KEEPING STUDENT RECORDS ON A COMPUTER SYSTEM

by

P. Arunasalam

A Graduation Exercise submitted in partial fulfilment of the requirements for the Degree of Bachelor of Economics in the Faculty of Economics and Administration

UNIVERSITY OF MALAYA

AUGUST, 1970

In presenting this graduation exercise in partial fulfilment for the Degree of Bachelor of Economics at the University of Malaya, I agree that the libraries of the University of Malaya shall make it available freely for reference and study or for restricted reference and study. I further agree that permission for extensive copying of this graduation exercise for scholarly purposes may be granted by the Dean of the Faculty of Economics and Administration or his representative. It is understood that copying or publication of this graduation exercise for financial gain shall not be allowed without my written permission.

....
P. Arunasalam

Faculty of Economics and Administration,
University of Malaya.
29th August, 1970.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER I

INTRODUCTION

The development of computers have made possible a new kind of Industrial revolution. The first revolution relieved man of much of his manual labour and gave him powerful tools that will help him accomplish great and otherwise impossible physical feats. The new revolution gives him powerful tools that will relieve him of much of his mental durdgery and make it possible for him to use his mind more profitably.

In the scientific field, for example, the computer has make it possible things that could not have been done otherwise. Problems that would require many man-years of human computation are solved by computers in a matter of minutes. The spilt second calculations required to direct the Appolo 13, when it had a mishap in space, safely back to earth would not have been possible if it had not been by computers. Humans may lose their efficiency under the grip of emotion, but computers being machines, will do their job without being "flustered".

Computers were originally developed for use in scientific research. But the tremendous advantages offered by them in other fields, mainly in business, have made computers popular. The computer is a great aid to man in performing routine and repetitive tasks more efficiently. The principles of uniformity, standardization, and numeralization have become the keys of successful performances in many fields. Imagination, creativeness and clever innovations may be used in building the machines, but once they are built, the "human" qualities must be replaced by the "inhuman" qualities surrounding standardization of performance. This is necessary if the machine, especially the computer, is to be used in accomplishing the work for which it was designed.

The electronic computer has taken the drudgery of work out of many fields; some of the more popular of which are the following:-

In police work: the speed and accuracy of results given by a computer is a great asset in police work. In America, computers are used to store and process arrest reports, recovery reports, traffic tickets, gun records and many other active fields records. Thus up-to-date information is made available in a matter of seconds.

The Police Department in Malaysia is to soon have a computer system of keeping records.

Computers are also used in Post Office and mail transportation in America. Book-composition for printing is also done by computers. They are also used in the real-estate business; for manufacturing bread (to see that correct proportion of ingredients are added properly mixed etc); in the clearing of checks in banks; in flight and hotel reservations; in air-traffic control which is a problem in the major air-ports; in weather forecasting and recently, even in compiling horoscopes.

To obtain an idea of the capability of the computer, the following example would be illustrative. In 1932, Richardson of England developed the necessary mathematics for predicting the weather; but it would have taken 60,000 mathematicians working together in a stadium to handle the calculation necessary to predict the weather for one small part of England alone. The advent of computers has made it possible for the United States Navy to use this method to plot weather prediction for the entire Northern Hemisphere in 40 minutes. Thus it can be seen the potentiality of computers is enormous.

Keeping Records

"Keeping Records" is here used as a general term to denote the following processes done to raw data. They are

1) classifying and coding;
2) recording;
3) sorting;
4) summarising;
5) calculating and communicating.

These are the processes involved in the keeping of any records, but specially so in the case of students records - where marks have to be added, averaged, put in order of merit, etc. The appropriate term for this is data processing. If the processing is done by an electric computer, it is known as electronic data processing.

For the purpose of this graduation excercise, "keeping student records" is taken to be synonemous with "Students' Data Processing".

Data can simply be defined as any bit of intelligence or piece of information; data may be any number, any letter or symbol or any combination of numbers, letters and symbols. All data would be the combination of the following ten numbers

0, 1, 2, 3, 4, 5, 6, 7, 8, 9,

twenty six letters

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

and about 14 symbols,

(& . - @ $ / , % # + ¢ =).

This fifty or so characters constitute the raw material that is presented as data. The data which is the source material for a data processing system is known as input data, and the data after processing is known as output data. The various steps involved in data processing can now be examined in detail.

Classifying and Coding

This basic data processing identifies the data according to its proper catagory (classifying) and converts it into short uniform equivalents (coding). Abreviation is the object of coding, in order to make subsequent processing steps more efficient.

The identification of a student by his matrix number instead of his name is an example of coding. It is easier to deal with a six digit number rather that a name involving perhaps twenty or thirty letters.

Data is classified - i.e. identified as part of a particular category - usually by alphabetic, numeric or alpha-numeric codes. Index number (numeric), car registration (alpha-numeric), exam. grades (alphabetic) are all examples of classification of information by menas of coding. In classifying and coding, accuracy is essential. All subsequent data processing steps are affected by the accuracy of this operation.

Recording

When data is "captured" in a data processing system; i.e. written on a report, punched into a card, etc. it is said to be recorded. Technically speaking, recording is the expression of data on a medium used in the data processing system. "Medium" is a technical term for whatever carries data in the system (punch cards, tapes etc). Each time data is put on a medium, it is recorded.

Sorting

When the data is arranged or put into sequence, it is said to be sorted e.g. sorting students by sex. Two or more files rearranged into one file is known as "merging". Another variation of sorting occurs when a record selected from one file is added to one selected from another file. This is known as matching.

Summarizing

This is basically a coding operation. Its objective, as reflected in the common use of the term summarizing, i.e. compression of data. The most common example of summarizing is the total or sum of a column of figures.

Calculating

This operation involves the application of arithmetic to data i.e. adding, subtracting, multiplication and diversion.

Usually most of the data processing work involves calculating e.g. payment of fees, payroll figures etc.

Communicating

Whenever data is transported from one point to another, it can be said to have been communicated. This movement of data may occur by human means, electrically as signals over wires, or mechanically as movement of cards through a machine.

Data processing with or without machines, is the manipulation of numbers, letters and symbols through one or more of the basic operations mentioned above, to convert input data into useful output.

This is what we are concerned in keeping student records.

Kinds of Data processing systems

Data processing can be carried out by several menas to accomplish the basic operation. Manual Data processing would mean that the data - processed by human beings.

When mechanical devices are used to assist man in converting input data into useful output data, the data processing is called Mechanical Data Processing. Hand operated cash register or adding machines are examples of mechanical data processors.

Electro-mechanical Data processing is the term applied to data processing systems that use electro-mechanical equipment. Desk-top calculations are of this category.

Finally, the type of processing we are interested in, electronic Data processing is processing with the use of computer. These are machines which electronic circuitery, tubes, or transistors in their design.

At the campus, we have also a Punched-Card System for data processing. However, a computer system is any time better than the punch-card system because of the following reasons:

(1) Speed: A computer, being an electronic device, can process with the speed of light and the design complication of using decimal systems of numbers is overcome by using the briary system of number.

(2) Intercommunication: In a punch-card system, each process would take place in a different machine and the machine (punch-cards) have to be moved by hand. However, in a computer system, all data can be moved across wires, since data is in pulse-train form in the equipment.

(3) Arithmetic Ability: is provided, i.e. addition, subtraction, division and multiplication can be carried out.

(4) Logicability turns out to be a by product of subtraction. Decision making can be based on comparison of one with another. Some relatively subtle alternatives can be chosen based on successive tests as to whether one number is bigger than, less than or equal to another.

(5) Storage, or memory, results from the aclaptation of common devices like tape recorder or ponograph record.

(6) Self control: In all punch card systems, an operator is necessary to carry out each process, however a computer, once correctly programmed, can carry out all the processing by itself without supervision. This have advantage in speed and less chance of human error creeping in.

Due to these factors, a computer system is at present the best means of keeping records (data processing).

Object and Scope of Study

This graduation excercise is an attempt to explain the computer system and how it can be made use of to keep records. "Students' Records" is taken to mean a general term meaning data. It is not specifically defined, and as such, I have not attempted to design a specific system of keeping records. This is merely a discussion on how to go about disigning a system.

Research Methodology

To go about doing the graduation excercise, some knowledge of the computer had to be gained. Hence I attended lectures on computer and their usage given by various lecturers both in the computer centre and also a course on systems analysis given by Dr. Jacobus of the Public Administration Division.

Specific details on the computers were obtained through manuals and various books. Here I spend a number of days in the Statistics Department to obtain some information on how they make use of the computers.

But the main research was done in the computer centre and the Library. One major problem I had to face was that no student's data were processed in the computer centre, othen than that of applicants to enter the University and on payment of fees. Their programmes on payment of fees were not available for study and hence I had to make do with a general study of keeping records.

Organization of Subsequent Chapters

This essay is divided into two major parts - one dealing with the computer, and the other with the system of keeping records. The second chapter is a description of the computer, with its various components. The third chapter deals with programming, and the fourth chapter is a sample programme to process marks of students.

The fifth chapter is an explanation on how to make use of the computer system for processing. The various steps involved are gone through. The sixth chapter is a discussion of file organization techniques which is the esence of record-keeping and finally, we end up with the concluding chapter.

CHAPTER II

THE COMPUTER

An automatic computer is a machine that manipulates symbols in accordance with given rules in a predetermined and self-directed manner. In other words, it is a high speed, automatic, electronic, digital data-processing machine.

The term digital in the defination refers to two features of the symbols in the automatic computer:

(1) all the symbol combinations used follow a modified counting rule, i.e., from one symbol combination it is possible to derive all the other possible symbol combination by some counting process

(2) some of the rules of logic and arithmetic are applicable to the symbol combinations

The term digital is also in contrast to to the term analog. Computers are of two types viz the digital and the analog. Analog computers operate with physical analogy, rather than digitally expressed information. Most computers have a work area on which an electrical analogy (not symbols) of the problem is physically contructed from a selection of electrical components. This analogy is then energised and all parts perform simultaneously to produce a response under given conditions (conditions which are also analogies). The result (response) takes the form of variation in electrical quantities and is often recorded on graphs. In short, an analog computer operates by a form of measuring, whereas a digital computer operates by applying rules in sequence to symbol combinations.

In our case, we are interested only in the digital computer. The IBM 1130 in the campus is a digital computer. For all administrative, business purposes, digital computers are used. Only in scientific field is the analog computer of importance.

It is fairly obvious that digital means to deal with numbers - digits. Our decimal system of numbers has ten different symbols, and this cannot be handled by a computer. We shall examine this more.

Here we have to bring in the term "state". A state can be loosely defined as one or two or more conditions or positions than a thing may be in. For example, a light has two possible state, on or off. The number of wheels in a desk calculator may have ten stable states, one for each of the digits 0 through 9 as well as the state of being in rotation (i.e. instable).

Many computers, including the Campus IBM 1130, are made to operate in two states, i.e. a binary manner. They alternate back and forth between two states. For example, a transistor or a vacuum tube may be either conducting electricity or not conducting, or it may be conducting at a high voltage or a low voltage. On, for example, a magnetic component may be magnetised in one direction or the opposite direction.

By convention, the state of off ("no pulse") can be represented by the symbol 0, and the state of on (pulse) can be represented by the Symbol I. The symbols 0 and I disignates states of a two state device, called bits (Binary digits). A bit serves as a basic unit or measure of information - e.g. about the state of a two state device.

As the computer deals with data processing, it has various components to carry out the different parts of the process. These tasks are input, arithmetic, logic, storage, output and control. The components which carry on these tasks in a computer can be illustrated by a computer block diagram.

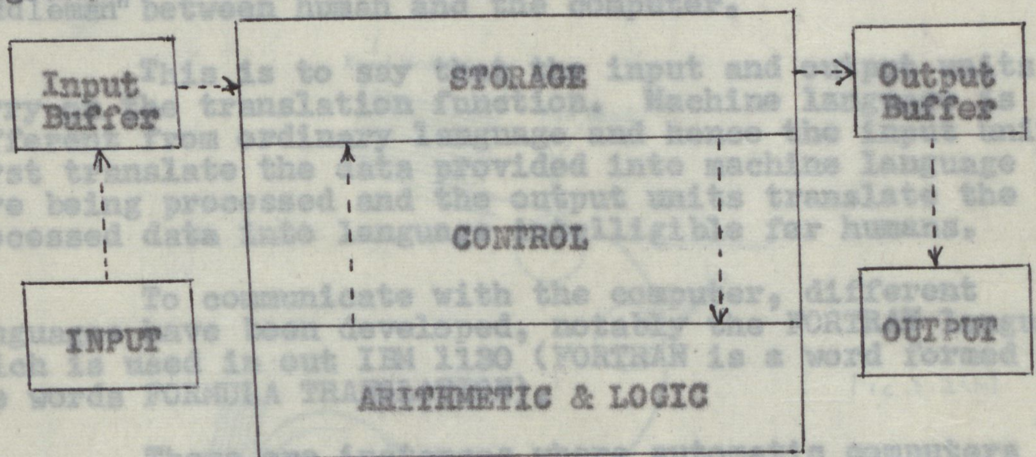


Fig.1 | Computer Block Diagram

There are different input units or devices for reading punch cards, punched tape, magnetic tape, magnetic ink characters, printed characters and even hand written characters.

There are also output devices which can punch cards, punch tape or give out information - the input units i.e. most of the input units can also be used as output units.

Arithmetic and logic units make up the 'computation' part of the computer, i.e. they add, subtract, multiply and divide and also perform logic operations.

Storage devices (memory devices) as their name implies store data or programmes (list of instructions). The most widely used devices are the drum disc, tape and cone storage devices.

The control unit is the most glamorous part of the computer. It has lights, coloured buttons and data flashing on the screen, all relating to the operation of the computer. We shall now examine the components in more detail.

Input and Output Units

The imput devices read the data that is to be processed. The output devices takes the data from the storage unit of the computer and provides the result on cards, tapes or printed on paper. The output and input units act "middleman" between human and the computer.

This is to say that the input and output units carry on the translation function. Machine language is different from ordinary language and hence the input units first translate the data provided into machine language before being processed and the output units translate the processed data into language intelligible for humans.

To communicate with the computer, different languages have been developed, notably the FORTRAN language, which is used in out IBM 1130 (FORTRAN is a word formed from the words FORMULA TRANSLATION).

There are instances where automatic computers receive their input data from other machines, or supply data directly to other machines. In these instances, only "machine languages" are involved. Even then, the input equipment translates the other "machine language" into its own "machine language" and the output unit does the retranslation.

