SPERRY RAND

UNIVAC

9200/9300 DISC SYSTEMS

INSTALLATION PLANNING GUIDE

FOREWORD

This Installation Planning Guide has been prepared to help the user install his new UNIVAC 9200 or 9300 Disc System as smoothly and as economically as possible. Proven conversion methods are outlined; procedures are described; charts and forms are provided. The user who takes full advantage of the experience behind this publication can maximize his efficiency not only in the planning and implementation stages but for many years to come.

Although this guide is specifically oriented to the UNIVAC 9200 or 9300 System and 8410 Direct Access Subsystem, most of the concepts presented would apply equally well to all Univac disc systems.

CONTENTS

FIGURES

1. DIRECT ACCESS STORAGE - GENERAL CONSIDERATIONS

INTRODUCTION

In a 9200/9300 Disc System installation, the direct access storage device is but one element of the total system. It is one of the most important elements, however, and many of the decisions made during the installation planning stage will be based on its capabilities. In order to get optimum use of the system, therefore, it is essential to be thoroughly familiar with the basic concepts of direct access processing before starting to plan the installation. For that reason, Section 1 of this guide is devoted to a general description of direct access processing as it relates to the 8410 Direct Access Subsystem (DAS). Additional information on the concepts of direct access storage can be found in Introduction to Direct Access Processing (UP-7565) and UNIVAC 8410 Direct Access Subsystem General Description (UP-7567 Rev. 1).

GENERAL DESCRIPTION OF THE 8410 DIRECT ACCESS SUBSYSTEM

The 8410 DAS is an auxiliary storage device which provides low-cost, removable mass storage for large quantities of information. It may contain data, application programs, software, or work areas.

Disc storage is much larger than main computer storage, but is much slower to access. Like main storage, the disc can be accessed directly or sequentially. Instructions and data cannot be operated upon in auxiliary storage; they must first be brought into main storage. Another important difference is that main storage is accessed in bytes, whereas disc storage is accessed in 160-byte sectors.

The 8410 DAS is designed for use with any UNIVAC 9200/9300* System having a minimum configuration of 12K main storage, a reader, punch, printer, and Multiplexer Channel. A modular subsystem, the 8410 can be tailored to the individual needs of each installation. The basic disc configuration consists of a Disc File Control and a Dual Disc File Master. The Disc File Control is housed in the Central Processing Unit; it provides a single eight-bit data path to the Multiplexer Channel and a single eight-bit data path to the 8410 Disc File. The Dual Disc File Master is a free-standing unit which contains two handlers, a nonaddressable core memory buffer and the Fastband Search feature. Figure 1–1 illustrates the minimum configuration for a 9200/9300 Disc System.

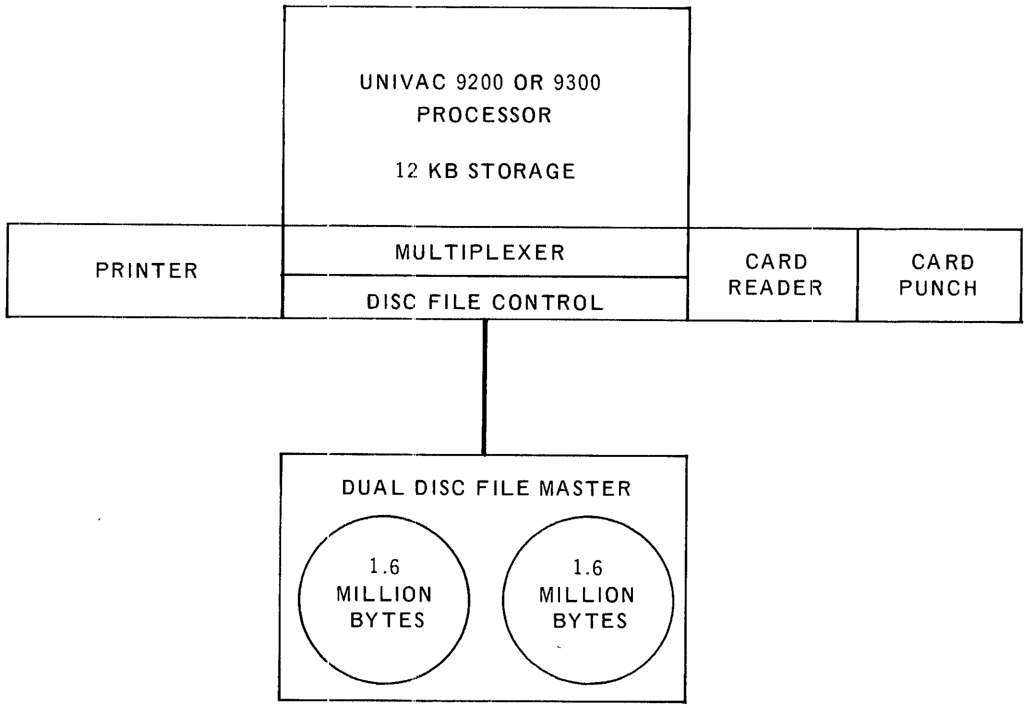

Figure 1–1. 9200/9300 Disc System: Minimum Configuration

As storage requirements increase, up to six slave handlers can be added, one at a time, to the basic configuration. The slave handlers are housed in Dual Disc File Slave cabinets; each cabinet can contain two handlers.

Each disc handler provides 1.6 million bytes of online storage, for a total subsystem capacity ranging from 3.2 million bytes to 12.8 million bytes. Figure 1–2 shows the expanded disc configuration.

