

## 情報工学系学科の計算機初期教育における EDSAC の活用の試みと効果†

清水 敬子\*\* 阿刀田 央一\*\* 高橋 延匡\*\*

情報工学を専攻する学生に対する計算機初期教育として、EDSAC を用いた機械語によるプログラミング教育を行っている。この教育の目標としては、(1)ストアード・プログラム方式の計算機で実現可能なことの限界を実感として理解させること、(2)計算システムに関する知的好奇心をわかせること、(3)技術の発展の方向を認識させること、などを目的としている。本報告では、とくに上記の実現のため、カリキュラム上の考慮からはじめて、実際の教育、とくに演習を通して、いかに実施しているかについて述べている。そのために、(1)ミニコン化 EDSAC の開発、(2)TSS による EDSAC の仮想計算機の開発、(3)EDSAC の CAI システムの開発、などの教育用ツールの開発と充実を行った。これら 3 種のツールの特徴を生かして使い分け、過去 4 年間、実際に初期教育を行った結果、学習内容およびツールの組合せ方などについて効果的な方法が確立され、教育目標も達成できたと思われる。その経緯を報告する。

### 1. はじめに

大学における情報工学系の学科で「計算機システムに関して、何を最初に教えるべきか」という問題は種々討論され、試みられているが<sup>1)-5)</sup>、本論文では、東京農工大学工学部数理情報工学科で行っている EDSAC<sup>22)</sup> (付録参照) によるプログラミング教育について述べる。

### 2. EDSAC を用いた教育

#### 2.1 教育理念とその実現<sup>8),15)</sup>

##### 2.1.1 教育目標

なぜ EDSAC を計算機システムの教育に選んだかを以下に述べる。計算機システムに関する教育で、かつ最初の教育の基本的な目標は、

- (1) ストアード・プログラム方式の計算機（機械語）で、実現可能なことの限界を実感として理解させること
- (2) 計算機システムに関する知的好奇心をわかせること
- (3) 実在するか、実在した計算機システムを用いて教育すること
- (4) 技術の発展の方向を認識させること
- (5) 教えることは実用上、最小限にとどめること

† The Use of EDSAC for an Introductory University Course on Computers and Programming by TAKAKO SHIMIZU, OICHI ATODA and NOBUMASA TAKAHASHI (Department of Information Science, Faculty of Technology, Tokyo University of Agriculture and Technology).

\*\* 東京農工大学工学部数理情報工学科

である。

#### 2.1.2 EDSAC 選定の理由

(1), (2), (3) を満足させる計算機システムとして、筆者らは、ストアード・プログラム方式の事実上の最初の計算機として、以後の計算機システムの方式設計に根本的な影響を与えた EDSAC を選んだ。

その理由は、

- (ア) 命令コードとそのアーキテクチャが単純である
- (イ) シングル・アドレス方式である
- (ウ) 数値の表現形式が 2 進固定小数点補数表示方式である
- (エ) ハードウェア技術の制限があり、細部で改良すべき点がある

など、EDSAC は現在の主流の計算機システムの基本となっており、基本的には単純で自然なアーキテクチャである。

(4) に関しては「計算機械 I」(後述) で EDSAC を理解させ、次年度の「計算機械 II」で、技術の発展の方向と、技術的問題をいかに解決したかを、計算機システムの歴史の中で位置づけて理解させることを試みている。おもなアーキテクチャとしては、STRETCH, System 360, MULTICS, ミニコンなどを採り上げ、その設計哲学を論ずる。その際の基本は比較設計学の見地からであり、EDSAC は原器の役割を果たしている。

(5) に関しても、命令数が少なく構造の単純な EDSAC は最適である<sup>17)</sup>。なぜなら、学生の知的的好奇

心を失わせる方法の一つは、すべての情報を与えることである。そうすると、何ら発見の喜びが得られず、教育 (Education) の効果は期待できない。自分なりに考えて納得することが大切である。

### 2.1.3 疑問および反論

EDSAC を用いたプログラミング教育に対する疑問と批判は以下に代表される。

(1) なぜ、そんな古い計算機を教える必要があるのか (ACOS や System 370 のアセンブラとかマイコンならすぐ役立つのに)。

(2) EDSAC のプログラミングはトリッキーなプログラミングの技法だ。現在はプログラムのリーダビリティやメンテナビリティが大切だ。凝ったプログラムを書かれては困る。

(3) EDSAC では計算機がないから実習できない。また、学生が書いたプログラムを読むのは大変だと思う。

これらの疑問や批判に対し、筆者らは前記の理由を回答している。さらに以下に補足しておく。

(1) 計算機の原点を学習することにより大型機にせよマイコンにせよそのアーキテクチャの基本的概念が容易に形成できる。最初に System 370 のような大型計算機システムのアーキテクチャを与えたのでは、学生は計算機の全貌を把握しきれまい。仮想記憶装置の必要性を感じない者に、その方式を説明する時機ではない。

新しい計算機システムは、すぐ陳腐化する。すぐ役立つものはすぐ役立たなくなる。

計算機システムの価格性能比は5年で4倍のペースで改善されつつけている。オペレーティング・システムを含めて、バージョン・アップが行われており、情報工学科の計算機システムの現状は、一世代ないし二世代前のシステムを利用していることすらありうる。その時点で最新鋭機を採用してもすぐ古くなるし、おそらく学生にはそのシステムのどこを改善すべきかに関しては理解できないのではないか。むしろ、一つの計算機を完全に理解することにより、他機種への移行がすこぶる容易になるのが常である。

(2) 第三者が読みやすいプログラムを書くように指導すること、信頼性の高いプログラムを書くことは大切なことであり、むしろ演習のときに、プログラムの書き方を指導する必要がある。これはいかなる計算機を用いようと同じことである。機械語のプログラムの場合はとくに、非構造的プログラムではデ

バッキングなど苦勞が大きい、構造的なプログラミングが大切となることを身にしみて納得させやすい。

(3) プログラミングの演習に労力がかかる。これもいかなる計算機でも同じことである。また、TSSの端末から EDSAC の仮想計算機を用いて多人数の同時実習は可能である。またアーキテクチャが簡単のため、「ミニコン化 EDSAC」の製作も学部卒研程度でも不可能ではない。

これらからわれわれが期待している効果は以下のとおりである。

- (1) 種々の計算機システムに関する学習意欲をわかすこと。
- (2) 計算機システムに関する本質的な理解をさせること。
- (3) 計算機の機械語の理解とプログラミング言語との関係の理解をさせること。
- (4) 新しい計算機への移行が容易にできるようになること。

## 2.2 カリキュラムとの関係

EDSAC を用いた教育は、計算機の本質的構造を理解させるための初期教育であるから、学生が専門課程に進学後、ただちに開始する。すなわち、2年次前期には講義 (「計算機械 I」2 単位) が行われ、またこれに密接に関連した形で2年次前期、後期を通して演習 (「数理情報工学演習 I」通年で2 単位) が行われる。演習 I の時間わくはすべて EDSAC 演習に使われるのであるが、全体から見れば時間数、単位数ともそれほど多いほうではない。なお、EDSAC 演習以外の著者らの学科の実験、演習は文献<sup>6)</sup> に詳しく述べてある。