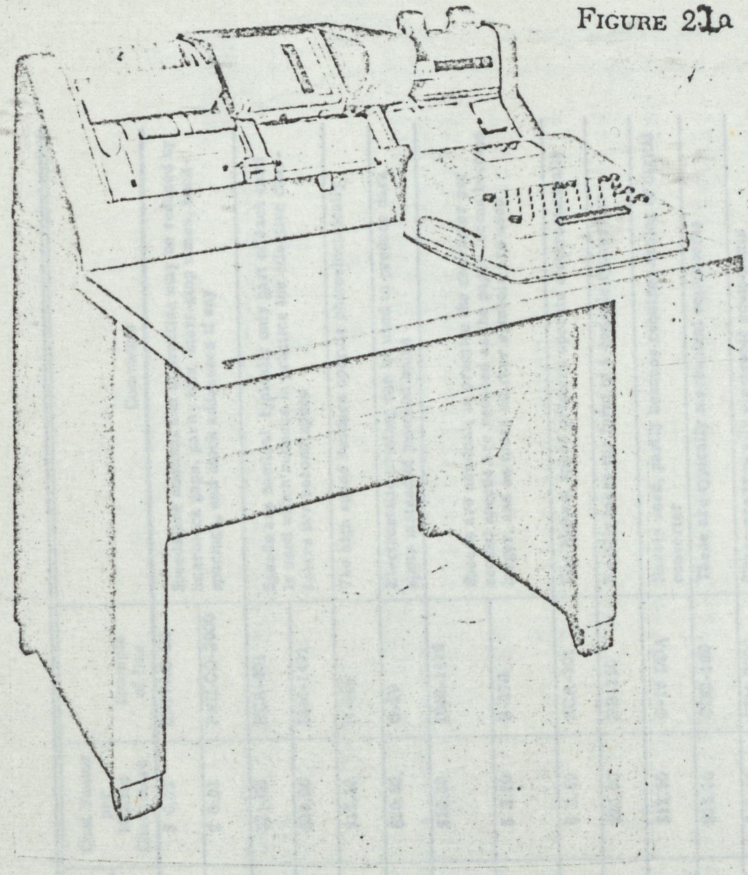


FIGURE 2.1a THE IBM CARD-PUNCH MACHINE. THIS PARTICULAR MODEL INCLUDES A FEATURE FOR PRINTING AT THE TOP OF THE CARD WHAT IS PUNCHED INTO THE CARD.

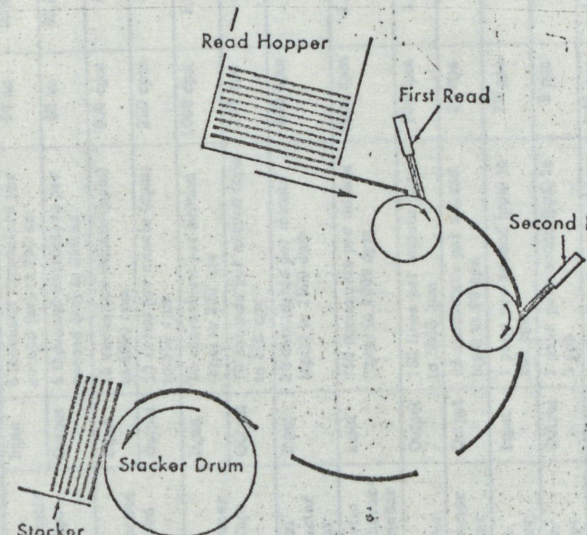


FIG. 2.2(b) THE PUNCHED CARD READER

Equipment	Use	Usual Range of Speeds	Representative Common Speed: As Quoted	Representative Common Speed: In Char. Per Sec.	Cost Factor per Million Characters	Example of Use	Comments
Magnetic Tape	Input	6 thousand characters per second (kc) to 200 kc	60 kc	60,000	$ 0.03	UNIVAC-III	Speeds are nominal and in practice may be reduced by interblock gaps, parity bits, start-stop times, back-spacings, and block addresses if any
	Output	6 thousand characters per second (kc) to 200 kc	60 kc	60,000	$ 0.03	PHILCO-2000	
Punched Card	Input	13 cards per minute (cpm) to 2000 cpm	600 cpm	800	$21.00	RCA-601	Speeds are nominal; typically only part of each card is used which reduces in practice the effective characters per second speed
	Output	13 cards per minute (cpm) to 350 cpm	250 cpm	333	$23.00	IBM-1401	
Paper Tape	Input	10 characters per second (cps) to 200 cps	1000 cps	1,000	$17.50	H-800	The high speed readers operate photoelectrically
	Output	10 seconds per second (cps) to 300 cps	100 cps	100	$20.50	G-20	Electrostatic printing can be used to produce dark spots instead of punched holes
Optical Character Reader	Input	25 documents per minute (dpm) to 1500 dpm	600 dpm	500	$10.00	IBM-1410	Speeds are nominal; in practice the character per second speeds are reduced as the paper forms become larger, and as more marker symbols are used
Magnetic Ink Character Reader	Input	750 documents per minute (dpm) to 2000 dpm	1600 dpm	1,450	$ 3.50	B-270	
Line Printer	Output	150 lines per minute (lpm) to 5000 lpm	900 lpm	1,800	$ 6.50	RCA-301	The highest speed printers operate electrostatically
Character Printer	Output	10 characters per second (cps) to 60 cps	10 cps	10	$20.00	PB-250	Usually takes the form of a console typewriter
Chart Reader	Input	1 point per second (pps) to 500 pps	10 pps	100	$12.50	G-15 DDA	Rarely used, partly because requires analog to digital converter
Graph Plotter	Output	1 point per second (pps) to 5 pps	2 pps	20	$25.00	CDC-160	These are typically mechanical equipments
Display	Output	1 point per second (pps) to 10000 pps	100 pps	1,000	$ 5.00	IBM-7090	These are typically electronic equipments
Transaction Recorder	Input	1 transaction per month to 6000 transactions per minute (tpm)	1 tpm	2	$30.00	NCR-315	Usually used in a satellite manner with a central buffer to improve the effective characters per second speeds
Tag Reader	Input	1 transaction per day to 10 transactions per minute (tpm)	1 tpm	2	$20.00	None on Line	
Keyboard	Input	1 key stroke per hour (sph) to 14000 sph	8000 sph	2	$55.00	NCR-390	Often found as part of a control console, commonly in the form of a typewriter
Direct Input	Input	1 measurement per month to 6000 measurements per minute (mpm)	10 mpm	1	See Comments	TRW-300	Usually requires an analog to digital converter; cost factor is not meaningful
Actuator	Output	1 setting per month to 6000 settings per minute (spm)	6 spm	1	See Comments	IBM-1710	Usually requires a digital to analog converter; cost factor is not meaningful

Fig. 2.3 Summary of major types of input-output equipment

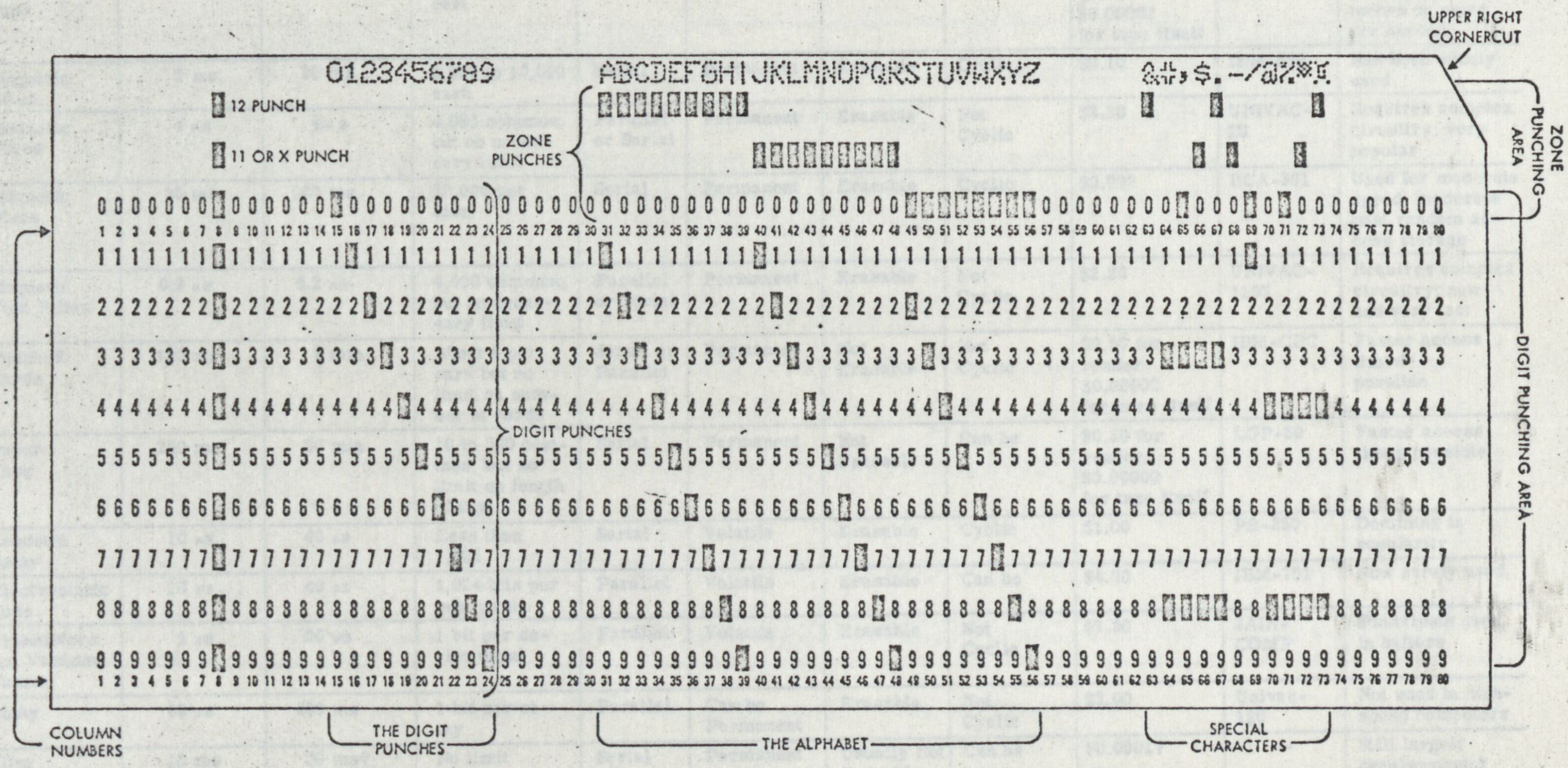


FIGURE 2.4 THE STANDARD IBM CARD AND CARD CODE.

Storage Device	Representative Access Time*	Representative Random Access Time*	Representative Capacity in Words*	Usual Mode	Permanence	Erasability	Cyclic Availability	Equipment Cost per Character Factor	Example of use	Comments
Magnetic Tape	7 ms	2 min	500,000 per reel	Serial	Permanent	Erasable	Can be	$0.06 for drive $0.00002 for tape itself	Honeywell-400	100 bits per channel per inch 50 inches or more per second
Magnetic Drum	2 ms	20 ms	2,000 to 16,000 each	Serial	Permanent	Erasable	Cyclic	$0.10	IBM-650	Has been widely used
Magnetic Cores	4 μs	4 μs	4,000 common, but no necessary limit	Parallel or Serial	Permanent	Erasable	Not Cyclic	$2.30	UNIVAC-III	Requires complex circuitry; very popular
Magnetic Discs	20 ms	300 ms	20,000 per disc	Serial	Permanent	Erasable	Cyclic	$0.008	RCA-301	Used for moderate speed, moderate cost random access storage
Magnetic Thin Films	0.2 μs	0.2 μs	4,000 common, but no necessary limit	Parallel or Serial	Permanent	Erasable	Not Cyclic	$2.25	UNIVAC-1107	Requires complex circuitry; new and very fast
Punched Cards	100 ms	2 min	About 6 per card but no limit on number of cards	Serial or Parallel	Permanent	Not Erasable	Not Cyclic	$0.50 for reader $0.00002 for card itself	IBM-CPC	Faster access times possible
Paper Tape	250 ms	20 min	10 to 500 common, but no limit on length of tape	Serial	Permanent	Not Erasable	Can be	$0.30 for reader $0.00002 for tape itself	LGP-30	Faster access times possible
Acoustic Delay	10 μs	40 μs	Less than 1,000	Serial	Volatile	Erasable	Cyclic	$1.00	PB-250	Declining in popularity
Electrostatic Tube	10 μs	40 μs	1,024 bits per tube usual	Parallel	Volatile	Erasable	Can be	$4.00	IBM-701	Now rarely used
Transistors and Vacuum Tubes	1 μs	20 μs	1 bit per device usual	Parallel	Volatile	Erasable	Not Cyclic	$7.50	JAIN-COMP	Sometimes used in buffers
Relay	50 μs	500 ms	1 bit per relay	Parallel	Can be Permanent	Erasable	Not Cyclic	$3.00	Univac-120	Not used in high-speed computers
Film	10 ms	50 ms?	No limit	Serial	Permanent	Usually Not Erasable	Can be	$0.0001?	—	Still largely developmental
Cryogenic Devices	0.2 μs?	0.2 μs?	No necessary limit	Parallel or Serial	Permanent if kept cold	Erasable	Not Cyclic	$10.00?	—	Still largely experimental

*For words ten decimal digits in length.

Fig. 25 Summary of major types of storage devices

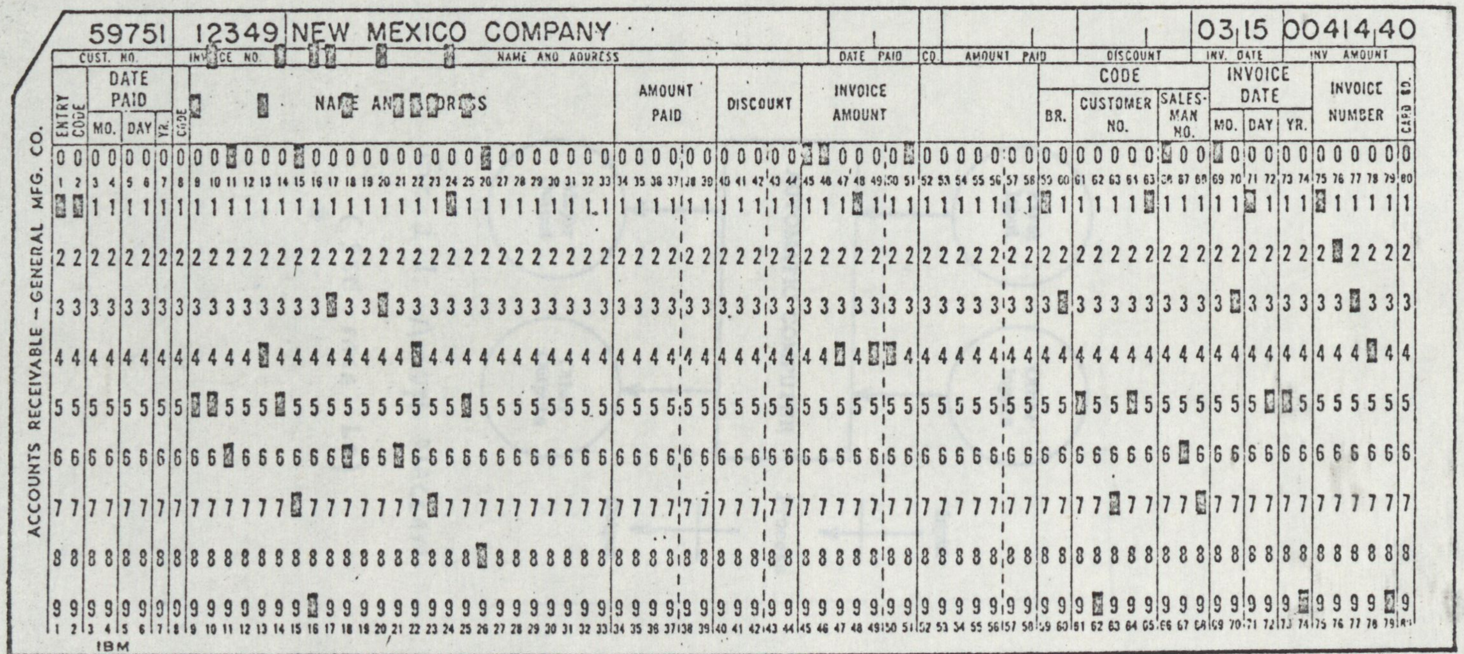


FIGURE 26 AN IBM CARD WITH INFORMATION WHICH IS PUNCHED INTO THE CARD ALSO PRINTED AT THE TOP OF THE CARD. NOTE THE PRE-PRINTED IDENTIFICATION OF VARIOUS SECTIONS OF THE CARD.