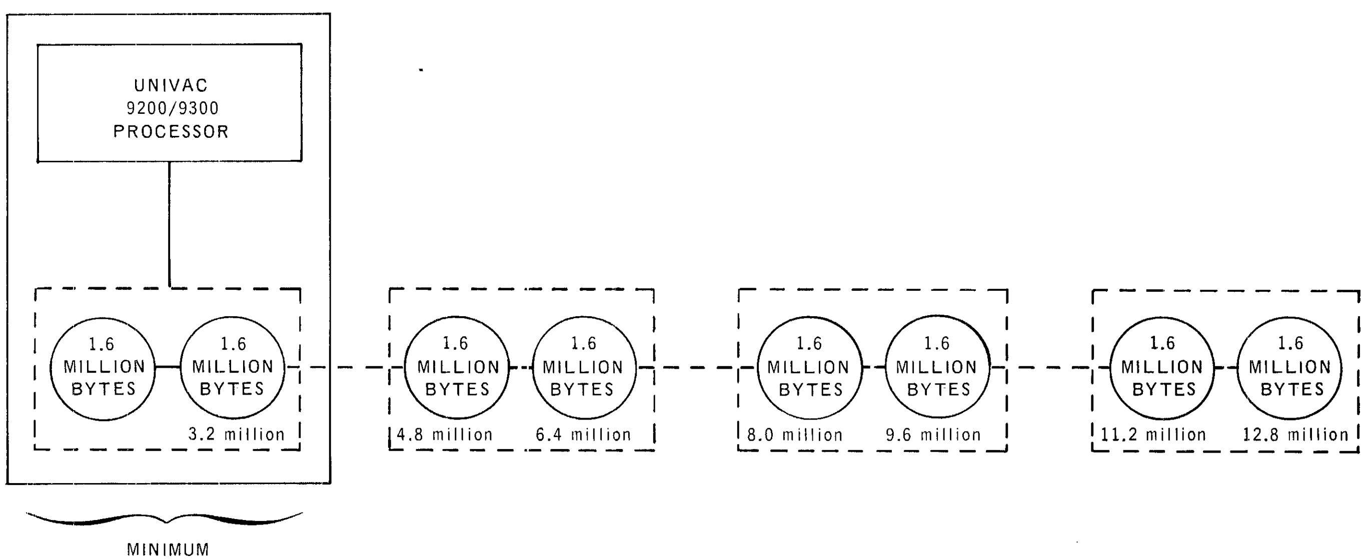

Figure 1–2. 9200/9300 Disc System: Expanded Configuration

USES OF DIRECT ACCESS STORAGE

The major uses of disc storage are as follows:

Data Files: This is the most common use of disc storage. Obvious candidates for direct access are files which (1) are used frequently and have a high percentage of inactive items during each individual use, or (2) are affected by transactions in unsorted form, or (3) are used to produce reports in different sequences.

Temporary Storage: Disc storage is often used as an extension of main storage for information such as tables, form headings, descriptive information, formulas, and summary totals. Use of disc storage in such cases may greatly enhance systems design.

Program File Storage: The UNIVAC 9200/9300 System will accept object programs stored on cards, magnetic tape (9300 only), or disc. Transition from one program to the next may be accomplished by calling the next program from the library in online disc storage.

Communications Buffering: When a computer is used for communications it is often desirable to set aside incoming messages for later handling –– for instance, to be transmitted to a different location. If a number of communications lines are simultaneously active, messages may be transmitted from and received into auxiliary storage, so that main storage can be devoted to the mechanics of communication. Actual processing of the communicated records may be done at another time.

Peripheral Buffering: Use of disc files as peripheral buffers frequently improves the efficiency of computer operation. A long computer run, which might be seriously slowed by a high output card punching requirement, can stack the output card images on disc files during the run for punching at a more convenient time. If sufficient memory is available for use of the Concurrent Operating System, the card punching can proceed concurrently with a later program which does not require the punch. Print line images can also be accumulated on discs to speed a run whose speed would be limited by the printer, or to accumulate a related report for later printing.

TYPES OF TRANSACTIONS SUITED TO DIRECT ACCESS PROCESSING

Direct access storage permits each record to be accessed directly, regardless of its file position. Because of this unique capability, disc files are particularly desirable in the following four types of transactions:

1. *When a small number of transactions must be processed against a large file:* Direct access techniques do not require examination of all the records in the file to determine whether they are active. For instance, a customer order involving only four items can be processed against a complete stock inventory without progressing item by item through the entire inventory list.

2. *When each transaction affects multiple files which are not normally kept in the same sequence:* Processing a single job card once can update an equipment usage file, a departmental cost file, and the production schedule, if these files are online in direct access storage.

3. *When transactions received in unsequenced order must be handled quickly:* Critical time requirements often will not allow card or tape sorting of transactions to precede processing.

4. *When an external transaction can trigger references to a number of items in a file:* For example, in exploding a Bill of Materials, an order for one end product might cause reference to several assemblies, which call for examination of parts records. Without direct access, each level of reference would have to produce output records for sorting before applying to the next level file.

THE DISC CARTRIDGE

General Description

The storage medium for the 8410 DAS is an aluminum disc, 14 inches in diameter, plated with nickel-cobalt. Each disc is enclosed in a plastic cartridge to protect it against accidental damage. The combined weight of the disc and cartridge is approximately 7.5 pounds.

Storage Design

The disc surface is divided into 100 logical tracks, each comprised of 100 sectors. Each sector has a storage capacity of 160 eight-bit bytes, and can contain one or more records or a partial record. When a disc is mounted on the handler, the 1.6 million bytes of data located on the underside of the disc are online, while those located on the upper surface are offline. The disc storage design is illustrated in Figure 1–3.