## 2.3 課題内容<sup>9)</sup>

情報工学専攻の学生としては、計算機がブラックボックスではいけない。間に何段階かのシステムを介在させずに、なるべく直接的に計算機の実体に触れるということが大事である。

具体的な課題の選択にあたって<sup>10)</sup>は、後述の表 1 に示すように EDSAC 仮想計算機およびミニコン化 EDSAC の両ツール (後述) の特色を生かすよう留意している。使用する言語は、EDSAC の初期入力ルーチンが受付ける言語 (A 100 F, S 101 F のような 1 文字ニーモニックのアセンブリ言語のようなもの。以後、機械語という) である。

初期の段階で学習のポイントとなるのは次の 2 点で

表 1 演習課題一覧

Table 1 The programming problems for student practice.

	課 題	内 容	使用するツール			
			EDSAC 仮想計算機	EDSAC 80	EDSAC 7620	CAI
前	数の表現	$n$ 進数 ( $n=2, 8, 10, 16$ ) について、相互の基数の変換を机上で行う。				
	加減乗算の計算機内部での進行のしくみ	アキュムレータ、乗算レジスタ、記憶装置、命令カウンタの働き、データの動きなどの習得。加減乗算を用いたプログラムの作成。擬似命令による定数の作成。	○	○ (拡大パネル)		○
	プログラムの制御符号の判定	絶対値、最小値などを求めるプログラミング。プログラムの制御指令、終端記号の与え方。	○	○	○	○
	プログラムの制御反復計算	Bレジスタ (インデックスレジスタ) に関する命令を使い反復、番地修飾の習得。同時に、初期値の設定やBレジスタを使わないときの命令の書換えも行う。	○	○	○	○
期	データ処理の問題	2組のデータを与え、分類、混ぜ合わせ、照合、寄せ集めなどのプログラミング。	○			○
	命令の合成の問題	番地部や反復回数を任意の値として与える問題のプログラミング。	○			○
	多項式の計算	多項式の計算の技法習得とプログラミング。桁ずらし、尺度因子、長語の把握。	○			○
	平方根表の作成	平方根を求める算法とサブルーチンのリンクのとり方の習得。平方根サブルーチンの作成と平方根表の印刷。	○			○
後	除算ルーチン	漸化式を用いて除算ルーチンの作成。相対番地、丸め、誤差の評価も扱う。	○			
	整数と小数の入力ルーチン	10進で表されている整数、小数を入力する技法の習得。数字、記号の入力符号の扱い。	○			
	整数と小数の出力ルーチン	整数、小数の内部表現 (2進) を 10進に変換して出力させる技法の習得。	○			
	初期入力ルーチンの解説	全員に異なるプログラムを与えて、一つのプログラムを初期入力ルーチンがどう処理するか解説させる。				
期	浮動小数点の加減算ルーチン	浮動小数点の加減算サブルーチンの作成。自作の入出力ルーチンを使う。	○			
	浮動小数点の乗除算ルーチン	浮動小数点の乗除算サブルーチンの作成。自作の入出力、除算ルーチンを使う。	○			
	コード変換ルーチン	EDSAC コードを ASCII コードに変換し、その逆も行う。紙テープを使用。	○	○	○	
	算術式のコンパイルルーチン	加減乗算のみの簡単な算術式をコンパイルする。	○			

ある。

(1) 機械語によるプログラミングを通して、計算機のしくみを体得させること。すなわち、命令を実行していくとはどういうことか。プログラムカウンタの働き、命令の実行に伴うデータの動き、レジスタ類の働きなどを納得させること。

(2) 計算機の中は2進法で動いているということをはっきり認識させ、プログラムの機械語とのつながりを納得させること。

後期には、一つのまとまったサブルーチンをいくつか作成させるが、その学習のポイントは次の2点である。

(1) 入出力装置と計算機本体とのつながり、プログラムやデータの入力、計算結果の出力、紙テープのコードの役割などはどうなっているか納得させるこ

と。

(2) 長いプログラム作成のときの設計および多くのサブルーチンをつなぐときの構成を合理的に行わせること。

演習での課題内容を表1に示す。問題は一課題に数～十数題出題し、数題を教室で解いた後、他の問題は1～2題ずつ異なる問題を各自に割り当てる。

演習では、初期の段階はEDSACのおおのこの命令の働きがわかるように丁寧に課題をこなし、段階ごとに到達度試験を行う。後期は全命令を使い、基本的な技法、算法を折りこんで課題を与えるが、しだいに個人差もでてくるので、多種類の課題を用意して選択させる。各課題は適宜レポートで提出させるが、内容には解法や考察などとともに、必ず完成したプログラムリストとデータも付けさせて、添削指導する。

### 3. 用意したツール

#### 3.1 EDSAC 仮想計算機

EDSAC 仮想計算機は ACOS 600 上に EDSAC の機能をそっくり再現し、TSS 端末機を通して一人の学習者が一台の EDSAC を専有する形になっている。後述のミニコン化 EDSAC とは本体の機能はもとより、操作パネルの表示や記号なども同じにしてある。ただし、ケンブリッジ大学の EDSAC では入出力装置に紙テープ読取機、穿孔機を使用していたが、仮想計算機ではファイルに対応させている。記憶装置としては、ファイルに 1 キロワードの記憶場所を各自、所有している。

仮想計算機の実行手順はコマンドにより与え、命令の実行状態はいろいろなモードを選択し、ディスプレイ装置に表示される。トレース・モードでの実行状態を図 1 に示す。

#### 3.2 ハードウェア実習のためのツール

##### 3.2.1 ミニコン化 EDSAC (EDSAC 7620)

EDSAC 教育の開始当時 (1978 年) は、教育用ツールとしては前述の EDSAC 仮想計算機のプロトタイプのみであった。当時の教育経験から、そのみでは以下の点が不十分であることが判明した。

- (1) 一部に、EDSAC を BASIC や FORTRAN と同列の単なる言語と思いつく学生がいた。
- (2) コードの概念、数値と命令の同値性などを実感として捉えにくかった。

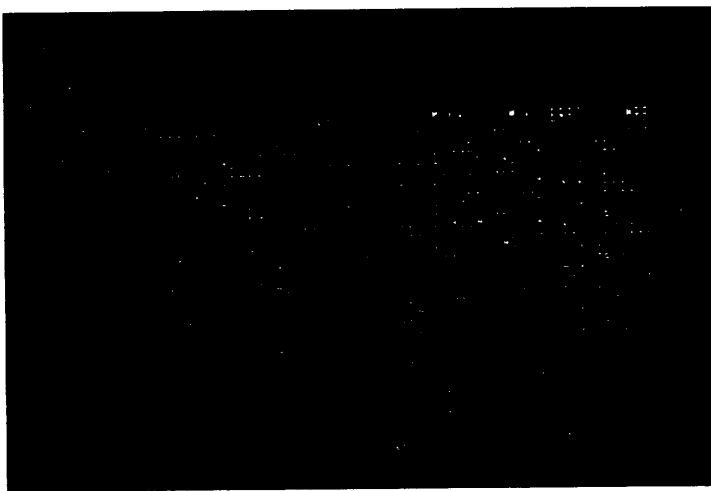


図 1 EDSAC 仮想計算機によるトレース例  
Fig. 1 An example of program tracing on virtual EDSAC machine.

