

アセンブラ言語CAP-Xのマイコン用 シミュレータの試作

宮 寄 敬**

1. ま え が き

最近のコンピュータの発達および普及に伴ってコンピュータ言語の進歩が著しく、コンピュータの使用が容易になってきている。こうした高級言語の利用によれば、コンピュータ内部の具体的動作について知らなくてもコンピュータの操作は可能であり、ある程度の処理を行うことも可能である。

しかし、処理内容が複雑化し、高度化してくると、これらの高級言語の利用だけでは十分なあるいは信頼のおける処理システムの開発が困難になってくる。このため、機械語やアセンブラ言語を使用してコンピュータの基本的な動作や内部機能について、習得する必要があるように思われる。情報処理技術者試験においてもアセンブラ言語が必須問題として出題されているのは、この点にあるのではないかと考えられる。ただし、この情報処理技術者試験において使用されているアセンブラ言語はCAP-Xと呼ばれ、これは仮想コンピュータCOMP-X用のものである。このため、CAP-Xで書かれたプログラムを実際のコンピュータを使用して実行させることは不可能である。従って、CAP-Xは小型によくまとめられたアセンブラ言語にもかかわらず、教育用として十分に利用されていないのが現状である。

そこで、本研究ではこのCAP-Xで書かれたプログラムの実行をマイクロコンピュータ上でソフト的に模擬することのできるシミュレータプログラムを試作してみた。本シミュレータは教育用という点を重視し、プログラムの実行過程の理解とデバックの手助けとなるようにトレース実行を可能にした。このシステムによるプログラム作成と実行を幾つか行ったら、ほぼ満足のいく成果が得られたため、ここに報告いたします。

2. システムの概要

CAP-Xはアセンブラ言語として最も基本となる命令12種類と擬似命令5種類とから成るアセンブラ言語で、入門用としては小型で適当なものである。本シミュレータシステムは、このCAP-Xの実習教育がマイクロコンピュータ上で可能となるように普及型のマイクロコンピュータのBASICにより作成した。本システムの概要を表わす状態図を図1に示す。その構成は、入力待ち、プログラム入力、コマンド処理及び実行処理の4状態から成る。まず、本システムを起動すると、プログラムのソースリストの入力かコマンド入力の待ち状態となる。プログラム入力の場合には、実行経過や編集が容易になるように行番号を付加して入力する。プログラムのソースリストの書式とその例を図2に示す。各項目の桁数について

* 昭和57年度高等専門学校情報処理教育担当者上級講習会課題研究発表会において発表

** 電気工学科助手

原稿受付 昭和59年9月28日

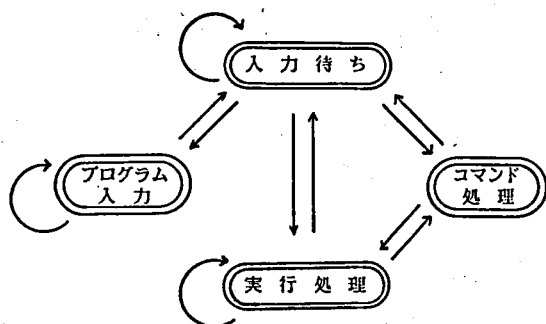


図1 本シミュレータの状態図

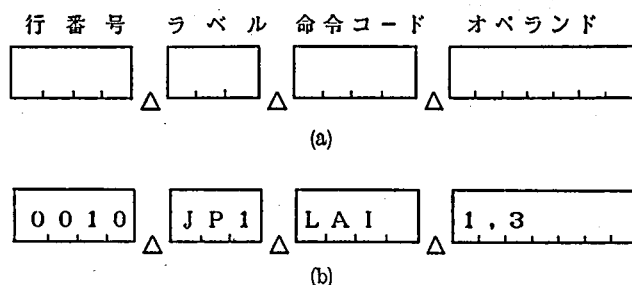


図2 プログラムの入力書式とその例

プログラムの実行が無限ループ状態になった場合にはシミュレータの実行を停止させる必要がある。

3. コマンドの説明

表1に本シミュレータのコマンド一覧表を示す。“S”、“/D”及び“/T”はプログラムの実行に関するコマンドである。“S”は通常の直接実行コマンドであり、“/D”は実行段