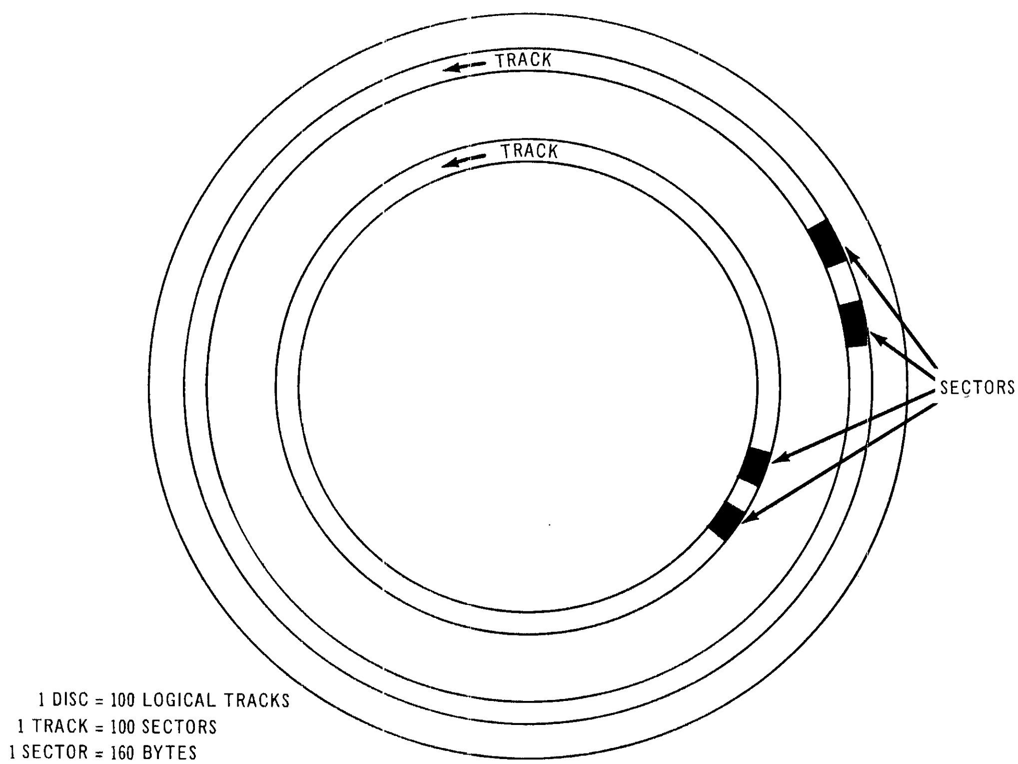

Figure 1–3. Disc Storage Design

For all practical purposes, a track can be thought of as one complete circle on the disc. In reality, however, the disc is divided into two zones, each containing 100 physical tracks. Each logical track consists of two parts: Sectors 00 through 54 are in Zone 1, and Sectors 55 through 99 are in Zone 2. Although each logical track occupies two physical tracks, the control circuitry causes the system to function as though the two physical tracks were continuous. Thus, for purposes of designing file layout, programming, and estimating time, the storage design should be regarded as 100 tracks of 100 sectors each.

Address Structure

As illustrated in Figure 1–4, each of the 10,000 sectors on the disc is identified by a unique four-digit address ranging from 0000 to 9999. The first two digits of the address identify the logical track and are symbolically referred to as TT. The last two digits identify the sector within the track and are symbolically referred to as SS.

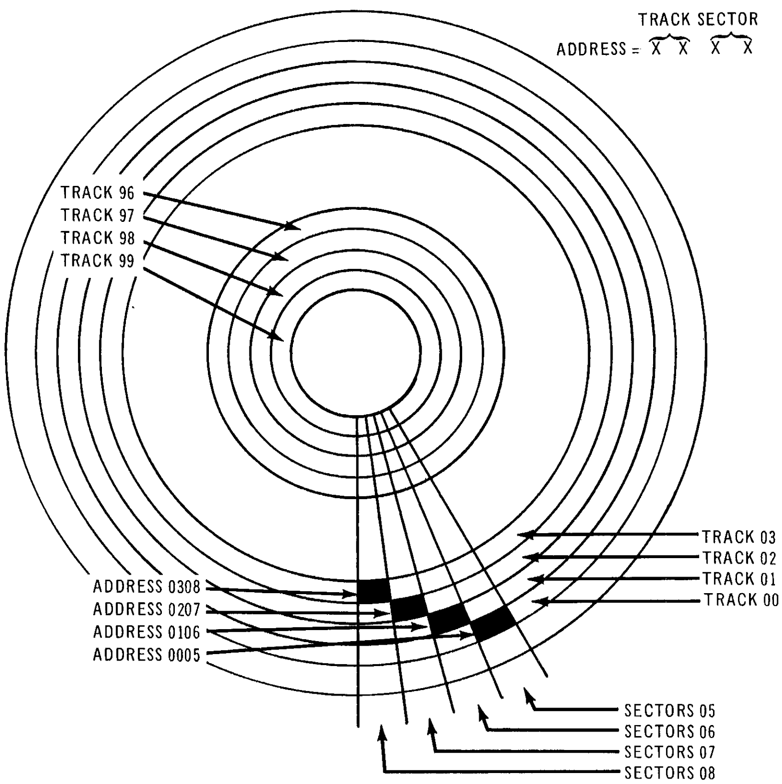

Figure 1–4. Disc Address Structure

The entire address of the form TTSS is recorded on the disc by the Disc Prep program (a Univac utility program). This process is called "prepping" the disc.

Since up to eight discs can be online at the same time, a fifth digit (0 to 7), symbolically referred to as U, is added to the left of the track-and-sector address to identify the disc unit. The complete address, therefore, is a five-digit number of the form UTTSS, ranging from 00000 to 79999, capable of uniquely identifying each of the possible 80,000 sectors online to the 9200/9300 System.

