

IBM

650 Data-Processing System

with

355 Random Access Memory

and

838 Inquiry Stations

MANUAL OF OPERATION: PRELIMINARY EDITION

©1956 by
International Business Machines Corporation
590 Madison Avenue, New York 22, N. Y.
Printed in U. S. A.
Form 22-6270-1

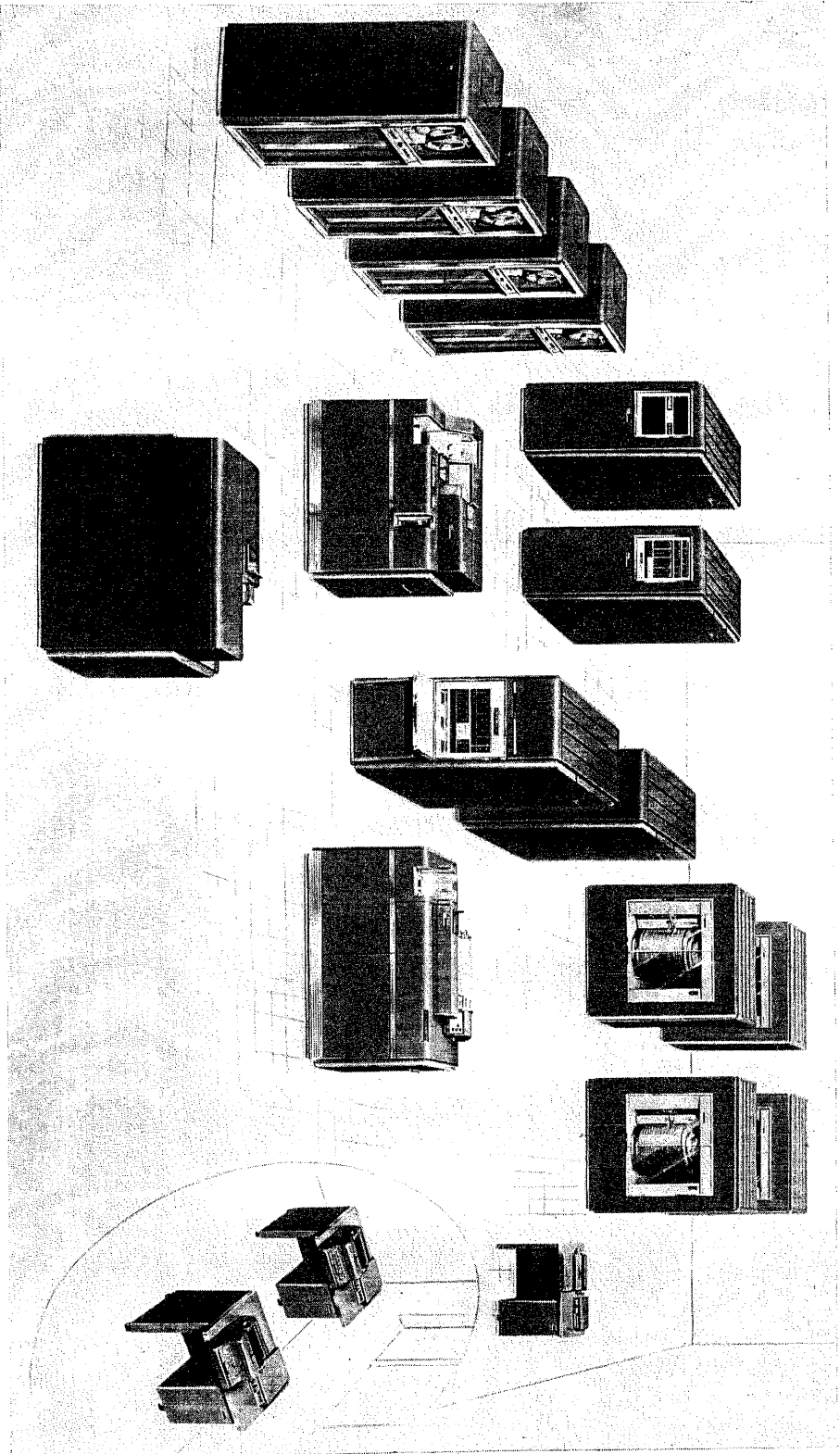
CONTENTS

INTRODUCTION - - - - -	5
IBM 650 DATA-PROCESSING SYSTEM - - - - -	6
650 Console - - - - -	6
Input Units - - - - -	6
Storage - - - - -	7
Output Units - - - - -	8
Inquiry Stations - - - - -	9
Application for 650 RAM - - - - -	10
Block Diagram and Record Layout - - - - -	12
355 RANDOM ACCESS MEMORY - - - - -	17
Storage Arrangement - - - - -	18
RAM Addressing - - - - -	18
Flow Paths - - - - -	22
Timing Considerations - - - - -	25
Over-all Job Timing Considerations - - - - -	27
DATA STORAGE AND ADDRESS MODIFICATION CONSIDERATION - - - - -	31
Table Lookup - - - - -	31
Arithmetic Operations to Convert Record Codes to RAM Addresses - - - - -	31
Packing Records - - - - -	32
838 INQUIRY STATIONS - - - - -	44
Physical Description - - - - -	44
Operation - - - - -	46
Operating Keys and Lights - - - - -	52
CHECKING AND FILE PROTECTION - - - - -	56

NOTE: This manual does not attempt to describe the entire 650 system. To fully understand the complete system, the following three manuals of operation should be studied prior to reading this publication:

- Form 22-6060. 650 Manual of Operation,
- Form 22-6265. 650 Additional Features (Magnetic Tapes, Printer),
- Form 22-6258. Indexing Accumulators, Floating Decimal Arithmetic.

Considerable detail is included here on timing considerations relating to the RAM file. The timings stated herein are approximate at the time of publication and are subject to change. They are included to assist in planning applications.



IBM 650 System with RAM

IBM 650 DATA-PROCESSING SYSTEM

with

355 Random Access Memory

and

838 Inquiry Stations

THE IBM 650 Data-Processing System with Random Access Memory combines the inherent data-processing capacity of the IBM 650 with the facility for large-capacity random access storage. This combination provides an efficient in-line data-processing machine.

The in-line method of data processing maintains the records of a business continually up to date. Any transaction affecting a business may be processed when it occurs, and all records and accounts affected will be updated immediately. The executives of an organization may have available at any time information representing the status of any account at that moment.

Records in the IBM Random Access Memory Unit are stored on the faces of magnetic disks. Each unit has the capacity for storing 6,000,000 digits of data. In the four units available for an IBM 650, there is the capacity for 24,000,000 digits of random access storage.

Only three programming instructions are necessary for communication between the random access storage and the 650.

The stored data in the Random Access Memory Unit are read and written by access arms. There are three independent access arms in each unit. Any record in random access storage is available to any of the arms. Speed in operation is realized because each

access arm operates independently, and all may be operated simultaneously.

Another feature of the IBM 650 Data-Processing System, important to in-line data processing, is the facility for quick communication with the system, to inquire into the status of records, or to enter new information. The inquiry stations for the 650 system provide stations from which inquiries and data may be sent manually to the system and to which the system may send replies and other data.

Each inquiry station consists of a modified IBM Electric Typewriter connected by electrical cable to the 650 system. As many as ten inquiry stations are available. These stations may be connected to the 652 by 50-foot standard length cables.

The function of each inquiry station is entirely under the control of the 650 programming. There is complete flexibility in the type of operation provided for each. Because the inquiry station is as simple to operate as a typewriter, the person making an inquiry from a station does not need to know the details of the 650 programming. However, the operator should be familiar with the function of the inquiry station as it applies to the system as a whole. The same positive accounting control that is available in all the 650 operations also pertains to the inquiry stations and RAM.

IBM 650 DATA-PROCESSING SYSTEM

A 650 system can be obtained in a capacity structure to fit any need in the intermediate data-processing area. The basic unit is composed of a 650 Console and the 655 Power Unit.

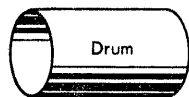
To this are added various input, output, and storage features, such as

- 533 Card Read Punch
- 407 Accounting Machines (Printers)
- 653 Storage Unit
- 652 Control Unit
- 727 Magnetic Tape Units
- 355 Random Access Memory
- 838 Inquiry Stations
- Indexing Registers
- Automatic Floating Decimal Arithmetic

These additional features permit selecting a system with the capacity to meet the needs of the accounting or scientific installation.

650 CONSOLE — 655 POWER UNIT

THE 650 Console is the heart of the system. It utilizes the stored program to direct the entire system. It contains the arithmetic unit where all accumulations and calculations occur. The logical decision features of the 650 test the arithmetic unit for sign status, test control data for branching to subroutines, and test the checking features for correction routines.



The major specifications of the 650 console are:

1. The 20,000 digits of storage, arranged as 2,000 words of memory on the magnetic drum (also available as 10,000 digits of storage in 1000 words).
2. Each word is 10 digits and sign.
3. The 650 system contains many self-checking features. All data and instructions are checked for validity as they are used. All operations are checked for correct execution.

4. One-word time on the drum is .096 ms. Access time to a word on the drum varies between immediate access (one-word time) and one drum revolution (4.8 ms).

5. Instructions and data can be optimally located on the drum to minimize access time.

6. It is a modified single-address machine where an instruction is a 10-digit word. Two digits specify the operation, 4 digits specify the data location, and 4 digits give the location of the next instruction.

7. Any combination of three 533's or 407's for a maximum of three units can be connected to the system. These machines are connected through input-output synchronizers.

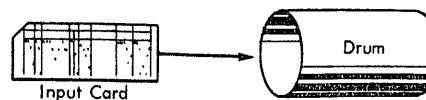
8. Alphabetic characters and special characters are stored and processed in numerical form. An alphabetic device is available for each input-output synchronizer and corresponding input-output unit. It automatically converts each alphabetic and special character entering the system to a two-digit number. This same device automatically performs a reverse conversion on alphabetic output data.

9. The control console provides manual control of the system. Signal lights on the console indicate validity errors and permit visual inspection of data and instructions in the machine.

INPUT

533 Card Read Punch

Card input to the 650 system is provided by the 533 Card Read Punch. The 533 reads cards at a rate of 200 cpm and is capable of entering 100 digits of information as each card is read. A control panel on the 533 provides for selection of card information and word arrangement.

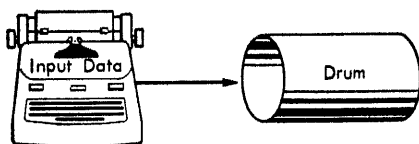


407 Accounting Machine

The 407 Accounting Machine can be attached to the system. Data from cards read in the 407 feed, as well as data from the 407 counters and storage units, can enter the system. The 407 feed operates at a maximum rate of 150 cpm, and like the 533, can enter up to 100 digits of information on each read operation.

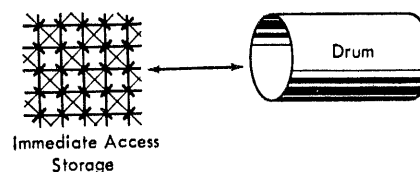
838 Inquiry Stations (available with 355 RAM)

Input to the 650 system can be from 838 units. These units can be placed at a data source, such as the receiving department, to update inventory records immediately. The typed input message and typed reply from the 650 indicate how the entry was processed and the resulting stock balances to provide an audit trail of the transaction.



727 Magnetic Tape and 355 RAM

Data from the magnetic tape and RAM can also be considered as input to the system but are basically storage mediums and are herein described as such.



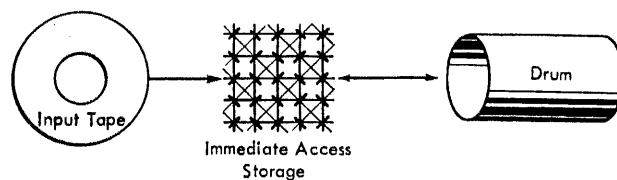
Additional operation codes are added to the programming structure which applies to immediate access storage. Operating features such as the following are provided by this unit:

1. Each of the 60 words is addressable.
2. Data or an instruction can be placed in any word location.
3. Block transfers between the drum storage and immediate access storage are provided. Block transfers can be of one word or up to 50 words.

Immediate access storage is required for any 650 system that contains magnetic tape or RAM units.

727 Magnetic Tape Units

Magnetic tape provides concentrated storage of data for high-speed reading and writing operations. Data written on tape by the 650 can be processed by the main frame of the 700 series systems. The 700 series systems can prepare tapes to be used by the 650 system.



STORAGE

IN ADDITION to the 20,000 digits (2,000 words) of general storage on the drum the following storage features can be added.

653 Storage Unit

When used with magnetic tapes or RAM, the 653 Storage Unit contains immediate access storage. This magnetic core storage unit has a capacity of 600 digits arranged as 60 words. Each word consists of 10 digits and sign.

Operation specifications of the 650 tape and the system are:

1. Tape density of 200 characters per inch.
2. Tape speed for reading or writing is 75 inches per second.
3. Maximum record length of 600 digits written as 60 words. The minimum record length is one word (10 numerical digits and sign).
4. Up to 23,000 ten-word records or 7,600 sixty-word records can be placed on one reel of tape 2400 feet long.

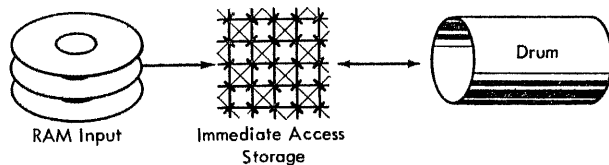
5. Alphabetic and special character data can be written in two-digit numerical form or in single-character alphabet and special-character form.

6. All tape records are written from and read into immediate access storage; therefore, simultaneous tape read-write is not possible.

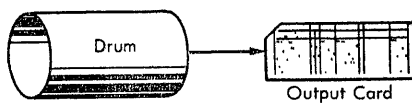
7. Up to six magnetic tape units (IBM 727 units) can be attached to a system. The 652 Control Unit is required to coordinate the operation of the tape units.

355 Random Access Memory

The IBM 355 RAM units provide extremely large storage capacity for data where any record is available in approximately one second. Up to four RAM units can be attached to the 650 to provide 24,000,000 digits of RAM storage. A RAM unit contains three independent access arms which seek, read, and write data under the 650 programming control.



The three access arms can provide a continuous flow of data between the RAM files and the 650 processing area. For example, while one access arm is recording (writing) an updated record in RAM, the 650 is processing another record. The concurrent operation of the independent access arms is initiated by 650 programming. Special characters and alphabetic data are stored in RAM in the same 2-digit numerical representation as is used on the drum. The 652 Control Unit is required to coordinate the operation of the RAM units.



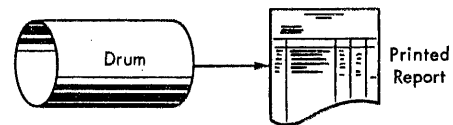
OUTPUT

533 Card Read Punch

The 533 Card Read Punch Units provide punched-card output for the system. The punch operates at a maximum rate of 100 cpm. One hundred digits of information are read out from the 650 on each punch operation. The 533 control panel provides for selecting this output information into the desired card fields for punching.

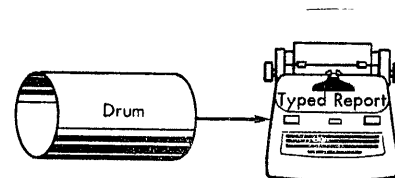
407 Accounting Machine

The 407 Accounting Machine provides a direct printed output for the system of up to 150 lpm. The 650 output consists of 100 digits of data on each print operation. The control panel provides for forms control, selection of data, and printing arrangement.



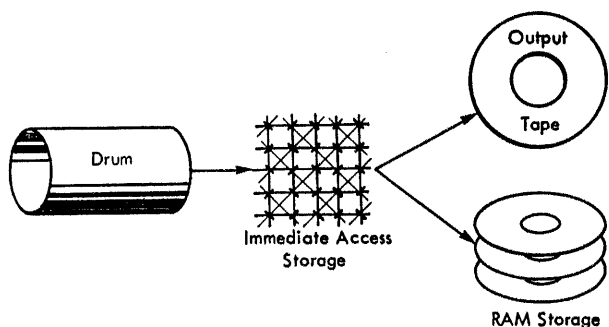
838 Inquiry Stations (available with RAM)

IBM 838 Inquiry Stations can be utilized as productive output printers. As such they type at a maximum rate of 10 characters per second. Format control is provided by a program tape. Further details on this unit are provided in another section of this manual.



727 Magnetic Tape and 355 RAM

Output results from the 650 system can be recorded on magnetic tape and/or in RAM files. The output tape can be read by the IBM 774 Tape Data Selector for independent tape-to-card or tape-to-printer operations. Alphabetic information to be read by the 774 must have been written in single-character coding. The output tape can also be re-entered into the 650 for a tape-to-printer operation, which is parallel-programmed with another 650 program.



CONTROL

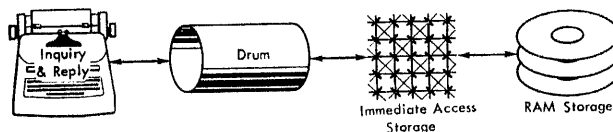
652 Control Unit

The 652 Control Unit contains circuitry that controls magnetic tapes, RAM and Inquiry Stations and must be included in a RAM or tape system.

INQUIRY

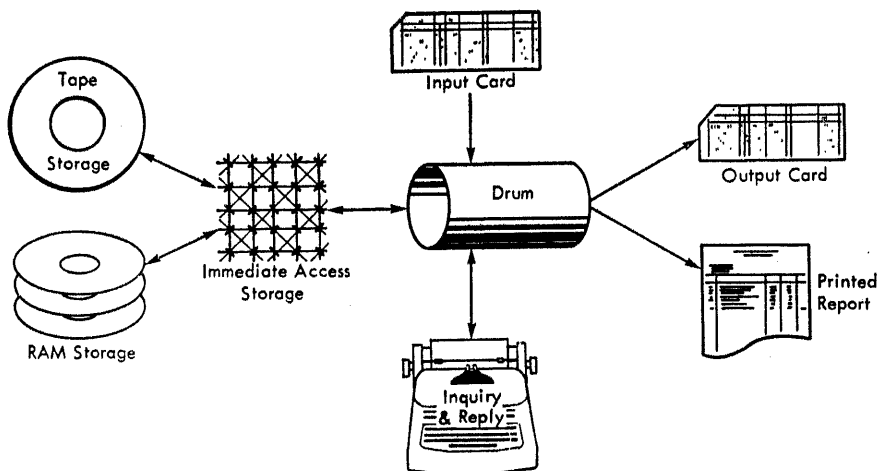
838 Inquiry Stations (available with RAM)

THE 838 Inquiry Stations provide for inquiry into data stored in any part of the 650 system. The 650 accepts the inquiry and formulates a reply that can be typed on the inquiry typewriter, or another inquiry typewriter if desired.



650 SYSTEM

THE IBM 650 Data-Processing System has the capacity and speed to meet many data-processing requirements. This versatility permits the 650 to move from role-to-role standard applications, in-line processing, scientific computing. It is truly a complete data-processing system.



APPLICATION FOR 650 RAM

MANY accounting applications can be combined into one 650 system with the addition of Random Access Memory. The application description that follows is based upon a company that serves 3,000 retail outlets and distributes 7,000 different items to these outlets. This application is used as an illustration. It does not fully utilize the capacity or flexibility of the RAM system. The 650 with RAM can be applied to combine inventory control, accounts receivable, billing, and a sales and cost distribution by commodity class into one operation.

The large capacity of RAM provides for storage of all the data required for the stated applications. The data necessary for each application can be stored into a separate section of a single RAM file. An appreciation for the magnitude of RAM storage capacity can be gained by the fact that this application utilizing 3,000,000 digits requires only one half of one RAM file. This is only $\frac{1}{8}$ of the total RAM storage capacity available. To utilize the maximum capacity available, additional applications can be integrated, or applications of much larger scope can be applied.

Random access to the section of the file containing inventory data permits each inventory item to be maintained on an up-to-the-minute basis instead of a daily, weekly, or periodic basis. This means that receipts can be immediately posted to the proper accounts, and subsequent withdrawals will then be made from current inventory balances. When the inventory balance reaches the minimum stock level, a re-order card is automatically punched. Similarly, back-order cards are automatically prepared if the stock is depleted.

As withdrawals are processed, their amounts are accumulated for posting to the accounts receivable section of the RAM file. Data in the accounts receivable file can be used to prepare supplementary reports, such as customer statements, aged trial balances, etc. Accounts receivable application processing is not completely described in this manual because of the periodic nature of reports prepared. Payments received for invoices rendered can enter the system as a punched card. This cash receipts card updates the customer current balance and inserts a paid signal in storage for that particular invoice. As customer statements are printed from the RAM file, the statement can reflect all paid and outstanding invoices.

The paid invoice is deleted from RAM storage at this time.

A billing operation requires the same data as that processed in the inventory and accounts-receivable operation. The billing is obtained as a by-product of the inventory processing. At the same time that withdrawals are updating inventory accounts, a line on the invoice can be printed or a card punched for later invoice preparation. Billing variations, such as tax computations, item quantity, customer discounts, freight-calculations, special notations, and bonus merchandise can be added to this write-up as needed. The tax, discount, and freight calculations can be integrated with the normal processing routine. Special notes and bonus merchandise can be added to the application as card input.

The same data can also be distributed and stored by general classifications. These classified data are then available to prepare various reports. One such report can be a sales cost distribution by commodity class prepared on a daily basis.

The foregoing applications are affected by an input of current transactions that update the accounts in storage and may result in a punched card output, or a printed output illustrated by the general flow chart (Figure 1). The input to the machine is an order card that contains the customer number, the item number, and quantity ordered for as many as six items. A receipt card is also an input card, and it designates the item number and the warehouse location where the item is stored.

