arithmetic	ADD	ADDA	2	20	
ADD Logical	＊	ADDL	2	22	
SUBtract Arithmetic	SUB	SUBA	2	21	
SUBtract Logical	＊	SUBL	2	23	
ADD Arithmetic	＊	ADDA	1	24	ADDA r1, r2（※１）
ADD Logical	＊	ADDL	1	26	ADDL r1, r2（※１）
SUBtract Arithmetic	＊	SUBA	1	25	SUBA r1, r2（※１）
SUBtract Logical	＊	SUBL	1	27	SUBL r1, r2（※１）
AND	AND	AND	2	30	
OR	OR	OR	2	31	
eXclusive OR	EOR	XOR	2	32	
AND	＊	AND	1	34	AND r1, r2（※１）
OR	＊	OR	1	35	OR　r1, r2（※１）
eXclusive OR	＊	XOR	1	36	XOR r1, r2（※１）

比較演算命令	旧命令（＊は新設）	新COMETⅡ命令コード	命令語長（Word）	hexコード番号	備　考
ComPare Arithmetic	CPA	CPA	2	40	
ComPare Logical	CPL	CPL	2	41	
ComPare Arithmetic	＊	CPA	1	44	CPA r1, r2（※１）
ComPare Logical	＊	CPL	1	45	CPL r1, r2（※１）

シフト演算命令	旧命令（＊は新設）	新COMETⅡ命令コード	命令語長（Word）	hexコード番号	備　考
Shift Left Arithmetic	SLA	SLA	2	50	
Shift Right Arithmetic	SRA	SRA	2	51	
Shift Left Logical	SLL	SLL	2	52	
Shift Right Logical	SRL	SRL	2	53	

分岐命令	旧命令（＊は新設）	新COMETⅡ命令コード	命令語長（Word）	hexコード番号	備　考
Jump on Plus	＊（JPZに似ている）	JPL	2	65	旧JPZの０を除いた
Jump on Minus	JMI	JMI	2	61	
Jump on Non Zero	JNZ	JNZ	2	62	
Jump on ZEro	JZE	JZE	2	63	
Jump on OVerflow	＊	JOV	2	66	
unconditional JUMP	JMP	JUMP	2	64	

スタック操作命令	旧命令（＊は新設）	新COMETⅡ命令コード	命令語長（Word）	hexコード番号	備　考
PUSH	PUSH	PUSH	2	70	
POP	POP	POP	1	71	

コール・リターン命令	旧命令（＊は新設）	新COMETⅡ命令コード	命令語長（Word）	hexコード番号	備　考
CALL subroutine	CALL	CALL	2	80	
RETurn from subroutine	RET	RET	1	81	

その他	旧命令（＊は新設）	新COMETⅡ命令コード	命令語長（Word）	hexコード番号	備　考
SuperVisor call	＊	SVC	2	F0	
No Operation	＊	NOP	1	00	

※１　旧CASLにはなかったレジスタ間の演算で，この場合には命令コード番号，命令語長が異なる。
（r1，r2はレジスタ　GR*）

でFRの設定がされることになった（ただしオーバーフロー時は０が設定される）。

具体的な命令語の変更点をまとめると表１のようになります。なお，この表にある「hexコード番号」は，これまで試験センターからは発表されていませんでしたが，シミュレータ等の作成を意識してか，参考ということで発表されました。

３．「Turbo CASL II」の変更点

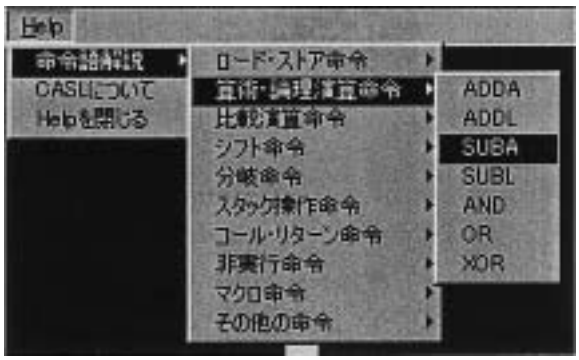
すでに「技能と技術」1999年２号で旧タイプの「Turbo CASL」を発表していますが，基本的な操作で大きく変わるところはありませんが，いくつか変更した点について説明します。