Two spare sectors are provided on each data track to allow for possible damage to the primary sectors. If a sector should become unusable, the disc is reprepped to maintain the full contiguous address structure.

Fastband

In addition to the 100 data tracks described above, each disc surface contains a fast access track called Fastband. The Fastband track contains 50 sectors of 160 bytes each, thus providing an additional 8,000 bytes of storage. It also contains several spare sectors for reprepping needs.

OPERATION OF THE 8410 DAS

Track Access

Data is read or written by two sets of magnetic read/write heads located in fixed positions on a movable access arm. The first read/write head is assigned to the outer portion of the disc, the second one to the inner portion. When a track is accessed, the access arm moves radially across the disc, both sets of read/write heads moving in tandem, until it reaches the correct track. The access arm can stop at any one of 100 discrete positions. Since the two read/write heads are placed exactly 100 tracks apart, one complete logical track is accessible at each position. When the first read/write head is positioned on Sectors 00 to 54 of a logical track, the second is automatically in place on Sectors 55 to 99 of the same logical track (see Figure 1–5). The section of the read/write head is made by circuitry.

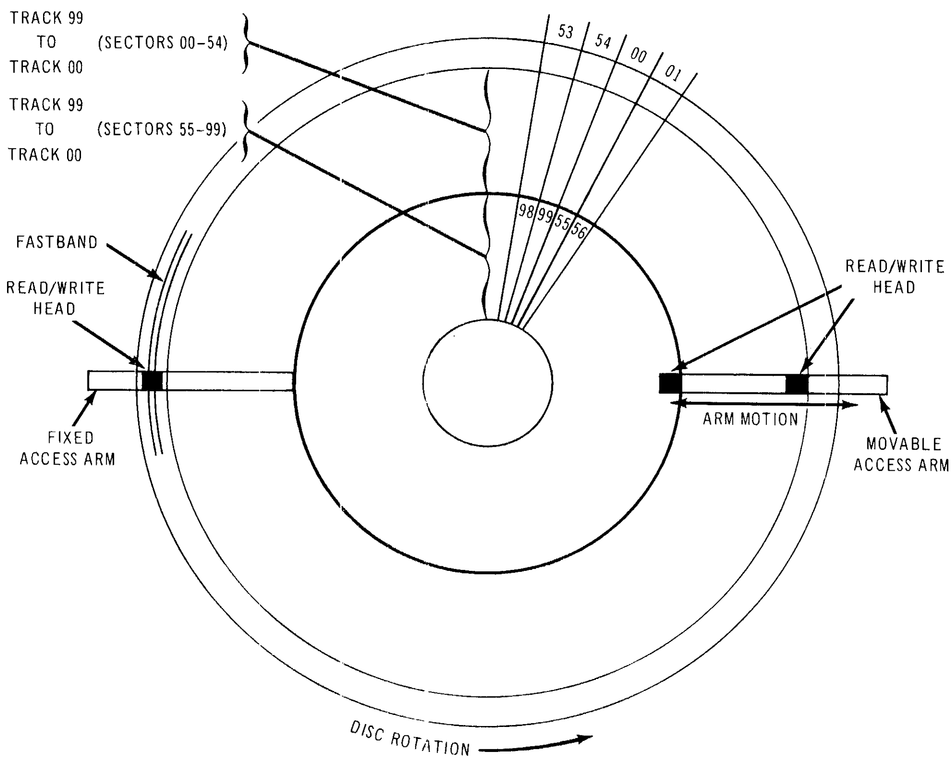

Figure 1–5. Track Access

The Fastband is accessed by a permanently assigned read/write head. Since that head is fixed, there is no arm travel time involved in selecting a sector on the Fastband. (Disc rotation time, however, is still a factor.)

Information Transfer

Information is exchanged between the Central Processing Unit and the 8410 Disc File through the 8410 Disc File Control and the 9200/9300 Multiplexer Channel, as shown in Figure 1–6.

Data is transferred one byte at a time (serially); all bits of a byte are transferred into memory at the same time (parallel).

The 8410 is a buffered subsystem; thus, the 9200/9300 System can continue processing while a record is sought or while information is moved between buffer and disc.

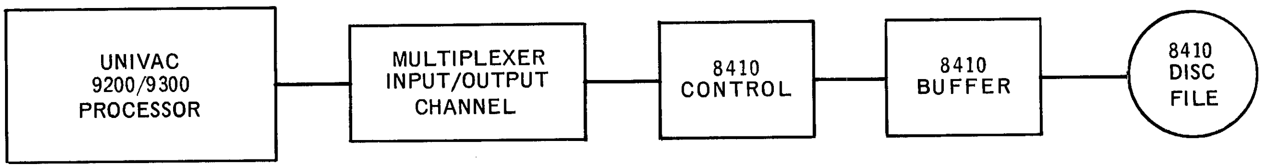

Figure 1–6. Data Transfer

Data Validation

Assurance of correct data and address recording is provided by a combination of parity, bit count, and phase error checking. As entries are made, check bytes are developed and stored in each sector. When data is read from a sector, the check byte is recomputed. The new check byte is then compared to the one stored in the sector when the data was written.

If the "Write and Check" command is used, data written on the disc is verified immediately; it is read back on the next disc revolution. At that time, the check bytes are recomputed and compared to the check bytes in the sector; in addition, each character in the sector is compared to the character received from main storage.

FILE ORGANIZATION

Sequential File Organization

A sequential file is one in which records are arranged in sequence, such as in ascending or descending order by key data fields. This is the same type of file organization used in a card or magnetic tape file.

In a sequential file, records are usually accessed consecutively, starting with the first record and proceeding through each succeeding one. Transaction records must therefore be sorted in record key sequence before processing. The file must be planned carefully, since additions can only be made by re-creating the file.