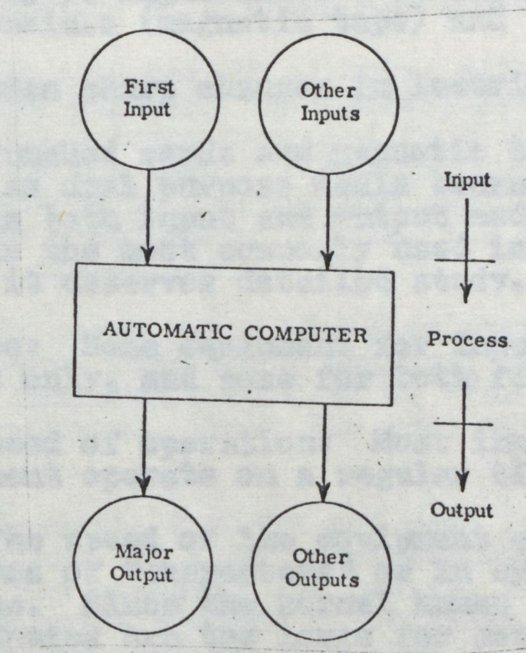


FIG. 2.7. A RUN DIAGRAM

(SEE PAGE 60)

There are six major attributes of input output equipment. They are the following:-

(1) How the units operate. All data which is communicated to the computer take the form of patterns or media. The most common patterns or media are:-

a) ink on paper (such as number, letters or special symbols)

b) holes punched into cards or paper tapes

c) spots of magnetization on magnetizable materials (magnetic tape) and

d) sudden sharp changes in lectrical voltages

Punched cards and magnetic tapes are known as dual purpose media because they are used as both input and output media. The punched card is the most commonly used input media and as such, it deserves detailed study.

(2) Its use: Some equipment for input only, some output only, and some for both functions.

(3) Its speed of operation: Most input and output equipment operate on a regular time-sequence.

The speed of the equipment can be measured in terms of "characters" or in cycles per unit of time. Since the normal human time of seconds and minutes are too large for computer speeds, milliseconds are used as measures of time. The speed taken using various input/output media are shown in FIG. 2.3

(4) Its built in checking facility. Some form of errors can be detected by the input-output equipment itself.

(5) Its buffering: The translation function of the input-output equipment is carried on by the buffers part of the equipment

(6) Its cost: Since there is no comparable basis for measurement of using various units, any such measurements are not really valid.

Punched Cards

Punched cards, as said earlier, is the most common input media. Its most popular because of its low cost.

Punched cards are pieces of cardboard with specific dimensions and thickness, 3 1/2 inches wide, 7 3/8 inches long and 0.007 inches thick. There are two major types of punched cards, Remington Rand and IBM. The IBM system, which is used in the campus, uses cards with rectangular holes; the other system uses cards with round holes. This is the only difference in the two types of cards. The card-code concepts, reading of cards, equipment and effectiveness of systems are similar in both cases.

The IBM card has 80 columns, with twelve position in each column for punching the rectangular holes. The Rand Cards have 90 columns arranged in 2 backs of 6 rows each, 45 column to the Back. Although these are the standard forms, variation in columns per card, size, colour, corner cut can be obtained. Of course, the physical dimension of the card cannot vary.

The IBM cards has numbers printed on it. The 80 columns are numbered with small numbers, while ten of the twelve rows are numbered from 0 to 9. The 11th and 12th rows, are not numbered. These two rows exist, just above the 0 row.

The eleventh, twelfth and zero positions are called zone punches, and the other nine positions are called digit punches. The eleventh and twelfth positions are not numbered because usually other information can be printed there for normal use, and the presence of numbers would make reading difficult (see fig. 2.4).

Card Codes

The numbers, alphabets and other symbols are coded into a hole or conbination of holes punched on the card. The pattern of which always represent the same characters. The position of the holes in the is the basis of the whole data processing systems, and hence the card dimensions are held in strict precision.

Numbers

Any of the ten digits (0 - 9) is represented in the card by a hole in the any of the column to represent the digit.

Example: A hole in number 9 position will represent the number 9 to any IBM Card machine.

This means that only one digit can be represented by any one column on the card. Any number of more than one digit will be represented by the same number of columns as there are digits in that number i.e. a four digit number say 4732, will be represented by 4 holes in 4 columns.

Letters

Alphabets are represented by two holes in any one column. One hole would be in any of the digit positions (1 - 9) and one in the zone positions (0, 11 and 12) in the same column.

Example: "A" is represented by a hole in position 12 and another in position one. The letter "B" is represented by a 12 and a 2 punch. The letter "C" is 12 - 3 and so through until 12 - 9, which is the letter I.

Then we move to the 11th zone and 1st column to represent J, and move consequetively until the 11 - 9 positions representing R. Now we have 9 more of the 26 alphabets to take care of. 0 - 2 represents S and we move down to 0 - 9 which represents Z.

The positions 0 - 1 is not used because the holes would be too close together and errors may occur as the holes could easily become one.

Symbols

Symbols may be represented by one, two or even 3 holes in a column. (See fig.

Summarizing, each IBM card has 80 columns and twelve rows. There are 12 punching positions. Punching position 1 - 9 are called digit or numerical punch position and the 0, 11, 12th positions are called zone punch positions.

The 11th and 12th positions are not labelled on the card. The other positions are numbered 0 to 9.

All digits are represented by holes in the corresponding positions 0 - 9. All alphabetic letters are represented by 2 holes in each column, one in the zone area and the other in the digital area.

Punch cards are "read" by the physical movement of a card part, a sensing mechanism, which may be electrical contacts or a photoelectric device. The deck of cards are placed on the feed hopper of a punched card reader. (See fig.). The cards are taken from the feed hopper one by one past the reading station.

The reading devices is a row of thin flexible springs or brushes, one for each column, which can pass through the holes in the card to form an electrical circuit for each hole.

The card may be taken for a second reading for checking purposes. After this, the card is taken to the stacker, where all the read cards are stacked.

In the output equipment, there is a card punch which punches cards. It operates like the "reader", but is reverse. Unpunched cards are taken from the bottom of a feed hopper one at a time and pass in front of the dics (the punching mechanism). Sharp rods, one for each column, are thrust through the cards one or more times in the required pattern. The pattern of holes of course depends on what data is punched.

The output mechanism in the IBM 1130 is "the line pointer". The pointer points an entire line at a time. It is a large device, about a yard high and has provision for keeping a large amount of paper inside to be printed. The paper is in long strips with spooked holes along the edge to assist feeding the paper though the printer.

<u>The storage Unit</u>

The storage unit is like a precisely ordered warehouse where information can be stored until it is needed either by the arithmetic and logic unit or by the output unit. The storage unit is frequently called the memory unit because it works somewhat like the human memory system. It is used during the processing of a problem and to act upon the data presented, just as the human memory is used to "think" out a logical solution to a given problem.

The information in the memory unit is obtained from the input unit, except for some information which is "built in" in the computer. The information in the memory unit is not permanent. That is to say, as each new problem is fed into the computer, it will replace the previous data which is stored in the computer, unless the information stored in a special address which is not used for pulling in the new data.

Information in the memory unit can be numeric, alphabetic or alpha-numeric. A single piece of information in a digital, alphabetic or symbolic form is called a character. A "word" in computer language, is group of machine language symbols (or a group of characters) which is treated by the computer as a unit of invariant length. The "address" is the location of a given character, word or field in the storage unit. A computer having say, 5000 character positions as primary storage, is capable of storing 500 "bits" of information. The address of the first character is 0000 and the address of the last character is 5000. Hence each character or word or field can be identified by its address.

Information can be "read in" into the storage unit, or read out from the storage unit. When a word is read out from storage (or written out), the work remains unchanged in the address it was read from; the information is merely read out. However, when information is read in, the ingoing information obliterates the previous contents of the address written into. Erasure occurs automatically at read in, and does not occur at all when read out.

Characteristics of Storage Devices

For the computer to work rapidly, any information it needs for its operation should be immediately available for use - from the storage unit. Hence to be effective, there should be high speed and high capacity storage i.e. a large number of words should be speedily available whenever they are required. This brings in the concept of access time.

Access Time

The access time of a storage unti is the lapse of time between the instant that data are required by the computer by the arithmetic and logic unit from the storage unit and the instant they are available. Or access time is the time required for data to pass from arithmetic and logic unit to storage.

"Random access time" is the time taken for the computer to read out any two randomly chosen words from the storage unit.

Capacity

Capacity of a storage unit is the number of words it can store. The IBM 1130 has a capacity of 512,000 words.

Mode

The mode may either be parallel, serial or serial-parallel or parallel-serial. If there is a difference between the mode of the Arithmetic Unit and the storage unit, some kind of buffer is needed between the two units.

Permanence

If the memory in a computer fades off when the electricity is switched off, then it is termed volatile. If it is still available after switching it off and then on again, it is termed permanent.

Erasability

In erasable storage devices, words in an address can be erased when new words are put in, in some this is not possible.

Cyclic availability

In some storage devices, a given word is available repeatedly at minimum access time at regular intervals, whereas in others it is not. If this availability is repeated, the computer has a cyclic memory, if not it is non-cyclic.

Reliability

The storage unit should not miss out any words in the storage. In volatile memories, errors may occur, but this should not be the case.

Compactness is also another favourable character in a storage unit. Compactness is the density of information in the storage device, i.e. capacity in words per unit area or volume.

A combination of these characteristics on the different types of storage mediums are shown in

There are a number of storage mediums viz magnetic tape, magnetic drum, magnetic cones, magnetic disc, magnetic films, punch cards, paper tape accountic delay etc. The medium used in IBM 1130 is a magnetic disc. Magnetic discs can store large quantities of information (512,000 words) at random access time of less than a second. It is a metal disc that rotates about its centre. The flat surface on both sides is coated with megnetic material of much the same type as that used on magnetic tape. The arrangement of data on a magnetic disc is usually in concentric circles.

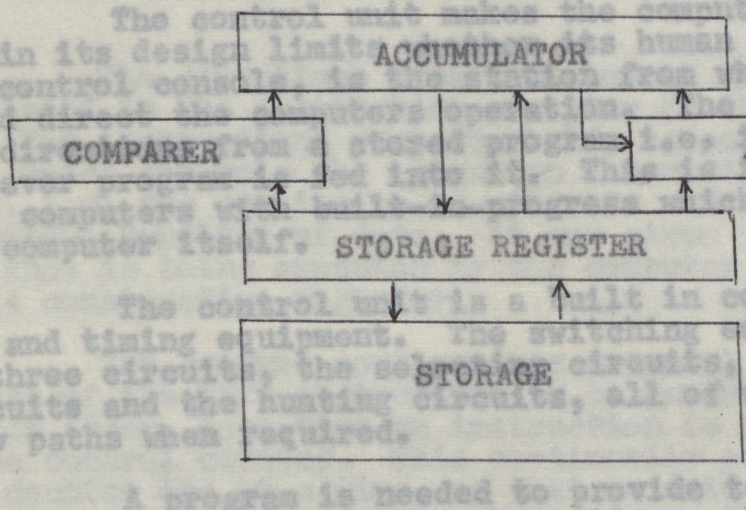
When the disc is rotating at constant speed, a combination reading and writing head can be positioned, or floated on a stream of air, above the magnetic coatings on either side of the disc at any distance from its centre. An address of data recorded on the disc is expressed in terms of the distance of the recording head from the centre of the disc, and the number of disc in question.

The magnetic discs provide permanent but erasable storage, just as does magnetic tape. These vast quantity of data can be stored by replacing discs.

The Arithmetic and Logic Unit

The arithmetic and logic unit is the actual "computing" part of the computer. However, this unit is not visible outside and is not so glamorous as the other components of the computer.

The flow of data in the arithmetic-logic unit of all computers is basically the same. Basically, there is a flow of information from storage to a distributor or storage register. For arithmetic operations, data from this register, together with data from the accumulator, go to the adder and results from the adder are returned to the accumulator. For logic operation, data may go to the comparer for an indication of comparison.



2.8 Basic data flow for Arithmetic Operation

The storage register temporarily store data from the storage. It has fast access-time and is limited in capacity to only one or two words.

The accumulator holds the results of arithmetic operations. It can hold either one word, one word plus one character, or twice either of the two lengths, or the sum of their lengths.

The adder, which is the main arithmetic element, obtains its data from either or both the storage register and accumulator. It performs addition, subtraction, multiplication, division, using number obtained from the other two components. The results of the airthmetic operation are returned in the accumulator for temporary storage.

The comparer is the major logic element in a computer. The compares produces and indication of whether the contents of the accumulator and the storage register, when compared character by character bit by bit as the case may require, are equal or unequal in some way.