Two kinds of records are stored in RAM: the item record, and the customer record. Each item record uses a separate track, and the track address is assumed to be the item stock number. The RAM capacity of 10,000 tracks is arranged so that the inventory data uses portions of each of the first 7000 tracks. Each customer is identified by adding 7000 to the customer number. The customer record that includes accounts-receivable data is stored on the remaining 3000 tracks.

Two output alternatives are presented. The output can be punched cards or it can be an invoice printed directly by an attached 407. Individual application considerations will determine which is best for a given situation. Both alternatives are presented to illustrate the flexibility of the system and the timing considerations that apply. Where a direct printed invoice is prepared by the 407 connected to the 650, exception cards are punched to signal back-orders and low-stock items that require re-ordering.

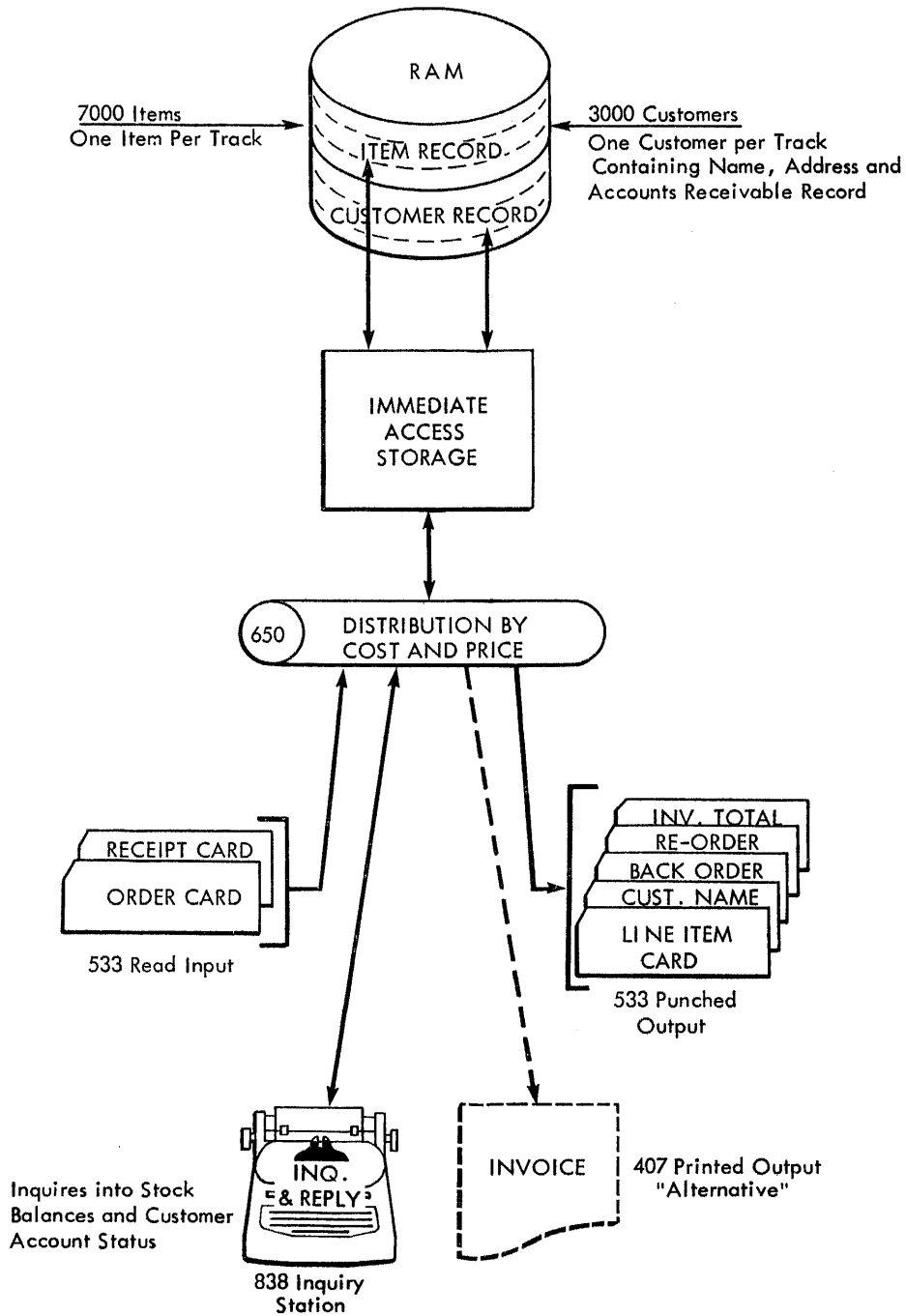


FIGURE 1. GENERAL FLOW CHART FOR 650 RAM APPLICATION

When a punched output is desired, the 533 punches a separate card for each line item, each heading line, and the invoice total. Exception cards indicating back orders and re-orders are also punched as needed. The output cards are then processed on a separate 407 or 402 Accounting Machine to print the invoice.

A sales and cost distribution is made to 100 item classes and is stored on the 650 drum. At periodic in-

tervals during the day, this distribution could be unloaded from the drum into punched cards for subsequent report preparation.

Manual inquiries into stock balances and customer account status can be made at any time. The inquiries can originate from any of the inquiry stations. The reply will be formulated by the system and returned to an inquiry station.

BLOCK DIAGRAM AND RECORD LAYOUT

VARIOUS MACHINE approaches can be used to apply the 650 RAM to this application. These variations depend on whether the output is in punched cards, or directly printed; whether each invoice contains few items or many items; whether the calculations per item are lengthy and complex or relatively simple. The block diagram and record layout (Figure 2) shows a punched-card output where one RAM access arm is used to seek, read, and write in the inventory section of the RAM file. Another access arm is used for the customer record. This is the simplest approach but not the fastest. Detailed charts, showing the timing relationships among RAM; the 650; the output punch; 407 printer; and inquiry stations for this application study, are described in later sections of this manual. The detailed timing charts illustrate the simple approach and the more complex methods that provide efficient operation.

Figure 2 contains four parts. Part one shows the main routine for processing inventory items from the order card. Part two shows a block diagram of the customer record processing for accounts receivable and billing operations. The block diagram of part three is the receipt card routine. Receipt cards can be placed in the feed hopper at any time during the processing. However, in practice they would be inserted in front of the order cards at the beginning of the run. Part four is a record layout showing the data included in the item record and in the customer record.

The descriptive write-ups of this block diagram follow the path taken by a typical receipt card followed by two order cards. The first order card is for a different customer than the second order card.

Block Diagram Description

Each part of the block diagram is further subdivided into sections. The sections are shown in brackets on the left side of the diagram.

At the beginning of the run, the last invoice number used on the prior run is set up in the machine. This is shown above the start routine at the top of part one in the diagram.

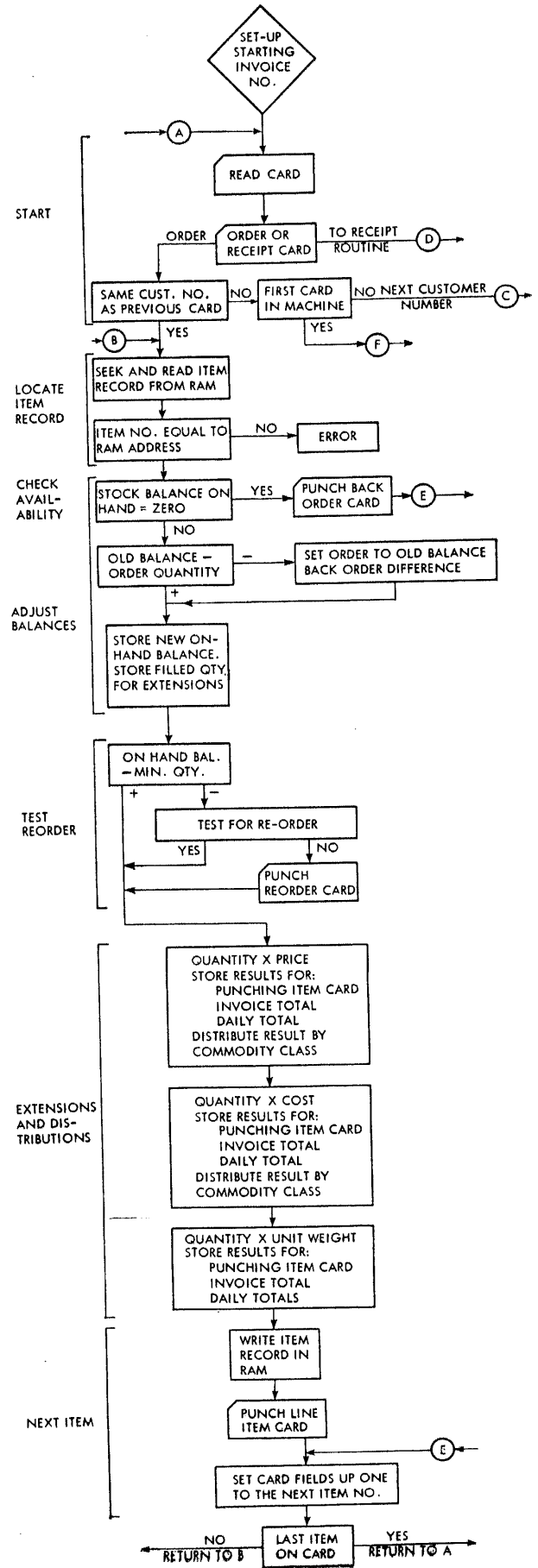


FIGURE 2, PART I. MAIN ROUTINE

If it is assumed the first card is a receipt card. The start routine would cause the program to branch out of the main routine to the receipt routine (D). The function of the receipt routine is to seek the item record in RAM, add the received quantities and write the up-dated item record on the same RAM track. If the receipt is for a new stock item, a complete new record must be constructed and written in RAM. This block diagram makes no provision for calculation of a new average unit price or the recording of this transaction on a register. Both of these can be included if necessary. The receipt routine is complete when the item record is updated, and the block diagram returns to A of the main routine.

The next card is read and analyzed by the start routine. If it is assumed that it is the first item card in the machine, the program branches to the customer record routine at point (F). The function of the customer record routine is to cause heading cards to be punched that can be used to write the heading portion of the invoice. Point (F) enters the customer record routine just above a seek operation that locates the desired customer record in the RAM file. A Read-RAM operation transfers it into immediate access storage.

As soon as the customer record is found and placed in immediate access storage, the heading cards are punched. Whether these heading cards are MLR cards (a card for "sold to" and a card for "ship to"), or whether one card for each heading line is punched depends upon the amount of alphabetic data in each address. Up to 30 positions of alphabetic data can be punched into each output card. The customer record is block-transferred to the drum, where it is stored until the end of the invoice is recognized. Upon completion of punching these cards, the program returns to the main routine at point (B).

Point (B) enters the main routine at the section titled *locate item record*. At this point, the first item on the order card is analyzed for part number, and the corresponding RAM seek and read are initiated. When the item record from RAM is placed in immediate access storage, its item number is checked to insure that the proper record has been secured.

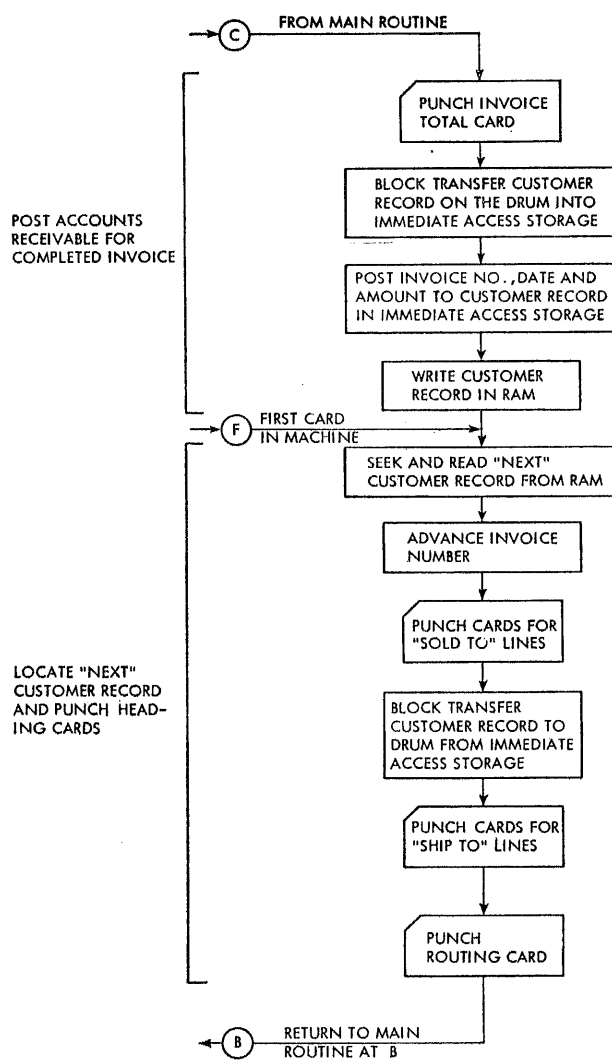


FIGURE 2, PART II. CUSTOMER RECORD ROUTINE

Now that the item selected has been verified as correct, the routine continues by checking the order quantity against the on-hand balance for availability. If no stock exists, a back-order card is punched, and the routine branches to point (E) near the bottom of the main routine. This branch bypasses a large part of the main routine; because if no stock exists, no

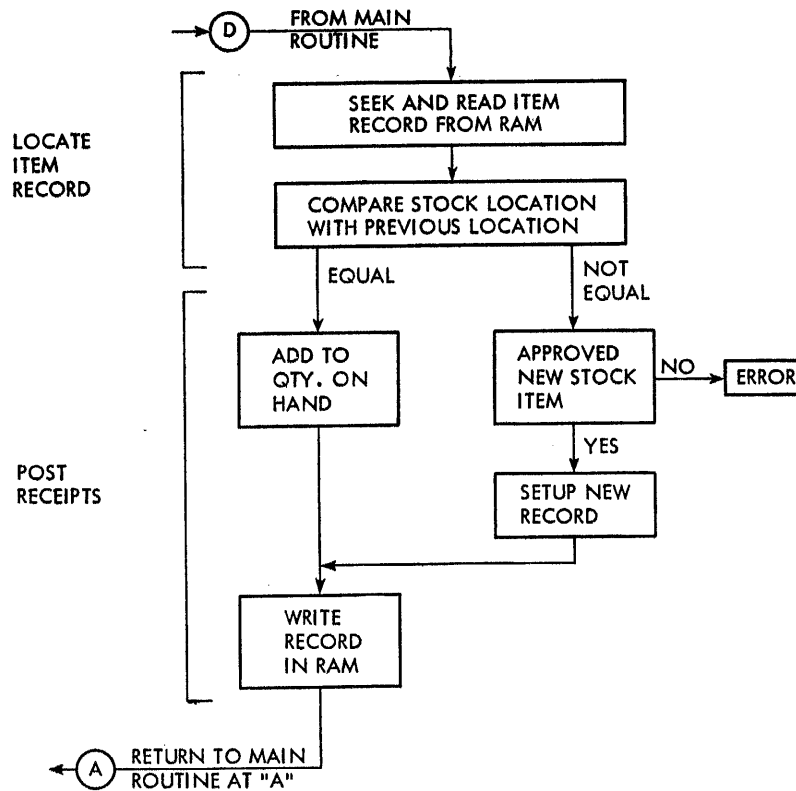


FIGURE 2, PART III. RECEIPT ROUTINE

further action is required for this item. Assuming that sufficient stock is available to fill the order, the routine progresses by subtracting the order quantity from the on-hand balance and tests the new balance against the minimum-quantity level to determine the need for punching a re-order card. The re-order card is used to request manufacturing or purchasing to secure additional quantities.

The routine proceeds into the extension and distribution section, where the filled-order-quantity extensions are made. Note that the distributions of price and cost by commodity class are made at this time. These distributions are made on the drum to the general storage areas reserved for these data. At this point, all the processing for a given inventory item is complete, and the updated item record in immediate access storage is written on its RAM track. A card is punched for this item, containing all the data needed to write a line on the invoice. The next field of the order card is then examined to determine whether two items were ordered on the same card. If the order card contains additional items, the routine branches to point (B) to begin processing the next item.

If no further items are included on this order card, the routine branches to A to read the next card.

Assume the next card is an order card for another customer. The program prepares to close out the invoice for the first order card and post the accounts-receivable totals. This is initiated by the start routine, which causes a branch to point (C) of the customer record routine. The customer record routine causes an invoice total card to be punched. The customer record is block-transferred into immediate access storage from the drum. The customer record layout (part 4) has capacity to record many unpaid invoices.

The data for this invoice are posted behind the last outstanding invoice in the customer record, and the total balance forward amount is updated. This "present" customer record is then written in RAM on the same track it was originally found. The invoice is now closed out and the routine begins to secure the customer record corresponding to the new or next order card. Heading cards are punched for this "next" customer and the program is transferred back to the main routine for processing the ordered items.

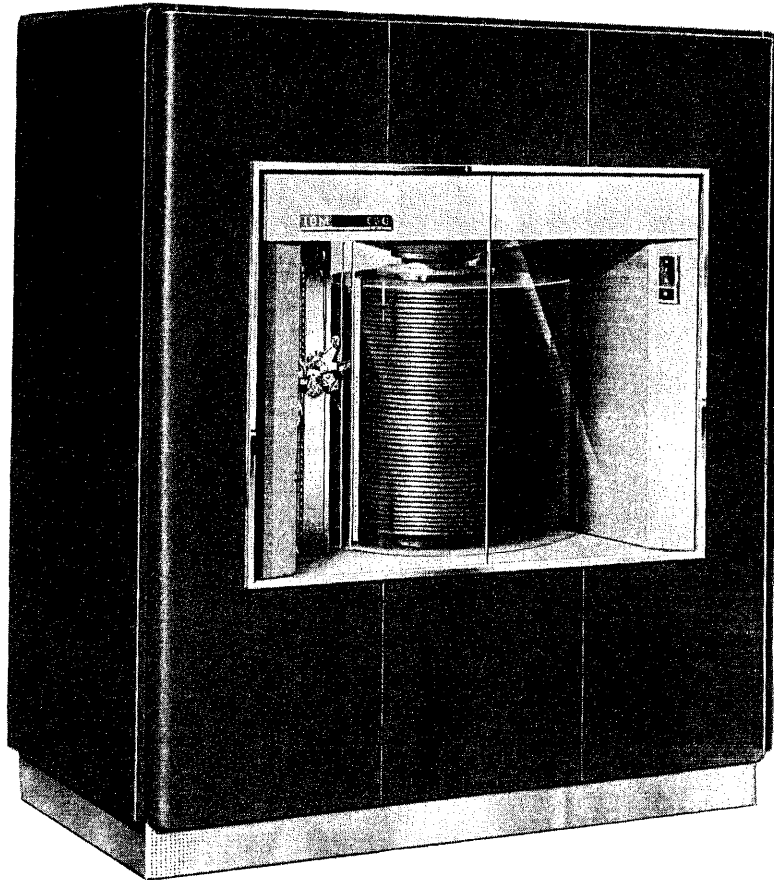


FIGURE 3. 355 RAM FILE

355 RANDOM ACCESS MEMORY

PHYSICAL DESCRIPTION

THE 355 Random Access Memory (RAM) is available in files, or units of six million digits each. Up to four such RAM units can be attached to a 650. This provides a total of 24 million digits of RAM storage.

Information is magnetically stored in RAM on circular disks. Each disk is 24 inches in diameter, and it is coated with a magnetic material on both sides. The fifty disks included in each RAM file are stacked on a vertical shaft and turn continuously at 1200 rpm. Because the disks are coated on both sides, each file contains 100 disk faces. The reading and writing of data on the surface of these disks are performed by read-write heads mounted on access arms. The access arms are mounted on a vertically movable carriage

alongside the disks and are capable of moving up or down to any disk. The magnetic read-write head on the access arm moves across the disk face to locate the desired track of data on the disk. Three mechanically independent access arms are provided for each file unit, and each arm can be independently directed to any track in the file. This independent operation of the three access arms permits concurrent seeking of three RAM records. Furthermore, it allows for overlap of seeking and processing operations.

Figure 3 shows the RAM file unit. Figure 4 is a schematic of the RAM disk face arrangement. Note the access arms; the arm straddles a disk as it moves across seeking the desired track. Each arm has two read-write heads; the upper head is for the top side of the disk, and the lower head reads and writes on the bottom side of the disk.

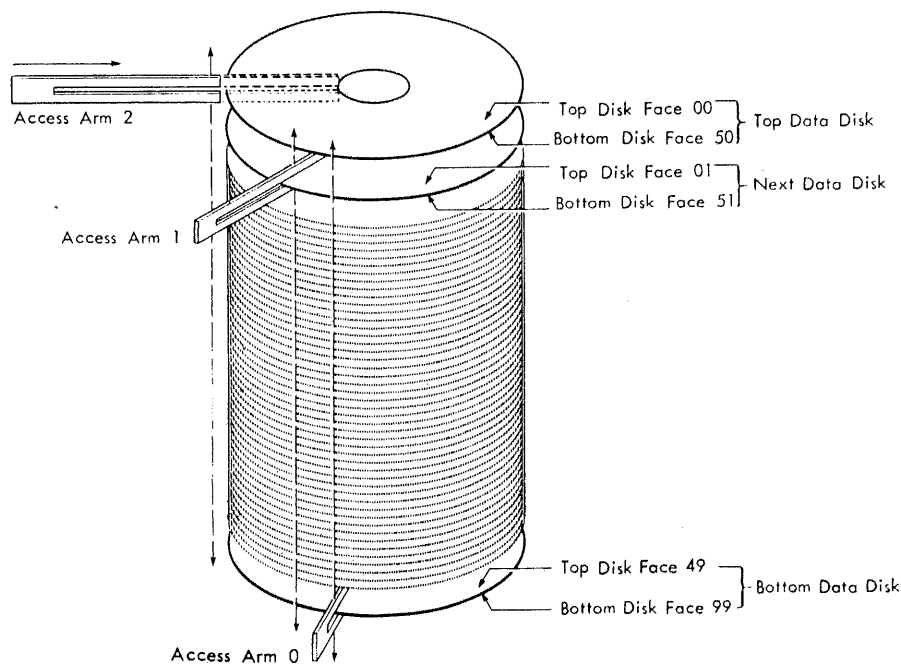


FIGURE 4. SCHEMATIC OF RAM DISK FACE ARRANGEMENT

STORAGE ARRANGEMENT

EACH of the 100 disk faces in the RAM file are further subdivided into tracks. Each disk face has 100 tracks, and each track has capacity for storing 600 digits of data plus 60 sign indications. The 600 digits per track are arranged in word lengths of 10 digits and sign per word. A track, therefore, has capacity for 60 words of data.