Sequential file organization is intended primarily for files in which most or all of the records are processed each time the file is used. Because head movement is minimal and there is no index to search, sequential accessing is the fastest method of handling files with a large percentage of active records. Another advantage of sequential file organization is that only one volume (disc) of a multivolume file is required online at a time.

There are many variations of sequential file organization. A few of the more common ones are described below.

- *Expanded Sequential:* This is similar to sequential file organization except that gaps are left during file construction to allow for anticipated additions to the file. In an expanded sequential file, a one-byte activity field is set aside in each record location to denote whether that location contains an active record or is a gap area.

  Figure 1–7 shows an expanded sequential file having a gap between records 00100 and 00168, and the same file when the gap has been filled by record 00101.

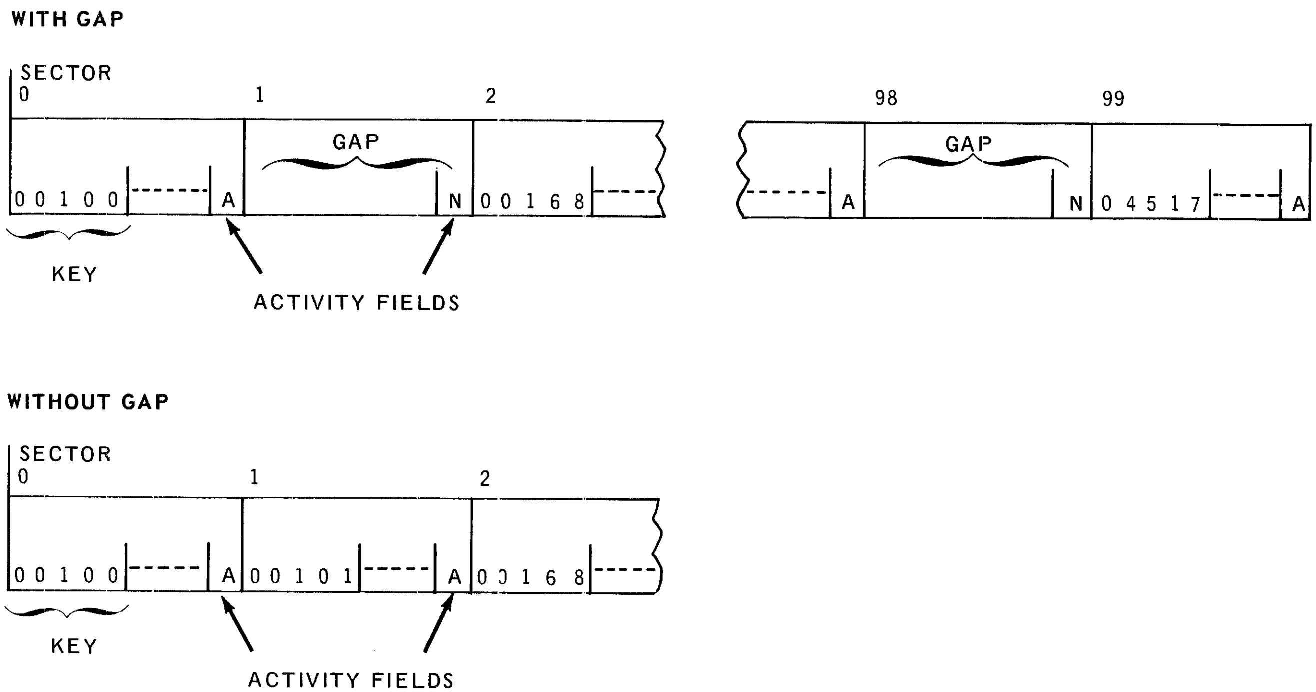

Figure 1–7. Expanded Sequential File Organization

- *Partitioned Sequential:* This type of file is partitioned into a number of sequential subfiles called members. Each member is processed sequentially. However, the file also contains a table that lists the record key of the first record in every member. As a result, processing of the file can begin with any member desired, and entire members may be bypassed during processing.

■ *Chained Sequential:* Here the records are arranged in sequence, but an overflow area is set aside for additions. The file is processed sequentially according to record key. Each record in the file contains a chain field in addition to a record key. The chain field is usually blank or zero, indicating that the record in the adjacent position in the file is the next one in the record key sequence.

When a new record is added to the file, it is put in the overflow area and chained to its proper place in the main area as follows: The address of the new record in overflow, Record 4, is put in the chain field of the record having the next lower record key, Record 3. In turn, the address of the record having the next higher record key, Record 5, is put in the chain field of the new record. When the file is accessed, the processor will read Record 3 from the prime area, Record 4 from overflow, then Record 5 back in the prime area.

An example of chained sequential file organization is given in Figure 1–8.