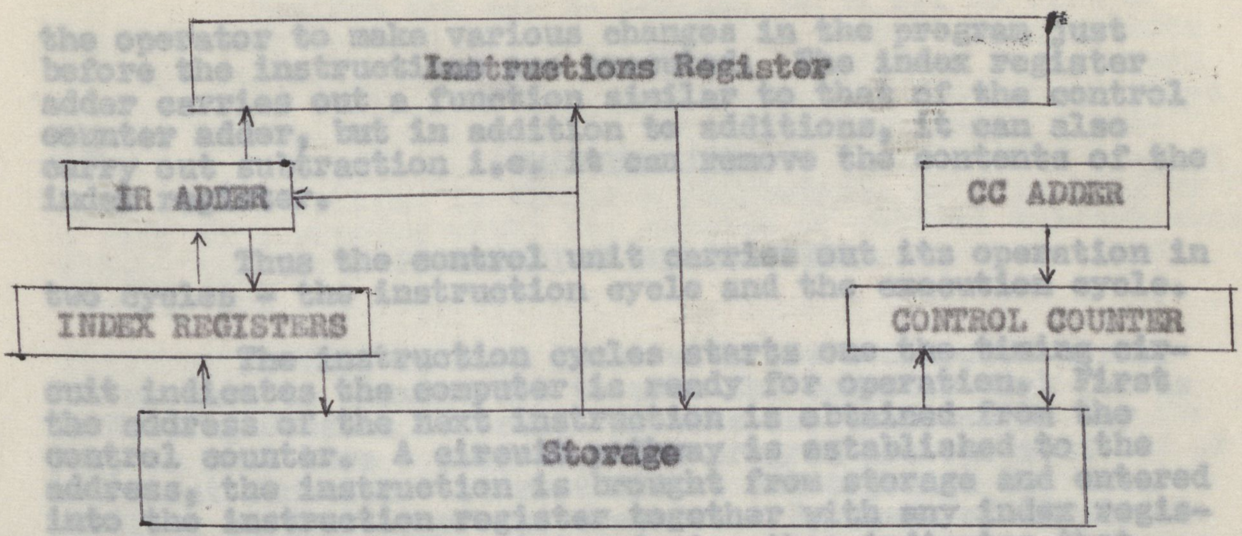
The presence or absence of plus or minus signs, or of a zero contents, may also usually determined by the comparer.

<u>The Control Unit</u>

The control unit makes the computer accomplish within its design limits whether its human operators direct. The control console, is the station from which an operator could direct the computers operation. The IBM 1130 obtains its directions from a stored program i.e. it will follow whatever program is fed into it. This is in contrast to some computers with built-in-progress which are fixed into the computer itself.

The control unit is a built in complex of switching and timing equipment. The switching equipment consists of three circuits, the selecting circuits, the connecting circuits and the hunting circuits, all of which form data flow paths when required.

A program is needed to provide temporary control over the general sequence of operations. Each programming instruction consists of two parts - a command of some type, and one or more operators which are usually represented by addresses. For example, a computer instruction may consists of a command "add" and the address of the word in storage which is to be added.



2.9. Basic data flow for control operations

The directions from the storage unit one at a time and placed in the instruction register or the control register. The instructions register hold, temporarily one instruction, which controls the operation automatically while it is there, i.e. the setting up of circuits and pathways which do the actual handling of data. At any given time, the instruction that is being executed by the computer would be the one in the construction register.

The control counter (see diagram) holds the address of the next instruction to be executed. The control counter adder, each time an instruction is executed, adds +1 to the control counter. This continually advances the control counter one at a time, so that as said earlier, the control counter always shows the address of the forthcoming instruction.

The index register carries out the function of adding some number to part of an instruction after it leaves storage to go to the control unit, and before it is entered into the control register. Thus the index register enables the operator to make various changes in the program just before the instructions are executed. The index register adder carries out a function similar to that of the control counter adder, but in addition to additions, it can also carry out subtraction i.e. it can remove the contents of the index register.

Thus the control unit carries out its operation in two cycles - the instruction cycle and the execution cycle.

The instruction cycles starts one the timing circuit indicates the computer is ready for operation. First the address of the next instruction is obtained from the control counter. A circuit pathway is established to the address, the instruction is brought from storage and entered into the instruction register together with any index register modification. The timing device then indicates that enough time has been allowed for the instruction cycle to be complete and the execution cycle begins.

The execution cycle includes both transport of a word (or field) and the making of a change in a word. If a word is to be brought to moved to some part of a computer, a pathway is established for the data flow. If addition or subtraction is to be done, these operations are performed. Whenever the execution is complete, the timing device indicates the completion of execution. During the execution-cycle, the control counter adder increases the contents of the control counter by 1 instruction length.

In the definition of data processing, it was pointed out it involved a number of steps of converting input data into output data. The computer carries on such a function and the operation of various components of the computer have been described. Input equipment transform data into pulses readable by the computer storage devices provide the means to hold reference data and all instruction, self-contained and fed, hence once instructions are fed into the machine, the computer is self-controlled.

The control component directs all other components to function. The arithmetic operations and logic operations and, finally, the output component translates the results into a form which could be read by humans. This then is the function of the electronic computer. We shall now go on to investigate how we could program a computer to carry out these functions.

CHAPTER III

PROGRAMMING THE COMPUTER

So far, we have dealt with how a computer works. Now we have to examine how to make the computer work for us in Keeping Student Records. To make the computer do any job, we must have a program. The program has the most important function - it must direct and organise the data handling capability of the computer. By means of different program, a general purpose computer can be made to perform a variety of tasks.

The IBM 1130, like other general purpose computers, has certain capabilities built into it by the designers and are called by various names - functions, codes, orders or commands. These capabilities are addition, multiplication, comparision, moving of data from one place in storage to another, movement of data from the arithmetic and logic units to storage and a variety of other tasks.

The commands specify the basic data maniputation operations which the computer can perform, but they do not specify the sequence in which they are used, nor do they identify the data to be manipulated. The command themselves are only latent capabilities until brought into play by some specifications of the sequence of operations and of the items of data.

The sequence in which the instructions are held in storage need not be the same as the sequence of operation. The instructions will be executed in an ascending order until or when the instructions specify a special order of execution.

The basic unit of a program is known as the instruction, consisting of one command, one or more operands, each associated with control character. An instruction thus specifies the data to be operated on, and the operation the computer is to perform. The assortment of commands are built in the computer, the command portion of the instruction simply calls the calls the capabilities into action (example of FOTRAN COMMAND statement is DO, READ or WRITE). The operand is the item of data on which the computer is to act on. It may be a parameter, an argument, a result, or an indication of the location of the next instruction.

Program can now be defined as "a set or list of instructions in such an order that the computer can execute the instructions automatically"[1]. A programmer is a person who prepares problem-solving procedures and flow charts (see below) and who may also write and debug routines. A subroutine is usually part of the program, or in itself a short program. A routine is either a subroutine, or a program or a part of a program which may consist of several subroutines. Often, the terms routine, subroutine and program are used loosely and interchangeably.

PROGRAMMING LANGUAGES

There are three major types of languages used for programming. The oldest languages were known as absolute or machine languages. Next came the symbolic languages, which were adapted to automatic coding and programming languages. ("problem orientated languages"). The machine language was made up of various combination of symbols, numbers and letters which specified various commands or operands. This was found to be too time consuming to construct programs with, and so symbolic language was developed. These languages used simple English, in short forms, to represent commands, operands. In symbolic language, a command ADD may be represented by the word ADD instead of some specified number as in the machine language. This was developed into automatic coding languages which are more easily understood and thus more easy to deal with.

The most popular automatic coding languages are FORTRAN AND ALGOL. In data processing, COBOL (Common Business Orientated Language) is most common. Our IBM 1130 programming language is FORTRAN and we shall look at FORTRAN LANGUAGE in more detail.

---

[1] Ned Chappin: INTRODUCTION to AUTOMATIC Computers. D. Van Nostrad Co. Inc., Princeton, New Jersey.

The Fortran Language - a brief introduction.

The Fortran language was developed to enable the programmer to specify a numerical procedure using a concise language like that of mathematics and obtain from this specification an efficient program to carry out the procedure. It was originally planned for the IBM 704 system but has since been adapted to most IBM systems.

The Fortran language can conveniently described by reviewing some examples:-(1)

Arithmetic Statements

Example 1 Compute: $\text{root} = -\dfrac{(B/2)+(B/2)^2 - AC}{A}$

Fortran Statement:

```
Root = (- (B/2.0) + SQRTF ((B/2.0)**2-A*C))/A
```

The desired program is a single FORTRAN statement, denoting the arithmetic formula. Its meaning is "Evaluate the expression on the right of the = sign and make this the value of the variable on the left". The symbol * denotes A^B). The program which is generated from this statement effects the computation in floating point arithmetic, avoids computing (B/2.0) twice and computes (B/2.0) **2 by a multiplication rather than by an exponential routine. [Had (B/20) ** 2.01 appeared instead, an exponential routine would necessarily be used requiring more time than multiplication.

Quantities both in floating points forms and interger forms can be used by the programmer. To indicate interger forms, (i.e. numbers without a decimal point), each interger variable has to be preceded by any of the alphabets, I, J, K, L, M or N. Any meaningful arithmetic expression may appear on the right hand side of an arithmetic statement, provided the following restriction is observed: an interger quantity can appear in a floating point expression only as a subscript or an exponent or as an argument of certain functions. SQRTF, denoting square-root, is computed automatically since it is built-in in the computer memory. Other functions if the programmer wants to refer to them at a subsequent time, have to be defined first.

---

(1) SAOL ROSSEN, Programming System and languages. McGrow Hill Book Co. N.Y.

Function Statements

Example 2:

Define a function of three variables to be used throughout a given problem, as follows:

```
ROOT F (A,B,C,) = (B/2.0) + SQRTF ((B/2.0)**2-A*C))/A
```

Function statements must come before the rest of the program. They are composed of the desired function name (ending in F) followed by any desired arguments which appear in the arithmetic expression on the right hand side of the = sign. The deprition of a function may employ any previously defined functions. Having defined ROOT F as above, the programmer may apply it to any set of arguments in any subsequent arithmetic statement. For example, a late arithmetic statement may be:-

```
THE TA = 1.0 + GAMMA * ROOT F (P1, 3.2*7+14.0, 7.63).
```

DO Statements, Dimension statements, and subscripted Variables

Example 3: Set Q max. equal to the largest quantity $P(ai + bi)/P(aj - bj)$ for some i between 1 and 1000 when $P(x) = Co + C_1x + C_2x^2 + C_3x^3$

```
1. POLYF(X) = Co + X * (C1 + X * (C2 + x * C3))
2. DIMENSION A(1000), B(1000)
3. QMAX = - 1.0E20
4. DO 5 I = 1, 1000
5. QMAX = MAXF(QMAX1 POLYF(ACI) + B(I)/POLYF(A(I)-B(I)))
6. STOP.
```

The program above is complete, except for the usual input and output statements. The first statement defines the polynominal (in factored form for efficient output program). The second statement informs the computer that there are 1000 elements in each A and B. Statement 3 assigns a large negative initial value for QMAX, $- 1.0 \times 10^{20}$ using special concise form for writing floating point constants. Statement 4 says, "Do the following sequence of set of statements down to and including the statement numbered 3 for successive values of I from I to 1000. In this case, the statement 5 is repeated 1000 times; the first time reference is made to A(1) and B(1), the second time to A(2) and B(2), etc. After the

1000th execution of statement 5, the function MAXF appears. MAXF may have two or more arguments and its value, by definition, is the value of its largest argument. Thus on each repetition of statement 5, the old value of QMAX is replaced by itself or by the value of POLYF (A(I)+B(I))/POLYF (A(I)-B(I)) whichever is the larger. The value of QMAX after the 100th repetition is therefore the desired maximum.

Example 4:

Multiply nxn matrix a(ij)(n 20) by its transpose, obtaining the product elements on or below the main diagonal by the relation

C(ij) = sum of a(ij) a(jk) from k = 1 to n
(i j and k are subscripts)

and the remaining elements by the relation

C(ij) = C(ij)

FORTRAN PROGRAM

```
      DIMENSION A(20, 20)  C(20, 20)
(a)   DO 2 I = 1 N
(b)   DO 2 J = 1 I
(c)   C (I,J) = 0.0
(d)   DO 1 K = 1 N
 1    C(I, J) = C(I,J)+A(I,K)*A(J,K)
 2    C(J,I) = C(I,JO
        STOP
```

As in the preceeding examples, the DIMENSION statement says that there are two matrices of maximum size 20 x 20 named A and C. The first do statement (a) says that the following statements down to statement 2 is to be carried out for I = 1, then I = 2 and so up to I = N. The second DO statement (b), directs the statements from (c) to 2 be carried out for I = N to J = I. Each execution of the second DO statement would involve N executions of the 3rd DO statement i.e. statement (d). (Note, (a) (b) (c) (d) are introduced for explanatory purposes and are not part of the program.)

This is an example of a "nest" of DO statements, means that the sequence of statements of one do statements contains other DO statements. Another example of such a next is by the following diagram.

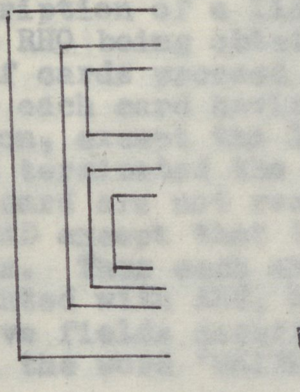


Fig. 3.1

READ, PRINT, FORMAT, IF and GO TO Statements

Example: For each case, read from cards two vectors, ALPHA and RHO, and the number ARG. ALPHA and RHO each have 25 elements and ALPHA(I) is less than or equal to ALPHA(I+1) I = 1 to 24. Find the SUM of all the elements ALPHA from the beginning to the last one which is less than or equal to ARG [assume ALPHA(I) is less than or equal to ARG which is less than ALPHA(25)]. If this last statement is the Nth set VALUE = 3.14159*RHO(N). Print a line for each case with ARG, SUM and VALUE.

FORTRAN Program:-

```
      DIMENSION  ALPHA(25), RHO(25)

1.    FORMAT (5F 12.4)

2.    READ 1, ALPHA, RHO, ARG.
      SUM = 0.0
      DO 3 I = 1,25
      IF(ARG-ALPHA(I)) 4, 3, 3.

3.    SUM = SUM + ALPHA(I)
4.    VALUE = 3.14159 RHO(I-1)
      PRINT 1, ARG, SUM, VALUE
      GO TO 2.
```

The FORMAT statement says that numbers are to be found (or printed) 5 per card, that each number is in fixed point form, that each number occupies a field of 12 columns wide and that the decimal point is located 4 digits from the right. The FORMAT statement is referred to by the READ and PRINT statements to describe the desired arrangement of data in the external medium.

The READ statement says "READ cards in the card reader which are arranged according to FORMAT statement 1 and assign the successive numbers obtained as values of ALPHA(I) I=1, 25 and RHO(I) I = 1, 25 and ARG". Thus "ALPHA, RHO, ARG" is a description of a list of 51 quantities (the size of ALPHA and RHO being obtained from the DIMENSION statement). Reading of cards proceed until these 51 quantities have been obtained, each card having 5 numbers, as per the FORMAT description, except the last which has the value ARG only. Since ARG terminated the list, the remaining four fields on the last card are not read. The PRINT statement is similar to the READ except that it specifies a list of only three quantities. Thus each execution of PRINT causes a single line to be printed with ARG, SUM, VALUE printed in the first three of the five fields described by FORMAT statement 1. For the IBM 1130, the word 'WRITE' is used instead of "PRINT".

The IF statement says "IF ARG-ALPHA(I) is negative go to statement 4, if it is zero go to statement 3, and if it is positive go to 3". Thus the repetition of the two statements controlled by DO consists normally of computing ARG-ALPHA(I), finding it zero or positive, and going to statement 3 followed by the next repetition.