In summary, a RAM file of one cabinet contains:

600 digits per track

100 tracks per disk face

100 disk faces per cabinet

a total of $600 \times 100 \times 100 = 6,000,000$ digits

Access to information of a track is obtained by programming in the 650. An operation code `SEEK` directs the access arm to the desired disk face and track. An operation code of `READ` or `WRITE` is then given. The `READ` instruction then results in data from the entire RAM track being placed in immediate access storage, where it is then available for processing under program control.

The `WRITE` instruction stores the contents of immediate access storage (600 digits plus signs) on the designated RAM track. Either alphabetic, numerical, or special-character data can be stored in RAM. Alphameric and special characters are stored in RAM, in the same two-digit numerical form as used in the 650. Checking of data is provided, which insures correct transmission to and from RAM.

Information from immediate access storage is checked for validity (missing or extra bits) whenever data are either read in or read out. After a track is written in RAM, the track is immediately read back and checked against what is in immediate access storage, thus insuring correct writing of the stored information. If for some reason an error is detected, the track is automatically rewritten and checked again until a correct check is obtained. Checking will be discussed more fully in another section of this manual.

RAM storage is permanent storage, because data will remain stored indefinitely and may be read out as many times as desired. Data in RAM storage are changed by a `WRITE` instruction, which erases each previous digit as the new digit is written.

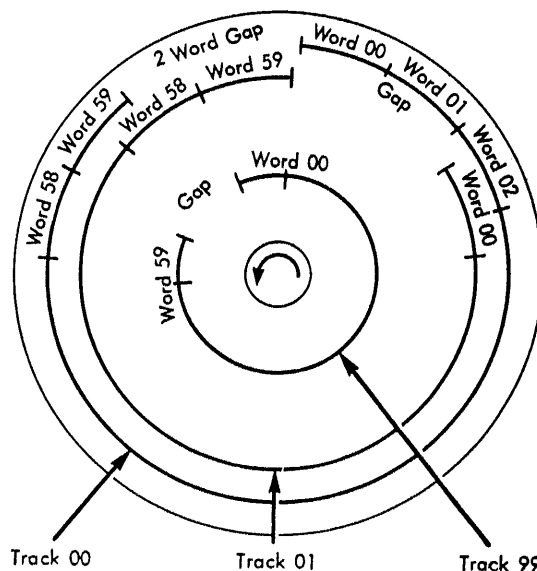


FIGURE 5. TRACK AND WORD ARRANGEMENT ON A DISK FACE

Information is recorded magnetically on each track. This information is stored serial by digit, and serial by bit. The conversion from serial-serial in RAM to serial by digit, then parallel by bit in immediate access storage is automatically handled whenever data are transferred between the two storage media.

The schematic (Figure 5) shows the track and word arrangement for a disk face. Track 00 is the outer track, and track 99 is the inner track. The total length of a track is 62 word lengths, where two word lengths constitute a gap, followed by the 60 words of data. Note that the gaps and data words on a track have no physical location reference to corresponding gaps and data words on the other tracks. This is explained in the sections describing the write and read operation codes.

RAM ADDRESSING

EACH RAM file contains 10,000 addressable tracks. The RAM address is a 6-digit code that specifies the file, disk faces, track and access arm, and is constructed as follows:

- File unit number (0-3)
- Disk face (00-99)
- Track (00-99)
- Access arm (0-2)

As an example, a RAM address of 2-04-91-1 specifies file unit 2, disk face 04, track 91, and access arm 1.

FILE UNIT	DISK FACE	TRACK	ACCESS ARM
2	04	91	1

Another example of a RAM address:

FILE UNIT	DISK FACE	TRACK	ACCESS ARM
0	00	00	0

This example specifies file unit 0, disk face 00, track 00, and access arm 0.

FILE UNIT	DISK FACE	TRACK	ACCESS ARM
3	99	99	2

The foregoing address will direct access arm 2 to the last track (99) on disk face 99 in file unit 3.

This RAM address is placed in the distributor by 650 programming whenever reference to the RAM file is desired. The six-position RAM address would constitute the low-order six positions of the distributor word. The subsequent RAM operation code automatically refers to the distributor for the RAM address and is explained in the *Operation Code* section of this manual.

The schematic of Figure 4 shows the disk face arrangement in a RAM file. Disk face 00 and 50 share the same disk. Disk face 01 and 51 are on the next disk below. Note that the disk face address of the bottom of a disk is always 50 greater than the address of the top face. This note is important because the seek access time between tracks on the same disk is considerably less than the seek access time between tracks on different disks.

Operation Codes

Three operation codes are added to 650 programming for RAM control: *seek*, *read*, and *write*.

In the case of the read and write operation codes, the D-address activates immediate access storage to function in the transfer of data from and to the disk file and must be 9000. Another address is needed when a RAM operation code is given. This is the RAM address, which was previously described in the RAM addressing section of the manual. The RAM address is a six-digit code, and this address must be in the 650 distributor when any of the three RAM instructions is initiated. Immediate access storage will be in use during the entire read or write operation.

85 S-RAM (Seek RAM)

The function of this instruction is to cause the access arm to move to the disk face and track specified by the RAM address located in the distributor. The time required for the access arm to seek and find the specified disk and track varies with the distance it has to move. If the RAM address specifies a new track on the same disk, the access time is about 150 to 300 milliseconds.

When the RAM address directs an arm to another disk, the access time is greater. The maximum access time for a seek operation is 800 milliseconds. To illustrate the variation in access time, recall that a RAM file contains 50 disks. A seek operation that directs the access arm to move 25 disks takes about 600 milliseconds. When an access arm moves the maximum distance of 49 disks on a seek operation, the access time can be up to 800 milliseconds.

Because data are recorded on both sides of a disk, track 00 of disk face 00 is directly above track 00 of disk face 50. Therefore, if the access arm is located on track 00 of disk face 00, the minimum access time for a seek operation would be to track 00 on disk face 50. This access time is about 30 milliseconds. This access time overlaps 650 programming inasmuch as the 650 is interlocked only for two word times (0.192 ms), which is the time it takes to transfer the RAM address from the distributor to the 652 control unit.

The D-address of the seek instruction must always be a valid address, and the 6-digit RAM address must be in the distributor for the two word times during which the seek instruction is initiated.

Seek Checking. Before any arm is placed in motion by a Seek instruction, the instruction address is checked for correct arrival at the RAM file. After the arm has moved to its new location, the position of the arm is compared with the seek address. When this check is satisfied the seek operation is complete.

87 W-RAM (Write-RAM)

The function of the Write-RAM instruction is to cause the entire contents of immediate access storage to be recorded on the RAM track. The D-address of the Write-RAM instruction must always be 9000, and the 6-digit RAM address must be in the distributor for the two word times during which the Write-RAM instruction is initiated.

The Write-RAM instruction can be initiated only if a prior Seek-RAM instruction has positioned the selected access arm at the correct RAM track. If the seek operation has not been completed, the Write-RAM instruction automatically waits until the access arm is in position on the designated track.

The Write-RAM operation begins by checking the RAM address in the distributor against the position of the selected arm. An error is signaled if the address of the Write-RAM instruction and the previous Seek-RAM instruction does not agree. An incorrect comparison is indicated by a checking light on the 652. This check safeguards the information stored in the file against errors in programming.

As soon as the RAM address is checked and confirmed, the Write operation begins immediately. Any previous data on the track are erased as new data are written. A two-word-length gap is written, followed by writing the 60 words of data from immediate access storage. The two-word gap is written to designate the starting point of the data on the track for future Read-RAM operations. Each time data are written on a RAM track, the previous data are erased and a new two-word gap and data recording are made. This writing of data takes one revolution of the RAM disk.

As each digit is read out of immediate access storage, it is checked for validity. To insure that this information is correctly written and that in the future it will be read exactly as it now exists in immediate access storage, an automatic write-check cycle

is taken. The RAM track and immediate access storage are read simultaneously while a digit-by-digit comparison is made. This requires an additional revolution of the RAM disk. The Write-RAM operation is complete when this check is satisfied. If the Write-RAM check is not satisfied, a comparing light on the 652 will indicate this condition, and the entire Write-RAM operation is repeated. This operation continues until the track is successfully written and checked or until the operator intervenes.

One Write-RAM instruction takes a total of 125 milliseconds. This is composed of two disk revolutions of 50 milliseconds each and 25 milliseconds of setup time. Immediate access storage is interlocked for the complete 125 milliseconds, but the 650 is interlocked for only two word times. Therefore, the 650 can be programmed to execute other instructions that do not refer to immediate access storage while the Write-RAM instruction is being executed.

86 R-RAM (Read-RAM)

The function of the Read-RAM instruction is to read the entire track of 600 digits of data and place it in immediate access storage. The D-address of the Read-RAM instruction must always be 9000, and the six-digit RAM address must be in the distributor for the two word times during which the instruction is being initiated in the 650.

The Read-RAM instruction can be initiated only if a prior Seek-RAM instruction has positioned the selected access arm at the correct RAM track. If the seek operation has not been completed, the Read-RAM instruction automatically waits until the access arm is in position on the designated track.

The Read-RAM operation begins by checking the RAM address in the distributor against the position of the selected arm. An error is signaled if the address of the Read-RAM instruction and the previous Seek-RAM instruction do not agree. An incorrect comparison is indicated by a checking light on the 652. This check safeguards against errors in programming.

As soon as the RAM address is checked and confirmed, the read operation begins searching for the two-word gap on the track. This gap indicates the starting point of data previously written on the track. Because the disks are continuously rotating, the gap could be immediately available or could be a maxi-

INSTR NO.	LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
		ABBREV.	CODE	DATA	INSTRUCTION	
	0000	LD	69	0501	0001	Place RAM address in distributor Location 0501 contains the commodity code 00000 1530 0
	0001	S-RAM	85	9000	0002	Direct access arm to RAM location
	0002	R-RAM	86	9000	0003	Read record into immediate access storage

FIGURE 6. SEEK AND READ A RAM RECORD

imum of a complete disk revolution away from the read head. When the track gap is recognized, the entire track of 600 digits of data is read and transferred to immediate access storage in the time it takes for one revolution of the RAM disk.

As each digit is read out of the RAM track, it is checked for validity. When all the digits have satisfied this check the Read-RAM operation is complete. If this check is not satisfied, a comparing light on the 652 will indicate this condition and the entire Read-RAM operation is repeated. This operation continues until the track is successfully read and checked or until the operator intervenes.

The total time for a Read-RAM instruction varies from a minimum of 70 milliseconds to a maximum of 120 milliseconds. This maximum limit assumes that the transmitted data satisfactorily pass the validity check. An average time for a Read-RAM operation is 100 milliseconds. The Read-RAM timing is comprised of 25 milliseconds setup time; 50 milliseconds to read the complete track and a variable time from 0 to 50 milliseconds until the track gap passes under the read head. This variable time is dictated by the proximity of the track gap to the read head at the beginning of the read operation.

Immediate access storage is interlocked for the entire Read-RAM operation. The 650 is interlocked for

only two word times and is free to execute other instructions that do not refer to immediate access storage.

Programming Example

The programming for selecting a commodity record stored in RAM is shown in Figure 6. This example illustrates the programming for a seek, a read, and a write operation in the RAM file. It presumes that commodity records have previously been stored in the RAM file, and each commodity is identified by a four-digit commodity code. Each commodity record is stored in a RAM location, whose address is identical to the commodity code. The commodity referred to in this example is coded 1530.

Instruction 0003 begins a 650 routine, which will process the data from the RAM record. The processing routine may change, add, or delete information in the RAM record. If so, the revised record in immediate access storage has to be rewritten in RAM. The programming (Figure 7) writes the record on the same RAM track it originally came from.

Note that a seek instruction is not repeated, because it is assumed that the RAM access arm is still positioned on the same disk.

INSTR NO.	LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
		ABBREV.	CODE	DATA	INSTRUCTION	
	0100	LD	69	0501	0101	Load Distributor with RAM Address
	0101	WRAM	87	9000	0102	Write Record in RAM on Track 001530
	0102					

FIGURE 7. WRITE A RAM RECORD

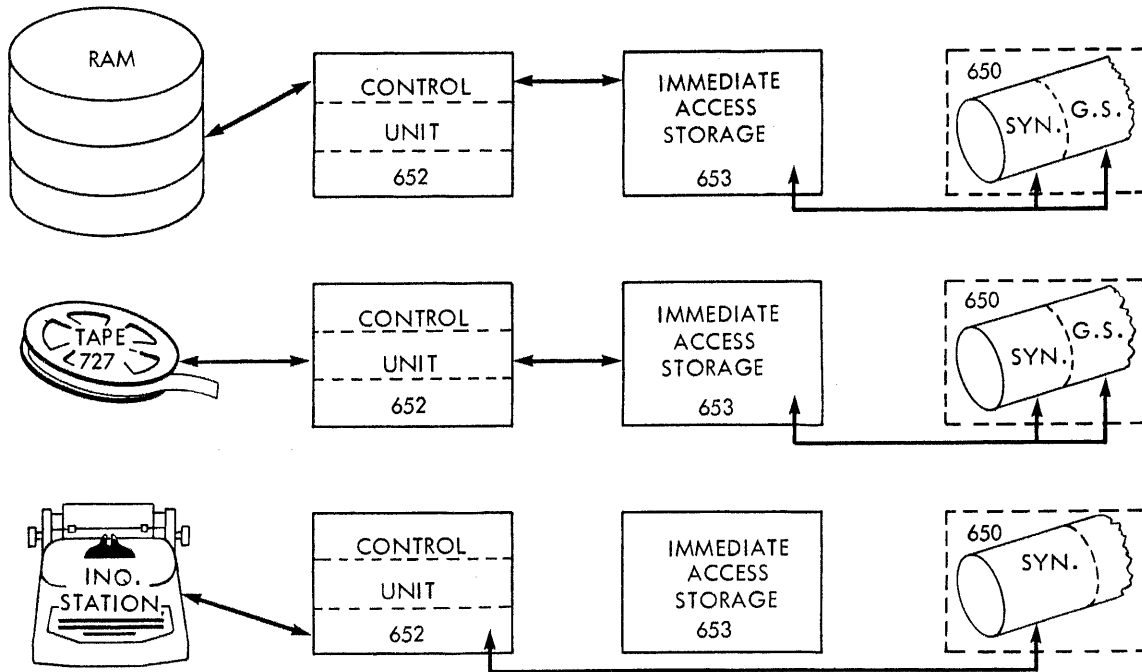


FIGURE 8. SCHEMATIC OF DATA FLOW

FLOW PATHS

FIGURE 8 shows the flow paths between the 650 and the additional features such as immediate access storage, tape, and RAM. The flow paths within the basic 650 for data and instructions are shown in Figure 9 but are described in detail in the 650 Manual of Operation. The additional-feature flow paths indicate that data read from magnetic tape and from RAM disks are always placed in immediate access storage. Data to be written on magnetic tape or RAM disks must come from immediate access storage. Immediate access storage is also capable of accepting information from the 650 drum, distributor, and accumulator (via the distributor). The output flow path

from immediate access storage can send data into the 650 to the drum, the distributor, program register, and the accumulator (via the distributor).

Validity check points are shown in this figure abbreviated as VC. These validity check points have been placed at strategic locations to make sure that all data are carefully checked. All data read from magnetic tape and RAM units are checked as they are read and before they are placed in immediate access storage. Information read out of immediate access storage is always validity checked as it is being transmitted. A Write-RAM operation goes through an additional cycle during which the written information is read from the RAM disk and compared digit for digit with the information in immediate access storage.

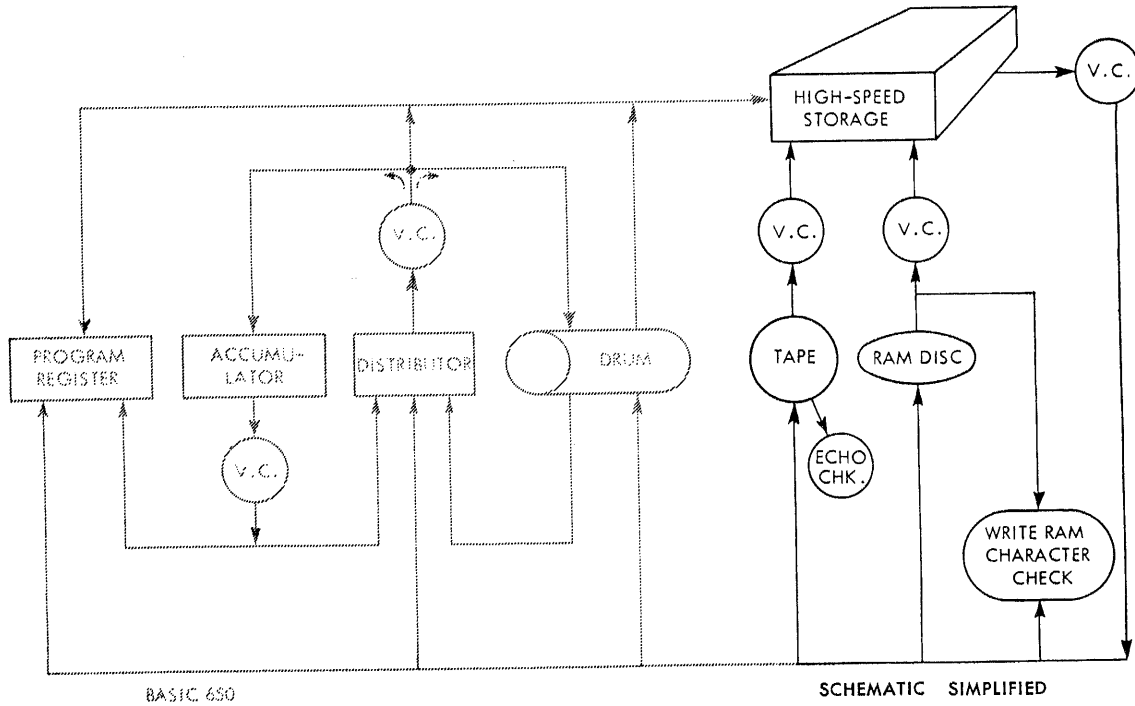


FIGURE 9. 650 SYSTEM DATA FLOW AND CHECKING

652 Control Unit

The RAM file is connected to the 650 through the 652 Control Unit and the 653 Storage Unit.

The 652 Control Unit combines the circuitry for control of Magnetic Tape Units, RAM Units and Inquiry Stations. It accepts operation codes and addresses from the 650 to direct the functions of the specified unit.

Figure 10 shows the arrangement of the lights and switches on the 652 Console. These are used to visually indicate the functions taking place in the various units controlled by the 652.

653 Storage Unit

The 653 Storage Unit must contain the 600 positions of immediate access storage for use with the magnetic tapes or RAM. Whenever a RAM record is consulted, the entire track of data is read to or written from immediate access storage. Immediate access storage serves as a link between the RAM and the 650. The capacity of immediate access storage is identical to that of a RAM track because each holds 600 digits of data as 60 words.

Figure 11 illustrates the light panel on the 653. It indicates when the units affected by the 653 are functioning in the system.