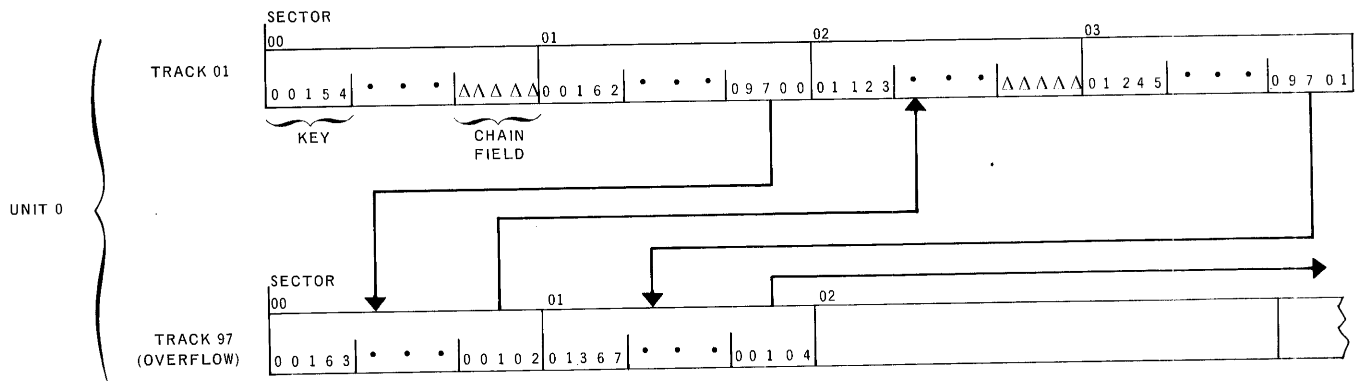

Figure 1–8. Chained Sequential File Organization

Indexed Sequential File Organization

FILE LAYOUT

An indexed sequential file is actually a variation of a sequential file. But, because it is one of the most frequently used methods of file organization, it is discussed here separately.

A file organized by the indexed sequential method has three parts:

First, a *prime area,* which contains all the original records of the file, arranged in sequential order according to their record key. Additional records may be added to the prime area if their record key is higher than that of any record previously in the file.

Secondly, an *overflow area,* which is used when new records are added to the file, if the prime area is filled or if the new record has a lower key value than the highest record in the prime area. All records placed in the overflow area are chained to their proper place in the prime area, much as footnotes in a book would have a reference number in the text. The overflow area thus allows new records to be added to the file without completely rewriting or reorganizing the file.

Thirdly, an *index to the data records,* which allows the file to be accessed directly as well as sequentially.

TYPES OF INDEXES

There are several different types of indexes. In one type, there is an entry in the index for every data record; each index entry contains a record key and the disc address of that record. Access to records is made by reference to the index. For efficiency, the index should always be in sequential order.

Another type of index is split in two parts, rough and fine. The rough index references the fine indexes; it indicates the range of record keys controlled by a particular fine index and gives the track address of each fine index. The fine index gives the sector address of the desired record. The rough index may be stored in the computer main storage or in the disc device. The fine indexes are usually stored on the same disc as the record they reference.

There are many variations to the rough and fine index technique. For example, indexes could be established in such a way that the first entry in the rough index equals Track 1 in the disc device, the second entry equals Track 2, and so on. Similarly, in the fine indexes, the first entry would equal Sector 1, the second entry Sector 2, and so forth. This would be a rough and fine index technique based on the principle of relative position.

A third type of index is the activity frequency index. Instead of being arranged sequentially, the index entries are put in order according to frequency of access, with the reference to the most active record appearing first. The file itself remains in sequential order according to record key. The index is searched sequentially – starting with the first entry and proceeding through each successive entry. The number of comparisons required is minimized because the entries which are referenced often are at the beginning of the index. When a hit is made, an activity counter is incremented. Periodically, the index is rearranged and the counter cleared.

ACCESSING THE INDEXED SEQUENTIAL FILE

An indexed sequential file can be accessed sequentially or directly. If it is accessed sequentially, the index is ignored, but the records in the overflow area are accessed in their proper record key sequence. If the file is accessed directly, the program first searches the index, then goes directly to the desired record or, if rough and fine indexes are used, to the appropriate fine index then to the desired record.

METHODS OF SEARCHING THE INDEX

There are several ways of searching an index. Some of the most common ones are:

1. *Sequential Search:* This is the most simple and straightforward method of index lookup. Starting at some index location (usually the first) the record key of each input record is compared to the key of each index entry until a match is found. When this method is used, the index entries could be in sequence or could be in any other order desired. This technique works well if the file is a small one.

2. *Merge Search:* This is an extension of sequential search. It requires that both the index and the input data be sorted in the same sequence on the key involved. The record key of the first data record is compared to the first index key. If they do not match, the record key of the data record is compared sequentially to the index keys until an equal or greater index key is found. If a greater key is found, there is no matching index key, and the next data record is engaged to be compared to the index keys. There is no need to return to the beginning of the index because it is arranged in the same sequence as the data records. If a match does occur between a data record and an index key, the data item is processed and the search for the next match begins. A big advantage of the merge search is that all the index entries need not be in main storage at once. They can be stored in a disc device. This is especially important when indexes are too large for main storage or when main storage requirements are critical.

3. *Binary Search:* This method requires that the index be sequentially organized. The record key is compared for equality and magnitude to the median index entry. When the key and index are unequal, the next comparison is made to the median index entry in the half of the entries indicated by the magnitude tests results. Binary search is quite fast for many applications, because the number of comparisons required to find an index item is much less than the comparisons required in a sequential search. (Note that binary search can also be used to search a sequential file.)

4. *Scan Search:* This method, also known as the computed position method, can be used to find an index entry when the data records in the file have a continuously numbered series of record keys, with no gaps. The address of the index entry which references a desired data record is found as follows:

   (a) $\dfrac{\text{Record key of desired record} - \text{Lowest record key in index}}{\text{No. of records per index entry}}$

   = Quotient A. Discard remainder.

   (b) Quotient A $\times$ No. of bytes per index entry = Quotient B

   (c) Tag of first index entry + Quotient B = Tag of desired index entry.

   To illustrate, suppose the file address of Record 1013 is desired. Let's assume that the index to that file starts with Record 1000 at the location of TAG and that each index entry has eight bytes and references five data records in the file. The index entry that references Record 1013 is found as follows:

   (a) $\dfrac{1013 - 1000}{5} = 2.$ Remainder of 3 is discarded.

   (b) $2 \times 8 = 16$

   (c) TAG + 16 = location of index entry that references Record 1013.