- (3) 演算速度、命令による演算時間の違いを学生に体験的に認識させることができなかった。

上記の問題点から、何よりも学生自身が紙テープあるいはスイッチボードから、自分で箱の中の仕掛けに命令コードをロードし、ロードされたプログラムを自分で 2 進のランプで確認し、このプログラムをシングル・ステップで逐一、あるいは高速で実行させて確認できるような存在感と実体感に満ちたツールが必要であるとの結論を得た。この結論のものに、「ミニコン化 EDSAC (EDSAC 7620)」を試作した<sup>10)</sup>。

EDSAC 7620 はいわば、初期の計算機や裸のミニコンなどを隅から隅まで熟知して使用した技術者の体験を学生に追体験させるという目的のものである。その外観や操作、速度は市販のミニコンに近い。CPU 本体のサイズは W 570 mm×H 1050 mm×D 540 mm である。これにアドレススイッチ、データスイッチ、各レジスタ表示用の LED などが実装された操作パネルが取り付けられている。入出力は、CPU 本体とは別のラックに PTR, PTP, キーボード、プリンタがインタフェース/コントローラとともに収納されている。この外観写真は、文献<sup>7)</sup>ですでに紹介した。図 2 に EDSAC 7620 の操作パネル面を示す。このパネルもやはり、ミニコンの平均的なものに近い。しかし、この計算機は実習用のものなので以下の点に考慮を払ってある。

- (1) スイッチや LED 表示のマルチプレックスは極力さける。すなわち、アドレスとデータは別々のスイッチでセットするようにし、各レジスタ (PC, Acc, M, B および命令レジスタ) およびメモリから読み出されているデータはすべて別々の LED で表示する。

- (2) 学生がプログラムをトレースするのに便利のように、シングル・ステップ・スイッチを設ける。

- (3) 計算機の命令取出し、命令の実行の各サイクルの動作、命令によるサイクル数の違い、入出力装置と CPU の速度の違い等を理解させるのが便利のように可変ロックおよびマニュアルロックを用意した。

その他詳細は図 2 によりほぼ明らかであろう。なお、“INIT IN” スイッチを押すと、かの歴史的な初期入力ルーチンが ROM からメインストレージにロー

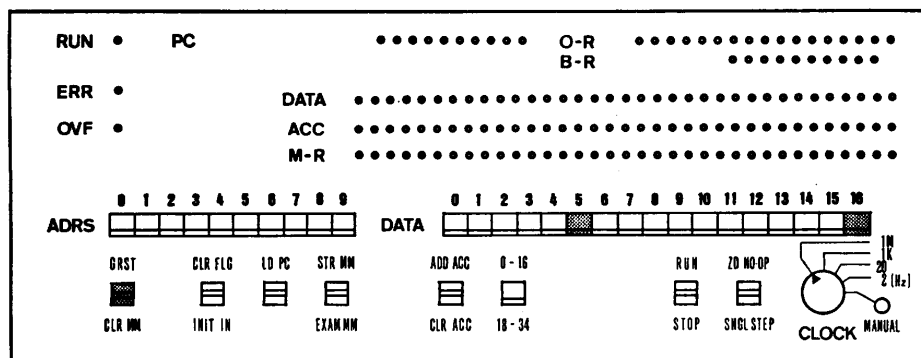


図 2 ミニコン化 EDSAC 7620 の操作パネル

Fig. 2 The console panel of EDSAC returned to life as a minicomputer in educational use (EDSAC 7620).

ドされる。

EDSAC 7620 は EDSAC と完全に命令コンパチブルであるが、内部演算は全ビット並列演算方式を採用している。クロックは 1 MHz, 加算  $2 \mu s$  でこれもミニコンに近い値である。論理演算素子は TTL の MSI, SSI を約 500 個使用して、 $185 \text{ mm} \times 140 \text{ mm}$  のユニバーサル基板 33 枚上に手組みで実装した。メモリは CMOS スタティック RAM (5101) で、実習上の便を考えて Ni-Cd 電池によるバックアップを行っている。全回路図は紙面の都合で示すことはできないので、Acc まわりの一部のみを図 3 に示す。

EDSAC 7620 は後述のように、1979 年度の卒業研究で 6 人の学生によって設計製作され、1980 年度から演習に使用を開始した。その結果、当初予定された教育効果は十分発揮することが示された一方、次のような問題点が明らかとなった。

- (1) 計算機の動きを多人数の学生に一度に見せるのにたいへん不便である。
- (2) EDSAC 7620 1 台では多人数の学生の実習を時間内に消化しきれない。

そしてこれらは結局、教育効果は高いが教官の負担を増加させるという結果をもたらした。

### 3.2.2 教室デモンストレータ (EDSAC 80)

上記の問題をいくぶんでも軽減すべく、第 2 のミニコン化 EDSAC (EDSAC 80) を試作した<sup>13)</sup>。EDSAC 80 は EDSAC 7620 と異なり、次の 2 項目を主目的としたものである。

- (1) 教室デモンストレーション、すなわち教室で教官がシングル・ステップ等で計算機の動きを追いながら説明する。
- (2) EDSAC 7620 と同様の実習。ただし、将来

の複製を考慮した設計としておく。

教室デモの使用に供するため、EDSAC 80 には「拡大ディスプレイパネル」と称する  $1200 \text{ mm} \times 900 \text{ mm}$  のパネルを接続することができる。この拡大パネルは、市販のホワイトボードを土台として、この上に CPU パネルを相似形に拡大した表示ランプを取り付けたものである。CPU パネルのスイッチに相当する部分には、大きなスイッチを取り付けることが技術的に困難なので、CPU パネルのスイッチの動きをコピーするランプを取り付けてある。すなわち、教官がパネルの上で操作したことが、そのまま拡大されてこのパネルの上に表示されるわけである。拡大パネルの外観を図 4 に示す。EDSAC 80 はハードウェアを単純化するため、マイクロプロセッサ (Z-80) によるエミュレータ形式で製作した。そのためミニコンなみのスピードは諦めざるをえないが、代償として可搬性 ( $W 430 \text{ mm} \times H 250 \text{ mm} \times D 430 \text{ mm}$ ) と複製の容易さをうることができた。しかし、このような構造上の違いにもかかわらず、CPU パネルの表示および操作は EDSAC 7620 とほとんど同じにして、学生の無用の混乱を防ぐこととした。EDSAC 80 に使用した IC はマイクロプロセッサ CPU, メモリのほかに TTL 約 170 チップ、エミュレーションプログラムは約 5 KB (スタティック)、加算命令のエミュレート時間は約 4 ms である。