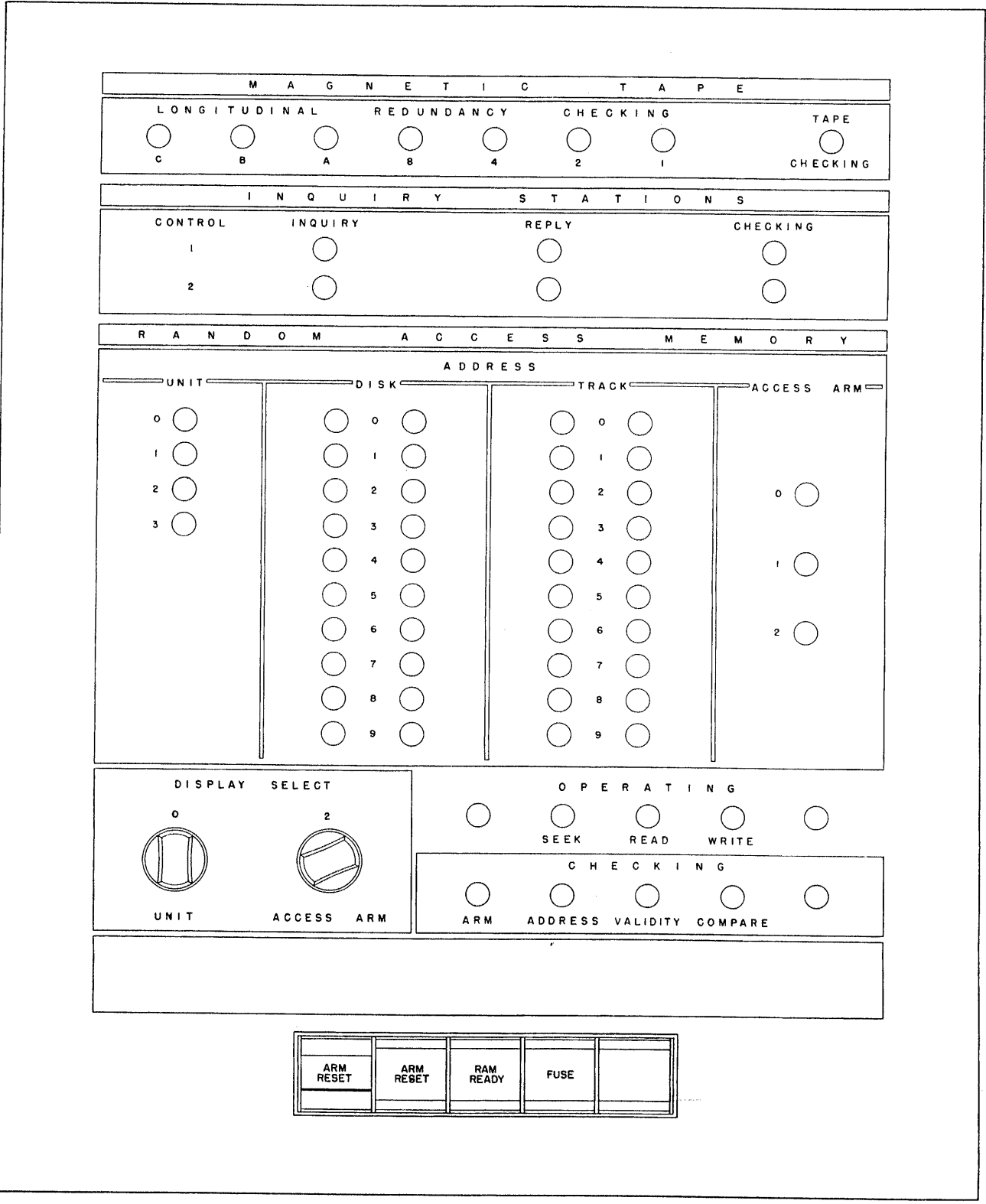


FIGURE 10. 652 CONSOLE

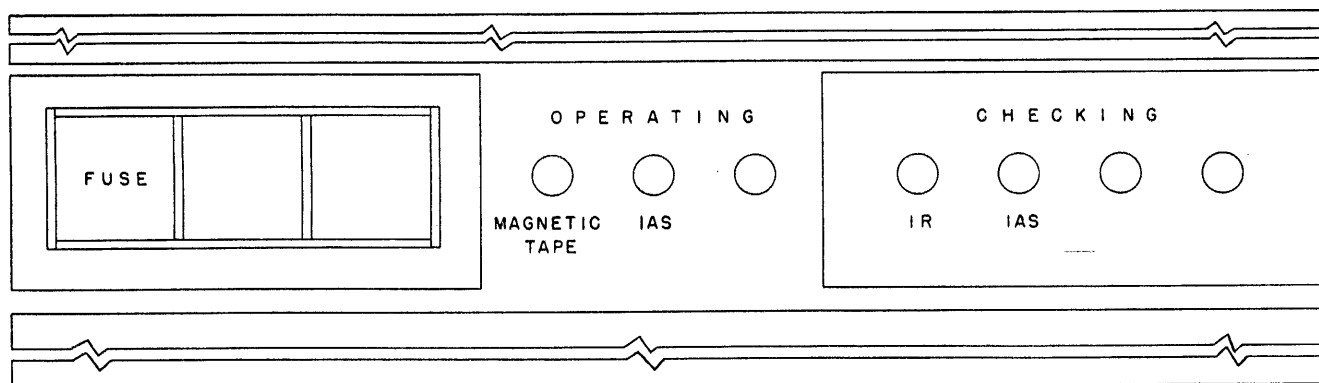


FIGURE 11. 653 CONSOLE

TIMING CONSIDERATIONS

Seek Access Time

In a previous section describing operation codes, the upper and lower limits of seek access time were stated. This section describes the seek time in more detail. Several charts show pictorially the range of access time for address changes involving a move of an access arm. These charts depict the action of any one access arm in a RAM file. They do not show the net effect on timing that results from three access-arm operations. Obviously when the 650 program integrates the operation of the three access arms, the processing unit of the system will have record data from RAM constantly available.

Figure 12 shows the range of access time for an address change made to another track on the same disk. Note that a change of 50 tracks takes about 200 to 250 milliseconds. A change of 25 tracks takes about 175 to 225 milliseconds.

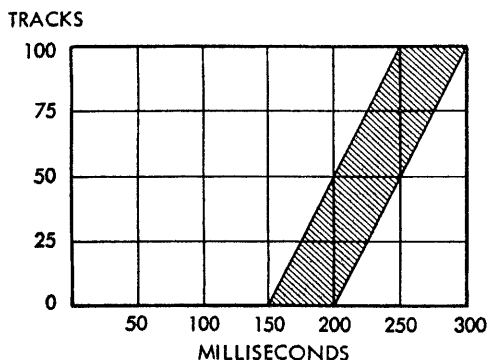


FIGURE 12. SEEK TIME FOR ACCESS ARM MOVING FROM TRACK TO TRACK ON SAME DISK

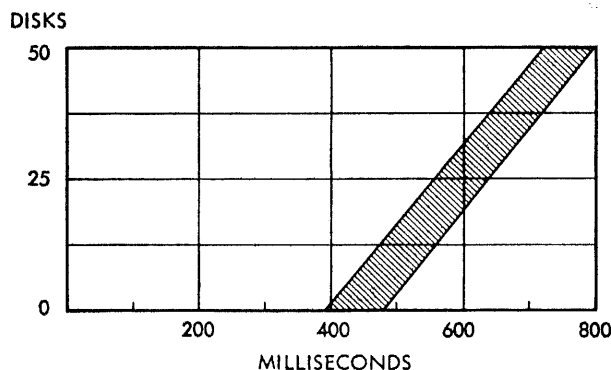


FIGURE 13. SEEK TIME FOR ACCESS ARM MOVING FROM TRACK ON ONE DISK TO TRACK ON ANOTHER DISK

Figure 13 depicts an address change involving a disk change; that is, an address change where the access arm leaves a specific track on one disk, moves up or down to another disk, and moves in to a specific track on the new disk. Note that the minimum move is about 400 milliseconds. A move of 25 disks takes approximately 600 milliseconds. A change of 37 disks takes approximately 650 milliseconds.

Estimating the Overlap of Access Time with 650 Processing

One of the objectives in programming the 650 RAM is to completely overlap seek time with processing of data. Where the processing time is longer than the net seek time (3-access arm operation) the RAM seek time is immaterial. Where the processing cycle is short, it is advantageous to know the average seek time. This, however, is dependent on several factors, such as how the data in the RAM file are organized;

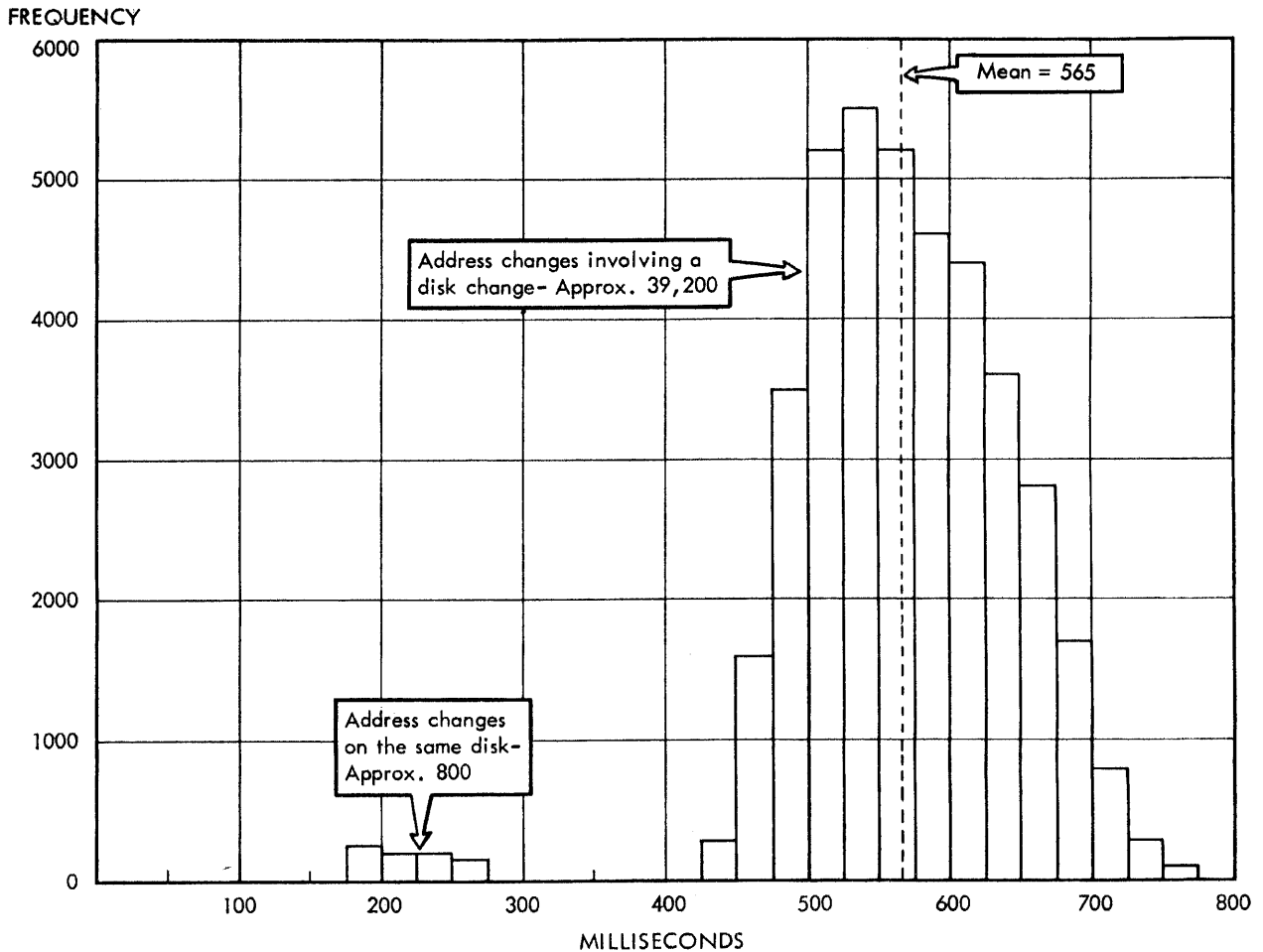


FIGURE 14. FREQUENCY DISTRIBUTION OF ACCESS TIME FOR 40,000 RAM CHANGES

the method used to convert the record identification code to a RAM address, and the possibly unknown pattern of address changes requested of the RAM file. If the access arm is continually being asked to move to extreme ends of the disk file, the average access time will be higher than if the access arm moves short distances for successive address changes. Therefore, an average access time to data in the file will vary from one application to another.

Statistical studies have been made of access time for address changes resulting from random number addresses. Figure 14 is a bar graph showing the frequency distribution for 40,000 random address changes. Note that 800 of the 40,000 changes were to an address on the same disk. Address changes of one to 49 disks

totaled 39,200. This averaged out to about 565 milliseconds for a random change in address. Again, keep in mind that the bar graph is plotted for one access arm operation only and is based on random seeks to the entire file of 100 disk faces.

Figure 15 shows a probability curve for access time involving a disk change. This is plotted against milliseconds. To interpret the curve assume that the programmer allows 565 milliseconds for access time to a record in the file. The probability curve shows that in approximately fifty per cent of the cases this will be sufficient time for one access arm to move anywhere in the file to find the record. When 680 milliseconds is allowed, the probability curve indicates approximately 90 per cent.

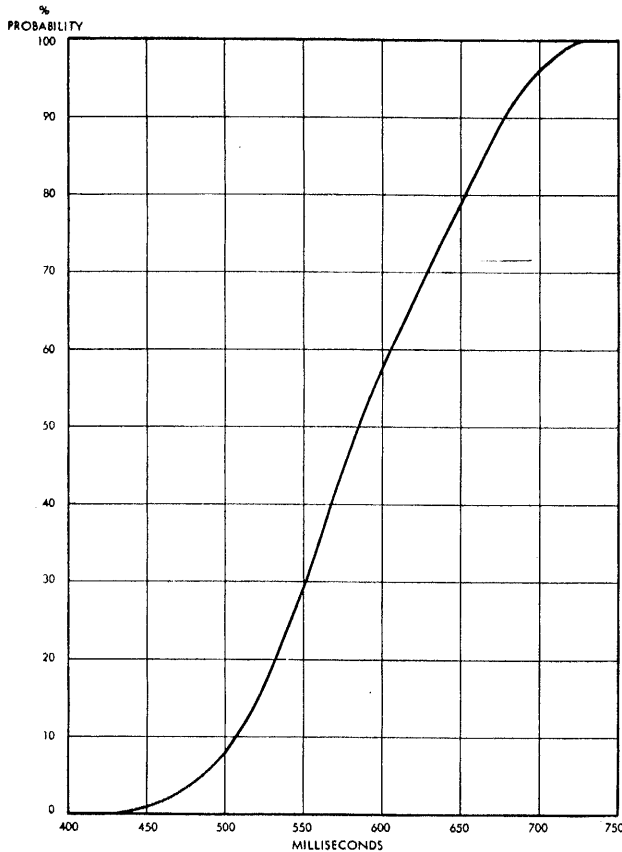


FIGURE 15. ACCESS TIME PROBABILITY INVOLVING A CHANGE IN DISKS

OVER-ALL JOB TIMING CONSIDERATIONS

THE ESTIMATING of the total time required for a data-processing system to handle an application is based on several factors: the input and output document volume, the internal processing time, and the frequency (daily or weekly, etc.) of output. All these factors make up the total time. These factors are directly affected by the speed and timing considerations of the individual components of the data-processing system. For example, the maximum output printing speed of the 407 is 150 lines per minute, and, where the 407 is the output device, this is the maximum output production. Each system contains a variety of components whose operation needs consideration; however, the three basic elements are input volume, processing time, and output volume. The most efficient operation will be determined by overlapping the operational time of the various components as much as possible. No estimated input or out-

put speed should be assumed until plotted as described herein.

This section of the manual describes a way to layout the operational time of the 650 RAM components. The application illustrated in Figure 2 is used as the example. The first chart shows the layout for processing an order card when only one RAM access arm is utilized. The second chart shows how the punch feed of the 533 Card Read Punch can be kept in continuous operation and two RAM access arms are used. The third chart shows the 407 printer operating at maximum speed of 150 lines per minute and three RAM access arms are utilized. The timing considerations for including manual inquiries are described in another section which follows the description of these units.

The timing charts show a separate horizontal line for each functional component. The length of each horizontal line segment is expressed in milliseconds and labeled to indicate the function performed in that segment of time. A millisecond (ms) is 1/1000 of a second and provides a convenient time scale, because all functions are very easily expressed in this medium. For example: $\frac{1}{2}$ second = 500 milliseconds; $\frac{1}{10}$ second is 100 milliseconds, etc. The 533 Card Read Punch operates at 100 punch cycles per minute; therefore, each punch cycle takes 600 milliseconds for the punch to complete the cycle. The 650 program is delayed for only a portion of this time and can be preparing data for the next punch cycle. In applications where this 650 processing is completed within the time required for the 533 cycle, the punch can be operated at its maximum speed.

Thus, a method of planning applications efficiently is to lay out the estimated timing considerations with as much overlapping of functions as possible within the programming structure. The layout may show a good balance between input, processing, and output times, or it may indicate that the operation is not functioning at its maximum speed, in which case further planning or a new approach may improve the efficiency.

The timing chart for one access arm operation (Figure 16) shows a separate line for the RAM access arm, the 650 programming, the 533 punch, and the 533 read operation. Each of the RAM and 533 functions is highlighted in the block diagram to show the

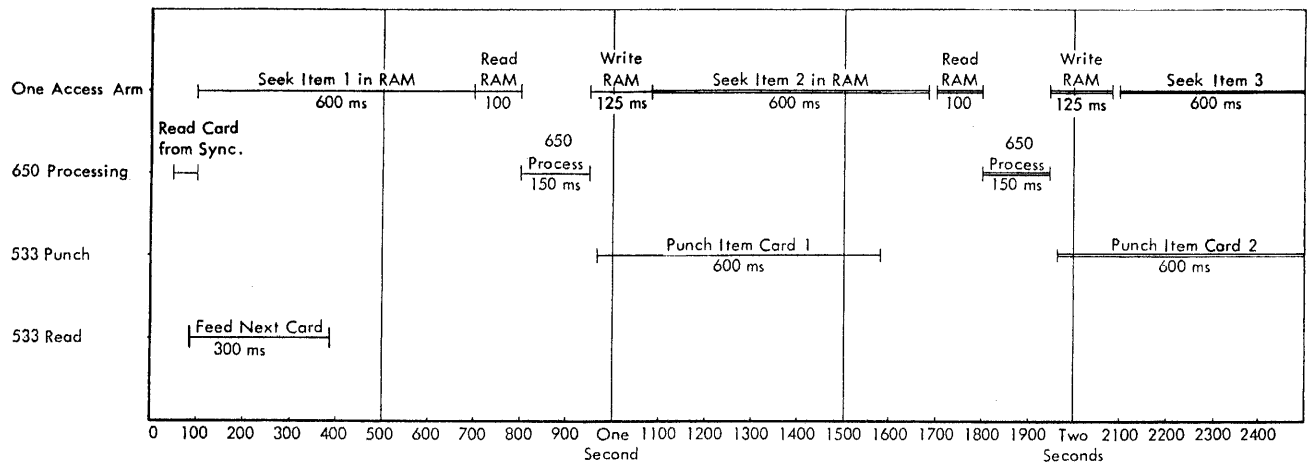


FIGURE 16. TIME CHART FOR ORDER CARD PROCESSING USING ONE ACCESS ARM

relation of the timing chart to the over-all programming. The timing charts assume a RAM-Seek time of 600 ms and a RAM-Read time of 100 ms. The chart begins by reading an order card and programming a RAM-Seek for the first-ordered item. The timing chart shows that the next card is being read by the 533 while the Seek is executed. At the completion of the Seek, the item record is read into immediate access storage.

The 650 processing time includes the programming required to update the inventory record for this item, calculate the price and cost, perform the distributions by class, etc. The Write-RAM instruction is then given to record the updated item record, an item card is punched, and the Seek for the second item on the order card is begun. Note the overlap of RAM-Write and Seek time with card punching time. A complete processing cycle for one item totals 975 milliseconds. This is composed of a 600 Seek, a 100 Read, 650 processing time of 150, and RAM Write of 125. The punching time is completely overlapped by RAM operations.

A better approach would be to use two access arms and keep the 533 punching output cards continuously. Figure 17 shows that access arm 1 seeks, reads, and writes item one, three, five, etc., of the order card. Access arm 2 seeks, reads, and writes item two, four, six, etc., of the order card. The 650 processing and immediate access storage time is short enough to be overlapped by a card punch cycle; therefore, the

punch runs continuously at its maximum rate for all the items on the order card.

The complete series of operations for each given item are identified by the same shading. This shading permits following one item from start to completion. Note that the pattern for a given item follows a diagonal line from the top to the bottom of the chart. By a vertical inspection at any point on the chart, all the units in operation at that time can be seen. By following each horizontal line the operating time of each component can be determined. A study of the timing chart from these various angles can be utilized to evaluate the machine approach that is planned for a given application.

As an alternate method the application output may be directly printed on a 407 attached to the 650; the timing chart in Figure 18 applies. The maximum operating rate of 150 lpm of the 407 is faster than the punching rate of 533. Therefore, the 650 processing and RAM operations must produce an output every 400 milliseconds in order to keep the 407 running at maximum speed (150 lpm). This can be accomplished in this application study by using three RAM access arms. Applications that require large amounts of 650 processing time obviously will reduce the printed output below the maximum rate.