表1 コマンドの一覧表

表 記	機 能
S	プログラムの直接実行を行う。
/D	“ ” のダンプ実行を行う。
/T	“ ” のトレース実行を行う。
/L	CRT 上にプログラムリストの表示を行う。また、1ステートメントずつの挿入、追加および削除を行う。
/E	プログラム中のステートメントの修正を行う。
/X	レジスタの内容の表示を行う。また、レジスタの値の設定を行う。
/M	メモリの内容の表示を行う。また、メモリの値の設定を行う。
/O	プログラムリストのディスクへの書き込みを行う。
/I	“ ” のディスクからの読み込みを行う。
/W	“ ” のカセットへの書き込みを行う。
/R	“ ” カセットからの読み込みを行う。
/P	“ ” プリンタへの出力を行う。

は、各項目の最長となるものから決定してあり、入力の際には桁数に注意する必要がある。コマンド処理状態では、入力されたコマンドがシミュレータに内蔵された11種類のいずれかを判断し、各コマンドに対応した処理を行うものである。直接実行コマンドを除いたすべてのコマンドは、先頭に“/”記号がつく。各コマンドについては次節で説明する。実行状態は入力待ち状態より直接実行コマンドが入力された場合と、コマンドのうちのトレース実行及びダンプ実行コマンドが入力された場合の3通りある。本シミュレータが起動されている場合には以上の4状態になる。

階における全レジスタの内容を表示しながら行うものである。"/T" は "/D" の実行を1ステップずつ行うことを可能にしたものである。以上3つの実行命令では、各ステートメントを何回ずつ実行したかを示すカウンタ機能を付加することも可能である。このダンプ実行やカウンタ機能を使用することにより、レジスタの値の実行の進行に伴った変化やループの回数を知ることができるため、プログラムのデバックや実行状況を確認するうえで非常に有効であると思われる。"/L" はプログラムリストを CRT 上に表示するコマンドであるが、1ステートメントずつの追加、挿入および削除の編集機能をもっている。"/E" はステートメントの訂正をするためのコマンドである。"/X" は CAP-X のもつ各レジスタの内容を調べるコマンドであり、また、各レジスタの初期設定も可能なため、プログラムの実行後におけるレジスタ値の変化をみることもできる。"/M" はメモリ内容の変更や表示を行うコマンドである。プログラムの実行結果等によるメモリ内容の確認のために使用する。"/O" 及び "/I" はプログラムリストのディスクへの書込み及びディスクからの読込みを行うコマンドである。"/W" 及び "/R" はカセットテープへの書込み及び読込みを行うコマンドである。"/P" はプログラムリストのプリンタ出力やダンプ実行やトレース実行の結果をプリンタに出力させるためのコマンドである。

