

2U-3

G<sub>MICRO</sub>/200アーキテクチャの機能の考察

— 文字列処理命令 —

海永正博\*<sup>1</sup>、野尻徹\*<sup>1</sup>、川崎郁也\*<sup>2</sup>

\* 1 日立製作所システム開発研究所 \* 2 日立製作所武蔵工場

1. はじめに

G<sub>MICRO</sub>/200はTRON仕様に準拠した最初のプロセッサチップである。TRON仕様チップは明らかにCISCに分類される。CISCとはComplex instruction Set Computerを意味している。TRON仕様チップの発想は可能な限りの論理をプロセッサチップに詰め込み、それにより高速化を図るものである。高速化の対象としては、一般の高級言語で記述されたプログラムが考えられ、特にCで記述されたプログラムが高速化の主対象となる。Cは今後ますますマイクロプロセッサに浸透し、その上で動作するオペレーティングシステムでさえ、大半がCで記述されていくことになるのが明白であるからである。

Cは高級言語であるが特殊な言語ではない。ざっくりいうと、システムミックスが高速であれば、Cで記述されたプログラムは高速に動作できる。さらにざっくりいうと、整数転送命令、整数加算命令、分岐命令が高速でありその他の普通の命令も同程度に高速であれば、Cで記述されたプログラムは高速に動作できる。

しかし、Cの機能を吟味すると、C(及びG<sub>MICRO</sub>/200の応用プログラム)がアーキテクチャに期待する

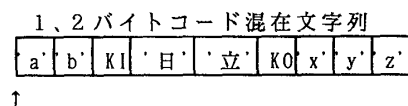
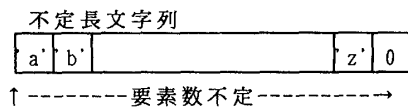
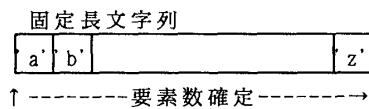


図1. 文字列の分類

る。例えば、「文字列処理」、「再帰呼出し」などが該当するが、ここでは文字列処理に焦点を絞り、考察を加える。

2. 文字列の扱い

要素数の性質から文字列は以下の2つに分類される(図1)。

(a) 固定長文字列

(b) 不定長文字列

ここで(a)の固定長文字列とは、要素数があらかじめ確定した文字列のことである。この文字列は、データベースにおけるレコードがその代表例であり、文字列先頭アドレスと要素数により文字列が指定される。

一方、(b)の不定長文字列と仮によぶ文字列は、要素数が可変であらかじめ確定していない文字列のことである。この文字列は非定型OA処理やワードプロセッサなどにおいて必要となるものであり、文字列先頭アドレスと要素の終了を指示する番兵文字(値0の要素など)で文字列が指定される。

また以下の文字列

(c) 1バイト、2バイト混在文字列

も現在広範に使用されている。

Cが想定するアプリケーション、更にはG<sub>MICRO</sub>/200が想定するアプリケーションにおいては(b)や(c)を旨く扱えることが必要条件となる。尚Cの言語仕

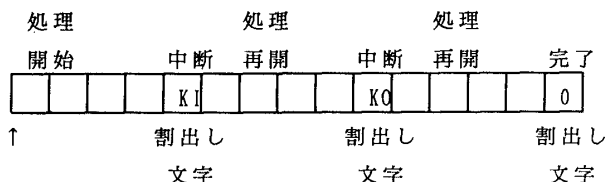


図2. 文字列処理の考え方

様自体は文字列をサポートしていないが、広い意味で言語仕様にふくまれる標準関数パッケージの中に文字列処理関数群がサポートされており、広範に使用されている。

さて G<sub>MICRO</sub>/200 においては、(a)の固定長文字列、(b)の不定長文字列、(c)の1、2バイト混在文字列もともに旨く扱えるように工夫してある(図2、図3)。

G<sub>MICRO</sub>/200 においては文字列を以下の様に捉える。

- (a) 文字列とは同じサイズ(バイト、半語、語)のスカラ要素が連続的に配置された配列体である。
- (b) 文字列は先頭アドレスで特定される。
- (c) 文字列はさらに配置方向(順方向/逆方向)を属性として持つ。これは逆方向処理のためのものである。
- (d) 文字列は要素数を属性として持つ。  
標準要素数は無限大である(要素数0を指定すると実効的に要素数無限大となる)。
- (e) 文字列はさらに、割り出し条件を属性として持つ。ここで割り出し条件とは、個別要素の値の状態を検査し、条件を充たしていれば命令処理を一旦終了するものである(この時、ポインタは割り出し文字の次の文字を指している)。

尚(d), (e)により不定長文字列などを旨く扱える。そして、このような文字列を扱う文字列処理命令を G<sub>MICRO</sub>/200 は提供している(図3)。

### 3. 評価

Cコンパイラは、G<sub>MICRO</sub>/200の文字列処理命令を直接に利用している。即ち、CコンパイラはCの文字列処理関数の呼出しをインラインに展開している。従って、両者の相乗効果によりCの文字列処理は3倍程度高速になる(表1)。

表1. 文字列処理命令の効果

例題 プログラム	文字列処理 命令使用時	文字列処理 命令不使用時
文字列転送 strcpy	1 0 + 6 n サイクル	4 + 1 5 n + α サイクル

α : 関数呼出しオーバーヘッド

### [文字列]



### [割り出し条件]

割り出し条件	ニモニック	eeee
要素値 < R3	LTU	0000
要素値 ≥ R3	GEU	0001
要素値 = R3	EQ	0010
要素値 ≠ R3	NE	0011
要素値 < R3	LT	0100
要素値 ≥ R3	GE	0101
無し	N	0110

### [文字列処理命令]

機能	アセンブリ表現	機械語表現
転送	SMOV/eeee/b.SS	00eeeeSS1110010b
比較	SCMP/eeee.SS	00eeeeSS11100000
検索	SSCH/eeee/r.SS	00eeeeSS1111010r
充填	SSTR.SS	000001SS11110111

/b : 文字列の方向

/r : /bに同様

SS : 文字列の要素サイズ

### [レジスタ上のパラメータ]

- R0 : ソース文字列先頭アドレス(入力/出力)
- R1 : デスト文字列先頭アドレス(入力/出力)
- R2 : 要素数 (入力/出力)
- R3 : 比較値 (入力)

図3. G<sub>MICRO</sub>/200の文字列型と文字列処理命令

### 参考文献

- 1) 坂村 : TRON Project 1987
- 2) カーニハン、他 : プログラミング言語C