However, when I has been increased to the extent that the first ALPHA exceeding ARG is encountered, control will pass to statement 4. Note that this statement does not belong to the sequence controlled by the DO. In such cases the repetition specified by the DO is terminated and the value of the index (in this case 1) is preserved. Thus if the first ALPHA exceeding ARG were ALPHA(20) then RHO(1a) would be obtained in statement 4.

The GO TO statement passes control to statement 2 which initiates reading the 11 cards for the next case. The process will continue until there are no more cards in the reader. The above program is entirely complete. When punched in the cards as shown, with of course the preliminary statements, the computer will execute the program. The Preliminary statements for a FOTRAN Program are the following.

```
    JOB
    FOR
*   LIST SOURCE PROGRAM
*   IOCS (CARD, DISK, 1132 PRINTER, TYPEWRITER )

    (Program Cards)

   CALL  EXIT
   END
    XEQ
```

These FOTRAN STATEMENTS are put in along with each program fed into the computer.

Other than these statements, there are about twenty three other types of statements in the language, many of them completely analogous to the above. They provide facilities for referring to other input output auxilliary storage devices (tapes, drums and card punch) for specifying present and computed branching of control. A complete description of the FOTRAN Language can be found in the IBM manuel on FOTRAN LANGUAGE.

Objectives of Programming

Programming is the preparation of the sequence of instructions for the computer. Each of the instructions should be in the programming language i.e. FOTRAN in our case. There are four major objectives which should be borne in mind when preparing a program. These objectives partially conflict with each other, hence programming difficulty is increased.

Firstly, the most important objective of program is accuracy. The program must be accurate so that the results obtained would be accurate. There can be no compromise in accuracy. Its importance could be realised, especially in the keeping of student records. If a student is graded as failed when in fact he has passed, it is possible to visualize how important accuracy is. Without accuracy, using a computer system is of no use. A program can be checked by partial printing out of output at different stages, and comparing them with results obtained manually.

A second objective is speed. Running a computer is terribly expensive. Hence any usage of the computer should take up the least time possible. Also, more work can be obtained from the computer if each job takes less time. This objective conflicts with that of accuracy, for to get accurate results with partial checks, would take much time. Hence this would be a problem for the programmer.

A third objective is economy of storage. In practice this is the most bothersome. The internal storage of the computer is limited (the IBM 1130 has a capacity of 512,000 words). Only part of the storage capacity can be used by the programmer because much of the storage is also used by other programs or data.

The fourth objective of programming is simplicity. A simple program is easier to write, easier to understand, easier to use and easier to revise when necessary. But simplicity in programming conflicts with the objectives of

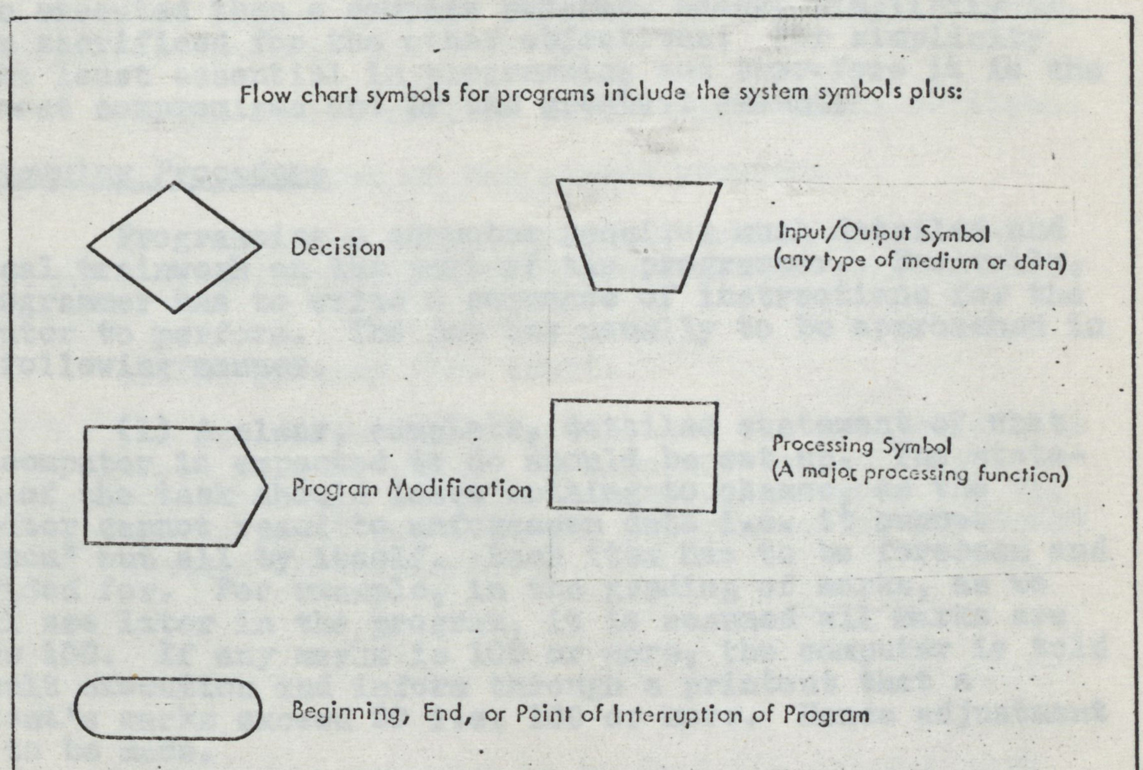


FIGURE 31 PROGRAM FLOW CHART SYMBOLS.

economy of storage, for a straightforward program usually makes use of more storage than an elaborate program. And surprising enough, a simple program may actually be slower to be executed than a complex program. Hence, simplicity is often sacrificed for the other objectives; for simplicity is the least essential in programming and therefore it is the one most compromised and by the greatest amount.

Programming Procedure

Programming a computer requires much detailed and logical brainwork on the part of the programmer. Basically, a programmer has to write a sequence of instructions for the computer to perform. The job has usually to be approached in the following manner.

(1) A clear, complete, detailed statement of what the computer is expected to do should be set up. The statement of the task should leave nothing to chance, as the computer cannot react to unforeseen data i.e. it cannot "reason" out all by itself. Each item has to be foreseen and provided for. For example, in the grading of marks, as we shall see later in the program, it is assumed all marks are below 100. If any marks is 100 or more, the computer is told to halt execution and inform through a printout that a student's marks exceed 99 i.e. 100 or more. Hence adjustment has to be made.

(2) A detailed flow diagram must be made for each program. All the alternatives must be specified completely and what the computer is to do in each alternative should be set down fully. A flow diagram is thus a graphic device listing the sequence in which, and the conditions under which, various operations are to be performed by the computer to transform the available input data into desired output data. The symbols used in flow diagram is illustrated in the opposite page.

(3) The instruction should now be written in FOTRAN Language, in our case. To simplify the translating process, into "set-up" instructions, "DO" instructions and "clean-up" instructions.

(4) The program is then run with a test-data. Where errors occur, the program has to be "debugged" i.e. corrected. This process is called "testing and debugging" a program.

The program is now ready for use and can now be applied to the regular data.

Finally, the program is documented i.e. the programmer writes a brief description of the program: What the output should be and what the input is; what are the limitations of output and the possible variations in output. The documented package consists of:-

1) The source program punched into cards or tapes.

2) A print-out of the source program.

3) A write up of the program, including detailed operating instructions, plus what the computer does and how he does it.

4) The program flow chart.

The reason for this is that now the program can be used by anybody else as well. We have now completed a introduction of how the computer can be programmed. In the next chapter, we shall see a sample program on the treatment of marks obtained in examinations.

CHAPTER IV

A SAMPLE PROGRAM(1)

Students' records are usually of their marks or grades they obtained in their examinations. This sample program is concerned with grading of marks.

<u>What the Computer is expected to do</u>

The computer is fed with the data, which is the students' names, followed by the marks he/she obtained in the subjects economics, statistics and mathematics. In the punched card, the name is punched between columns 1 and 20, the economics marks on the 36th and 37th columns, the statistics marks on the 40th and 41st columns and the mathematics marks on the 44th and 45th columns. (See figure 4.1).

Firstly the computer is to find the average mark for each student. Then the students' names are to be arranged in the order of their average marks; i.e. the student with the highest average first, followed by the one with the next highest etc. The marks the students obtained in the three subjects are also given.

Secondly, another three tables are drawn up, one in order of economics marks, the next in order of statistics marks and the final one in order of mathematics marks.

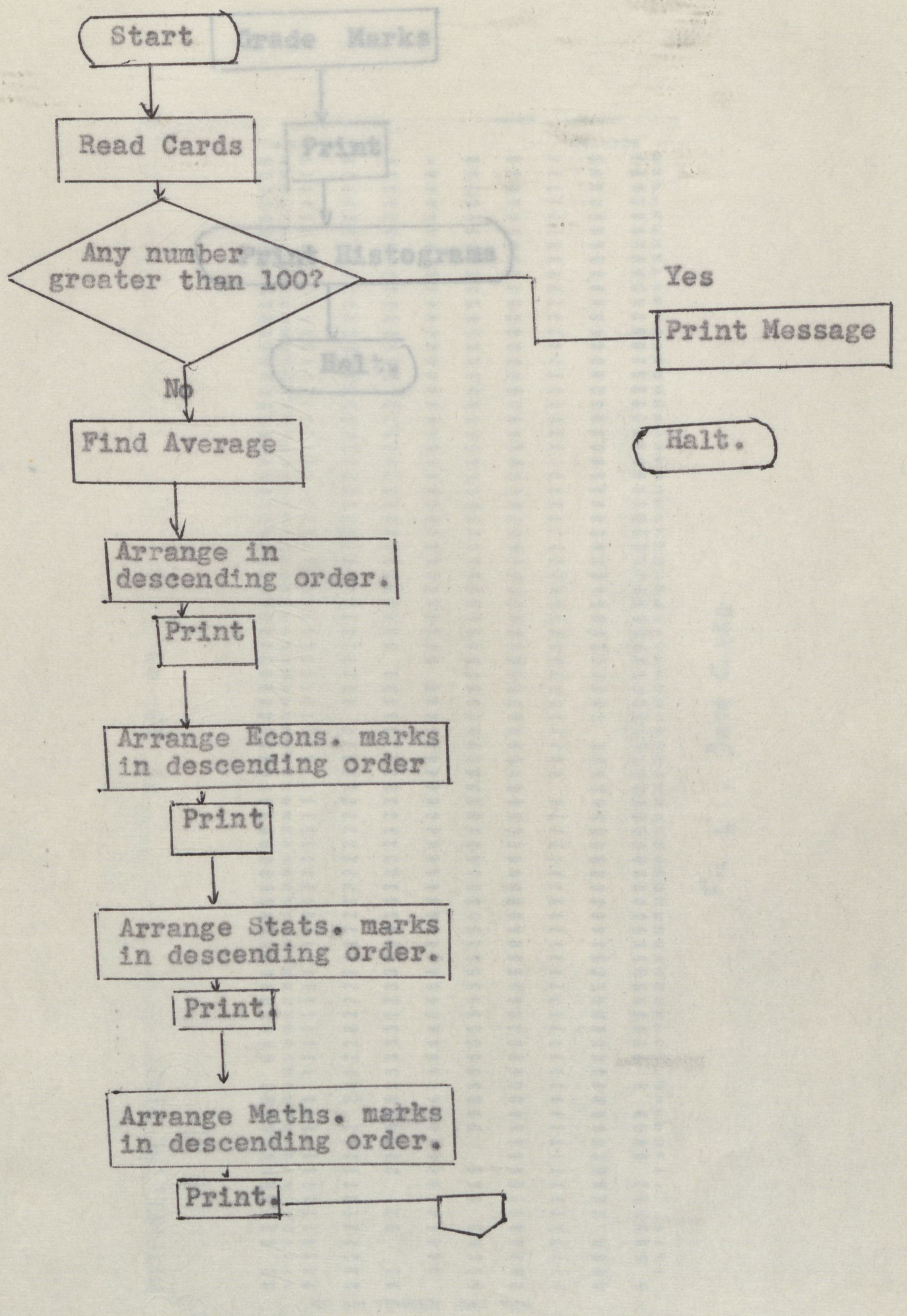
Next, the computer grades the marks and prints out how many students obtained distinction, credit, pass or fail in each subject.

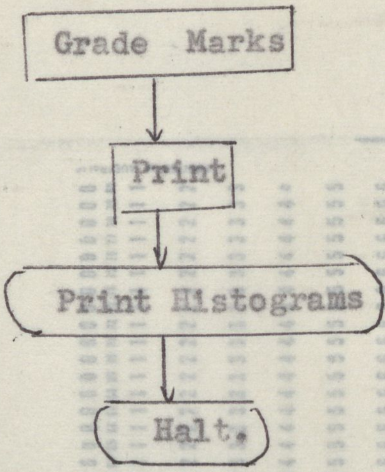
Finally, a histogram is drawn for the marks obtained in various subjects, showing how the marks are distributed.

---

(1) This program was written by Mr. John Slaugh of the Computer Centre. I have made some minor adjustments to his original program.

The Flow Chart




Grade Marks
Print
Print Histograms
Halt.