SPECIAL CONSIDERATIONS FOR THE INDEXED SEQUENTIAL FILE

The indexed sequential file offers several important advantages: Storage space is used efficiently, because records can be tightly packed regardless of gaps in the record key series; the file can be accessed directly as well as sequentially; additions can be made without recopying the entire file; and processing may start at any point in the file.

On the other hand, all volumes of the file must be online at the same time. Furthermore, access time of the file is increased because an index must be searched before a data record can be read or written.

Random File Organization

GENERAL

In a random file, records are stored according to a logical or mathematical relationship between the record key and the file address. The file is accessed directly, without proceeding sequentially through an ordered set and without referencing an index.

A random file requires no index, no overflow area, and no file support routine for reorganizing the file. It does require that all volumes of the file be online simultaneously. Random organization provides the fastest way of obtaining data, since there is only one access per record. Storage utilization is usually not as efficient as for sequential or indexed sequential files; but the file can be tightly packed if the series of record keys used to calculate the address contains few gaps.

While sequential and indexed sequential files are normally used when a large percentage of the file is to be accessed, random files are chosen primarily when a small percentage (40% or less) of the file will be referenced and rapid access is desired. Random organization is also highly efficient in real-time applications and in applications where the sequence of reference to the file is unpredictable.

ADDRESSING THROUGH A DIRECT RELATIONSHIP TO RECORD KEYS

The simplest way of achieving random organization is to use a record field, with little or no manipulation, as the disc storage address. For example, the record of inventory item number 12345 would be put in disc storage address 12345 (Drive 1, Track 23, Sector 45). Record fields commonly used include employee number, part number, invoice number, account number, and policy number.

In some cases, the record key may be modified somewhat before it is used. For instance, the file address might be found by using the following formulas:

(a) If there is only one record per sector:

$$\frac{\text{Record key} - \text{Lowest record key in file}}{\text{No. of sectors per track}} = \begin{array}{c}\text{Track number (quotient)} \\ + \\ \text{Sector number (remainder)}\end{array}$$

(The lowest record in the file is subtracted from the key of the record to be addressed, to obtain an address relative to 0, since in computer logic the file begins at 0.) Thus, the disc address of Record 2000, where the lowest record key in the file is 0001, would be:

$$\frac{(2000 - 0001)}{100} = \frac{1999}{100} = 19 + \text{remainder of } 99$$

or *Track 19, Sector 99.*

(b) If there are *n* records per sector:

$$\frac{\text{Record key} - \text{Lowest record key in file}}{\text{No. of records/sector}} = \text{Quotient A} + \text{Record number in sector (remainder)}$$

$$\frac{\text{Quotient A}}{\text{No. of sectors/track}} = \text{Track number (quotient)} + \text{Sector number (remainder)}$$

Assuming four records per sector, the disc address of Record 2000, where the lowest key in the file is 501, would be:

$$\frac{(2000 - 501)}{4} = \frac{1499}{4} = 374 \text{ with a remainder of } 3$$

$$\frac{374}{100} = 3 \text{ with a remainder of } 74$$

or *Track 3, Sector 74, third record in sector.*

The direct relationship between record key and file position is often used for relatively small data files. It provides the fastest way of accessing records, since few calculations are required and there is no index to search. It also minimizes the amount of main storage needed for programming space, since little coding is required to interpret the file address of a desired record.

This method of filing, however, works only if the identifier system matches the address structure of the 8410 Disc Subsystem, and is efficient only if most of the numbers in the range are used so that there is little wasted space. Furthermore, deletions may leave gaps which cannot readily be filled. Nor can additions be easily accommodated, unless they are already covered by the existing series of numbers, and storage areas have been allocated to them. New records must be assigned available record keys; for that purpose, a manual record of available numbers in the series must be maintained.

ADDRESSING THROUGH RANDOMIZING TECHNIQUES

The alternate method of achieving random organization is to convert the record key to a track and sector number through a mathematical operation. The mathematical formula used to transform the record key into a valid file position is known as a "randomizing" formula.

Depending on the randomizing technique employed, storage utilization can reach between 80% and 90% efficiency. Generally speaking, a technique that achieves high disc storage utilization generates more duplicate file positions, or synonyms, and thus increases access time. The randomizing technique chosen, therefore, depends to some extent on the relative importance of disc storage utilization versus access time.

Once a randomizing technique has been selected for possible use, it should be evaluated with a sample selection of actual record keys. This evaluation provides information on the efficiency of disc storage utilization, the frequency and distribution of synonyms, the processing time required for calculation, and the evenness with which the generated addresses are distributed. The results enable the user to select the technique most suited to his particular requirements and data pattern.

COMMONLY USED RANDOMIZING TECHNIQUES

A few of the most commonly used randomizing techniques which are economical in processing time and core memory requirements are outlined in the following paragraphs. There are many possible variations of these methods, and many other methods not discussed here.

1. *Prime Number Division:* A divisor is selected, equal to the number of record spaces allocated for the file. This divisor should be a prime number (that is, evenly divisible only by itself or one). If the number of record spaces is not a prime number, the first number above it that ends in 1, 3, 7, or 9 is used. The record number, or relative position of the desired record within the file, is then calculated as follows: The record key of the current data record is divided by the number selected. The quotient is discarded, and the remainder becomes the record number.

   Thus,

   $$\frac{\text{Record key}}{\text{Prime number}} = \text{Quotient (discard)} + \text{Remainder (record number)}$$

   This relative record number is then supplied to the Random Processor (IOCS), which in turn will compute the desired track and sector address.