拡大パネルのランプは  $28 \text{ V } 40 \text{ mA}$  の E-10 タイプの白熱球を約  $24 \text{ V}$  で点灯しているが、50 人程度のクラスの最後部でも十分な視認性が得られる。これらのランプの ON-OFF ももちろんエミュレーションプログラム内で行う。

教室内で多数の学生に計算機の動きを示すのに、次のような方式も考えられる。すなわち、教室内に多数

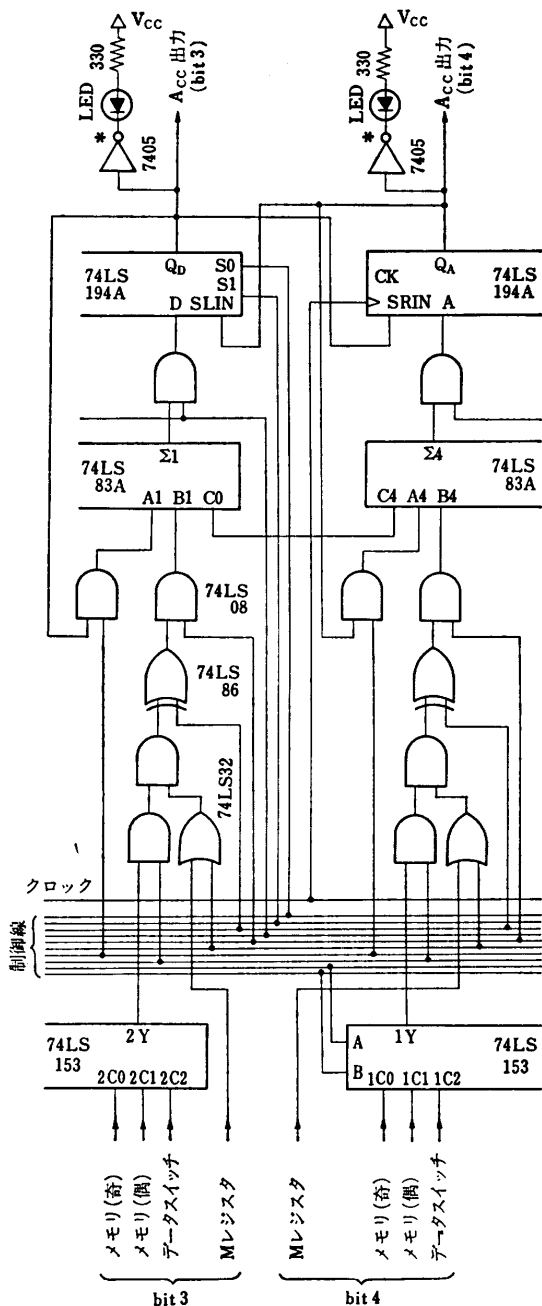


図3 ミニコン化 EDSAC 7620 の回路図の一部 (Acc の一部)  
Fig. 3 A partial circuit diagram of EDSAC 7620.

の TV 受像機を置き、それに計算機のパネルの絵を写し出す方式である。これはエミュレーションプログラムによって Video-RAM 上に画像を作り出し、その出力の標準ビデオ信号を受像機に分配することにより容易に、比較的安価に実現できる。しかし、この方法は学生にとって、鏡の中の世界のように臨場感にと

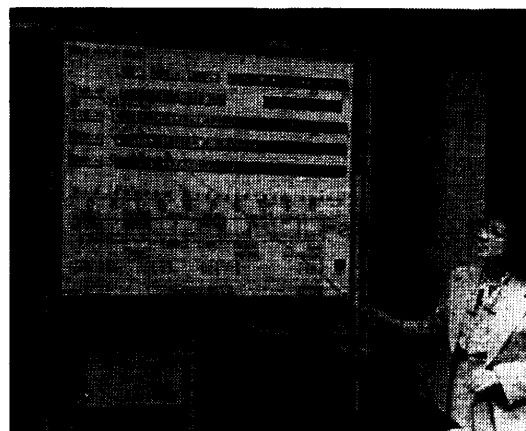


図4 ミニコン化 EDSAC 80 の拡大パネル  
Fig. 4 An enlarged copy of EDSAC console panel for classroom demonstration using conventional white board.

ぼしくないか。学生の視線が前方に集中せず、教室が散漫にならないか。こういった危惧からこの方法は採用せず、ここでは最も古典的で単純な方法—パネルそのものを拡大してしまう方法を採用したわけである。しかし、CRT スクリーン上の絵を用いる方法の考え方は、次に示す EDSAC CAI に活かされている。

拡大パネルの使用実績はまだ十分ではないが、少なくともこれを使用した授業に対する学生の注目度は高く、彼らの興味をかきたてているようである。

なお、ハードウェア実習における紙テープの作成には、専用のオフライン穿孔機を2台用意してある。

### 3.3 CAI

CAI は演習の時間以外に個人の好みと進度に応じて、予習にも復習にも自由に使用できる。演習中に落伍しそうな者や進度の速い者など個人差が生じることへの対応策として、ACOS 600 の端末機用に作成したもの<sup>12)</sup>である。

システムとして目ざしたことは、個人が専有する端末機の良さを生かして、

- (1) 一人で始めたら面白くて止められないような興味をもたせるものにする
- (2) 視覚的にわかりやすいよう図を多用することである。

構成はコースウェア、学習管理プログラムおよび学習者個人のデータファイルから成り、全体をシステムの管理プログラムが制御している。

それぞれの機能を以下に示す。

#### (1) コースウェア

学習内容は演習の前期分の課題に対応しており、16

項目に大別される。各項目は内容に関する説明と演習問題から成り、1項目約10問題、全体で141問題を含む。問題には穴埋め、誤り訂正およびプログラム作成の3種があり、プログラム作成問題を重視している。

#### (2) 学習管理プログラム

このプログラムは、学習者が課題に取り組むとき、学習者との質疑応答を制御する。プログラム作成問題では、学習者のプログラムをその実行結果によって検証する。すなわち、予想される分岐の道すじに対応した2種以上のデータを用い、初期設定の正しさを調べるため2回以上実行させ正答と比較する。誤りがあれば、誤りの種類と個数を、正しければ、プログラムのステップ数、計算時間などを学習者のデータファイルに記録する。

#### (3) 各学習者に関するデータファイル

学習者個人ごとに、学習の過程、学習状況を次の3種のファイルに記録していく。

- (i) 学習者が手がけた問題の表
  - (ii) 各項目ごとの命令別(加算, 減算, 格納, …), 種類別(番地部, 司令部などの書き換え, 削除, 追加など)の誤り件数の表
  - (iii) 完成したプログラムのステップ数, 計算時間, 結果の誤差などの表
- #### (4) システム管理プログラム