As in the previous chart all the operations for a given item are identified by a specific shading. The chart assumes six items are ordered on the one order card. The chart carries the processing through the reading of the next card. Note that the 650 process-

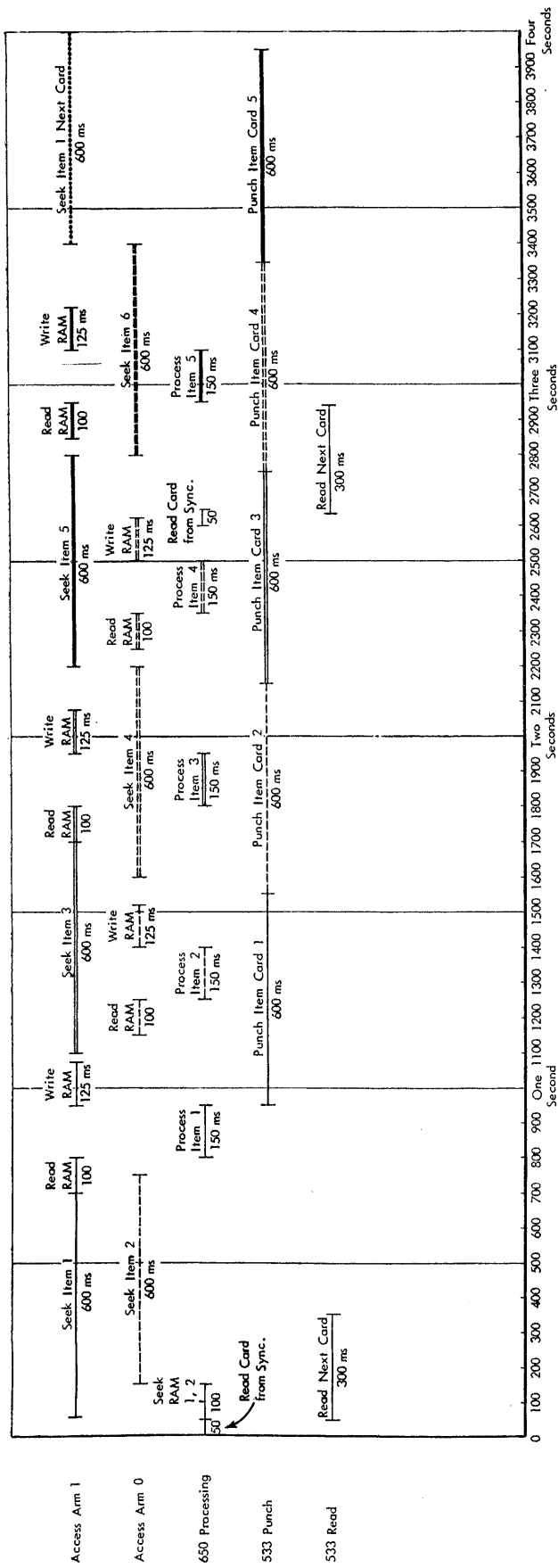


FIGURE 17. TIMING CHART FOR TWO ACCESS ARM OPERATION

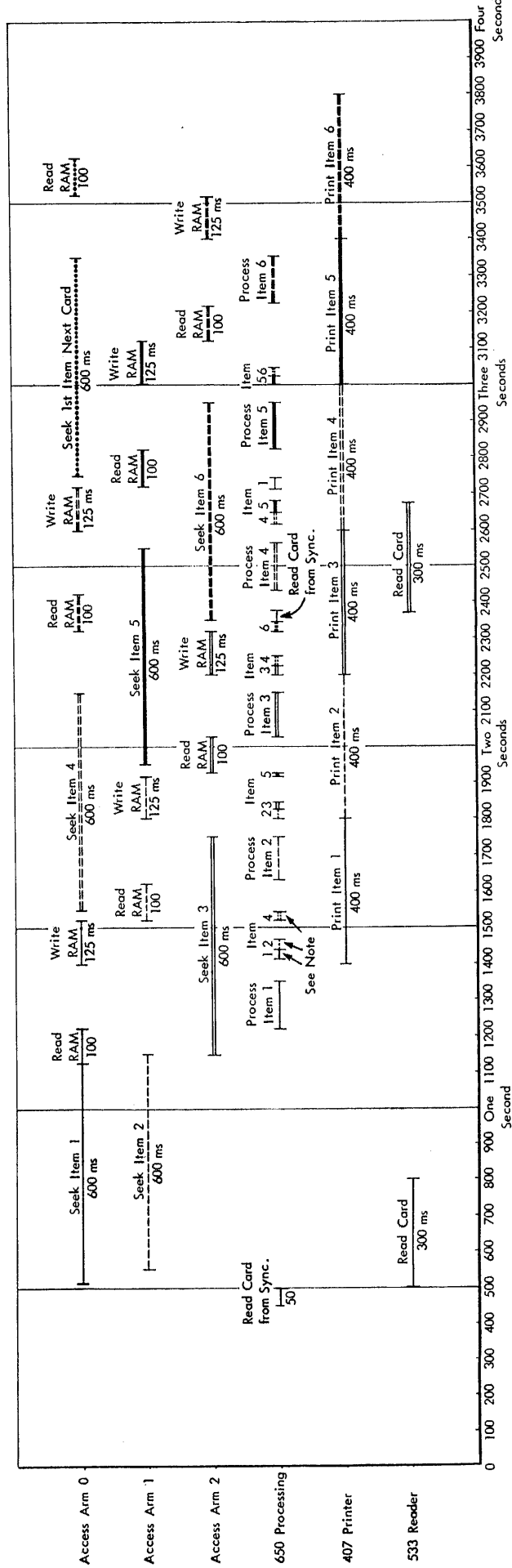


FIGURE 18. TIMING CHART FOR PRINTING 150 LPM AND UTILIZING THREE ACCESS ARMS

NOTE: Make sales distribution on drum for item 1. Develop next seek address item 4. Prepare to accept next RAM record item 2. During each write operation this process is repeated.

ing time is shown as several line segments. The basic processing for an item occurs between the Read-RAM and Write-RAM operations. Other 650 processing can occur during the Read and Write-RAM operations.

Note the two-part line segment of 650 processing, which occurs during a Write-RAM operation. The first part is used to make the sales distribution on the drum for the previous item processed. The second part is preparation to accept the next record read by RAM. During the Read-RAM operation the Seek address for the next item is developed. By arranging the 650 processing time properly, the maximum output for an application can be planned and block diagrams of the programming prepared accordingly.

Unloading the RAM File

The integration of magnetic-tape units with RAM files offers considerable flexibility for application planning. One advantage is that periodically it may be desirable to unload the RAM file so that it can be used for another application. It is also desirable to

unload the RAM file so that an independent record of the status of data in RAM at a given time is available. This independent record can be utilized to create periodic reports of inventory, such as a monthly stock-status report and others. The independent record offers a valuable system check on the data in the RAM file.

The RAM file can be unloaded to punched cards or to magnetic tape. Unloading to magnetic tape is more desirable because it is much faster. A complete RAM file can be unloaded to magnetic tape in about 30 to 40 minutes. Unloading to cards would take approximately 7 hours if two card punches are connected. Figure 19 shows the programming to unload the RAM file to magnetic tape. Note that it utilizes two access arms which keep immediate access storage operating continuously except when an arm moves to the next disk. A program written for three access arms would completely overlap the disk change seek time. As a general rule when using multiple arms, it is desirable to send an arm seeking a new address as soon as it has completed its functions.

INSTR NO	LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
		ABBRV.	CODE	DATA	INSTRUCTION	
						Start Routine
0051		RAI	65	0001	0052	Lower Accumulator used for Arm 1
0052		AU	10	0002	0053	Upper Accumulator used for Arm 0
0053		ID	69	8002	0054	
0054		SRAM	85	9000	0055	Even tracks developed in Lower Accumulator
0055		ID	69	8003	0056	
0056		SRAM	85	9000	0057	Odd tracks developed in Upper Accum.
						Arm 1 used for even tracks only
						Main Routine
						Arm 0 used for odd tracks only
0057		ID	69	8002	0058	
0058		RRAM	86	9000	0059	Overflow indicates RAM is unloaded
0059		AI	15	0003	0060	
0060		ID	69	8002	0061	
0061		SRAM	85	9000	0062	
0062		WT	06	8010	0063	
0063		BNTS	25	0065	0064	
0064		BNEF	54	Error	End File	
0065		ID	69	8003	0066	
0066		RRAM	86	9000	0067	
0067		AU	10	0003	0068	
0068		ID	69	8003	0069	
0069		SRAM	85	9000	0070	
0070		WT	06	8010	0071	
0071		BROV	47	End RAM	0072	
0072		BNTS	25	0057	0073	
0073		BNEF	54	Error	End File	
						Constants
0001			00	0000	0001	Selects Arm 1
0002			00	0100	0010	Selects Arm 0
0003			00	0200	0020	Up dates tracks for next Seek.
						High order 2 keeps track of
						Seeks, and Branch Overflow
						indicates End of Unload.

FIGURE 19. UNLOAD COMPLETE RAM FILE ON TAPE

DATA STORAGE AND ADDRESS MODIFICATION CONSIDERATION

MOST applications are such that it is not practical to consider the part number, policy number, or account code as the RAM address. This may not be practical for several reasons: the part number may be larger than six digits, or numbers are loosely assigned so that there are large gaps of unused numbers in the code. Where large amounts of data are to be stored, it will be desirable to "pack," or condense, the file. This is done by storing several records on a track.

The storage arrangement can be in any fashion as long as a method is devised which will find the record and make it available for 650 processing. This processing flexibility provides a variety of methods and is limited only by the imagination of the programmer. Four general approaches are outlined here that will serve as a base for many applications. These approaches are described through illustrative examples.

1. Table lookup method.
2. Arithmetic operations upon the record identifying code.
3. Use portions of the record identifying code itself as the RAM address.
4. Process the record identifying code through a randomizing formula and extract digits for the RAM address. The third approach is the most desirable, because it should locate a specific record in a minimum of time. However, many applications will utilize several of these general approaches to convert the record identifying code to a RAM address.

TABLE LOOKUP

THE table lookup feature of the 650 presents a most powerful tool for automatically converting part numbers, account numbers, policy numbers, etc., to a RAM address. This section illustrates the basic pattern utilizing the table lookup feature to find RAM addresses. The section on packing records further describes the flexibility provided for locating records in a large file. As an example consider an application that consists of 500 items, each having a record length of 400 digits. Each item is identified by a five-position alphabetic code.

The record for each item can be easily stored on a single RAM track; however, the alphabetic identification code cannot be the RAM address, because the five-position alphabetic code is represented by a ten-digit number in the 650.

For example: AGKLS = 6167727382

Assume the record for item code AGKLS is stored in RAM address 006931 (file 0, disk face 69, track 31). The function of the table lookup feature is to provide this RAM address whenever item code AGKLS calls for it. This is a conventional 650 table lookup application and consists of placing the item code table and the RAM address table on the drum. A given RAM address is placed a fixed number of locations away from its corresponding alphabetic item code.

When a RAM record is desired, the first operation is to do a table-lookup on the alphabetic item table. Then modify the item table address resulting from the TLU by the increment between the two tables stored on the drum. This modification gives the drum location that holds the desired RAM address. The next step is to load the distributor with the RAM address, Seek and Read RAM.

ARITHMETIC OPERATIONS TO CONVERT RECORD CODES TO RAM ADDRESSES

RECORD identifying codes can be arithmetically manipulated to create a RAM address by adding, subtracting, multiplying or dividing techniques. As an example, consider an application of 20,000 account numbers ranging from 00000 to 19,999. When the record length is 150 digits for each record, the 20,000 records can be stored on 5000 tracks (4 records to a track). This is accomplished by dividing the actual account number by 4. The division will create a 4-digit number (in the range of 0000 to 4999), which then becomes the RAM address. This method automatically assigns a space in RAM memory for each of the 20,000 account numbers.

When an account number record is to be updated by an input card, the programming procedure takes the account number from the card source, divides it by 4, and inserts the RAM file number and access arm digits to create the 6-digit RAM address. This RAM address is used in the seek operation to bring the correct track of data into immediate access storage. However, because four records are grouped on a track, the record for the desired account number is one of four records that are brought into immediate access storage. Several ways of selecting the one record of four can be utilized. A comparison of the four account numbers to the desired account number can be programmed. Another way stems from the divide operation that created the RAM address. The remainder from the divide operation can be used to specify if the desired record is the 1st, 2nd, 3rd, or 4th record on the track.

Record Identifying Code as the RAM Address

As an example, consider an insurance rating application. This consists of storing the data from an agent's rating book in the RAM file. The object of the application is to determine the correct policy premium, agent's commission, and policy form number for each new policy or renewal. The parameters of the application are:

1. 110 insurance plans in effect.
2. Ages 1 to 80 with different rates for each age, plan, and sex.
3. Policies are written on different forms depending on the plan and state.

To summarize: the number of unit records needed can be determined by multiplying 110 (plans) x 80 (ages) x 2 (sex) = 17,600 unit records. Although only 10 different policy forms exist, the state and plan must be considered to select the actual form number of the policy. Recall that one RAM file contains 10,000 addressable tracks. At first glance it appears that two files will be needed to house the 17,600 records. Such is not the case, however, because each unit record requires only 5 words for the plan, age, and sex data.

The 17,600 records can be stored on 55 disk faces or in approximately one-half of one RAM file.

$$\begin{aligned} \text{Disk address} &= \frac{\text{Plan No.}}{2} \\ \text{Track address} &= \text{Age} \end{aligned}$$

Thus, by dividing the plan number by 2, we utilize 55 disks. Each of the 80 ages is assigned one track on a disk. The following example shows the pattern for the RAM addresses. Figure 20 shows the layout for several tracks.

<u>DISK</u>	<u>PLAN</u>	<u>TRACK</u>	<u>AGE</u>
	02		
01	03	01	01
	04		
02	05	02	02
		80	80
	108		
54	109		
	110		
55	001		

The record layout for a typical track is shown in Figure 21. Note that the words 00 to 04 on the track are for plan 002, age 01, sex code 1. Words 05 to 09 are for the same plan (002) and age (01) but are for sex code 02. Words 30 to 34 are for plan 003, age 01, and sex code 01; and words 35-39 are for plan 003, age 01, and sex code 02.

Note that words 10-29 of plan 2 and words 40-59 of plan 3 store the four-digit policy form number that applies to each of the 48 states covered by the respective plan.

PACKING RECORDS

EFFICIENT use of RAM storage is gained when the file is tightly packed. Because most code structures provide for expansion, the assignment of a RAM space to each possible code in the structure leaves many empty RAM tracks. These empty (though assigned) tracks reduce the number of records that can be stored. A packed file leaves virtually no empty tracks (or space within a track) for inactive codes, and compresses the records for active codes.

IBM.

RANDOM ACCESS MEMORY LAYOUT FORM
FOR IBM 650 DATA PROCESSING SYSTEM

Subdivide into fields and label

Unit	Disk	Track	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59																																																																																																																		
1	01	01 AGE	PLAN #2 MALE																													PLAN #2 FEMALE																													POLICY FORM # FOR PLAN 2 IN 48 STATES																													PLAN #3 MALE																													PLAN #3 FEMALE																													POLICY FORM # FOR PLAN 3 IN 48 STATES																												
		02	PLAN #2 MALE																													PLAN #2 FEMALE																													PLAN #3 MALE																													PLAN #3 FEMALE																																																																																						
		03 AGE	PLAN #2 MALE																													PLAN #2 FEMALE																													PLAN #3 MALE																													PLAN #3 FEMALE																																																																																						
		04 AGE	PLAN #2 MALE																													PLAN #2 FEMALE																													PLAN #3 MALE																													PLAN #3 FEMALE																																																																																						
1	01	80 AGE	PLAN #2 MALE																													PLAN #2 FEMALE																													PLAN #3 MALE																													PLAN #3 FEMALE																																																																																						
1	02	01 AGE	PLAN #4 MALE																													PLAN #4 FEMALE																													POLICY FORM # FOR PLAN 4 IN 48 STATES																													PLAN #5 MALE																													PLAN #5 FEMALE																													POLICY FORM # FOR PLAN 5 IN 48 STATES																												
1	55	01 AGE	PLAN #55 MALE																													PLAN #55 FEMALE																													POLICY FORM # FOR PLAN 55 IN 48 STATES																													PLAN #01 MALE																													PLAN #01 FEMALE																													POLICY FORM # FOR PLAN 01 IN 48 STATES																												

FIGURE 20. GENERAL LAYOUT OF FILE

IBM.
IMMEDIATE ACCESS STORAGE LAYOUT FORM
FOR IBM 650 DATA PROCESSING SYSTEM

Problem Name		PLAN #2 MALE		PLAN #2 FEMALE	
01020111	ADDRESS	ADDRESS	ADDRESS	ADDRESS	ADDRESS
XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX
PLAN PAGE 1	DATE FIRST CASH VALUE AVAILABLE	POLICY CASH VALUE PER 1000	COMMISSION RATE PAYABLE	RATE PER 1000	DATE FIRST CASH VALUE AVAILABLE
01020111	0102	0102	0102	0102	0102
XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX
STATE 01	02	03	04	05	06
01	02	03	04	05	06
07	08	09	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	40	41	42
43	44	45	46	47	48
PLAN PAGE 1	DATE FIRST CASH VALUE AVAILABLE	POLICY CASH VALUE PER 1000	COMMISSION RATE PAYABLE	RATE PER 1000	DATE FIRST CASH VALUE AVAILABLE
0103	0103	0103	0103	0103	0103
XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX
STATE 01	02	03	04	05	06
07	08	09	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	40	41	42
43	44	45	46	47	48
PLAN PAGE 1	DATE FIRST CASH VALUE AVAILABLE	POLICY CASH VALUE PER 1000	COMMISSION RATE PAYABLE	RATE PER 1000	DATE FIRST CASH VALUE AVAILABLE
0103	0103	0103	0103	0103	0103
XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX	XXXXXXXXXX
STATE 01	02	03	04	05	06
07	08	09	10	11	12
13	14	15	16	17	18
19	20	21	22	23	24
25	26	27	28	29	30
31	32	33	34	35	36
37	38	39	40	41	42
43	44	45	46	47	48

DISK
01
TRACK
01

POLICY
FORM #
48
STATES

POLICY
FORM #
48
STATES

FIGURE 21. TYPICAL TRACK LAYOUT

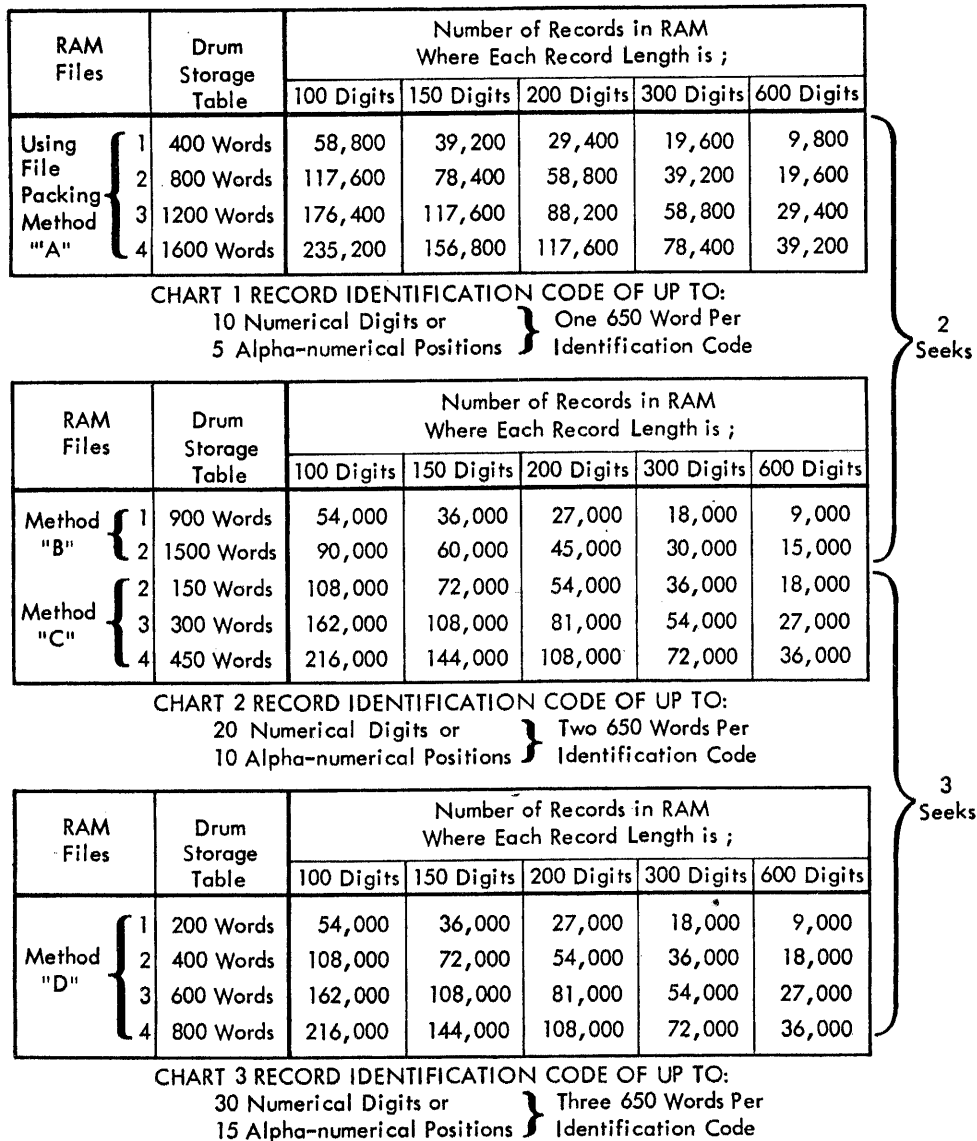


FIGURE 22. CHART FOR TABLE LOOKUP METHOD OF PACKING A FILE

The number of records that can be stored in a packed file varies with the length of the record. For example, a RAM track can contain six records, each 100 digits long, or three records each 200 digits long, etc. The identifying code of a record can be either alphabetic, numerical or alphanumerical.

The packed-file technique described here applies to any number of RAM files and permits storage and location of up to 235,200 records, where each is 100 digits long and four RAM files are used. Three charts are shown in Figure 22. The charts show the number of records that can be stored based on the three variables:

1. Number of RAM files used.
2. Length of record.
3. Length of record identification code.

Note that the charts indicate four file-packing methods: a, b, c, d. A detailed explanation of these methods follows the description of the charts.

Chart 1 applies to those records that are identified by a code of up to 10 digits; for example, an eight-digit numerical part number or a five-position alphanumerical item code. Consider an application where 75,000 parts are carried in inventory, and a part record requires 190 digits of memory. Chart 1 shows that this entire inventory record can be contained in

three RAM files. An insurance file of 75,000 policies, each requiring 150 digits of memory requires only two RAM files. Reference to this file is obtained by consulting a table on the 650 drum. The table provides the address of an index track on a specific RAM disk face. The RAM index track is consulted, and this provides the address of the RAM track that actually contains the desired record.

Chart 2 is based upon a record identifying code of over 10 digits and up to 20 digits long. This identifying code includes alphanumerical codes over five-positions and up to ten-positions long. Reference to the chart shows that 50,000 records, each with 100 digits, can be stored in one RAM file; but 50,000 records, each 200 digits long, require two RAM files; 50,000 records each 300 digits long, require three RAM files.

The charts also indicate the size of the drum table that is utilized to begin the search for a record in RAM. Chart 2 differentiates between two packing methods, B and C. Method B is acceptable when one RAM file is used but creates a large drum table when two RAM files are used. Method C greatly reduces the size of the drum table but increases the access time to a record by approximately 200 milliseconds. Estimated access time to any RAM record using the various packing methods is explained in the description of the method.

Chart 3 is based upon a record identification code of over 10 and up to 15 alphanumerical positions or up to 30 numerical digits. The capacity of this packing technique can be appreciated by noting that up to 144,000 records, each 150 digits long, can be stored in 4 RAM files where the record identifying code can be a 15-position alphanumerical code. The drum table which provides the initial seek address to the file uses 800 words of the 2,000 word drum capacity.