シミュレータの起動画面

画面上のメニューや表示で変更されたのは次の３点です。

① Modeメニュー上に「unsigned Dec表示」を追加した。

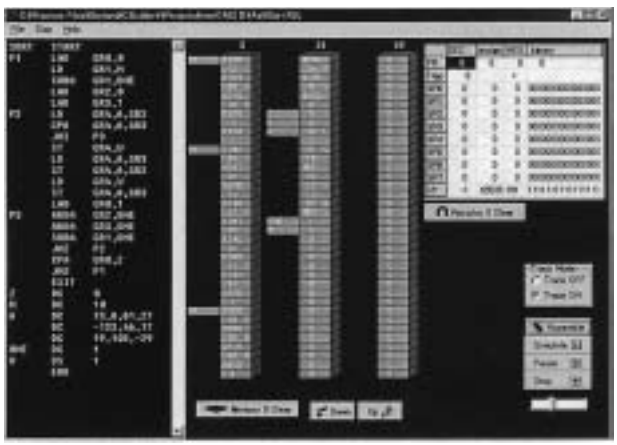
② Helpメニューの「命令語解説」では，命令語がいくつか追加されたので，命令語の機能ごとに分類して表示させる形式をとった。



③ レジスタが増えたことに伴って，レジスタのモニタ部分を変更した。

	DEC	unsign	HEX	binary
PR	0	0	0	0
Flag	0		+	
GR0	0	0	0	0000000000000000
GR1	0	0	0	0000000000000000
GR2	0	0	0	0000000000000000
GR3	0	0	0	0000000000000000
GR4	0	0	0	0000000000000000
GR5	0	0	0	0000000000000000
GR6	0	0	0	0000000000000000
GR7	0	0	0	0000000000000000
SP	-1	65535	ffff	1111111111111111

④ 追加・変更された命令（擬似命令）についての動作の変更を行った。



シミュレータ全体の画面

４．CASLⅡのプログラム例

最後に今回の変更によって，これまでのCASLプログラムとCASLⅡのプログラムの相違点を簡単なプログラム例で示します。

```
SAMP    START
        LAD GR1,0
        LAD GR2,0
LOOP    ADDA GR2,DAT,GR1
        LAD GR1,1,GR1
        CPA GR1,N10
        JMI LOOP
        ST  GR2,ANS
        EXIT
DAT     DC 54,32,81,45,30
        DC 12,29,75,82,47
N10     DC 10
ANS     DS 1
        END
```

DAT番地に定義された複数（10個）の数値データの和を求めてANS番地に格納する。

このプログラム例での変更点としては，

1）LEAがLADになった。

2）ADDがADDAになった。

3）DCによる値の定義は，（カンマ）で区切って複数でも定義可能になった（数値・文字列混在可）。

4）EXIT（マクロ）が廃止になった。しかし，このシミュレータでは，EXITを残してある。他のプログラム例では，このEXIT部分はRETになることが多いのだが，シミュレータとして，CALL命令としてのRETと区別するためにあえて残した。あるいはSVCでもよいのかとも思われるが，SVCについては，まだ，その意味をどのように解釈すればよいのか不明な点も多いので，はっきりわかった段階でシミュレータにも反映させたいと思う。

```
SAMP    START
        LAD GR1,0
        LAD GR2,0
        LAD GR3,0
LOOP    ADDL GR1,GR2
        JOV EX
        ADDA GR2,V100
        LAD  GR3,1,GR3
        JUMP LOOP
EX      ST GR3,ANS
        RET
V100    DC 100
ANS     DS 1
        END
```

このプログラム例での変更点としては，

1）ADDLという符号なし演算の命令が追加された。このことは，２バイトで扱える数値の範囲をFFFFまでで65535と解釈することが可能になったことを示している。

2）JOVというオーバーフロージャンプ命令が追加された。これは，フラグに新たにオーバーフロービットが追加されたことによりこれを検知してジャンプする命令を新設したものと考えられる。