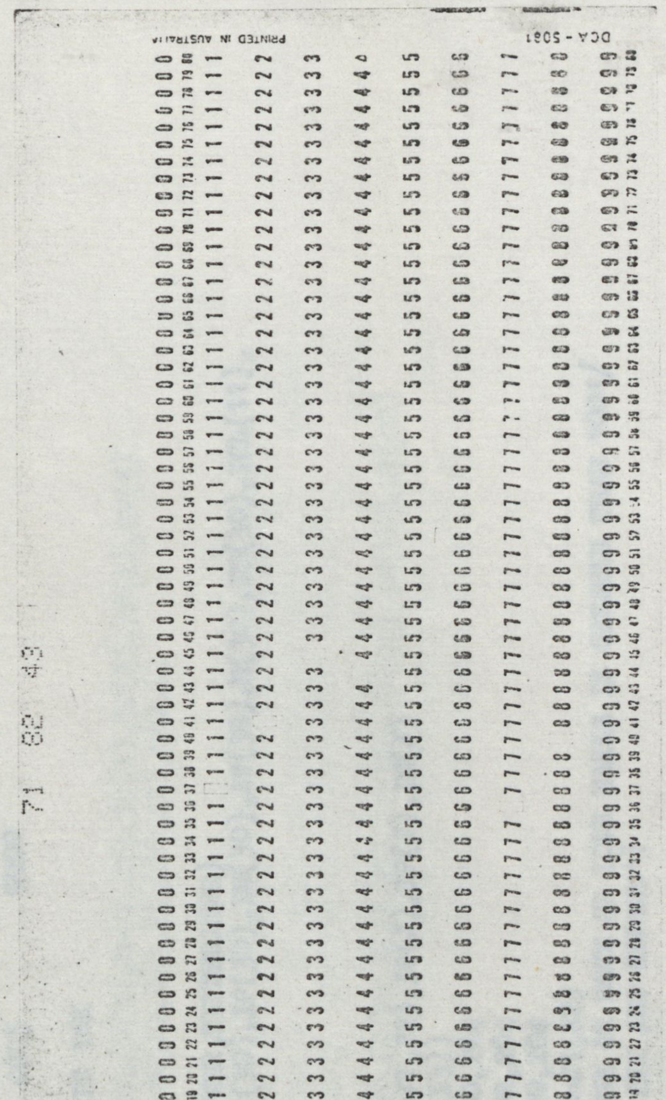


Fig. 4.1. DATA CARD.

```
// JOB T

LOG DRIVE   CART SPEC   CART AVAIL  PHY DRIVE
  0000        0C05        0C05        0000

V2 M06   ACTUAL 16K  CONFIG 16K

// FOR
*LIST SOURCE PROGRAM
*IOCS(CARD,DISK,1132 PRINTER,TYPEWRITER)
      DIMENSION X(240),IC(30),IP(30),IM(30),AV(30),M(30),IT(30),ICL(11),
     1MC(7),MP(7),MM(7)
      DO 10 J=1,30
      J1=8*J-7
      J2=J1+7
  250 READ(2,100)(X(I),I=J1,J2),IC(J),IP(J),IM(J)
  100 FORMAT(8A4,1X,3(1X,I3))
      IF(IC(J)-100)200,200,204
  200 IF(IP(J)-100)202,202,204
  202 IF(IM(J)-100)210,210,204
  204 WRITE(1,206)(X(I),I=J1,J2)
  206 FORMAT(1X,8A4,' , ONE OR MORE OR YOUR MARKS IS GREATER THAN 100'/
     1 ' PLEASE FIX AND REFEED')
      PAUSE
      GO TO 250
  210 AV(J)=(IC(J)+IP(J)+IM(J))/3.0
   10 CONTINUE
      WRITE (3,102)
      J=1
   25 IEQ=16448
      L=1
      I=1
      DO 12 K=2,30
```

```
      IF (AV(I)-AV(K)) 21,22,20
   20 GO TO 12
   21 I=K
      IEQ=16448
      L=1
   12 CONTINUE
      IF (L-1) 23,23,24
   23 J1=8*I-7
      J2=J1+7
      WRITE (3,103) J,IEQ,(X(N),N=J1,J2),IC(I),IP(I),IM(I),AV(I)
  103 FORMAT ('    ',I2,A2,2X,8A4,I8,I10,I11,F12.1)
      AV(I)=-1.0
      GO TO 11
   24 DO 13 N1=1,L
      N2=M(N1)
      J1=8*N2-7
      J2=J1+7
      WRITE (3,103) J,IEQ,(X(N),N=J1,J2),IC(N2),IP(N2),IM(N2),AV(N2)
      AV(N2)=-3.0
   13 CONTINUE
      J=J+L-1
   11 J=J+1
      IF (J-30) 25,25,27
   22 IEQ=32320
      L=L+1
      M(1)=I
      M(L)=K
      GO TO 12
   27 DO 30 J=1,30
      AV(J)=(IC(J)+IP(J)+IM(J))/3.0
   30 CONTINUE
      LOOP=0
   40 LOOP=LOOP+1
      GO TO (46,47,48,53),LOOP
```

```
46  WRITE (3,104)
    DO 50 J=1,30
50  IT(J)=IC(J)
    GO TO 49
47  WRITE (3,105)
    DO 51 J=1,30
51  IT(J)=IP(J)
    GO TO 49
48  WRITE (3,106)
    DO 52 J=1,30
52  IT(J)=IM(J)
    GO TO 49
102 FORMAT  ('1  TABLE 1   IN ORDER OF AVERAGE MARK  ',//'   PLACE
   1   NAME',30X,'ECONOMICS  STATISTICS  MATHEMATICS AVERAGE'/)
104  FORMAT (///'   TABLE 2   IN ORDER OR ECONOMICS MARK',//'   PLACE
   1   NAME'30X,'ECONOMICS  STATISTICS MATHEMATICS AVERAGE'/)
105 FORMAT  ('1   TABLE 3    IN ORDER OR STASTICS MARK  ',//'   PLACE
   1    NAME',30X,'ECONOMICS  STATISTICS  MATHEMATICS AVERAGE'/)
106 FORMAT(///'   TABLE 4    IN ORDER OF MATHEMATICS MARK',//'     PLA
   ICE   NAME',30X,'ECONOMICS  STASTICS  MATHEMATICS AVERAGE'/)
49  J=1
39  IEQ=16448
    L=1
    I=1
    DO 31 K=2,30
    IF(IT(I)-IT(K)) 32,33,34
34  GO TO 31
32  I=K
    IEQ=16448
    L=1
31  CONTINUE
    IF (L-1)  35,35,36
```

```
35  J1=8*I-7
    J2=J1+7
    WRITE (3,103) J,IEQ,(X(N),N=J1,J2),IC(I),IP(I),IM(I),AV(I)
    IT(I)=-1.0
    GO TO 37
36  DO 38 N1=1,L
    N2=M(N1)
    J1=8*N2-7
    J2=J1+7
    WRITE (3,103) J,IEQ,(X(N),N=J1,J2),IC(N2),IP(N2),IM(N2),AV(N2)
    IT(N2)=-3.0
38  CONTINUE
    J=J+L-1
37  J=J+1
    IF (J-30) 39,39,40
33  IEQ=32320
    L=L+1
    M(1)=I
    M(L)=K
    GO TO 31
53  DO 61 J=1,7
    MC(J)=0
    MP(J)=0
61  MM(J)=0
    DO 60 J=1,30
    M1=(IC(J)-25)/15+2
    MC(M1)=MC(M1)+1
    M1=(IP(J)-25)/15+2
    MP(M1)=MP(M1)+1
    M1=(IM(J)-25)/15+2
    MM(M1)=MM(M1)+1
60  CONTINUE
    MC(1)=MC(1)+MC(2)
    MC(5)=MC(5)+MC(6)+MC(7)
```

```
      MP(1)=MP(1)+MP(2)
      MP(5)=MP(5)+MP(6)+MP(7)
      MM(1)=MM(1)+MM(2)
      MM(5)=MM(5)+MM(6)+MM(7)
      WRITE (3,110)
110   FORMAT ('1   TAVLE 5   CLASSIFICATION BY GRADES'//25X,'ECONOMICS
     1 STATISTICS  MATHEMATICS'/)
      WRITE (3,111) MC(5),MP(5),MM(5)
111   FORMAT (3X,'DISTINCTION 70-100',I9,I10,I11)
      WRITE (3,112) MC(4),MP(4),MM(4)
112   FORMAT (3X,'CREDIT          55-69 ',I9,I10,I11)
      WRITE (3,113) MC(3),MP(3),MM(3)
113   FORMAT (3X,'PASS            40-54 ',I9,I10,I11)
      WRITE (3,114) MC(1),MP(1),MM(1)
114   FORMAT (3X,'FAIL             0-39 ',I9,I10,I11)
      LOOP=0
98    LOOP=LOOP+1
      GO TO (66,67,68,300),LOOP
66    WRITE (3,134)
      DO 70 J=1,30
70    IT(J)=IC(J)
      GO TO 99
67    WRITE (3,135)
      DO 71 J=1,30
71    IT(J)=IP(J)
      GO TO 99
68    WRITE (3,136)
      DO 72 J=1,30
72    IT(J)=IM(J)
      GO TO 99
300   CALL EXIT
99    DO 180 J=1,11
180   ICL(J)=0
```

```
      DO 90 J=1,30
      L1=IT(J)/10+1
   90 ICL(L1)=ICL(L1)+1
      ICL(10)=ICL(10)+ICL(11)
      MAX=0
      DO 93 J=1,10
      IF (MAX-ICL(J)) 92,93,93
   92 MAX=ICL(J)
   93 CONTINUE
      DO 94 J=1,MAX
      DO 95 K=1,10
      IF (MAX-J+1-ICL(K)) 96,96,95
   96 GO TO (81,82,83,84,85,86,87,88,89,91),K
   81 WRITE (3,141)
      GO TO 95
   82 WRITE (3,142)
      GO TO 95
   83 WRITE (3,143)
      GO TO 95
   84 WRITE (3,144)
      GO TO 95
   85 WRITE (3,145)
      GO TO 95
   86 WRITE (3,146)
      GO TO 95
   87 WRITE (3,147)
      GO TO 95
   88 WRITE (3,148)
      GO TO 95
   89 WRITE (3,149)
      GO TO 95
   91 WRITE (3,150)
   95 CONTINUE
   94 WRITE (3,140)
```

```
141   FORMAT (1H+,20X,'*****')
142   FORMAT (1H+,24X,'*****')
143   FORMAT (1H+,28X,'*****')
144   FORMAT (1H+,32X,'*****')
145   FORMAT (1H+,36X,'*****')
146   FORMAT (1H+,40X,'*****')
147   FORMAT (1H+,44X,'*****')
148   FORMAT (1H+,48X,'*****')
149   FORMAT (1H+,52X,'*****')
150   FORMAT (1H+,56X,'*****')
      WRITE (3,151)
151   FORMAT  (23X,'0  10  20  30  40  50  60  70  80  90'/23X,'9  19
     1 29  39  49  59  69  79  89  100'/)
140   FORMAT (' ')
      GO TO 98
134   FORMAT (//'  TABLE 6    HISTOGRAM FOR ECONOMICS'//)
135   FORMAT (//'  TABLE 7    HISTOGRAM OR STATISTICS'//)
136   FORMAT (//'  TABLE 8    HISTOGRAM FOR MATHMETICS'//)
      END
FEATURES SUPPORTED
 IOCS
CORE REQUIREMENTS FOR
 COMMON     0  VARIABLES    942  PROGRAM   2060
 END OF COMPILATION
// XEQ
```

The Output

TABLE 1 IN ORDER OF AVERAGE MARK

PLACE	NAME	ECONOMICS	STATISTICS	MATHEMATICS	AVERAGE
1	SIVANATHAN	95	85	75	85.0
2	IVANCVITCH ZAK	87	78	79	81.3
3	BOH OH DOH	55	88	99	80.6
4	ANGELA TAN	75	85	64	74.6
5	ELIZABETH WONG	52	72	82	68.6
6	MITCHELLINE BIAZ	71	82	43	65.3
7=	JUSTIN LUGG	64	38	92	64.6
7=	RAMANATHAN	68	71	55	64.6
9	SHERLOCK DUNCAN	82	43	68	64.3
10	ADRIAN SIM	52	79	61	64.0
11	ALUDDIN JAF	62	73	56	63.6
12	DAVID TONG	85	62	42	63.0
13	HOPPY ULIATZ	45	56	83	61.3
14	WONG SWEE LAN	58	49	76	61.0
15	AMY TEH	31	64	87	60.6
16	MAHENDRAN	44	71	61	58.6
17	AKBAR SAMSUDDIN	28	73	72	57.6
18	RAMAN JACOBS	98	25	47	56.6
19	INDRAN	46	64	56	55.3
20	TIMMY CHOO	24	75	62	53.6
21	FREDDIE LIM	63	52	41	52.0
22	GNORAMUS LIAN	73	28	48	49.6
23	JAMES LOW	23	45	78	48.6
24	ZUKEFFLI IBNI	53	42	48	47.6
25	WONG HO KAM	75	24	29	42.6
26	WHATSIT LING	31	25	69	41.6
27=	HERRINDA SARVAN	36	39	46	40.3
27=	LIM AH LEK	64	31	26	40.3
29=	REBECCA SIM	54	44	22	40.0
30	ALEXANDER PHILIPS	29	14	35	26.0

TABLE 2 IN ORDER OR ECONOMICS MARK

PLACE	NAME	ECONOMICS	STATISTICS	MATHEMATICS	AVERAGE
1	RAMAN JACOBS	98	25	47	56.6
2	SIVANATHAN	95	85	75	85.0
3	IVANOVITCH ZAK	87	78	79	81.3
4	DAVID TONG	85	62	42	63.0
5	SHERLOCK DUNCAN	82	43	68	64.3
6=	WONG HO KAM	75	24	29	42.6
6=	ANGELA TAN	75	85	64	74.6
8	GNORAMUS LIAN	73	28	48	49.6
9	MITCHELLINE BIAZ	71	82	43	65.3
10	RAMANATHAN	68	70	55	64.6
11=	JUSTIN LUGG	64	38	92	64.6
11=	LIM AH LEK	64	31	26	40.3
13	FREDDIE LIM	63	52	41	52.0
14	ALUDDIN JAF	62	73	56	63.6
15	WONG SWEE LAN	58	49	76	61.0
16=	BOH OH DOH	55	88	99	80.6
17=	REBECCA SIM	54	44	22	40.0
18	ZUKEFFLI IBNI	53	42	48	47.6
19=	ELIZABETH WONG	52	72	82	68.6
20	ADRIAN SIM	51	79	61	64.0
21	INDRAN	46	64	56	55.3
22	HOPPY ULIATZ	45	56	83	61.3
23	MAHENDRAN	44	71	61	58.6
24	HERRINDA SARVAN	36	39	46	40.3
25=	AMY TEH	31	63	87	60.6
25=	WHATSIT LING	31	25	69	41.6
27	ALEXANDER PHILIPS	29	14	35	26.0
28	AKBAR SAMSUDDIN	28	73	72	57.6
29	TIMMY CHOO	24	75	62	53.6
30	JAMES LOW	23	45	78	48.6

TABLE 3 IN ORDER OR STASTICS MARK

PLACE	NAME	ECONOMICS	STATISTICS	MATHEMATICS	AVERAGE
1	BOH OH DOH	55	88	99	80.6
2=	ANGELA TAN	75	85	64	74.6
2=	SIVANATHAN	95	85	75	85.0
4	MITCHELLINE BIAZ	71	82	43	65.3
5	ADRIAN SIM	52	79	61	64.0
6	IVANOVITCH ZAK	87	78	79	81.3
7	TIMMY CHOO	24	75	62	53.6
8=	AKBAR SAMSUDDIN	28	73	72	57.6
8=	ALUDDIN JAF	62	73	56	63.6
10	ELIZABETH WONG	52	72	82	68.6
11=	MAHENDRAN	44	71	61	58.6
12	RAMANATHAN	68	70	55	64.6
13=	AMY TEH	31	64	87	60.6
14	INDRAN	46	63	56	55.3
15	DAVID TONG	85	62	42	63.0
16	HOPPY ULIATZ	45	56	83	61.3
17	FREDDIE LIM	63	52	41	52.0
18	WONG SWEE LAN	58	49	76	61.0
19	JAMES LOW	23	45	78	48.6
20	REBECCA SIM	54	44	22	40.0
21	SHERLOCK DUNCAN	82	43	68	64.3
22	ZUKEFFLI IBNI	53	42	48	47.6
23	HERRINDA SARVAN	36	39	46	40.3
24	JUSTIN LUGG	64	38	92	64.6
25	LIM AH LEK	64	31	26	40.3
26	GNORAMUS LIAN	73	28	48	49.6
27=	WHATSIT LING	31	25	69	41.6
27=	RAMAN JACOBS	98	25	47	56.6
29	WONG HO KAM	75	24	29	42.6
30	ALEXANDER PHILIPS	29	14	35	26.0