A study of these charts provides not only an appreciation for RAM capacity but also a quick reference in estimating the applications that can be stored. Note that the charts state the number of records: 100,000

records can be comprised of 40,000 inventory records, 30,000 accounts receivables, and 30,000 factory accounts, etc. The in-line processing concept incorporates many applications into one operation, and thus the record capacity indicated on the charts applies to records in general and not those of a specific application.

File Packing Methods A, B, C, D

The preceding charts indicated four methods of packing the file. Actually, the four methods are identical in concept and differ slightly in arrangement. Method A provides for the largest number of records but also creates the largest table on the 650 drum.

The following description explains the concept of packing records by method A. This description is based on an application using one RAM file. Applications using multiple RAM files are merely an extension of this description. Methods B, C, D are subsequently discussed. Up to 39,200 records, each 150 digits long, can be packed into one RAM file. Access to any one record requires a drum table lookup and two RAM seeks when the following method is used.

Because the record length chosen is 150, four records can be stored on a track (Figure 23). Each disk face could conceivably contain 400 records, and a RAM file could contain 40,000 records. However, some means of determining the RAM address is needed, because the item code and RAM address will not be identical. Assume that the item code is a 10-position numerical code.

This method stores the records in numerical item-code sequence and groups tracks, where 50 tracks constitute a group. Forty-nine tracks contain data (four records per track). The other track is an index track for the 49 data tracks. The index track contains the highest item code from each of the other 49 tracks in the group. Each RAM disk face holds two groups, and a RAM file can hold 200 groups.

Ram Track	Record 1	Record 2	Record 3	Record 4
	Word 00-14	Word 15-29	Word 30-44	Word 45-59

FIGURE 23. RECORD GROUPING ON ONE DATA TRACK (METHOD A)

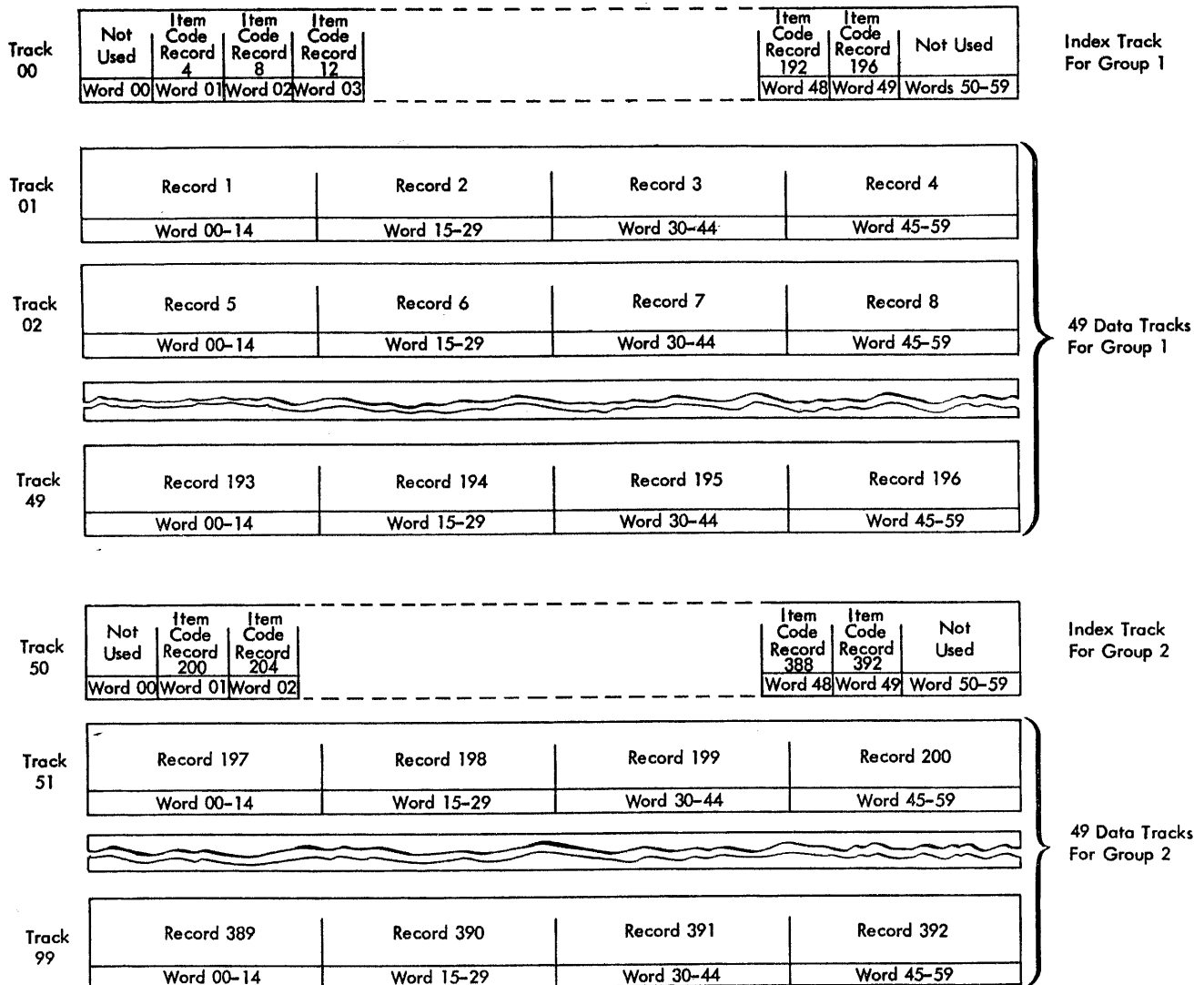


FIGURE 24. DISK FACE LAYOUT (METHOD A)

Figure 24 shows the record layout for a disk face. The highest item code from each of the 200 index tracks is then placed in a table. This table consists of 200 item codes and the 200 RAM index track addresses can easily be stored on the drum. Thus, the grouping technique results in several levels of item identification tables rather than one large unwieldy table.

In summary, the file is packed as follows:

- a. 4 records per track
- b. 49 data tracks to a group (4 x 49) = 196 records

c. An index track for each group that holds the code for the highest record from each of the 49 tracks

d. Two groups per disk face (196 x 2 = 392 records per disk face)

e. 100 disk faces per file

(100 x 2 groups = 200 groups)

(100 x 392 = 39,200 records per file)

To find a given record, three objectives must be met as follows:

1. Find the correct group (one of 200 groups)
2. Find the correct track (one of 49 tracks)
3. Find the record (one of 4 on a track)

The 200 groups can be arranged as a two-part drum table. One part of the table consists of the 200 item codes; the other part contains the 200 RAM address of the group index tracks and is located a fixed distance away (assume 200 words away) from its corresponding item code in the drum table. Thus, 200 drum locations are needed for the item code table and 200 locations for the RAM addresses of the index track for each of the 200 groups.

Figure 25 shows a block diagram for locating a given record in the packed file when file packing method A is used.

The complete process of locating one record out of 39,200 takes about one second. The majority of this time is used in the two RAM seeks.

Figure 26 provides the programming to recognize the item identification code in the card, locate its record in the RAM file and make it available for processing by the 650. The programming is divided into three sections. The first section includes the drum table lookup and a seek for the index track. The second section reads the index track into immediate access storage, does a table lookup on the index track contents, and seeks the data track. The last section reads the data track, processes the record and writes the updated track back in RAM. Note that this grouping into three sections provides three independent routines. However, these three routines apply to the operation of one access arm. The programming for multiple access arm operation will interweave the routines for the various arms to provide overlapping RAM seek, read or write, and 650 processing operations.

Method A is based on a record identification code of one word. Code structures that require more than one word, such as an 8-position alphanumerical part number, can use method B. Method B provides two words for each record identifying code, and each index track refers to 30 data tracks. The search routine to find a record is similar to the method A routine, and is not described; however, method B results in a large drum table when multiple RAM files are needed.

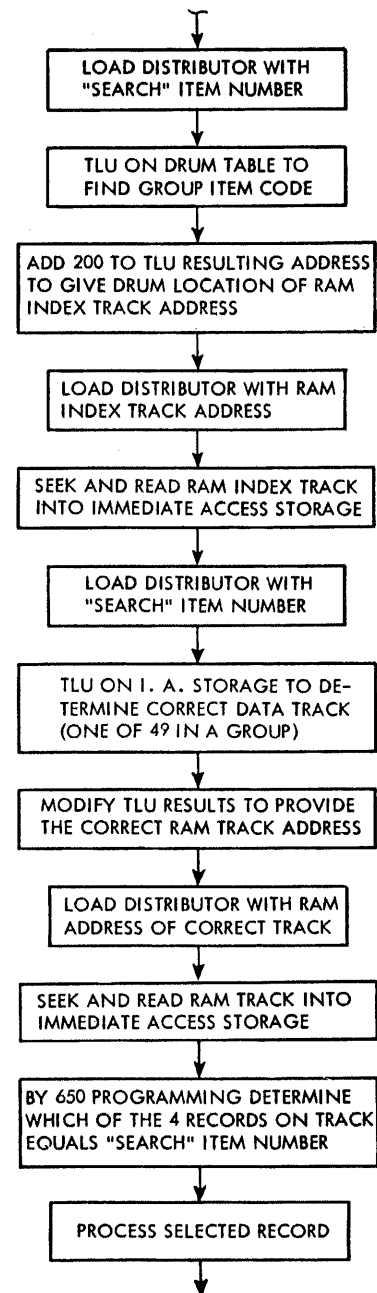


FIGURE 25. BLOCK DIAGRAM FOR LOCATING A RECORD METHOD A

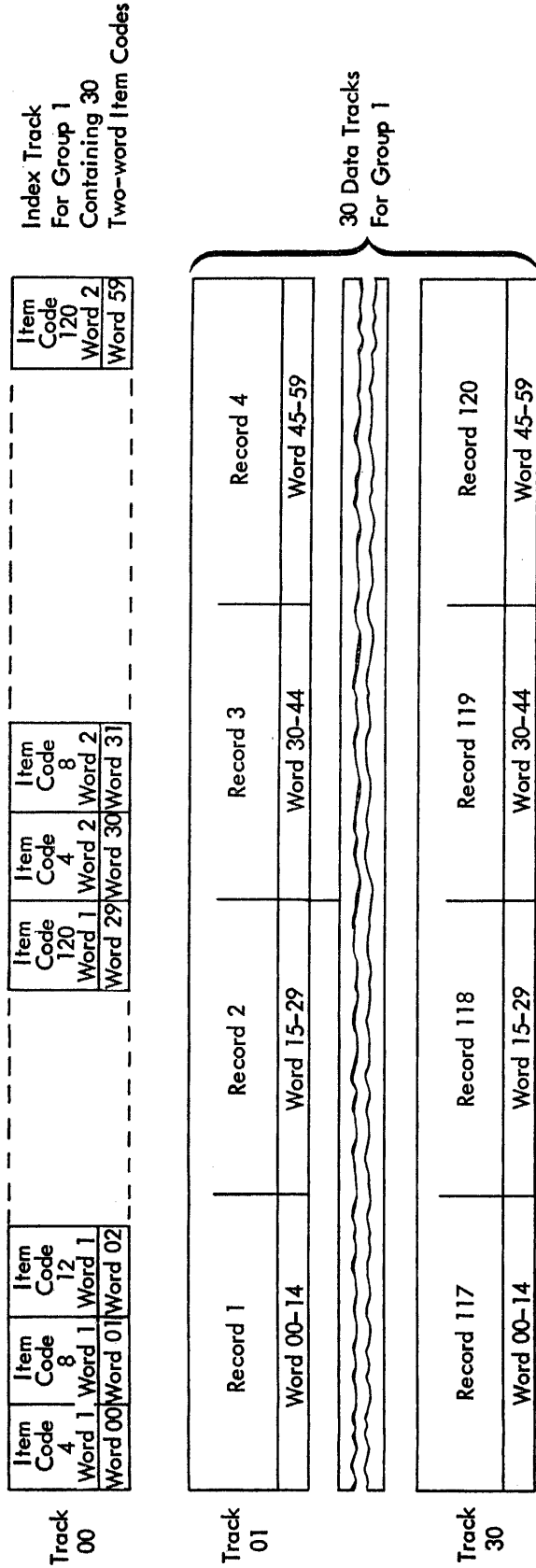


FIGURE 27. LAYOUT FOR ONE GROUP (METHOD B)

Figure 27 shows a record layout for method B. The record item code consists of two words for each record. The index track layout shows that all 60 words are utilized. Words 00 to 29 contain the first word of each item code corresponding to the last record on each data track. The second word of each of the 30 item codes is placed in words 30-59 of the index track. This storage arrangement permits a table lookup on the 30 first-words of the item code. The address resulting from the table lookup is modified by adding 30, and the second-word item code area is analyzed to select the second part of the desired second-word item code. At this point the data track address is known, and a RAM seek is begun to locate the actual track that contains the desired record.

Methods C and D are identical to each other except for the size of the record identifying code. These two methods differ from methods A and B, because they contain a relatively small drum table, an intermediate indexing track on each disk, and an index track for each group of data tracks. Thus, by placing an intermediate index track on each disk, the size of the drum table is reduced, but another RAM seek of approximately 200 milliseconds is added to the access time of a record.

The drum table provides storage for selecting the intermediate index track (one of 50 disks per file). The intermediate index track is brought into imme-

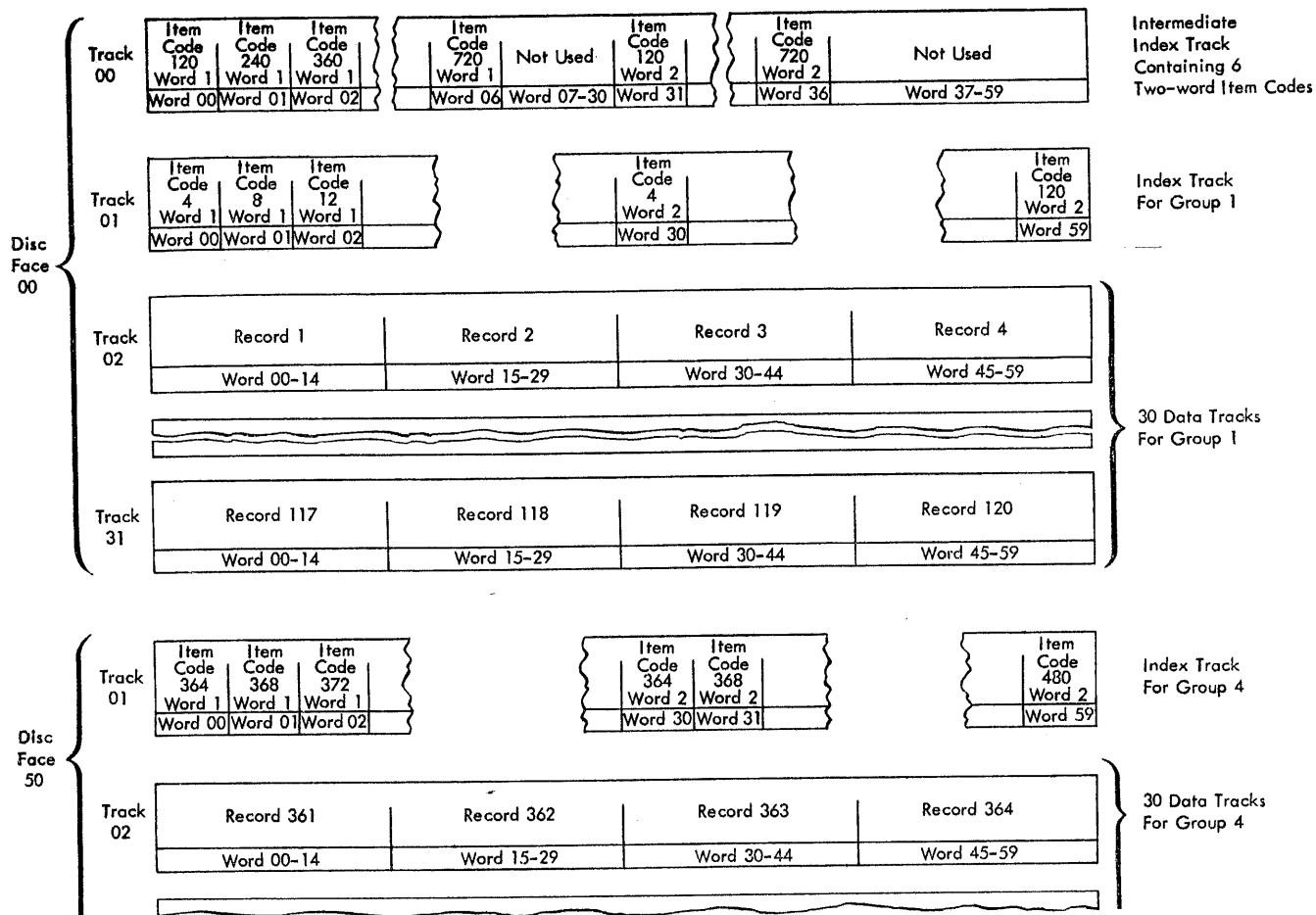


FIGURE 28. LAYOUT FOR DISKS (METHOD C)

mediate access storage and used to select the proper index track (3 per disk face, 6 per disk). A Seek is executed after which a Read-RAM operation brings the proper index track into immediate access storage. The index track permits the data track address to be developed, and a RAM Seek and Read is performed to bring the data track containing the desired record into immediate access storage.

The record layout for method C is shown in Figure 28. The upper track shown is the intermediate index track for the top disk in a RAM file. Two of the six index tracks associated with the intermediate index track are shown. Track 01 is the index track for group 1 of 3 located on disk face 00. The lower two tracks shown are on disk face 50 (bottom side of disk face 00 in the RAM file). Track 01 of disk face 50 is the index track for group 4 of the 6 index tracks associated with the intermediate index track.

A record layout for method D is not shown because it is similar to Figure 28 in arrangement. Method D differs slightly because a three-word item code identifies each record and only 15 data tracks can be included in a group. Twelve groups are associated with each intermediate index track (six groups on the top of each disk, and six groups on the bottom of each disk).

A summary of the four file packing methods follows:

METHOD A

1. Two index tracks per disk face; 49 data tracks per index group.
2. 200 index tracks per file
9800 data tracks per file
400 words per file for the drum tables
No unused tracks per file
1.1 seconds estimated access time to a record.

METHOD B

1. 3 index tracks per disk face
30 data tracks per index group
2. 300 index tracks per file
9000 data tracks per file
900 words per file for the drum tables
700 unused tracks per file
1.1 seconds estimated access time to a record

METHOD C

1. 1 intermediate index track per disk
3 index tracks per disk face
30 data tracks per index group
2. 50 intermediate index tracks per file
300 index tracks per file
9000 data tracks per file
150 words per file for the drum tables
650 unused tracks per file
1.4 seconds estimated access time per record

METHOD D

1. 1 intermediate index track per disk
6 index tracks per disk face
15 data tracks per index group
2. 50 intermediate index tracks per file
600 index tracks per file
9000 data tracks per disk face
200 words per file for the drum tables
350 unused tracks per file
1.4 seconds estimated access time per record

Randomizing Formulas

The table lookup method previously described provides for efficiently packing the maximum number of records in a given RAM area. The table lookup method, however, requires several seek and read RAM operations to find the exact record and thus is not the fastest method.

Most identification code structures are designed to express distinct classes of information. These classes may constitute a prefix, or an assigned block of numbers, which make up only a fraction of all the possible number combinations in a code structure. The unassigned portions of the code structure leave large gaps in a numerically ordered sequence. This is caused by variations of size among classes and because of

choosing blocks of numbers large enough to provide ample room for future assignments. Because of this structure many more digits are included in the identifying code than would be required to count the number of actual items.

As an example: accounting systems of 100,000 account numbers are often identified by a 10-digit or 10-position code structure (which has a capacity for identifying 9,999,999,999 accounts). Only 1/100,000 of the possible identification codes in the structure would actually be used to represent the 100,000 active accounts. The function of a randomizing formula would be to shrink the 10-digit code to a smaller number more nearly representative of the total account numbers actually assigned and at the same time delete as much as possible the structural characteristics of the 10-digit code to smooth distribution in order to arrive at a set of numbers more truly random.

This process of shrinking and randomizing can be carried out by various means. A few of these are listed:

1. Using alternate or selected digits of the record code.
2. Deriving a new number by,
 - a. partitioning the record code into pieces, which are then added together to form a new number.
 - b. multiplying the record code by itself (squaring) and extracting fixed positions of the new product to serve as a RAM address.

To illustrate the listed methods consider a 10-digit account number such as:

<u>MAJOR ACCOUNT</u>	<u>STANDARD EXPENSE</u>	<u>DEPARTMENT</u>
CODE 985	CODE 0261	CODE 423

Method 1 could select a specific four-positions of the account number and consider this the RAM address. If the units position of major account, the tens-position of standard expense, and the high-order and low-order positions of department number were utilized, the resulting address would be:

5 - 6 - 4 - 3

Thus, 5643 would be a random number representing the account number 985-0261-423. The 56 specifies the RAM disk, the 43 specifies track 43 on the RAM disk.

Method 2A could add the 3 elements of the 10-digit code together to form a 4-digit number such as:

$$\begin{array}{r} 985 \\ 0261 \\ \underline{423} \\ 1669 \end{array}$$

The resulting sum 1669 would be the random number used to specify the RAM disk and track where the data for account number 985-0261-423 are stored.

Method 2B can be illustrated as follows:

$$(985-0261-423)^2 \quad 97,027,650,101,441,984,929$$

Extracting the center four digits of this 20-place product gives the number 1014. This new number 1014 is used as the RAM address to specify disk and track. It is a random number and represents a cross multiplication of the various digits of the original 10-digit account number.

Record identification code structures that consist of mixed alphabetic and numerical characters are expressed as a number in the 650 system and thus are handled in the same fashion as a straight numerical code.

