

共通的計算機モデルに基づく機器制御用言語 $e1(\alpha)$ の開発

4 J - 7

エルアルファ

渡辺 坦†、佐久間和子†、荒井秀之††、梅谷孝文†††
 † 日立製作所システム開発研究所、†† 同社小田原工場、††† 日立マイクロコンピュータエンジニアリング

1. はじめに

計算機言語は計算機と人のインタフェースである。最近の言語開発は、そのインタフェースを人に合わせる方向で行われてきた。従って、計算機とのギャップを埋めるコンパイラの負担が重くなり、計算機資源の利用や性能等に対する制約もあって、今もアセンブラを使っている分野がかなりある。機器制御専用計算機等がその例である。そこで、これまでの行き方とは逆に、実際の計算機に合わせることを重視し、それを人に受け入れ易い形式にすることを目的として、現行の計算機に共通の計算機モデルを設定し、それに基づく言語 Essential Language $e1(\alpha)$ [エルアルファ]とそのコンパイラを開発した。これは次の特長を持つ。

- (1) 言語仕様が簡素で構文は2ページに図示できる。(2) 実行性能の良いコンパイラを作れる。(3) コンパイラによる誤り検出が容易である。(4) プログラムの分担開発が容易である。(5) 計算機資源を有効活用できるにもかかわらず、機種非依存性が高い。

2. 計算機の基本モデル

現在広く使われている計算機は、演算装置とメモリ、入出力装置を持つ。メモリはレジスタや主記憶等の種類ごとに番地付けされている。メモリの最少単位はビットであるが、アクセスの単位としてバイトやワードがある。メモリの構成要素は、種類や区域を m とすると、 $m(i)$ と表現できる。

演算装置は、操作名を f 、操作対象を $x1, x2, \dots, xn$ として、機械操作 $f(x1, x2, \dots, xn)$ を行う。機械操作にはデータを対象とする演算操作と、プログラムの一部を対象とする制御操作がある。一つ一つの機械操作は機種依存であるが、機械命令列として見ると、機種に依存しない操作にまとめられることが多い。入出力装置の機能も $g(x1, x2, \dots, xn)$ と表現できる(但し、副作用を持つ)。

3. 計算機言語の基本概念

言語の基本概念としては、操作、操作対象、属性があ

Development of Embedded Computer Language $e1(\alpha)$
 based on Common Machine Model, by T. Watanabe†, K. Sakuma†, H. Arai†, K. Umetani††, † Hitachi, Ltd.,
 †† Hitachi Microcomputer Engineering.

り、それらの定義と参照、並びに言い替え機能がある。操作対象の種類には、数値、文字、ブール値等があり、その配列と構造体がある。ビット列は桁数の指定された整数やブール値列とみることができる。操作 f を操作対象 $x1, x2, \dots, xn$ に施すことは、 $f(x1, x2, \dots, xn)$ と表現できる。操作 $s1, s2, \dots, sn$ の列を一つの複合操作とするには `do; s1; s2; ... sn; end` とすればよい。分岐、反復等はこれらを対象とする制御操作である。副プログラムとマクロは言い替え機能で表現できる。属性は操作や操作対象の扱い方の違いを表す。型宣言は属性を指定する。プログラムを分割した時の分割部分の使い方を示すのがインタフェース仕様であり、共通変数や型、副プログラム仕様等を示す。

4. 計算機モデルと言語概念の対応付け

2節で述べた計算機モデルの構成要素は言語要素としての操作や操作対象とうまく対応する。言語の機種非依存化上の問題点として、メモリ番地やレジスタがある。メモリ番地は、(1)外部インタフェースの指定、(2)データ詰合せの指定、(3)複数の操作対象の一括処理、(4)同一対象に対する異なる見方からの処理、(5)操作対象の動的変更等の目的で使う。(1)のためには番地を指定せざるを得ない。(2),(3),(4)は、同一メモリ区域に異なる型の変数を重ね合わせて使う「領域」を導入すると、機種によらずに処理できる。(5)はポインタで対処できる。

レジスタは高速のメモリと考えることができる。変数のアクセス頻度レベルを指定して、特定レジスタを指定せずにコンパイラに使い方を任せれば、機種非依存な形でレジスタを利用できる。