TABLE 4 IN ORDER OF MATHEMATICS MARK

PLACE	NAME	ECONOMICS	STASTICS	MATHEMATICS	AVERAGE
1	BOH OH DOH	55	88	99	80.6
2	JUSTIN LUGG	64	38	92	64.6
3	AMY TEH	31	64	87	60.6
4	HOPPY ULIATZ	45	56	83	61.3
5	ELIZABETH WONG	52	72	82	68.6
6	IVANOVITCH ZAK	87	78	79	81.3
7	JAMES LOW	23	45	78	48.6
8	WONG SWEE LAN	58	49	76	61.0
9	SIVANATHAN	95	85	75	85.0
10	AKBAR SAMSUDDIN	28	73	72	57.6
11	WHATSIT LING	31	25	69	41.6
12	SHERLOCK DUNCAN	82	43	68	64.3
13	ANGELA TAN	75	85	64	74.6
14	TIMMY CHOO	24	75	62	53.6
15=	ADRIAN SIM	52	79	61	64.0
15=	MAHENDRAN	44	71	61	58.6
17=	ALUDDIN JAF	62	73	56	63.6
17=	INDRAN	46	64	56	55.3
19	RAMANATHAN	68	71	55	64.6
20=	GNORAMUS LIAN	73	28	48	49.6
20=	ZUKEFFLI IBNI	53	42	48	47.6
22	RAMAN JACOBS	98	25	47	56.6
23	HERRINDA SARVAN	36	39	46	40.3
24	MITCHELLINE BIAZ	71	82	43	65.3
25	DAVID TONG	85	62	42	63.0
26	FREDDIE LIM	63	52	41	52.0
27	ALEXANDER PHILIPS	29	14	35	26.0
28	WONG HO KAM	75	24	29	42.6
29	LIM AH LEK	64	31	26	40.3
30	REBECCA SIM	54	44	22	40.0

TABLE 5 CLASSIFICATION BY GRADES

		ECONOMICS	STATISTICS	MATHEMATICS
DISTINCTION	70-100	9	12	10
CREDIT	55-69	8	4	9
PASS	40-54	6	6	7
FAIL	0-39	7	8	4

TABLE 6 HISTOGRAM FOR ECONOMICS

```
                    ****
                    ********
        ****        ************
        ****************************
        ********************************
        ********************************

0   10  20  30  40  50  60  70  80  90
9   19  29  39  49  59  69  79  89  100
```

TABLE 7 HISTOGRAM OR STATISTICS

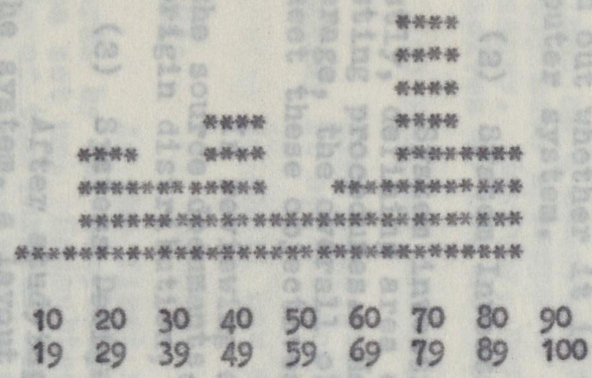
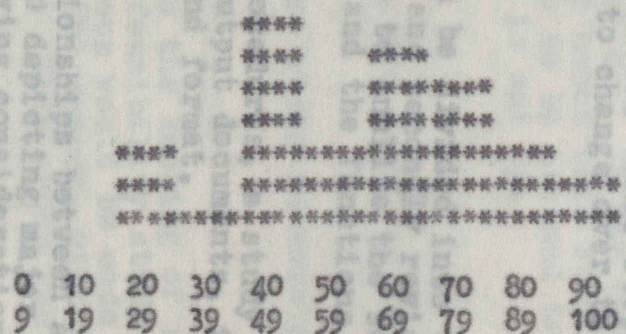


TABLE 8 HISTOGRAM FOR MATHMETICS



CHAPTER V

THE APPROACH TO SETTING UP A COMPUTER SYSTEM OF KEEPING RECORDS

In this chapter we shall attempt to describe the steps involved in preparing an application of the computer to keep records.

There are seven major steps to go through before we have an operative system. These are the following.

(1) Survey.

The survey is the determination of whether the proposed computer data processing system, i.e. in our case keeping of student records, is feasible. This step is to find out whether it is worthwhile to change over to the computer system.

(2) System Investigation.

System investigation can be divided into two parts: firstly, defining area objectives and secondly reviewing existing procedures. Here we have to analize the scope of coverage, the overall objectives, and the functions required to meet these objectives.

In reviewing existing procedures, a study is made of the source documents, and the output documents, in terms of origin distribution, material and format.

(3) Systems Design.

After studying all relationships between all parts of the system, a layout is prepared depicting major relationship in the proposed system and giving consideration to the reference record that must be created. Then an analysis of the reports to be made is made. The source and reference data required to produced reports are defined. After this process chart is constructed showing all processing needed. Finally, step one descriptions which describe the operation of the computer are prepared.

(4) Programming.

From the help of the step-run descriptions, the programmers will prepare flow charts picturing in detail how the operation is to be carried out. The next step is to code the flow chart into FOTRAN Language. The program is then debugged i.e. each of the individual runs are checked for accuracy. A run book and the computer centre procedures are prepared.

(5) File Making.

File making involves the construction of records used for reference to obtain information or to accumulate data. Since our student records would be kept in this manner, a special chapter on file organization techniques is found after this chapter. In file making, data gathered is converted to tape and an audit of the file is made. Prior to using the file in operation, it is brought up to date.

(6) Preparation of clerical procedures.

This step is the organization of the new clerical routines as a computer system is set up. In our case, we already have a computer centre, and hence this step is not necessary. However, personnel have to be trained to adapt themselves to the new system if it is set up.

(7) Program Testing.

This last step is testing the drawn up system prior to making it operational.

We shall now examine these steps in more detail.

SURVEY.

As the first step towards the keeping of student records by the computer system, a feasibility study group has to be set up. Undoubtedly this group would be made up of the heads of the various faculties in the campus, with the head of the computer centre as Chairman. This group can now study the idea in greater detail.

The needs of the project are defined i.e. what records are to be kept: the grades and marks of students together with their addresses? What has to be done? How far reaching would be the computer system be? Would the computer system centralization of some other organization? i.e. would the different faculty officers be dependent on the computer centre? What equipment would be necessary? Would the administration cough up the money required?

When the project needs are clear, then the potentiality of the system is to be investigated. Would it save costs, time? All results are published in the months of February-March-April. Would the computer centre be able to process out these results in the short time available? Would it be accurate - for the student's future may depend on it.

A block diagram may be set up showing the data processing system. The finer points in the relationship to various faculties, departments are studied. The areas of greatest potential are pinpointed and fully outlined. The results of this survey would then be given to the administration. If the administration is in favour of the system, then the designing of the system is carried out.

SYSTEM INVESTIGATION

What a system designer needs to know before entering the data producing area scheduled for computerization is what is the output required, in terms of content and timing. If the area is of the fees of students, then the output would be a statement of accounts, and timing would be the beginning of each term.[1] Or if it is grading of marks obtained in the examination a 'report card' on each student would be required at the end of the session.

We shall deal with a case where record of grades are to be kept by the computer system. The systems designer will have to study how the present records are kept.

Information to be gathered:

The systems designer must gather information on the following aspects.

(a) The objectives of the system.

(b) The existing procedures.

(c) The organizations doing the job i.e. whether it is carried out by the different departments or by the faculty or by the administration etc.

(d) The policies governing the area.

(e) The working papers used

---

(1) At present, the only area where student records are kept by the computer system is records on payment of fees.

(1) reports
(2) forms
(3) records.

(f) The effectiveness of the system.

The following information concerning records is to be gathered:-

(1) Information recorded in each column or space.

(2) Source of each entry.

(3) Volume of postings.

(4) Frequency of posting i.e. termly or yearly.

(5) Responsibility for maintenance.

(6) Man hours required.

(7) Methods of verifying posted data.

(8) Method and frequency of summarising posted data.

(9) Equipment in which record is filed.

(10) Filing arrangement including type and frequency, of visual indexing.[1]

(11) Purpose of record: what is the nature and frequence to it and of the inquiries for the information it contains. What reports are to be prepared from it? Are copies made? What are they used for?

(12) The time the record must remain on the file before it is destroyed.

(13) Legal or policy requirements of the University governing retention of records. To the extent that files or forms, correspondence, examination papers are referred to and analysed, they become a type of record, and together with this check-list of information on them, should be included in the forms binder.

---

[1] See Chapter on filing techniques.

QUALITY

Information on quality is the information on degree of error the procedure can tolerate. Where mistakes are least important and where they are crucial. The means of crosschecking and their effectiveness should be studied.

Briefly, the purpose of system investigation is to gather information on the data processing area. This information is concerned with the output the system produces; the input required, the organization of performing and processing; the policies the system operates under; the quality of the output produced, the suggestions made for improvement of the system; the concept of the ideal system and the problem areas in the system. This would provide the base on which a computer system can be designed.

SYSTEMS DESIGN

The systems design approach is concerned with how the data specified and documented in systems analysis can be provided. The data must be provided as useful output and would involve time availability, format and content of data.

When designing a data processing system, there are certain objectives that should be kept in mind.

1) Attempt to achieve standardization among like units. For example, the grading system in the different faculties should be the same. This will eliminate complications in the program.

2) Eliminate unnecessary functions.

3) Eliminate unnecessary reports, records and forms.

4) Eliminate superflows data.

5) Establish necessary controls.

6) Eliminate unnecessary detailed data.

7) Eliminate duplication of function. For example, it is not necessary for both the faculty and the administration to have a life record of the student.

8) Eliminate duplication of purpose, operation and duplication of information and forms.

9) Smooth out work flow by:-

reduction of waiting time, elimination of bottlenecks, advancing the cut-off data for source documents so that data enter the computer system earlier and finally, attempt to stick ahead of schedule.

With these objectives, the system designer has to design a system somewhat like this.

<u>A General Design of a computer-data processing system</u>.

A computer data processing system can be likened to a company manufacturing a good from raw materials, the 'good' being output data and the 'raw materials' the input data.

From source documents, which may be mark sheets, record cards etc. data is gathered for input controls are set up to protect against loss, duplication and consumption of information.

The information is then punched on cards and verified with coded information corresponding to data in the document. They may pass through some pre-computer processing or tabulating equipment.

The punch cards are then transferred to the computer centre, where the information is converted to magnetic tape or disk storage. The data is then processed. Report on grades or fee accounts are given out as output.

When processing is complete, the output files are printed out in a form suitable for usage.

At all points in the system - precomputer processing, output checking - unacceptable information is rejected. Error correcting procedures must be established to deal with any errors. Thus the design should take into account the following procedures:-

Name of Operation
Source Document
Control
Encoding
Keypunching
Verifying
Precomputer Processing
Transmission.

All these deal with input organization.

Conversion to magnetic tape/disk
Computer processing
Conversion from magnetic tape/disk

All these are computer operations.

Handling
Review
Distribution
User.

All these are output organization.

The designer has to take into account all these aspects. Hence the tools used by the system designer are these described below.

The Run Diagram.

The run diagram sets down on paper the overall sequence in which data handling work isto be done.

A run in computer language is the transformation of a given amount of input data into a specified amount of output data. Thus the updating of one master file from the charges reported in say, the payment of fees file, to produce an updated master file can be referred to as one run.

A pass, one or more of which can be found in a run, is one movement of data from input into the automatic computer. For example, in a sort run, (i.e. sorting records in a given order) a number of passes of the input file are required as the computer gradually moves the records from the original order to the desired order.

There are three parts in a run diagram. The input files, the computer and the output files, one below the other (usually).

The output and inputs are shown as to which media is used. In our diagram, the circles represent magnetic tapes.

Flow Diagram.

The second major tool in systems design is flow diagram. It gives in detail the transformation of input into output data. For each run, a detailed flow diagram is prepared.

The Data Sheets:-

Formats or data sheets are used to explain in specific terms each field that is to appear as input or output, including the number of characters in each field, the number of times the data will appear on record, the sequence of the fields, the position of a decimal point etc.

Format sheets, together with flow diagrams and run diagrams, would be the basis on which the computer program would be drawn up.

The Procedure Outline and Procedure Manual.

This manual is a description in general terms the non-computer operations which are performed with the computer operation. The computer operation is only one part of a whole set of operations to process the data. The Procedure manual is a detailed description of how such work is done.

This would show the relationship of the various operations with the computer operations and hence would facilitate in the preparation of input and output data.

PROGRAMMING.

Programming has been described in detail in Chapter Three. Briefly, it can be mentioned that programming would involve the implementing of a systems design in so far as the operations of a computer i.e. to carry out the actual processing of the data in the computer, a set of instructions in FOTRAN Language is written for the computer to operate on.

FILE MAKING.

File making is the most important operation in record-keeping. At present, information about a student may be kept in a number of ledgers (e.g. the Matric No. ledger) and for index card files or in various other binders and files. From all these files, information about the student has to be converted to magnetic tape storage. One advantage of this type of storage over the present system is that all records can be found in one spot i.e. in the specific area in the magnetic tape, and not all over the place.

Since the system design must specify the location of all these information, it is necessary to specify what information will appear in what format. Knowing what output is required, the systems designer has to design which of the data must go into record. For example, he must decide whether students are to be arranged by Faculty and Name or by Matric Number.

The creation of a master file would involve one or more runs in the computer to accept the information, from punch cards, into the magnetic tape. By the time the master file is complete, it may be out of date for example, students may change courses, or change place of residence, or change marital status etc.

Hence there must be some procedure to continually update the master file. This is more relevant if the master file contains information regarding payments of fees.

CLERICAL PROCEDURES.

If a computer system is set up to keep the records, there should be training for personnel to adapt to the new system. Instead of typists, more punch card operations may be necessary. However this need not concern us over much and so we proceed to the final item in systems design, namely that of Testing the System.

TESTING THE SYSTEM.

Once we have drawn up the system and prepared the necessary programs, the final step would be to test the system. This would involve three stages - run (or program) debugging, system testing and parallel running.

Run Debugging.

Each run is a program drawn up for a specific number of computer operation. The program is then run on the computer to whether it works. In the usual case, it would not work for the first time. Hence it has to be corrected and re-run. Then again, it is possible the output data is less or more than is required, or the time taken may be too long. Hence the program has to be modified until it is satisfactory.

Systems Test.

When programming is complete, the systems designer has all the runs necessary to complete the system. The next step is to see whether the various runs would, together, produce the results required i.e. whether they fit together as a functioning system. The individual programs may be inconsistent with each other, or there may be duplication in the runs.

In the systems test, the sequence of runs is run out with the actual data as test or an invented data. The advantage of actual previously processed data used in the test would mean the results also can be checked. However, invented data would do just as well.

Preparation of the test data is responsibility of the systems designer. The data should be exhaustive and should take into account all variations possible. File maintenance checks should be run first so that these files can be used in the checking of the other runs.