コースウェア16項目のうち、はじめの4項目は必修であるが、第5項目以降は、学習者の段階に応じてシステムが学習課題を選んで与えることも学習者の希望により選択することも可能である。このプログラムは学習者の段階に応じて、ある規則に従って課題を選択し、学習者に提示する。

CAI を使用しての分析結果については4.1節で後述する。応答に時間がかかりすぎるとの難点があったが、図による解説は好評であった。

今後、学習の励みにするため、囲碁のような段級制度を取り入れ、学習効果を個人個人にフィードバックする予定である。ある学習内容まで到達すると何級かを与えると表示し、CAI 終了時が基礎コース終了で初段に対応するものとする。

## 4. 教育の効果

ここでは、EDSAC を用いた計算機システムに関する基本的な教育が学生にどのような効果をもたらしたかについて述べる。

教育の効果を論ずる場合に、短期的な効果と長期的な効果の双方の視点から評価する必要がある。

### 4.1 学習効果について

2.2節で述べたように、EDSAC を用いた教育は、授業として「計算機械 I」と「数理情報工学演習 I」と組になって行われる。大多数の学生にとっては、演習を段階的に行わなくては目的とする事柄の理解はできない。

以下、われわれの演習の特色である3種類のツール活用の効果、演習の課題とその学習効果、授業の効果について述べる<sup>11), 14)</sup>。

#### 4.1.1 ツール活用による効果

EDSAC の演習は1981年度までで4年間(ただし、履修学生は延5年分)行われ、最初の2年間は仮想計算機のみ、後の2年間はミニコン化 EDSAC が加わり、1年間は CAI も併用した。

EDSAC 仮想計算機では EDSAC を操作するときの状態をそのまま再現しているので、命令実行時のレジスタ類の内容は2進や8進で表示している。しかし、EDSAC 仮想計算機のみ使用していたときは、プログラム修正による番地のつけかえを BASIC の行番号つけかえのようにエディタで簡単に行いたいとかモニターの形でトレースしてほしいなどという、計算機そのもののしくみを理解させたいという意図するところとずれた意識をもつ者もあった。これは高級言語を扱う場合と同じ端末機が媒体であることからくる弊害と考えられ、それを除く意図もあってミニコン化 EDSAC が製作された。この導入による利点は、

(1) 計算機の中では2進法で命令が実行されていくことを把握させたこと。前期にミニコン化 EDSAC を初めて使うときは、機械語のプログラムを2進表示に変換してスイッチで入力し実行させるが、パネルに点滅する2進のランプを見て、命令の実行の状態、レジスタ類の働きを一目瞭然に理解させたこと

(2) パネル上の可変クロックを使うことにより、命令はいくつかのサイクルに分かれて実行され、さらにサイクル数は命令により異なることを把握させたこと。同時に計算機の処理速度も実感させたこと

(3) 後期にミニコン化 EDSAC を使うときは、入出力装置に紙テープ読取機、穿孔機を使うので、入出力コードの意味を目で確認させたことなどである。

EDSAC 仮想計算機で表示される内容がミニコン化 EDSAC の動作をシミュレートしたものであることに

気が付くと、学生は単に高級言語と比較しての機械語によるプログラミングの不便さをあげつらうことの無意味さを悟るにいたる。

ミニコン化 EDSAC の使用に先がけて、拡大パネルを教室に持ち込んで命令の実行手順を表示したときは、2進で点滅するレジスタ類のランプにどよめきが起り、自分で操作してみたいとの興味をかきたてたようである。

しかし、使用できるミニコン化 EDSAC の台数は2台と制約されるので、要する時間は1プログラムにつき1人約30分、50人で約25時間にもおよぶ。またオフラインでの紙テープの作成、修正にもかなりの時間を要する。実際にはミニコン化 EDSAC による学習には、前期、後期ともに3~4週間をあて、その間、週3日ぐらい演習時間外にも使用している。

一方、演習では、実際にプログラムを作り問題をこなしていくことが大切である。そのためには TSS による仮想計算機をフルに活用すればよい。

ミニコン化 EDSAC で計算機のポイントをおさえ、納得した上で EDSAC 仮想計算機で問題量をこなす、理解の定着をはかる。この一人一人が納得したときが、次の飛躍の基礎となっていく。ミニコン化 EDSAC と EDSAC 仮想計算機の両ツールは車の両輪の働きをしている。

つぎに、CAI については、過去一年間併用したので、その長所、改良点、対象として適当な学習者などが検討できた。

使用した学習者側からは、CAI の中でデータの動きなどを動的に示す図はわかりやすいと好評であった。しかし、現在 TSS の同時使用者が増えると、CAI と学習者との応答時間が、かなり遅くなるという問題がある(1端末のとき約3秒のものが、5端末で約1分に達することがある)。

学習者の項目別、命令別などの誤り表を分析した結果、乗算命令(精度や桁おちなど)、分岐命令(分岐先でのおおの異なる処理方法の誤りと統合時点でのレジスタの処理方法の不一致など)、反復演算(反復回数や初期値設定、ループ脱出など)を学習する場合には誤りが多

いことがわかった。CAI によりこれらの誤りの検出が容易になり、教育へのフィードバックがしやすくなった。

つぎに、CAI はどういう者が使い、どのような効果があったか調べてみる。演習の前半に基礎課程の試験を行ったが、復習が必要と思われたので、前期内容に対応する CAI の活用を奨励した。活用後、後半に試験を行い、点数の伸びを比較した。試験の結果を図5に示す。この図の横軸は前半の試験の正規化された点数(いわゆる「偏差値」)、縦軸は点数の伸び(後半の偏差値-前半の偏差値)を示す。●印は CAI をよく活用した者(30課題以上履修)、×印はあまり活用していない者を表す。この結果をみると、全体的に CAI を活用した者の点数の伸びがよく、その伸びの平均値が活用しない者の伸びの平均値を上回るということをや約90%の信頼率でいうことができる。

さらに図5を見ると、活用した者しない者を含めてほぼ三つのクラス(A, B, C)に分割できそうである。これに他の実験、演習などの成績、態度などを加味し、さらに CAI についての学生の感想を考