シフトやローテイトも機種依存性の強い操作であるが、これらはビットフィールドを表現できる変数があれば殆ど使わなくてすむ。

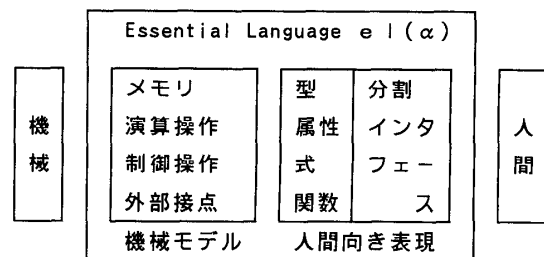


図1. $e1(\alpha)$ 言語概念図

5. e1(α)の言語仕様の要点

以上の考察に基づいてe1(α)の言語仕様を定めた。これは一口で言うと、図1のように、共通の計算機モデルの外部表現に、使い易さを向上させるための宣言文とインタフェース指定を加えたものである。仕様設定に際しては、実行性能と、保守性、信頼性の向上を重視し、予想外の副作用を防止するようにした。

プログラムは、共通仕様の宣言からなるインタフェースユニットと、処理操作を表す処理ユニットで構成され、インタフェースユニットを他のインタフェースユニットや処理ユニットにconnect文で結合すると、そこで指定された共通仕様を利用可能になる。基底のスカラー型は整数と文字、プル、ポインタであり、整数とプル型にはビット幅を指定できる。集合体としては配列と構造体、領域があり、詰合せ指定denseが使える。ポインタはポイント先の型を限定しており、演算の対象にできないが、組込手続きINCR, DECRにり前進、後退させることができる。型変換は組込関数で明示的に行う。変数にはアクセス頻度レベルとアクセス形態(refer/update)を指定でき、引数には in/out/in out の区別を指定する。実行文は代入、if, case, loop, goto, exit, return, none, 手続き参照、ブロックである。

言語を機器制御に使えるようにするには、機種に依存せざるを得ない点もある。e1(α)では、それを極小化し他と分離するため、番地やdenseの指定はphysical指定のあるインタフェースユニットでのみ記述する。外部ハードや外部ソフトとの特定命令系列による接続部分は、機械語副プログラムとして記述する。

e1(α)のプログラム例を図2に示す。

```
unit PRIMFACT ; -- プログラムの構成単位はユニット
connect IOLIB ; -- 入出力インタフェースの接続
variable
  NUMBER, FACTOR, QUOTIENT : INTEGER ;
do ; -- 実行部
  GETI(NUMBER) ;
  FACTOR := 2 ;
  loop do ;
    QUOTIENT := NUMBER/FACTOR ;
    if NUMBER=QUOTIENT*FACTOR then do ;
      PUTI(FACTOR) ;
      NUMBER := QUOTIENT ; end
    else FACTOR := FACTOR + 1 ;
    if FACTOR*FACTOR > NUMBER then exit ;
  end ;
  PUTI(NUMBER) ;
end PRIMFACT ;
```

図2. e1(α)プログラム例

6. 評価

e1(α)の言語仕様を他言語と比べると、図3のように、機能は名前の通り本質的なものに限定しているが、仕様記述に近い所から機械操作に近い所まで扱える特長をもつ。構文規則はLL(1)文法に従っており、JIS BASIC並みの大きさである。コンパイラはC言語より大幅に小さい。言語仕様設定時には、アセンブラでないと難しいような多くの機器制御処理を記述し、それらが十分記述できることを確認した。コンパイラでは型の整合性とアクセス形態の整合性、並びにインタフェースユニットとの関係を綿密に検査する。コンパイラの性能もほぼ満足できる状態にあり[1]、既に機器制御専用マイクロプロセッサのソフトウェア開発等に実用している。

比較項目		Ada†	C	e1(α)
記述範囲	インタフェース仕様	↓	↓	↓
	誤り検査			
	操作対象			
	演算操作			
	機器制御			
実行時負荷		制御情報設定 スタック管理 レジスタ管理	スタック管理 レジスタ管理	レジスタ管理

図3. 他言語との比較 (幅の広い方が機能が豊富)

7. むすび

e1(α)は、多くの計算機に共通の本質的機能を抽出し、それに使い易さ、分割開発し易さ、誤り防止の観点から人間寄りの衣を被せた、機種非依存型システムプログラミング言語である。このコンパイラは複数の計算機用に開発され、シミュレータとの接続やプログラムからのドキュメント自動生成も実現されている。今後、プログラミング環境の拡充と多機種展開を行う予定である。

最後に、貴重な意見を頂き開発を推進して頂いた日立製作所の早川、上田、宮崎、桑原、増井、堂免、神野の諸氏、開発を分担して頂いた同社山下、日立マイコン宇野、日立SK加藤、日立KE小島の諸氏に深く謝意を表す。

8. 参考文献

- 佐久間、他：機器制御用言語e1(α)のコンパイラ開発、情報処理学会第40回全国大会予稿集(1990/3)

† Adaは米国政府の開発した言語であり、米国国防省の米国における登録商標である。