4. システムの処理手順

図3に本シミュレータの処理手順の概要を表わすジェネラルフローチャートを示す。システムを起動すると、キー入力待ち状態になり、プログラムリストかコマンドの入力が可能になる。1行入力がなされると、プログラムかコマンドかが判断され、プログラムリストの場合には再び入力の続行が可能となる。コマンドの場合には、実行コマンドか他の処理コマンドかを判断し、処理コマンドの場合には更にどの処理コマンドかが判断されて各コマンドに応じた処理がなされ、再びキー入力待ち状態になる。ただし、処理コマンドのスペル等の誤りがあればその表示を行う。実行コマンドの場合には、初めに1ステートメントずつの行番号、ラベル欄、オペレーションコード及びオペランド欄の分割処理が行われる。CAP-X はラベル使用が可能なため、本システムの実行の処理は2パス方式で行っている。従って初めにプログラム中の各ラベルと行番号の対応テーブルの作成を行っておく。実行時にはこのテーブルを参照しながらプログラムの処理が進められる。次に5種類の擬似命令の処理に移る。初めに "START" 命令の処理が行われ、プログラムの先頭番地の値が記憶される。次に "END" 命令の処理が行われ、オペランド欄の値によりプログラムの実行開始番地が算出され、プログラムカウンタに格納される。実行時にはこの先頭番地より開始される。続いて "CONST", "RESV" 及び "ADCON" 命令のいずれかが判断され、"CONST" 命令であれば、オペランドの16進定数が10進定数に変換されてラベル名とともに登録される。"RESV" 命令であれば、オペランドに書かれている数だけメモリを確保し、確保されたメモリの先頭番地とラベル名が登録される。"ADCON" 命令であれば、オペランド欄にあるラベル名の番地が求められ、これがラベル欄に書かれているラベル名とともに登録される。以上の処理が終わるとプログラムリスト及び擬似命令処理により登録された内容が表示される。続いて実行コマンドのうち "S" か "/T" か "/D" かが判断され、各実行処理が行われる。この時に構文上の誤りがあるとそのエラーが表示され、キー入力待ち状態になる。エラーが

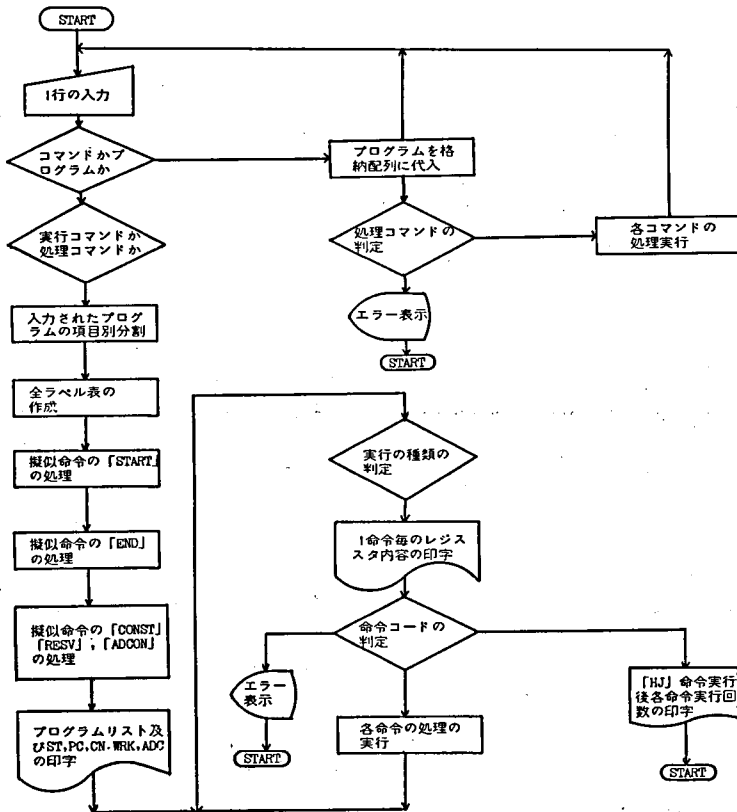


図3 システムのジェネラルフローチャート

なく実行が終了すると各ステートメントの実行回数が表示されて、再びキー入力待ち状態になる。以上のように大きく分けて、各コマンドの処理を受けもつ部分と実行処理を受けもつ部分との2つの部分より構成されている。