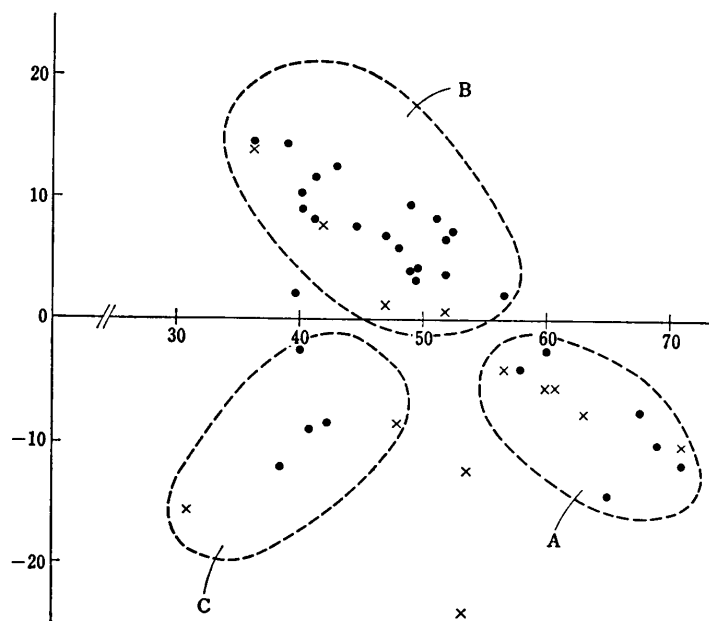


図5 試験の点数の伸び

縦軸：前半の試験の偏差値、横軸：試験の点数の伸び(後半の試験の偏差値-前半の試験の偏差値)、●印：CAIをよく活用した者、×印：CAIをあまり活用していない者

Fig. 5 Improvement in test scores.

Vertical axis represents standardized test scores before exercising and horizontal an improvement of test scores (difference between standardized test scores before exercising and those after exercising). ●: Students well exercised by EDSAC CAI. ×: Students unexercised by EDSAC CAI.



え合わせると、各クラスタの特色がはっきり浮き出てくる。Aクラスタは成績上位の者で、ほとんど CAI を使っていない。これは基礎的部分の理解はすでに済んでおり、応答時間の遅い CAI を使う必要がなかったからであろう。そのなかで CAI を使っているのは、課題履修よりは CAI のシステムそのものに興味をもち、その欠点を列挙してくれたりした者などである。事実、このクラスタ内では、活用した者と活用しない者の伸びの平均値の間に有意な差はない。Bクラスタに属するのは出発は遅々としていたが、少しずつ努力を重ね、必要ならば CAI などのツールも根気よく利用を試み、またその効果も大きかった成績中位の者である。Cクラスタは CAI を使用するとしないにかかわらず、演習のポイントを把握しきれず、勉学にあまり意欲をもてなかった者である。

今後 CAI を改良するときは、コースウェアなどにこの分析結果を反映させたい。

#### 4.1.2 授業全般からの効果の考察

「計算機械 I」の授業は演習とよく同期をとって行われている。具体的には「演習 I」は授業に引き続いた時間で行うといった教務上の工夫も行われている。

「計算機械 I」の授業の目的の一つは、計算機システムに関する原理を理解させるとともに、引き続いて現れる計算機システムを容易に理解させるためのメジャーを確立させることにある。EDSAC 自身の講義は夏休み以前に終了させ、夏休みのレポートとして、「EDSAC の初期入力ルーチン<sup>16)</sup>を解読し、それを FORTRAN で記述し考察せよ」というテーマを与えている。初期入力ルーチンについては解読の仕方、とくにコードの概念を与えるだけである。また、この時点では学生には FORTRAN は教えていないので FORTRAN は独習することになる。

その結果、たった 40 語の初期入力ルーチンが展開してくれる能力に対し、優秀な学生ほど、知的好奇心を大いに燃やす。結果のレポートには、後述のような自分流に工夫した多数の初期入力ルーチンや単純なアセンブラが誕生する。と同時に、授業で、「現在はこのような機械に従属したプログラムを組んではいけない」と教えているが、このようなプログラムが書けるくらいの実力をつけたいという感想をもつ学生が多い。

つぎに、命令数が少ないこと、アドレス空間が少ないことから、こんな命令があったらよいと思うとか、アドレス空間の拡張方式といったアイデアを生み出す学生も多い。これは、引き続き「計算機械 II」の授業

に好影響をもたらしている。

#### 4.1.3 課題進行中の学習効果

学生は与えられた課題をこなしながら、どのように変化していくか、つぎにあげてみよう。

(1) 読みやすく何回も使用できるプログラムを書くようになる。以前のマシンステートによらず正しく動くプログラムの作成には、プログラムの初期化が大切であり、他人も読みやすく保守しやすいプログラムを書くことも大切であることを認識できるようになる。

(2) 計算の精度への関心をもつようになる。計算方法やプログラムの組み方により、打ち切り誤差や桁おちなどが生じることから、数値の扱いは慎重にしなければならないと気付いてくる。

(3) EDSAC にある種々の洗練されたユーティリティルーチン解読により当時の世界のトップレベルの人達の業績にふれることができる。すなわち、演習時に、そのユーティリティルーチンの一部を作成させるので、その際、学生は自主的にこれらのルーチンを解読しながら、EDSAC の設計の影響をうけているようすや、それを巧みに利用して解決しているさまなどを観察する。その結果、種々の制御指令の存在理由やプログラム技法を深く理解し納得するようである。解読による推理が的中するという喜びを味わえるのは実在した計算機だからこそであろう。しかし、誤解のないように付け加えると、課題で学習するプログラムでは決して凝ったプログラムを奨励はせず、読みやすい保守性のあるものを指導している。

(4) 自分で創造する意欲と発想力がつくようになる。演習で与えられた課題、先人のプログラムの解読などに触発されて、より発展させた問題や自分独自に考えついた問題のプログラミングに挑戦していくようになる。初期入力ルーチンに触発されて、現在のミニコンのものに近いアセンブラを作ったり、精度を上げるため倍精度ルーチンを作ったり、種々のダンブルルーチンを作ったり、さまざまな問題にとりつく意欲がでてくる。また、演習の課題で簡単な算術式のコンパイラも作るが、高級言語と機械語との関係、コンパイラの役割も認識できるようになる。

#### 4.2 長期的効果

##### 4.2.1 計算機ハードウェア技術の理解への波及

EDSAC 教育は、もちろん電子回路レベルのハードウェア教育とは関連をもたないが、その上の論理レベル、アーキテクチャレベルのハードウェア教育に対し

ては大きな効果を上げている。とくに、現在最も需要の多いマイクロプロセッサチップを使用してシステムを組み上げるなどの能力は、EDSAC 教育による効果が非常に大きい。

この効果がまず最初に現れるのが、2年次後期実験内の「マイクロコンピュータ」というテーマの実験である。この実験は、スイッチボードと LED 表示のみのついたマイクロコンピュータ (Z-80 使用) に、Z-80 のサブセットの命令体系を使ったプログラムを作ってロードし、走らせるというものである。ここで学生は、アーキテクチャの違いにもかかわらず、ほとんど抵抗なく実験を理解し、進めることができる。ただし、スタックなど EDSAC にない概念については補足説明を行う。