Printouts at different parts of the operation would simplify checking for errors. If every part of the system is checked at intermediate stages, errors can be accurately located. Once the system is checked out to be correct, the final step in the implementation of the system is parallel running.

Parallel Running.

In parallel running, both the computer system and the previous system (manual or whatever), are carried out simultaneously. This is to check whether the results produced are the same or not. This is a check to see whether the computer system is as accurate as the old system. This may involve a terrific amount of work, but it is necessary to achieve an efficient system.

This would then complete the work of the systems designer. The rest of the procedures can be carried out by the regular staff in the computer centre.

Briefly, the introductory of the computer system for keeping records would involve a feasibility study, a systems analysis and design, and final operational preparations. The implementation would be complete with testing and making the system operational.

CHAPTER VI

FILE ORGANISATION TECHNIQUES

It is obvious that a system of keeping records is essentially, the keeping of files. File organization is the process of relating the identification numbers of a file record to the address of a file in the storage unit. The basic objective of file organization is to keep the data in storage in a systematic way so that it can be retrieved in the fastest way possible when needed. The methods used varies, depending on the different applications.

In setting up a file for a direct access device, there are a number of questions considered. What addressing technique is to be used? How frequently will the file be referred to? Are disk files or magnetic tapes or both should be used? What are the inquiry requirements. What will be the predominant reporting sequence? In updating records, are all records and information for an application kept together or are they kept individually. Will there be random processing, batched sequential processing or both. What are the file maintenance requirements.

All the questions have to be looked into before setting up a filing system. Insofar as storage is concerned, it is felt that a combination of disk storage and magnetic tape is the most advantageous, since it would be less costly to keep for the mass of records that are to be kept in the University with 7,600 students.

It is also an advantage to keep all information about a student in one mailer-record. With a computer system, this is possible. Reporting sequence is also important in file organization. For example, it may be more convenient to keep records in terms of matric numbers rather than names. But this may involve the problem of students from different faculties getting mixed up. So a decision may have to be made to find a solution.

In setting up mass in storage files, it must be decided how it should be organised, basing the decision on the speed of access, convenience of application etc. We shall now look at file composition.

File Composition - Random and Sequential.

A file of records can be arranged in two major ways - randomly and sequentially. In a random file, each record is at an address computed by a randomizing routine - i.e. a program which calculates the address from an item's control number. The arrangement of the records is not sequential. To find a record in such a file, its address is simply computed from the ID (identification) by the same formula that was used to put it there. Through this, the necessity to have index tables can be avoided.

In a sequential file, records are sorted and stored in the disk storage with successively higher ID numbers will have successively higher addresses. However, it is not necessary for both the ID number and the address number to be the same.

Both these methods have techniques to minimise number of accesses to the file.

Random-order Techniques.

Activity sequence. In activity sequence, the records which are likely to be most used are put in the beginning of the storage medium i.e. the records which would be involved in the most activity would be at the beginning and the records which involve the least activity would be furthest away. By this method a lot of time-saving is made possible. To maintain the file in the correct activity sequence, it is necessary during accounting period to tally number of references to each record. Periodically, the file is revised.

Correspondence Method: In this method, when applications are related to one another, the first application is occupied in one part of the file. For the next application, the corresponding record will be in the next section of the file, and it will be relative to the initial record, i.e. if one record is at the address 000604, if one application occupies 0 - 100 positions, then the next corresponding record would be at 000104 i.e. an increment of 100 addresses away. Thus related records would be in corresponding positions.

With this method, great saving of time is made. In the example above, if the related records are not in arranged in corresponding method, but are 100 addresses apart, each of the 100 addresses have to be checked before arriving at the correct one. But in this case, there is no need to check through as it would be immediately known where the record is.

Associative Method: This is similar to the corresponding method. In this case however, the addressed are kept in the main record and are not incremented i.e. there is no ordered arrangement.

Tree Method: In setting up chain records, number of accesses can be reduced by using the tree method. The first ID to create an address is stored in the same manner as before. When a duplicate address appears however, the original ID is tested to see whether it is even or odd. If even, an address from the open list will be selected and stored in the seven characters of the record just in front of the last seven. If it is odd, the address from the open list is stored in the last seven characters of the original record. As more duplicates occur, the even, odd path is followed.

Sequential Order Techniques

Assigned file address. Using the direct address and indexing techniques, the input file to be stored on the disks is sorted and read into memory, followed by the building of fine tables. Thus the fine tables need not carry all possible ID's - only the ones which are going to be used. For example, only 30 numbers out of 100 may be found in the fine tables. As soon as a range has been read and the number listed, the list is stored on the disk as the fine table and the procedure continues until all the addresses have been assigned.

Unassigned file address. In this method, the disk file is treated as if it were continuous reels of magnetic tape with all records in sequence. First, input transactions are presented in batched sequence. Then, as many sequences as possible are read into the memory of the computer. One input transaction at a time is read and matched to a memory segment for updating purposes. When the appropriate segment is updated, that segment is rewritten in the disk file. To update an entre file, all segments must be read into the memory, but only those segments which have input transactions matching them will be rewitten in the file.

Linkage. In the linkage technique, a number of records are updated with one output i.e. if students names are arranged in one record according to Faculty, in another according to State, and yet in another according to sex, the updating of one record would automatically be followed by the updating of all the other-linked records.

Sequential Versus Random-file Organization.

There are some of the comparisons which can be made between the 2 methods of file organisation.

1) The sequential (table) method allows denser packing and permits a higher percent disk storage space for use.

2) The random method is faster for random processing.

3) The sequential method is more efficient for generating reports that depend on a search of the file in control number sequence. To accomplish the same thing with random organization, a finder file (a list of control numbers of all the records in the file) is needed.

4) The random method more readily handles addition to and deletions from the file. With a sequential file, a large number of additions and deletions forces frequent re-organization.

5) With sequential organization, sequential processing is preferable. However, when it is batched, if the input is sorted into randomised formula sequence instead of control number sequence, both are equally efficient.

From the above comparisons, it can be observed that if the storage capacity is very limited and files must be tightly packed, and/or if there are many reports to be run and control numbers are long, the sequential method is preferable.

If random output and time is the most important and/or there are frequent and numerous addition, deletions, then random method should be used.

Thus when both file organization techniques should be looked into and the most suitable one used when organizing a file.

Record Composition:

The way the records are compiled in a file is also important. It should be arranged in a manner as to save time or make updating easier.

Abstracting: When mass storage is suitable, but not justifiable in terms of size, an abstracting method may be used to reduce the size of the system. In abstracting, only essential record information is carried on direct access file. The rest is on reels of magnetic tape which are processed together with direct access files at regular accounting periods.

Consolidated files: In this method, rather than having several files on each aspect of a student, say, only one file is used. That is, all information pertaining the student will be in one file, rather than in many files.

This saves storage space.

Multiple records per segment.

When size of a record is smaller than a segment, it is better to cram more records into it rather than waste space. In a direct addressing system where the ID's of two related application areas coincide, the 1st part of the record could be used for one application and the last part for the other.

Random-Access and On-Line Systems.

"On-line" refers to the operation of input/output devices under direct control of the central processing unit. When this can be accomplished, it eliminates the need for human intervention between input origination and output destination within computer processing. "On-line" can be applied to the units near to and under the direct control of the L.P.V. (e.g. on line printer), or for the units which are not located near the central processing unit, but which require a communication link.

In batch-handing installations, many operations are reserved for "off-line" handling. Transfer of data from cards to tape or vice versa are done in separately and not on the computer, since they would hold up the work of the central processing unit, thus costing money.

Mass (Direct Access) Storage and Responsive Systems

"A mass-storage" system responds to changing priorities and requirements. Rather than processing data on a first come, first served basis, a mass-storage system can respond effectively on a controlled "first things first" basis. Thus the capability of altering priority according to the immediate needs of daily activity is present. Thus there can be reduction in program modification or introduction of new programs, and thus there is saving of time and effort.

Selective Updating.

This is another form of "activity sequence" method. The most likely information to be updated is kept in one segment, and the least likely to be updated (the static data) is kept in another. Hence when these segments are read into memory, only one need to be rewritten.

We have now examined some of the techniques of file-organization. Out of these, the sequential method would be better for our purpose because, in student records, there is not much need for updating, and also, there is a great amount of data to be stored.

CHAPTER VII

CONCLUSION

So far, we have studied the computer system and have discussed a method of setting up records by this system.

We have studied the computer in detail, discussed programming procedures, made up a sample program, and have examined in detail how to go about making use of the system to keep records. In the previous chapter, we have discussed file keeping techniques.

There are two major reasons why computers would be used in an organization. They are (1) to build a competitive lead - to have better and more timely information to improve decision making, and (2) to save on operations, by using computer as a cost cutting tool.

The University is not a business concern and therefore there is no reason for it to fight for any competitive lead, but it would like to have timely and better information for decision making for processing data on 7600 students is quite a job. However, the major reason computer would be justifiable in the University would be if cost saving is obtained.

The cost of running the University has grown into gigantic proportions and is increasing every year. Thus a reduction in any cost would be welcome. Estimates would be required and of course, a feasibility study would present the estimate costs.

The feasibility study would study processing requirements, examine the various computers on the market, estimate saving on costs. A detail cost tabulation would be something like this:-

Computer Costs.

General

ITEM	No. or value.	Any Change Expected: No — No Change	Any Change Expected: Yes — Cost Increase	Any Change Expected: Yes — Cost Decrease
LABOR				
Department A				
Supervision				
Type C Employees				
Type D Employees				
Type E Employees				
Department B				
Supervision				
Type K Employees				
Type L Employees				
Fringe Benefits				
MATERIAL				
Forms				
Office Expenditure				
General Expenditure				
Other Materials				
EQUIPMENT				
Rentals:				
Type P Machines				
Type Q Machines				
Loss of Capital Value				
Type R Machines				
Furniture & Fittings				
Total of Cost Increases, Decreases				

Computer Costs.

General

Since in our University cost would be the main decisive factor, we shall examine the various costs involved. There are two types of costs involved. One is the outlay for preparation and implementation, and for initial application, to acquire and install a computer. We already have a computer, but it may be necessary to set up another.

The second cost is the periodic outlay for maintenance. These are operating expenditures.

Investment Expenditures.

The IBM 1130 at our computer centre cost about half a million dollars for buying and installation. This is about the smallest computer available for our purpose. The basic computer itself could be about $100,000 but the additional input, output equipments add up to the total amount.

Most manufacturers rent out their equipment as well. Hence it may be economical to rent rather than buy.

Another large item in investment expenditures is change-over costs. These include the costs of system analysis, systems design, programming, debugging, etc. These also include the increase in expenditure due to changeover. People unfamiliar with the new system would be slow in adapting themselves and this would lead to more expenditure due to time wastage.

How long the changeover period will last depends on how will the changeover is executed. If the arrangement has thought boldly, planned well and executed firmly and decisively, the changeover costs can be held low. If on the other hand, management is unsure of the new system and is cautious, changeover time would be longer.

The next largest cost would be cost of installation. As we already have a computer, this would not be much. However, if a feasibility study is carried out and it is found another computer is required, then another half-a million dollars may be needed.

The fourth cost would be modelling of a place to keep the computer, new furniture fixtures etc. These costs can vary from nothing to a million dollars.

Finally another cost would be that of buying inventories and test equipment, punch card etc. Spare parts and test equipment would be necessary if the computer company is not doing maintenance, but in our case, all maintenance is done by the IBM.

Operating Costs.

The major cost of operating a computer is the cost of input. The cost of input is the cost of getting information and putting it into a form that can be accepted by the automatic computer. This cost is high because it includes labour (for example, the time of key punch operators); it includes materials and supplies (forms, punched cards, etc.) and it includes machine rental or charges (cost of key punches, typewriters etc.).

The second largest cost is the amortization of the investment expenditure. If it is leased from manufacturer, amortization would be charged only in part, but not fully, by the rental charge. This cost is difficult to calculate, but for write off purposes, a life of seven to ten years is used.

The third largest cost of operating is usually the labour cost for operating personnel (excluding those in input preparation). In the University, the computer men key punch, program and also operate the computers. There are also a number of key-punchers. However, to get better service, specialists may have to be employed and that would increase costs.

There are about seven classes of personnel for a computer centre. They are console operators, periphereal-equipment operators, systems personnel (prepare problems and applications for programming); programmers, clerical personnel, maintenance men, and finally the managers of the computer centre. The salaries of these personnel ranges from $27,000 to $72,000 a year.

The fourth largest cost is taxes and insurance. Taxes of course are not paid by the University, but insurance has to be paid.

The fifth largest cost is that of output. These include cost of supplies (preprinted forms etc.) and some labour and miscellaneous charges. Output costs are more variable than input costs.

The sixth largest cost is that of floor space. This would be rental charge or opportunity cost.

The other operating costs include use of power (electricity) and some other minor costs.

For the 1967-68 session, the amount of money spent on the computer centre was \$50,000[1]. This gives some idea of the costs involved in maintaining a computer.

The question faced by the authorities would be whether it is less costly to keep records by the computer system or go on with the present system. In the long run, computer system would be more advantageous, but due to lack of funds, it is not possible to set up such a system at present. The computer centre is not capable to deal with the record-keeping as it is already fully used.

Perhaps with a twenty-four hour working period, it may be possible to run a computer system of keeping records. However, costs would still be high. The University is already committed to various projects which require more funds than are available, hence, a computer system of keeping records is not feasible in the near future.

---

(1) University of Malaya Nineteenth Annual Report 1967-68.

BIBLIOGRAPHY

Books

1. Awas, Eias. M. Automatic Data Processing: Principles and Procedures. Prentice-Hall Inc. N.J. 1966.

2. Biocchino, William A. A simplified guide to Automatic Data Processing. Prentice Hall Inc. N.J. 1967.

3. Chappin, Ned., Ph.D. An Introduction to Automatic Computers. D. Van Nostrand Co. Inc. N.J. 1966.

4. Laden, H.N., Ph.D. and T.R. Gildersleeves: Systems Design for Computer Applications. John Wily & Sons. Inc. N.Y. 1967.

5. Lytel, Allan: Fundamentals of Data Processing. Howard W. Sams. Photofact Publications, Indianapolis, New York 1966.

6. Price, Wilson T. Elements of IBM 1130 Programming. Holt, Rinchart & Winston N.Y. 1968.

7. Sammet, Jean E. Programming Languages: History and Fundamentals. Prentice Hall Inc. N.J. 1969.

8. Scott G. Theordore G. Basic Computer Programming. Doubleday & Co. Inc. Garden City, N.Y. 1967.

Reports and Manuals

1. IBM 1130 Fundamental Characteristics. IBM Reference Library. File No. 1130-01. Form A26-5881-1.

2. IBM 1130 Fortran Language. IBM Reference Library. File No. 1130-25. Form C26-5933-3.

3. Standards for Systems and Program Documentation. Jabatan Perankaan Malaysia.

4. University of Malaya 19th Annual Report. University of Malaya.