5. プログラムの実行例

図4に本シミュレータによるプログラムの実行例を示す。図の例は簡単な乗算問題であり、被乗数を1レジスタに格納し、乗数を定数として与え乗数分だけ被乗数を加算して求めるといものである。このプログラムをダンプ実行した結果である。プログラムのラベル及び擬似命令の処理が終了すると、プログラムリスト及びラベル関係の結果(a)が表示される。次にステートメントの処理に入ると各ステップにおけるレジスタの内容(b)が表示されていく。最後の“HJ”命令を実行後、各ステートメントの実行回数(c)が表示されて終了する。

```

stp=0010      START 32
stp=0020      LAI    2,3
stp=0030      SUB    2,C01
stp=0040      LAI    1,0
stp=0050 JP1   ADD    1,AA
stp=0060      SUB    2,C01
stp=0070      JC     2,JP1
stp=0080      ST     1,ANS
stp=0090      HJ
stp=0100 AA    CONST 0008
stp=0110 C01   CONST 0001
stp=0120 ANS   RESV  1
stp=0130      END
cn=AA 8
cn=C01 1
cn=
wrk=ANS 100
wrk=
adc=
pc= 2          st= 32
0010 1 GR(0)= 0 GR(1)= 0 GR(2)= 0 GR(3)= 0 CC=0
0020 2 GR(0)= 0 GR(1)= 0 GR(2)= 3 GR(3)= 0 CC=0
0030 3 GR(0)= 0 GR(1)= 0 GR(2)= 2 GR(3)= 0 CC=0
0040 4 GR(0)= 0 GR(1)= 0 GR(2)= 2 GR(3)= 0 CC=0
0050 5 GR(0)= 0 GR(1)= 8 GR(2)= 2 GR(3)= 0 CC=0
0060 6 GR(0)= 0 GR(1)= 8 GR(2)= 1 GR(3)= 0 CC=0
0070 7 GR(0)= 0 GR(1)= 8 GR(2)= 1 GR(3)= 0 CC=0
0050 5 GR(0)= 0 GR(1)= 16 GR(2)= 1 GR(3)= 0 CC=0
0060 6 GR(0)= 0 GR(1)= 16 GR(2)= 0 GR(3)= 0 CC=0
0070 7 GR(0)= 0 GR(1)= 16 GR(2)= 0 GR(3)= 0 CC=0
0050 5 GR(0)= 0 GR(1)= 24 GR(2)= 0 GR(3)= 0 CC=0
0060 6 GR(0)= 0 GR(1)= 24 GR(2)= -1 GR(3)= 0 CC=1
0070 7 GR(0)= 0 GR(1)= 24 GR(2)= -1 GR(3)= 0 CC=1
0080 8 GR(0)= 0 GR(1)= 24 GR(2)= -1 GR(3)= 0 CC=1
*** End Program ***
++++ シェック カイスク ++++
0010      0
0020      1
0030      1
0040      1
0050      3
0060      3
0070      3
0080      1
0090      1
0100      0
0110      0
0120      0
0130      0

```

(a)

(b)

(c)

図4 プログラムの実行例

6. あ と が き

アセンブラ言語 CAP-X のシミュレータを普及型のマイクロコンピュータの BASIC 言語を使用して作成を試み、一通り動作可能なものが得られた。本シミュレータは単にアセンブラのプログラムを実行するだけでなく、訂正や編集機能及び周辺装置との入出力機能のコマンドを何種類か付け加えられたので、CAP-X のプログラムの作成や実行が BASIC のプログラム並みの手軽さで行うことが可能である。このシミュレータを活用し、実際のマイクロコンピュータ上で CAP-X によるプログラムを作成し、実行してみることで、コンピュータの基本的動作の確認ができ、また、CAP-X の習得も容易になると思われる。更に、CAP-X によるプログラムが実際に実行可能となったため、アセンブラ教育の入門用として取入れられやすくなると思う。

しかしながら、本シミュレータについて十分な検討を行っていないため、完全なものとは言い難い。例えば、ADD や SUB 命令によるオーバーフローあるいはアンダーフロー対策や HJ 命令のオペランドは省略する条件が加えられているので、これらを除去する点等である。これ以外にもまだ見つからない誤りがあるかと思われるので今後十分な検討を加えていくことが必要であると思われる。

最後になりましたが、今回の CAP-X 用シミュレータの試作にあたり、御指導していただいた東京大学の清水留三郎教授に厚くお礼申し上げます。

参 考 文 献

- (1) 甘利直幸：COMP-X プログラミング，(1981)
- (2) F.R.A. Hopgood：Compiling technique，(1969)
- (3) 岸田孝一：プログラミング序説，(1969)