In the preceding examples, a smaller number is derived from a larger number by various methods. In these examples a 4-digit number resulted from a 10-digit account number. Obviously a 4-digit number cannot provide a unique address for more than 9999 items. However, each 4-digit address represents a RAM track that has the capacity for 600 digits of

data. Several records can be placed on a track provided their total length does not exceed the 600 digits' capacity.

The disadvantage of selecting or shrinking to a smaller code lies in the fact that several dissimilar record codes will result in the same RAM address. These duplications or synonyms *cannot* be avoided and must be provided for. Consider the following example:

Using Method 1, account code 985-0261-423 was stored in location 5-6-4-3. These digits were selected from specific positions of the account code structure. However, account code 325-4769-483 would also produce address 5-6-4-3 using the same method of selecting specific positions of the account number. Thus, even though the two account codes are different, the address conversion to a RAM address generates the same address for the two account numbers. This duplication is handled in one of two ways. Either the RAM address has capacity to store the data for all the account numbers that try to utilize the same address, or a separate overflow storage area is planned. The other consideration is that some addresses may not be used at all, and, therefore, a portion of the file capacity is unoccupied.

The foregoing implies that each record identification code structure should be carefully analyzed to determine the best approach for addressing the RAM file. The flexibility of the 650 system, the power of the stored program machine, and its arithmetic operations, provide capability for many addressing techniques. Select or develop a method that provides the fastest access to the record, furnishes a minimum of duplicate addresses, and efficiently utilizes the storage space.

IBM 838 INQUIRY STATION

PHYSICAL DESCRIPTION

THE IBM 838 Inquiry Station typewriters provide transmission of data to the 650 system and automatic typing of data replies from the system (Figure 29). Up to 10 inquiry stations are available. They can be arranged in one control or in two controls. Each Control independently communicates with 650 system through its own inquiry station synchronizer. Thus, by having two Controls up to twice the volume of inquiries can be handled. By proper programming the two Controls, inquiries and replies can be functioning from both Controls at the same time.

The typewriter has the following standard specifications:

Keyboard — 44 character billing arrangement

Single case type

Ten pitch

Type styles

Manifold No. 10

Pica Gothic

Accounting Gothic

Carriage, 12 inch

Conventional platen

Line spacing, either 4 or 6-to-the-inch

2-color ribbon device

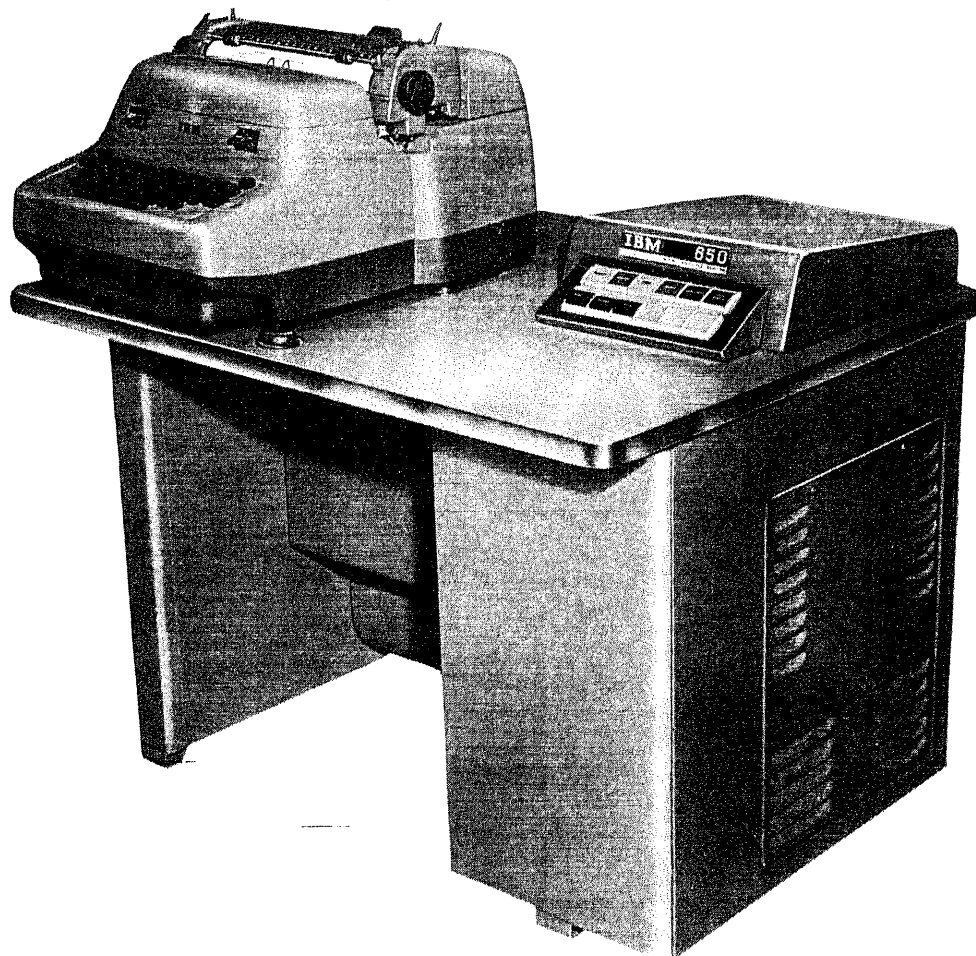


FIGURE 29. 838 INQUIRY STATION

Additional control keys and lights are mounted on the unit at each station and are described in a later section of the manual. The typewriter can be used as a standard single-case typewriter when not being used as a part of the 650 system.

The inquiry stations are completely flexible in that they are capable in inquiring into any record held in the 650 system. By means of the typewriter key-

board, data, instructions, alterations to existing records or programs can be provided to the system. Automatic typing of replies to inquiries, miscellaneous messages, or productive output printing can be accomplished.

Each typewriter is addressable from the 650 program, thus an inquiry received from one station can reply at a different station if desired. A program tape on each inquiry station provides for format arrangement of the inquiry and reply. The program tape also contains a control word that identifies the station and specifies the 650 program routine to be followed for the particular inquiry.

Connection to the 650 System

The inquiry stations of a Control can be connected to the system in two strings. Each string originates at the 652.

Control 1 uses connections labelled A and B and Control 2 uses connections C and D. Separate transmission channels are provided for inquiries going to the 650 and for replies coming from the 650 to the 838 unit. Thus, while one station is making an inquiry, another station can be simultaneously typing a reply for a previous inquiry. The 652 Control Unit also contains the 838 checking circuitry.

The 838 inquiry station controls are arranged as follows:

Control 1 corresponds to drum words 1-10 (or 51-60) for inquiry and 27-36 (or 77-86) for reply.

Control 2 corresponds to drum words 39-48 (or 89-98) for inquiry and 13-22 (or 63-72) for reply.

When Inquiry Station Control 1 is installed on a 650 system either input-output synchronizer 3 or an inquiry station synchronizer must also be installed.

When Control 2 is ordered, it is installed in addition to control 1.

A 650 system can have an input-output unit such as a 533 or 407, and inquiry station control 2 connected to synchronizer 3. However, for proper operation only one of these units can be connected to the system at any one time.

The synchronizers used for inquiries and replies are designed to accept and transmit one character at a time. On an inquiry the 838 sends one character at a time (as it is typed by the operator) to the inquiry synchronizer. A reply to an 838 from the 650 con-

Synchronizers		0	
Input Sync. 1 (OP 70,72)	Inquiry Control 1 (OP 26)	00	50
		01	51
		02	52
		03	53
		04	54
		05	55
		06	56
		07	57
		08	58
		09	59
Output Sync. 2 (OP 74)	Output Sync. 3 (OP 77) or Reply Control 2 (OP 79)	10	60
		11	61
		12	62
		13	63
		14	64
		15	65
		16	66
		17	67
		18	68
		19	69
Output Sync. 1 (OP 71)	Reply Control 1 (OP 79)	20	70
		21	71
		22	72
		23	73
		24	74
		25	75
		26	76
		27	77
		28	78
		29	79
Input Sync. 2 (OP 73,75)	Input Sync. 3 (OP 76,78) or Inquiry Control 2 (OP 26)	30	80
		31	81
		32	82
		33	83
		34	84
		35	85
		36	86
		37	87
		38	88
		39	89
40	90		
41	91		
42	92		
43	93		
44	94		
45	95		
46	96		
47	97		
48	98		
49	99		

FIGURE 30. DRUM STORAGE AND SYNCHRONIZER LAYOUT

sists of transmitting one character at a time from the inquiry synchronizer to be typed by the 838. Figure 30 shows drum storage and synchronizer layout.

OPERATION

THE KEYBOARD communication to the 650 system provided by the IBM 838 units begins with the operator requesting permission to send the inquiry or data to the 650. The 650 signals the 838 operator to proceed. The proceed signal is given immediately upon determination that the inquiry synchronizer is free to accept an inquiry. The proceed signal conditions the 838 keyboard and the message or inquiry is then typed. The typing provides visual verification to the operator. If an error is detected, the entire message can be cancelled by the operator and a new inquiry begun by re-requesting the 650 and retyping the message.

When the message is typed and visually verified, the operator depresses a release key to signal the 650 that the inquiry is complete or the program tape reaches the start output punch. At this point the 838 pauses until the reply is sent by the 650. Estimates indicate that the reply should be available in approximately 2 seconds. This timing is dependent upon the programming of the 650 to recognize that an inquiry is waiting.

If the 650 immediately branches to a sub-routine to develop a reply, the 838 will begin typing a reply in less than the estimated two seconds. The reply is automatically typed at a rate of 10 characters a second. Formulation of a reply may take considerably longer than two seconds where an involved calculation is requested or if the 650 program does not immediately recognize that an inquiry is waiting. The inquiry time will vary depending on the speed of the operator and length of the message. Approximately 15 to 25 seconds is estimated for an average inquiry time.

Another aspect of 838 operation is format control. This is provided by a plastic program tape on each 838 unit. A section of the tape is punched to provide format control for an inquiry. Similarly, another tape section is punched for output or reply format.

Because an installation can consist of many inquiry stations, there will be occasions where several stations

may simultaneously request permission to make an inquiry. The system is designed to remember each request but accept only one request at a time. For example, assume that stations 4, 1, and 6 simultaneously request to make an inquiry. The 650 will accept the inquiry from station 1 and remember that stations 4 and 6 are waiting. When the reply is sent for the first inquiry, the 650 signals station 4 to proceed. In other words, the 650 stacks the requests and processes them in sequence based upon the station number.

650 Operation Codes

Two operation codes have been added to the 650 for inquiry stations: a Branch-on-Inquiry code (Op code 26) and a Reply code (Op code 79).

The inquiry stations are capable of inquiring at irregular intervals; for example, during the course of a day's operation perhaps 50 or 1000 inquiries may be made. To meet this irregular demand, the programming routines for inquiry stations should be available for 650 processing but not as a part of the main routine; that is, the inquiry station program routines are sub-routines that the 650 will branch into if an inquiry is waiting.

26 BIN Branch on Inquiry

Control 1 Inquiries. As previously stated, inquiry stations can be connected to one or two controls. Control 1 is connected to the inquiry synchronizer corresponding to drum words 1-10 (or 51-60). Control 2 is connected to the inquiry synchronizer corresponding to drum words 39-48 (or 89-98). However, the one set of operation codes (BIN 26, and RPY 79) applies to both synchronizers. Which control will be used for an inquiry will be determined by the ten's position of the D-address (and it must be a valid address). For simplification a Control 1 operation will be discussed and the following example used as an illustration.

INSTR NO.	LOCATION OF INSTRUCTION	OPERATION		ADDRESS	
		ABBRV.	CODE	DATA	INSTRUCTION
		BIN	26	0061	Main Routine

This instruction, referring to Control 1 inquiry stations, would be located at a convenient place in the normal processing routine.

INSTR NO.	LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
		ABBRV.	CODE	DATA	INSTRUCTION	
	1406	Normal Routine			1410	
	1410	BIN	26	0111	1413	Control 1 Inquiry Stations
	1413	BIN	26	0149	1417	Control 2 Inquiry Stations
	1417	Resume Normal Routine				

FIGURE 31. TWO CONTROL BRANCH ON INQUIRY

The example illustrates a control 1 input. This is determined by the 1 or 6 in the ten's position of the d-address. This instruction can contain any data address corresponding to words 10-19 (or 60-69) and could branch to any one of these drum locations. This allows an instruction to be located in a word corresponding to drum word 10 (or 60), which is entered from the inquiry station, as well as drum words 11-19 (61-69). To insure that Control of the program remains with the 650 it is recommended that a Branch-on-Inquiry instruction (26) branch to a location that is outside the 10 words of the inquiry synchronizer and which cannot be altered by an inquiry station.

The start of the subroutine in the example shown is located in word 61.

The inquiry message can be a maximum of ten words. The last word (word 10) is the inquiry control word and specifies the station number as well as being used for data.

The inquiry control word is entered either from the program tape or the keyboard. It must enter the tenth word of the inquiry synchronizer. The station number code, for ten stations, ranges from 00 to 09. Two positions of word 10 should be reserved for the station number code and entered from the program tape. The remaining positions of the control word are completely flexible and can be used for data or to designate the address of a desired subroutine.

An example of the inquiry control word for station 1 asking for reply number 3, would appear as follows:

	STATION NUMBER	SUB- ROUTINE
0000	01	0003

Assuming the Branch-on-Inquiry (26) instruction has directed us to a subroutine, this subroutine can determine whether this station has the right to request subroutine 3 information. If the inquiry is valid, the

program branches into the reply routine; if invalid, it sends back a reply indicating an invalid inquiry and returns to the main routine. Figures 35 and 36 show the program and tape layout for this application.

Control 2 Inquiries. Control 2 inquiries operate the same as Control 1 inquiries except that they are independently connected to the system and utilize synchronizer 3. The BIN (26) Instruction data address, tens position, must contain a 4 or 9. This d-address provides that words 40-49 (or 90-99) can be specified as the beginning instruction for the inquiry subroutine. Thus, an instruction, such as the following, specifies Control 2 inquiries.

INSTR NO.	LOCATION OF INSTRUCTION	OPERATION		ADDRESS	
		ABBRV.	CODE	DATA	INSTRUCTION
		BIN	26	0149	XXXX

This instruction functions identically to a Control 1 instruction, except that the ten words from the inquiry synchronizer are transferred to general storage words applying to Control 2.

Branch-on-Inquiry for Two Controls of Inquiry Stations

When both Control 1 and 2 inquiry stations are installed, two Branch-on-Inquiry instructions are inserted in the normal program routine. One instruction is inserted for each Control (Figure 31).

An inquiry through a Control must be followed by a reply through the same control.

79 RPY Reply

Every inquiry must be followed by a reply. This is a system control requirement, which automatically insures a recording of the inquiry. The system is designed in such a manner that the Branch-on-Inquiry

Op Code cannot branch to its D-address until the previous reply has been sent.

Formulation of data for the reply is accomplished by normal programming.

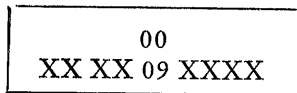
The reply instruction transfers the data from general storage to the inquiry synchronizer and automatically starts the typing of the reply on the designated typewriter.

The Reply (79) instruction requires that the Control through which the reply will travel, be designated by the 10's position of the data address. For Control 1 this must be a 3 or 8 and indicates that output words corresponding to 27-36 (or 77-86) contain the reply.

A 2 or 7 is used in the 10's position of the data address on a Reply (79) instruction to denote a Control 2 reply. This indicates that output words corresponding to 13-22 (or 63-72) contain the reply.

Because the reply can be sent to any of the typewriters in the Control, the address or station code must be placed in the 5th and 6th positions of the 10th word of the reply.

STATION CODE



This corresponds to the positions of the inquiry control word previously described.

Either a long reply or a short reply can be sent. This is controlled by the program tape on the inquiry station. It is not recommended that inquiry stations be used as output printers for the system as typing speed is not fast enough to keep up with 650 processing speed.

Timing Considerations

Figure 32 illustrates the timing relationships, which would occur when an inquiry is made for information from the RAM application, shown in the block diagram (Figure 2).

Inquiry Station Program Tape

Each inquiry station has provision for a program tape that provides vast flexibility for:

1. forms control through carriage tabulations and line spacing,
2. entry and exit arrangement for data transmission,
3. control and identification of data for 650 processing, and
4. zero suppression.

The program tape is a 16-channel perforated plastic tape similar to the paper tapes that are used on IBM tape-controlled carriages and the IBM 884 Typewriter Tape Punch. The program tape can be a maximum length of 4 feet (48 inches) and must be a minimum length of 1 foot (12 inches). Figure 33 shows a rear

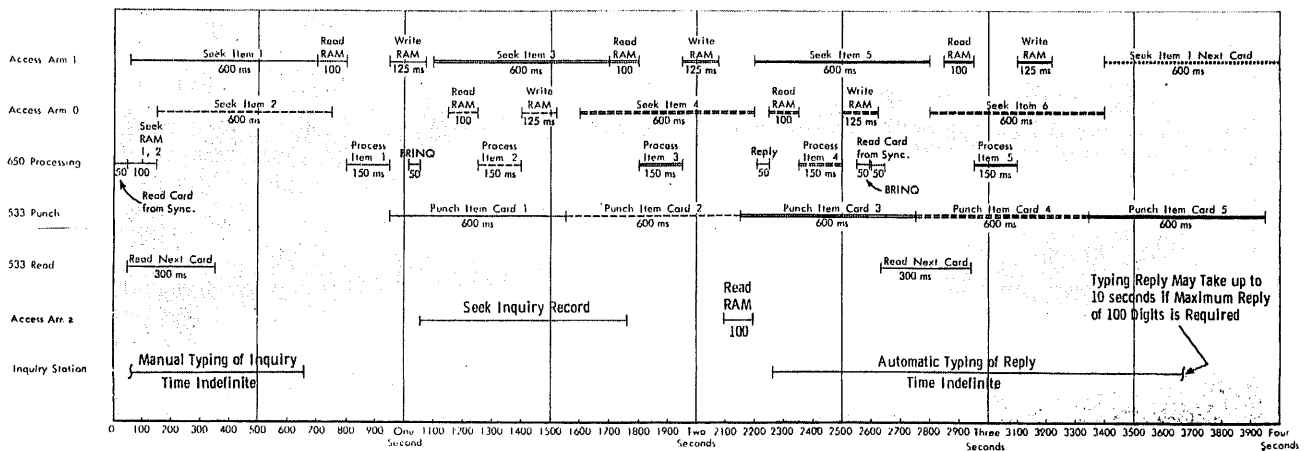


FIGURE 32. TIMING CHART SHOWING INQUIRY STATION OPERATION INTEGRATED WITH 650-RAM APPLICATION

view of the 838 and the program tape location on the machine.

Each of the 16 channels has an assigned significance. Holes are punched in the tape channels to control the assigned functions. The tape is advanced in conjunction with the programmed movements of the carriage or the control keys on the inquiry station console. Separate program tapes are prepared for each application and are easily interchanged by the operator.

Because the program tape can move in conjunction with the typewriter-carriage movement, each column of the tape is punched to correspond to a given position of the data being sent to the 650 and all characters typing the reply. On an inquiry routine, as each character is typed, the program tape designates the word and position of the inquiry synchronizer where the typed character is to be transmitted. On a reply from the 650 to an inquiry station, the program tape is again moving. The tape punching selects the word and position within the inquiry synchronizer that is to be typed in that location on the paper.

In an application where an inquiry station serves as an input and as an output device, both programs are punched into one tape. The ends of the tape are then joined together to form a closed loop. This permits the inquiry station to proceed automatically

from an input mode to an output mode and then be available for the next input mode inquiry.

Each inquiry and reply combined with the automatically inserted zeros fills ten words of the inquiry synchronizer. An inquiry must furnish information that is in addition to the message. This additional information (control word) consists of a code to identify the station, the 650 subroutine desired, etc.

Program-Tape Channels

The 16 channels of the program tape are assigned the following functions:

<u>CHANNELS</u>	<u>FUNCTION</u>
1-5	Inquiry synchronizer word (digit emitter if channel 12 is also punched).
6-10	Digit position of synchronizer word
11	Alphabetic character
12	Control word (inquiry), zero suppression (reply)
13	Inquiry format begins
14	Reply format begins
15	Carriage return
16	Tabulate
15 and 16	Space
7, 8, 11	Test for negative sign indication for word designated by channels 1-5.

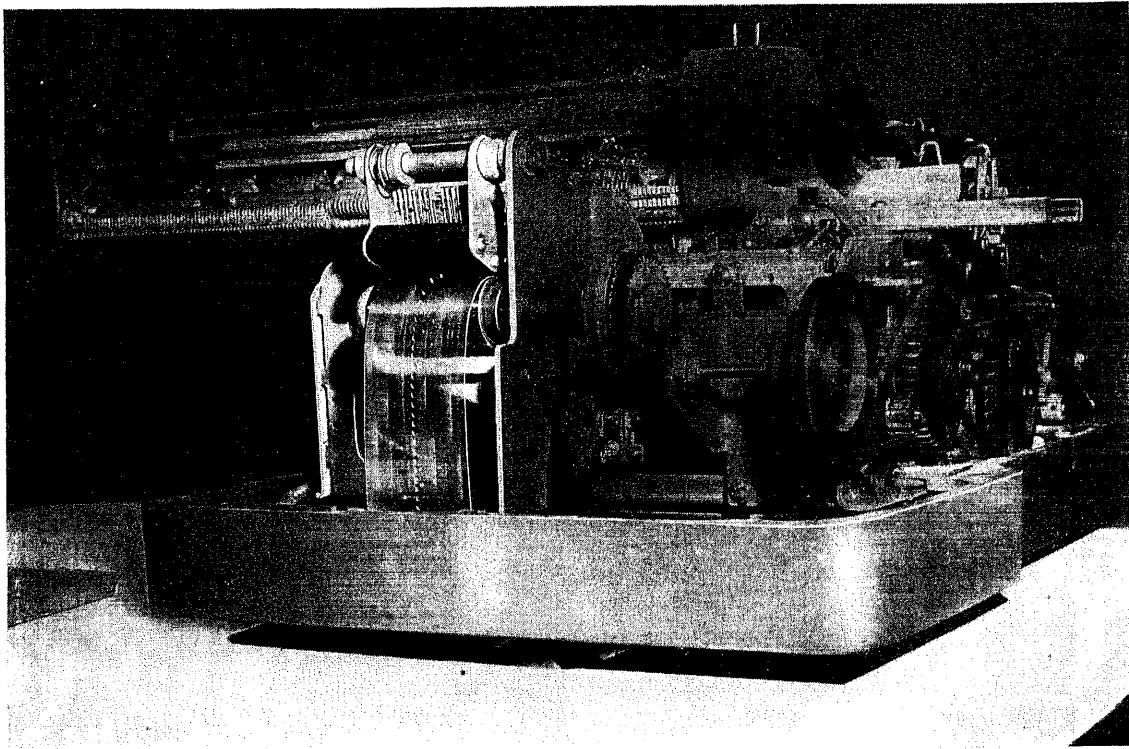


FIGURE 33. REAR VIEW OF TYPEWRITER SHOWING LOCATION OF PROGRAM TAPE

Channels 1 through 10 are punched to designate the 650 word, and position within the word, which is to receive (on an inquiry) or transmit (on a reply) the characters for the specific printing position being typed.