3）ADDL GR1,GR2のようなレジスタの値を直接扱えるように仕様が変更された。このことは，これまですべて命令語にはアドレスが対になって2Wordのコードを生成していたが，この例のように示すアドレスがない形式が出てきたため，このような記述のコードは1Wordで生成されることになった。したがって，同じADDLであってもこれまでのように対応するアドレスを持つ記述のものとそうでないもので，命令コードが異なる（ADDLでは22と26）。

4）無条件ジャンプ命令がJMPからJUMPにスペル変更された。

×（正しく動作しない）

```
SAMP    START
        LAD GR1,10
        LAD GR2,0
LOOP    ADDA GR2,GR1
        LAD GR1,-1,GR1
        JPL LOOP
        RET
        END
```

1）この例は１から10までの和をダウンカウントでループして求める基本的な問題である。しかし，このプログラムは正しく動作しない。その理由は，今度の新しい仕様によるとLADでは，フラグの設定がなされないことになっているからである。したがって，このプログラムを動作させるには，一例としては，次のようなプログラムが考えられる。

2）これまで，JPZという正または０のときジャンプするという命令が，JPLという正のときのみジャンプするということに変更された。この変更については，それほど大きなプログラム変更を要求されるものではないが，微妙に違う点は理解しておく必要がある。

○（正しく動作する）

```
SAMP    START
        LAD GR1,10
        LAD GR2,0
LOOP    ADDA GR2,GR1
        SUBA GR1,ONE
        JPL LOOP
        RET
ONE     DC 1
        END
```

```
SAMP    START
        LAD GR4,#17
        LAD GR7,#7
        XOR GR4,GR7
        NOP
        RET
        END
```

1）この例では，これまで，10進定数（アドレス）でしか記述できなかった部分に，＃で始まる16進定数が記述できることを示している。

2）これまで，排他的論理和の記述はEORであったが，XORにスペル変更された。

3）NOPという何もしない命令が追加された。実行としては何もしないが，1Wordの領域を00というコードで占有する。

```
SAMP    START
        LD GR1,DAT
        RET
DAT     DC 'MIKIYA',256
MN      DS 10
PD      DC 16,'SAORI',128
        DC 'TAKERU',543,#DE
```

1）この例では，先の例でも触れたが，DCの定義記述が，（カンマ）区切りで，数値，文字列，アドレス等を混在で可能になったことを示している。

　ただし，シミュレータの機能上は，エディターの画面（右端）からデータがはみ出すような記述はできないので，あくまでも，この例の程度の定義しかできない。それで足りないときは，これまでのように改行して定義していくことになる。

```
MD      DC '  ', #2DF, ', '
        END
```

```
SAMP    START
        LD GR1, =28
        LD GR2, =#18
        LD GR3, DAT
        LD GR4, MN
        LAD GR5, =#256
        RET
DAT     DC 12, 13, 14, 15
MN      DC 'ROBO', 'SUMOU'
AN      DS 8
        END
```

1）この例では，LD命令やLAD命令などで，リテラルに続く数値をDCで定義したように定義し，それをアドレスの中身やアドレス値としてレジスタに読み込むことができるものである。この方法によって定義した値は，プログラムコードの後のDC，DSで定義されたデータや領域の後にまとめて配置されることになる。

5．おわりに

今回のCASLの言語仕様の大幅な改訂の理由は，「時代にマッチしたプログラミング」を可能にするためであると考えられます。この改訂によって予想できることは，命令語の機能拡張やレジスタの増加によってこれまでのCASLよりもプログラムが簡単にしやすくなるため，試験問題では比較的行数の多い，中身の濃いプログラムの出題です。

また，今回は改訂発表間もないこともあり，例えば，SVC（スーパバイザコール）を具体的にどのように使うのかなど，まだその改訂部分の解釈がはっきり理解しがたい部分もあります。いずれにしても，アッセンブリ言語教育の根本には何ら不利益のあるものではないので，これまで同様，基本はしっかり理解させたいものです。

●シミュレータの無償配布について

このシミュレータとマニュアルは無償で配布しておりますので，ご希望の方は，下記のメールアドレスまたは，校へ直接電話で申し出てください。フロッピーディスク（プログラムとマニュアル（Word文書））にてお送りいたします。

◇

Eメールアドレス　mikiya@jody.svtc.ac.jp
宮城県立塩釜高等技術専門校　情報処理科
電話　022-362-3958