2年次に行った初期教育の効果が、最も大きく現れるのが4年次の卒業研究においてである。卒業研究においては、Z-80 や MC 68000 のようなマイクロプロセッサを使って計算機や制御装置を試作するテーマもある。通常マイクロプロセッサを初めて扱う者は、学生、技術者によらずアーキテクチャや機械語を理解するのに苦しむことが多い。しかし2年次にEDSAC教育を受けた学生の大多数は、マイクロプロセッサを扱うのが初めてであるにもかかわらず、マニュアルの簡単な読合せのみでそのプロセッサに精通できる。そのため、卒研の指導教官は使用しようと思うマイクロプロセッサについて教える手間がほとんどいらない。

EDSAC 7620 や EDSAC 80 を設計、製作した学生も、2年次に EDSAC による初期教育を受けた学生である。彼らが教育を受けた時代は、ツールとしては EDSAC 仮想計算機のプロトタイプのみであった。EDSAC 教育自体、ブートストラップ的に充実してきた。EDSAC 7620 は 1979 年度に 6 人の卒研学生により、EDSAC 80 は 1980 年度に 3 人の卒研学生により設計、製作された。

EDSAC 7620 の設計は、マイクロプロセッサを使うという立場ではなく、計算機自体を作るという立場であった。この場合も卒研開始当初からすぐに、「符号とオーバフローの処理回路はどうするか」というような議論ができるような状態にあった。この研究に取り組んだ学生は全員理解度が非常に高く、積極的に仕事を進めていった。なお、デジタル回路の設計、製作の能力に関しては著者らの学科の、3年次の製作実験<sup>6)</sup>に負うところが大きく、この卒研ではこれと EDSAC 教育が連続的に補完しあった形で効果を発揮した。

このように EDSAC 教育は、今後重要度がますます高くなるハードウェアとソフトウェアの接点を取り扱う能力の育成に関して、大きな効果をもっていることが示される。

#### 4.2.2 計算機ソフトウェア作成技術の理解への波及

3年次以降のソフトウェア教育においても、EDSAC の学習の成果が現れてくるのは、直接的にはマイコン、ミニコンなどでアセンブリ言語を使うときであり、間接的にはソフトウェアを作成、設計するときであろう。

4年次の卒業研究では ACOS 600, HITAC 10 II, MC 68000, Z-80 などのアセンブリ言語も使用するが、抵抗なく理解でき、プログラミングできる。この時点ではマニュアルを与えるだけで命令の働きを理解でき、EDSAC にない命令などの働きも容易に把握する実力をつけている。また、演習課題でコンパイルの意味や浮動小数点演算などの内部表現についても取り扱うので、卒業研究で必要なサブルーチン作成などで直接役に立つ。たとえば、卒業研究でコンパイラ作成中、二つの浮動小数点方式の比較を行った場合もあるが、コンパイル自体についても、浮動小数点演算の取扱いや精度に関する考察も<sup>19), 20)</sup> すぐに本題に入ることができた。また MC 68000 のアセンブラ、シミュレータ、さらに MC 68000 自体のアセンブリ言語学習の CAI<sup>21)</sup> なども卒業研究で作成したが、出発点におけるアセンブリ言語の学習も難なく行っていた。

また卒業生が企業などでアセンブリ言語を使用する場合も、とりかかりが容易であり、EDSAC 使用の経験が大変役立ったとの感想をきく。

次にアセンブリ言語にかぎらず、高級言語などを用いてソフトウェアを作成するとき、その設計をいかにするか、プログラミングの基本的姿勢にも EDSAC 教育の成果は現れている。

EDSAC の機械語のプログラム作成を学習する場合には設計方針を重視して、構造的なプログラムを書くこと、計算時間を考慮することなどを訓練するが、これらの方針は高級言語におけるプログラム作成にも、そのまま適用できる。

また、EDSAC のプログラム作成の場合もプログラムはデバッグして必ず完成したものを提出させているが、卒業研究や製作実験でも完成品を作ること、作ったものが使用にたえることをめざし、現実によくの完成品が役立っている。

たとえば、前述の EDSAC の CAI や MC 68000 の CAI はいずれも卒業研究で作成され、それ以後の学生の自習に役立っている。他にもアセンブラやコンパイラなどいろいろなソフトウェアが完成し、実際に使用されている。これら実際に使用するものを作成したということは、学生にとって実社会へ出ていくときの自信につながっていく。

## 5. 今後の課題

EDSAC による学習を過去 4 年間、試行錯誤してきた結果、学習効果をあげるための課題内容の配列、ツールの活用法などが確立できた。学習内容は段階別にどこを要所におさえて進行させるか、その場合、3 種のツールをどう効果的に組み合わせしていくなどの基礎的なルールは敷設できたと思う。学習効果も十分あったと思われるので、さらに学習内容の充実をはかりながら、計算機初期教育として続行する予定である。

ただ、ミニコン化 EDSAC に実際に触れることが非常に効果的であることはわかったが、現在は時間に追われるように使っているのも、もっとゆっくり自分のペースで操作してみたいと希望する学生が多い。このことを考えると、多人数学生を相手にした場合、技術的な問題もあるが、現在の 2 台よりさらに台数の増設が望まれる。

また、CAI も効果があると判断できるので、これは課題内容をさらに豊富にして充実をはかる一方、現在の CAI システムの難点である学習者との応答時間の長さを改良しなければならない。しかし、データの動きなどを示した動的な画面の多用は効果的であった。そこでグラフィックディスプレイを使用すれば、図形による表示の改良ができると思われるので試みてみたい。さらに、人工知能的な考えも取り入れ、個人学習用として活用できるように強化していきたい。

## 6. 結 言

EDSAC の機械語によるプログラミングを通して、計算機の初期教育を行ってみて、計算機のしくみに関することなどのハードウェアとソフトウェアの密接に関係する部分について、その学習効果を確認できた。学習にあたって、ミニコン化 EDSAC、EDSAC 仮想計算機および CAI の 3 種のツールの威力を十分に駆使できた結果である。多人数学生に効果的に初期教育を行う基本的路線はできたと思う。すなわち、直接ミニコン化 EDSAC に触れさせて、しくみを体得させ

るとともに、多人数同時進行の TSS による EDSAC 仮想計算機を用いてプログラミング技術を定着させる。

1 年後には、計算機で可能なこと不可能なことを理解し、設計の大切さを認識し、さらに自力で種々の命令の存在理由を発見し、新たにほしい命令や機能なども考えられるように発展してくる。これらは、前述の目標にかかげた内容が実現できたことを示している。教えることを最小限にとどめ、一番素朴な形で理解できたことは、さらに一般的なものへの拡張を容易にしている。すなわち、計算機の原点を学習したことは、技術の発展の方向を認識できるようになり、それ以後のソフトウェア、ハードウェア両方の技術獲得への応用力を培い、自力で新たに発想する力や意欲が育ってくる。専門課程へ進む学生の最初の関門であり、学生、教官ともに根気のいる地味な学習であるが、以後の教育の基盤となる有用性を確認できた。