The code punched in channels 1-5 designates a specific word of the ten inquiry or reply words; channels 6-10 designate a specific position of the word (units, tens, hundreds position, etc.). Each of the channels is assigned a digit value of a 5-bit code. This 5-bit code is the same code that is used internally by the 650 to record data in general storage. It is a self-checking code in that each digit must be comprised of 2 out of 5 bits. Any digit which does not meet the requirements of the 2 out of 5 code is recognized as an error to stop the machine on input. On output a validity check error prints an asterisk in red in place of the character.

Channels 1-5, Word

Channels 1-5 specify the 650 inquiry synchronizer word that is being used for the typed position both on inquiry and reply. It is coded as follows:

<u>WORD DESIGNATION</u>	<u>CHANNEL PUNCHING</u>
Word 1	1, 2
2	1, 3
3	1, 4
4	2, 4
5	3, 4
6	1, 5
7	2, 5
8	3, 5
9	4, 5
Word 10	2, 3

The foregoing table illustrates that word 2 is specified by punching channels 1 and 3; word 7 is specified by punching channels 2 and 5. Thus, each word is specified by punching two channels of the program tape.

The function performed by channels 1-5 is slightly different when channel 12 is also punched in the same tape column. This is explained in the channel 12 description.

Channels 6-10 Digit Position

Channels 6-10 are punched to designate the position within a word of the inquiry synchronizer on an inquiry or reply.

<u>DIGIT POSITION</u>	<u>CHANNEL PUNCHING</u>
1 (units)	6, 7
2 (tens)	6, 8
3	6, 9
4	7, 9
5	8, 9
6	6, 10
7	7, 10
8	8, 10
9	9, 10
10 High- Order	7, 8
Test Sign Position for Neg.	7, 8, 11

For example: the hundreds position of word 4 is specified by punching tape channels 2, 4, 6, 9. The fifth position of word 8 is specified by punching channels 3, 5, 8, 9.

Channel 11 Alphabetic

Channel 11 is punched to designate that an alphabetic character is being transmitted or received by this column. Alphabetic data are processed by the 650 system as a two-digit number. On an inquiry operation, channel 11 changes the typed alphabetic character to a two-digit number for storage in the 650 system. On a reply operation, a punch in channel 11 causes two positions of the synchronizer to be analyzed and converted to an alphabetic character for typing. Only one program-tape column is needed to specify the alphabetic character even though two digits of the 650 word are involved. The program tape is punched to specify the low-order digit in the 650 word utilized for the alphabetic character.

While this channel specifies whether a single character is alphabetic or numerical, it is necessary that all other positions of the same word are similarly defined by channel 11. That is, all positions of a word are in either alphabetic representation or in numerical representation.

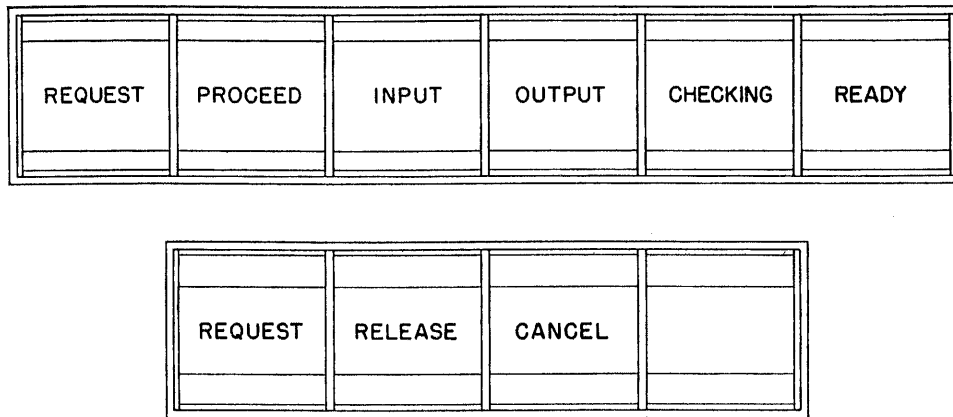


FIGURE 34. 838 CONSOLE

Channel 12, Control Word (Inquiry); Zero Suppression (Reply)

Channel 12 performs two functions. On an input (inquiry) program, it identifies the column as a position of the control word (word 10). On an output (reply) program, it performs a zero suppression function for output typing format.

Input program tape. A punch in channel 12 identifies this column as a position of the control word on an inquiry. The control word must be a numerical word and is automatically directed to word 10 of the drum input synchronizer. This means that channels 1-5 of the program tape are not needed to specify the 650 drum-input word. Therefore, channel 12 punching changes the function of channels 1-5 from "coded word" to that of a digit emitter for that tape column.

Channels 1-5 are punched to designate the actual digit that is to be automatically transmitted to the 650. The channels 6-10 are punched to designate the position within the control word. The following example shows the coding for two positions of the control word.

CHANNEL	PUNCHED	DESCRIPTION
1-5	Code 6	Transmits a digit 6.
6-8	Code 2	To the tens position of control word 10 in the synchronizer.
11	Unpunched	Numerical character.
12	Punched	This column is designated as a position of the control word.

When channel 12 is punched and 1-5 are used as a

digit emitter, the program tape advances automatically.

Zero Suppression. On output format, channel 12 functions to suppress typing of high-order zeros in a field until a significant digit is typed. Channel 12 is punched in the high-order column of each numerical field. When this column is reached, the hole in channel 12 turns on the zero-suppression device, and it remains on until the first significant digit is reached.

Channel 13 Input

Channel 13 is punched to indicate the start of the input format on the tape. It also operates to signal that the end of an output format has been reached. When a hole in channel 13 is reached, the input signal light glows to indicate to the operator that the tape is in position for an inquiry message.

Channel 14 Output

Channel 14 is punched to indicate the beginning of the output format punching on the tape. It prepares the Inquiry Station to accept a reply from the 650 and turns on the output light to signal the operator that the tape is in position for a reply. Channel 14 also signals the system that the end of the input format has been reached.

Channel 15 Carriage Return

A hole in channel 15 causes an automatic carriage return. The tape suspends data transmission with the 650 until the carriage return is complete.

Variable field length applications may require the operator to depress the carriage-return key before channel 15 is reached. In this case the program tape will automatically advance to the channel 15 column.

Channel 16 Tabulate

The function of channel 16 is to cause the typewriter to tabulate automatically to the next tab stop. The advance of the tape and manual operation of the tab key for variable-length fields is identical to the previously described carriage-return operation by channel 15. Tabulations should not be programmed for fewer than three spaces.

Channel 15 and 16, Space

When both channels 15 and 16 are punched, the typewriter is caused to space one column. The spacing function suspends any transmitting or receiving of data with the 650. Spacing from the program tape functions on both the input and output modes of operation. Channels 13-16 must not be punched in columns with word or digit punching.

CHECKING

On Inquiry

As each character is sent from the inquiry station to the synchronizer, an automatic character check is made to insure that the character sent matches the one in the synchronizer.

A validity check is also made on all characters entering the inquiry synchronizer.

The station number entering an inquiry is automatically checked against the number entered in the 5th and 6th positions of the inquiry control word.

On Reply

Inquiry station number is checked for a valid number.

A validity check is made on all characters sent from the inquiry synchronizer.

OPERATING KEYS AND LIGHTS

SEVERAL operating keys are located at each inquiry station, which allow the operator to control the various functions of the machine. Figure 34 illustrates the arrangement of these keys and lights.

Keys

Request Key

When an inquiry is to be made, the operator depresses the request key. The request key checks that the program tape is stopped in a column designated as the beginning of the input format (hole in channel 13). If not, the unit automatically advances the tape to the input hole. If the request key is held depressed when the tape reaches the hole in channel 13, it then sends a signal to the 650, asking permission to transmit a message. As soon as the 650 recognizes and accepts the request signal from this station, the PROCEED light glows to inform the operator that the message can now be sent.

When this request is received, the 650 erases all information in the synchronizer from the preceding inquiry and places zeros in all positions of the 10 words. It then automatically checks that all 10 words of the synchronizer contain zeros and proceeds to accept the message from the requesting inquiry station. As each character is transmitted by the 838 and recorded, it is then automatically read and checked against the character sent by the station. This is both a validity check and a character check.

Release Key

The release key is used to signal the 650 when the complete inquiry message has been sent. As the last character is typed and the end of the input format of the program tape is reached the tape advances to the next tape column. This column should be punched in channel 14 (output mode) to signal the beginning of the output format and cause the output light to glow.

The operator visually verifies the message and depresses the release key to signal the end of the message and permits the 650 to process the inquiry. If the release key is depressed at a time when the program tape is not reading a column punched in channel 14, an error is signaled. The error signal requires that the message be cancelled and retyped. This checking feature insures that the operator sends only complete messages and that the last character typed coincides with the end of the input format on the program tape. If the operator types after the output format (14) has been reached, an error is signalled.

Cancel

Depressing the cancel key extinguishes the error light. It also drops the typewriter out of the request status. The operator can then re-request and retype the message.

Lights

Indicator lights provide the operator with a visual indication of the status of the operation of the machine.

Request Light

This light glows when the operator depresses the request key and is waiting for the 650 to accept the inquiry. It is automatically extinguished when the PROCEED signal is received.

Proceed Light

The proceed light glows when the 650 accepts the

inquiry request signal. It stays lit while the inquiry message is being sent.

Checking Light

The checking light glows when an error in transmission of data is recognized. Depressing the cancel key extinguishes the checking light.

Input Light

The input light glows when the program tape recognizes a hole in channel 13. It indicates to the operator that the tape is at the beginning of the input format.

Output Light

This light glows to indicate to the operator that the program tape is at the beginning of the output format section of tape and ready for a reply.

Ready Light

This light when on indicates the system is ready for normal operation.

EXAMPLE OF INQUIRY STATION PROGRAM AND TAPE LAYOUT

Two inquiry stations are connected to Control 1. These stations are numbered 0 and 1. Three different types of inquiries can be processed and they are numbered 1, 2 and 3. Both stations are allowed to make inquiry 3. In addition to inquiry 3, station 0 can only ask for or receive a reply for inquiry 1 and station 1 can only ask for or receive a reply from inquiry

2. Any other types of inquiries will cause ERR to print on the output format and the program will return to the main routine.

This subroutine keeps the 650 program always in control of the system.

In this example the typing format for all types of inquiries and replies should be the same.

The index registers were included in this program to illustrate their use. By their utilization processing speed is increased and stored constants on the drum are reduced.

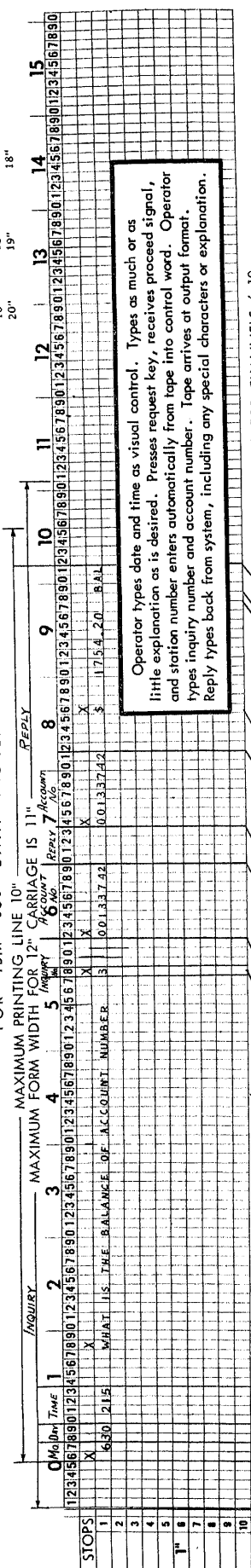
INSTR NO.	LOCATION OF INSTRUCTION	OPERATION		ADDRESS		REMARKS
		ABBRV.	CODE	DATA	INSTRUCTION	
	0100	BIN	26	0061	Main	
	0061	RAI	60	0060	0101	Read control word in upper accumulator
	0102	RAA	80	0003	0103	Load index register A with 0003
	0103	RAB	82	0001	0104	Load index register B with 0001
	0104	SA	51	8003	0105	Check for a 3 inquiry. If a 3,
	0105	BNZA	40	0106	Reply 3	process reply 3; if not, branch.
	0106	SRT	30	0004	8003	Shift control word to use
	8003	NOOP	00	0000	0001 or 0	Station number as instruction address of sub routine
	0001	SA	51	8006	0107	Check for a 2 inquiry from
	0107	BNZA	40	Error	Reply 2	Station 1--Branch to error if not a 2.
	0000	SA	51	8006	0108	Check for a 1 inquiry from
	0108	SA	51	8006	0109	Station 0--Branch to error if not a 1
	0109	BNZA	40	Error	Reply 1	
		Error Routine				
	Error	LD	69	0125	0126	Abbreviation for Error (ERR) stored
	0126	STD	24	0128	0137	in numeric code in location 0125=
	0137	RAL	65	8002	0138	0065797900. ERR is written on
	0138	STU	21	0127	0139	the Inquiry Station if an invalid
	0139	STD	24	0129	0140	Inquiry was made. Program goes
	0140	LD	69	0060	0141	back to main routine.
	0141	STD	24	0136	0142	
	0142	RPY	79	0136	Main	
	After processing any of the 3 correct replies and loading words 0127 - 0136 routine will go to location 0140 for its next instruction.					

FIGURE 35. PROGRAM FOR INQUIRY STATION. USING INDEX REGISTERS

INQUIRY STATION - - FORM AND TAPE LAYOUT

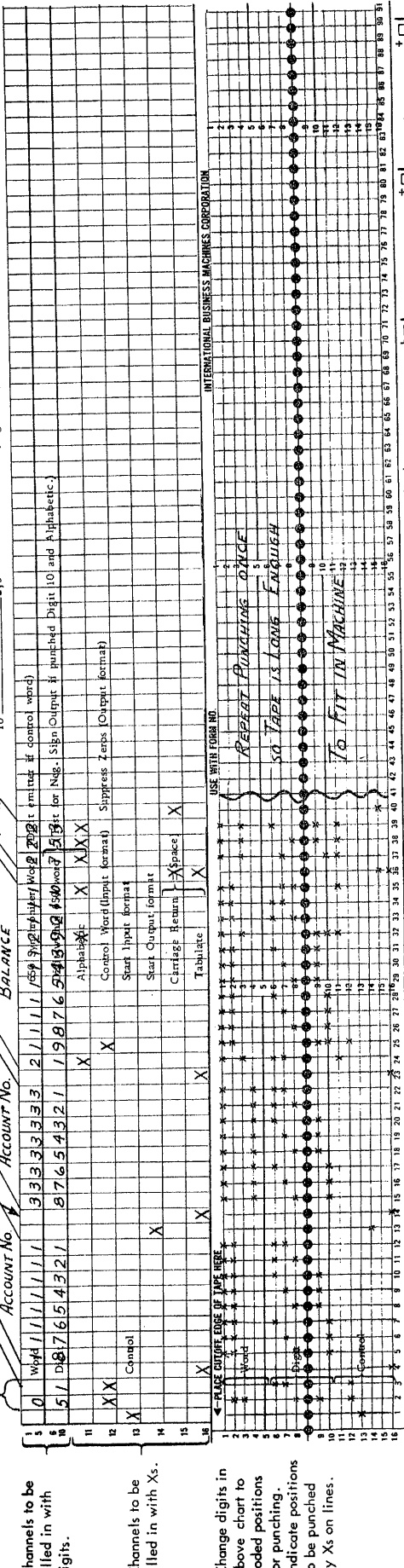
FOR IBM 650 DATA PROCESSING SYSTEM

Carriage Width: 12", 15", 16", 20"
 Maximum Form Width: 10", 14", 18"



CODING FOR CHANNELS 1-5

650 Word	Prog. Tape Channels	Position Within Word	Prog. Tape Channels
1	1,2	1	6,7
2	1,3	2	6,8
3	1,4	3	6,9
4	2,4	4	7,9
5	3,4	5	6,9
6	1,5	6	6,10
7	2,5	7	7,10
8	3,5	8	8,10
9	4,5	9	9,10
10	2,3	10 (High order)	7,8



ADDRESS	0051	0060	0127	0128	0129	ADDRESS
General Storage Locations	0000133742	0000010003	0010175420	0000133742	0000133742	
	INQUIRY ACCOUNT NUMBER WILL BE PROCESSED TO FORM RAM ADDRESS	STATION INQUIRY NUMBER WILL BE PROCESSED TO FORM RAM ADDRESS	REPLY ACCOUNT BALANCE	REPLY ACCOUNT NOTATIONS	REPLY ACCOUNT NUMBER	

FIGURE 36. SAMPLE PROGRAM TAPE LAYOUT

INQUIRY STATION CHECKING AND FILE PROTECTION

THE inclusion of inquiry stations in an IBM 650 system raises the possibility that an operator can alter or delete a record in the RAM file. The inquiry stations are designed to include checking features, which, if they are put into effect, will control any unauthorized alteration of a RAM record. These checking features consist of internal machine checks and system checks.

On an inquiry message, as each character is character checked, it is also checked for validity.

The system's checks at the inquiry station utilize the program tape flexibility to identify the inquiry unit, the 650 routine to be followed and the type of reply required. Another system's check to prevent unauthorized information requests is to limit a station to only certain program tapes. The 650 program will normally be written to accept only certain types of requests from a given inquiry station.

An application may be such that certain stations have permission to alter an inventory balance by reducing quantities or placing a reservation on the

stock item. Verification of the message from the station is quite important and can be handled in various ways. The operator has an opportunity to make a visual verification of the message before releasing it to the 650. The system's internal checking features insure that valid characters are sent. The programming provides additional flexibility to check the operation further. If the part number structure conforms to the self-checking numbering system of the IBM card punch, a part number verification can be programmed within the 650.

Another method is to have two operators send the same message. When the 650 receives the message the first time, it can be placed in a work area of the drum. On receipt of the message the second time, the 650 program can compare the two messages; and, if they are identical, begin processing the request. It is also recommended that a card be punched to make a record, possibly of every inquiry and certainly of every alteration to the file or program instruction. This system check provides a unit record separate from the document on the inquiry station.

INDEX

Application (Inventory and Billing)	10	Magnetic Tape Units (727)	8
Access Time (RAM)	25	Inquiry Stations (838)	8
Account Number Conversion to RAM Address	31	Output Light	53
Address Modification	31	Packing RAM File	36
Addressing RAM File	18	Proceed Light	53
Arithmetic Methods of RAM Addressing	31	Programming Example	21
Block Diagram Description (Inventory Application)	12	Program Tape Channels	
Branch on Inquiry (26 BIN)	46	Channels 1-5 Word	50
Cancel Key	53	Channels 6-10 Digit Position	50
Checking (838)	56	Channel 11 Alphabetic	50
Checking Light	53	Channel 12 (Control Word)	51
Console		Channel 13 (Input)	51
650	6	Channel 14 (Output)	51
652	23	Channel 15 Carriage Return	51
838	51	Channel 16 Tabulate	52
Controls 1-2 Inquiry Stations	47	Channels 15 and 16 Space	52
File Packing	36	RAM Addressing Example	18
File Protection	56	Random Access Memory	17
Flow Paths	22	Randomizing Formulas	42
Input		Read RAM (86 R-RAM)	20
Random Access Memory (355)	7	Ready Light	53
Accounting Machine (407)	7	Release Key	52
Card Read Punch (533)	6	Reply (79 RPY)	47
Magnetic Tape Units (727)	7	Request Key	52
Inquiry Stations (838)	7	Request Light	53
Input Light	53	Seek Access Time	25
Inquiry Stations	9, 44	Seek RAM (85 S-RAM)	19
Inquiry Station Connection to 650 System	45	Storage	
Inquiry Station Operation	46	Random Access Memory (355)	8
Inquiry Station Program Tape	48	Immediate Access Storage (653)	7, 23
Inquiry Station Two Controls	47	Magnetic Tape Units (727)	7
Inventory and Billing Application	10	Storage Arrangement	18
Job Timing	27	Table Lookup Address Modification	31
Keys	52	Timing Considerations	25
Lights	53	Unloading RAM File	30
Operation Codes	19, 46	Write RAM (87 W-RAM)	20
Output		Operation Codes	
Random Access Memory (355)	8	26 BIN Branch on Inquiry	46
Accounting Machine (407)	8	79 RPY Reply	47
Card Read Punch (533)	8	85 S-RAM (Seek RAM)	19
		86 R-RAM (Read RAM)	20
		87 W-RAM (Write RAM)	20

IBM[®]