**謝辞** おわりに、これら学習効果を高めている強力なツールの開発にたずさわってくれたのは、1979 年度卒業生でミニコン化 EDSAC 7620 を開発した、赤木京一、荒木邦行、井上 賢、江原 守、大串卓夫、深田正美、1980 年度卒業生でミニコン化 EDSAC 80 を開発した、五十嵐昭、沖田 勉、加藤雄司、1980 年度卒業生で EDSAC の CAI を開発した岩澤京子の 10 人である。すでに 2 年次に EDSAC の教育うけていた人達であるが、これらのツールが演習用として実用化され効果をあげているのは、彼ら卒業生達の逞しい行動力と発想力や意欲があっはじめて可能になったものである。心から感謝する。

## 参 考 文 献

- 1) 阿部圭一：情報学科協議会パネル討論「情報実験」報告，*bit*，Vol. 11，No. 10，pp. 98-99 (1979)。
- 2) Kimura, I.: A Historical, Generalistic, and Complementary Approach in Introductory Computer Science Education, ACM, Inc. from *SIGCSE Bulletin*, Vol. 10, No. 3, pp. 93-99 (Aug. 1978)。
- 3) 島崎真昭他：情報工学科におけるプログラミング実習の一例，情報処理学会論文誌，Vol. 21，No. 2，pp. 83-90 (1980)。
- 4) 宇津宮孝一他：TSS 環境におけるプログラム実行演示システムの作成とその検討，情報処理学会論文誌，Vol. 23，No. 1，pp. 25-34 (1982)。
- 5) 有山正孝他：パネル討論会「ソフトウェア技術者の養成」，情報処理，Vol. 21，No. 10，pp. 1088-1106 (1980)。
- 6) 小谷善行他：情報工学科における実験・演習の一設計例，情報処理学会論文誌，Vol. 22，No. 5，

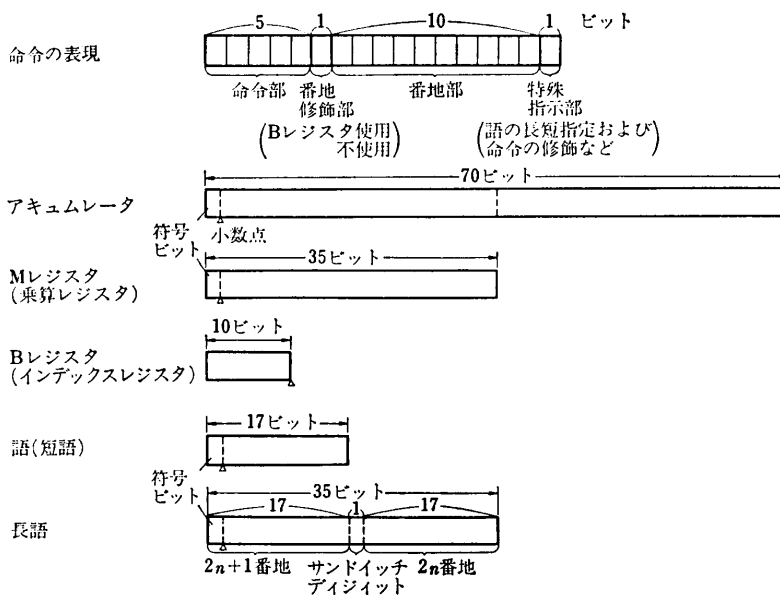
- pp. 402-410 (1981).
- 7) 高橋延匡他：工学系学科におけるソフトウェア・ハードウェア実験, *bit*, 臨時増刊「情報工学の教育・研究」, pp. 56-59 (1980).
  - 8) 高橋延匡：EDSAC によるプログラミング教育の方針, 情報処理学会第21回全国大会講演論文集, pp. 1221-1222 (1980).
  - 9) 清水敬子, 高橋延匡：EDSAC によるプログラミング教育の実際, 同上, pp. 1223-1226 (1980).
  - 10) 井上賢, 阿刀田央一：教育用 EDSAC ハードウェア・シミュレータ, 同上, pp. 1225-1226 (1980).
  - 11) 清水敬子, 高橋延匡：EDSAC の演習のトータルシステムについて, 情報処理学会第22回全国大会講演論文集, pp. 1073-1074 (1981).
  - 12) 岩澤京子他：EDSAC の演習援助用システムの基本設計, 同上, pp. 1075-1076 (1981).
  - 13) 加藤雄司他：教室デモンストレーション用 EDSAC シミュレータ, 情報処理学会第23回全国大会講演論文集, pp. 1129-1130 (1981).
  - 14) 清水敬子：EDSAC の演習システムにおいて効果的な方法, 情報処理学会第24回全国大会講演論文集, pp. 1217-1218 (1982).
  - 15) 高橋延匡：大学で独創力を養おう, *bit*, Vol. 10, No. 6, pp. 666-669 (1978).
  - 16) 和田英一：EDSAC のイニシャル・オーダー, *bit*, Vol. 4, No. 5, pp. 381-389 (1972).
  - 17) 清水留三郎：私が影響をうけた一冊の本 “The Preparation of Programs for an Electronic Digital Computer”, 数学セミナー, Vol. 6, No. 4, p. 43 (1967).

- 18) 森口繁一, 清水留三郎：OKISIP 基本プログラミング第4版, p. 64, 東京大学出版会, 東京 (1964).
- 19) 篠田佳博他：MC 68000 上での浮動小数点演算の実現—IEEE 方式と浜田方式の実現—, 情報処理学会第24回全国大会講演論文集, pp. 933-934 (1982).
- 20) 藤森英明他：MC 68000 上で実現した浮動小数点演算方式の比較—IEEE 方式と浜田方式の比較—, 同上, pp. 935-936 (1982).
- 21) 本間幹男他：MC 68000 教授用 CAI システムの基本設計, 情報処理学会第22回全国大会講演論文集, pp. 1087-1088 (1981).
- 22) Wilkes, M. V. et al.: *The Preparation of Programs for an Electronic Digital Computer*, 2nd ed., p. 238, Addison-Wesley Publishing Company, Inc., U. S. A. (1957).

付録 EDSAC について

EDSAC<sup>22)</sup> (Electronic Delay Storage Automatic Calculator) は, 1949年イギリス, ケンブリッジ大学の M. V. Wilkes 教授らによって開発された世界ではじめてプログラム内蔵 (stored program) 方式を採用した計算機である。入出力装置は紙テープによる読取機, 穿孔機を用い, 記憶装置には水銀遅延線を利用していた。記憶容量は1024語, 1語17ビットから成り, レジスタ類, 命令および数値の表現は図6に示すとおりである。

(昭和57年5月14日受付)  
(昭和57年11月8日採録)



命令数: 23種 (加減乗算, 論理乗積, Acc およびBレジスタを条件とした分岐命令など) 記憶容量: 17ビット×1024語, 実行速度: 加算命令 1.4ms, 構成素子: 真空管 3,000本

図6 EDSAC のレジスタ, 命令, 数値表現など  
Fig. 6 Accumulator, registers, instruction format, word organization and other specifications of EDSAC.