2. *Squaring, Enfolding, and Extracting:* In this randomizing technique, the record key is squared; the resulting number is split in the middle and the two parts added together; and the middle digit(s) of that sum are extracted to become the record address. (Other rules for determining the digits to be extracted are also acceptable.) To illustrate:

   Key: 45678

   Square: $(45678)^2 = 2086479684$

   Enfold: 20864 + 79684 = 100548

   Extract: (middle four digits) = 0054 = Record Address

   This method of randomization requires extensive manipulation and usually results in many synonyms.

3. *Radix Conversion:* Radix conversion is applied to strictly numeric keys. A radix is the arbitrary base of a system of numbers. For example, the decimal number system has a radix of 10; the binary number system has a basis of two. In the radix conversion method of randomization, each decimal digit of the record key is treated as if it were a radix 11 digit. Each digit is then "converted" to a radix 10 digit, and the high order digits are discarded, to obtain a field of the desired length. Radix conversion is performed as follows:

   Key: 301,283

   Radix 11: $(3 \cdot 11^5) + (0 \cdot 11^4) + (1 \cdot 11^3) + (2 \cdot 11^2)$

   $+ (8 \cdot 11^1) + (3 \cdot 11^0)$

   $= 483{,}153 + 0 + 1{,}331 + 242 + 88 + 3$

   $= 484{,}817$

   To obtain the relative file position, various options are acceptable; one might be to truncate the three leftmost digits in this example making 817 the relative record position. The variety of ways in which a record key can be manipulated and still produce the same result is an advantage of radix conversion. This method of randomization, however, tends to produce more synonyms than does prime number division.

4. *Nonnumeric Control Numbers:* When record keys are comprised of purely alphabetic or special characters, or of alphanumeric characters, it may be useful to treat the number as a binary number and to perform binary arithmetic on it. This has the advantage of allowing zone bits to be retained, so that unnecessary synonyms are avoided. When alphabetic record keys are converted to file addresses, zone bits generally should not be dropped, since that immediately gives rise to three possible sets of synonyms: (G,H,I) (P,Q,R) and (X,Y,Z). However, if the record keys are largely numeric, with only a few nonnumeric characters scattered throughout the series, zone suppression may be acceptable. Record addresses could then be calculated by using decimal arithmetic.

FREQUENCY ANALYSIS

As a preliminary step in selecting a randomizing technique, it is often desirable to perform a frequency analysis on the key field of the file. The frequency analysis gives a pattern of distribution for each key position; that pattern shows which positions are best to use when truncating or extracting addresses from the record keys, and often indicates whether a particular type of randomizing technique is suitable to that file.

To begin the analysis, the record keys of each record in the file are examined to determine how often any digit appears in any one position of the record key. For example, if there are 16,050 records in the file, the fifth key position might contain a 0 in 5,168 record keys; digits 4 through 9 might not occur in the fifth position in any record key. This count gives the actual distribution of digits occurring in each key position.

If the distribution were perfectly even, each of the ten digits would occur one tenth of the time. With 16,050 records, therefore, each digit ideally would occur 1,605 times in any one key position.

The deviation from the ideal distribution is determined by taking the difference between the actual number of times a digit occurs in the key position and the ideal number of times it should occur (in this case, 1,605). Thus, if 0 actually occurs in the fifth key position of 5,168 different records, the deviation would be 5,168 minus 1,605, or 3,563. The deviation is calculated for all ten digits, then all the deviations are summed to find the total deviation for that key position. The results for a six-digit record key of a file containing 16,050 records are shown in Figure 1–9. The lower the total deviations, the more even the distribution, consequently, the lower the access time for any given record.

	Key Position 1		Key Position 2		Key Position 3		Key Position 4		Key Position 5		Key Position 6	
Digit	No. of Occurrences	Deviation	No. of Occurrences	Deviation	No. of Occurrences	Deviation	No. of Occurrences	Deviation	No. of Occurrences	Deviation	No. of Occurrences	Deviation
0	16,050	14,445	0	1,605	0	1,605	1,852	247	5,168	3,563	1,807	202
1	0	1,605	0	1,605	4,408	2,803	3,147	1,542	5,638	4,033	2,120	515
2	0	1,605	2,198	593	3,792	2,187	1,174	431	4,958	3,353	1,745	140
3	0	1,605	581	1,024	2,236	631	2,724	1,119	286	1,319	1,689	84
4	0	1,605	1,195	410	2,459	854	1,194	411	0	1,605	1,378	227
5	0	1,605	12,076	10,471	3,155	1,550	1,267	338	0	1,605	1,647	42
6	0	1,605	0	1,605	0	1,605	1,243	362	0	1,605	1,560	45
7	0	1,605	0	1,605	0	1,605	1,228	377	0	1,605	1,329	276
8	0	1,605	0	1,605	0	1,605	1,227	378	0	1,605	1,415	190
9	0	1,605	0	1,605	0	1,605	994	611	0	1,605	1,360	245
Total Deviation		28,890		22,128		16,050		5,816		21,898		1,966

Figure 1–9. Example of Deviation Calculation for Six-Digit Key

HANDLING DUPLICATE FILE POSITIONS (SYNONYMS)

Ideally, the randomizing formula transforms every record key in the file into a unique file address. In reality, though, duplicate positions are occasionally assigned. A good randomizing formula minimizes them, but usually cannot eliminate them entirely. These duplicate positions are known as synonyms. Two of the most common ways of handling synonyms are:

- *Random Chained Format:* When synonyms are created in a random chained file, one record is stored at the generated position. Other records are linked to this record through a chain field. They are stored in an overflow area or in unused areas of the file.

- *Random Sequential Format:* This method handles synonyms in a manner similar to the random chained format except that synonyms are stored sequentially at the first available position following the generated position. The file is loaded in two passes. On the first pass, all records are loaded to generated positions. On the second pass, all synonyms are loaded in the unused areas of the file.

RELATIVE ADDRESS VARIATION OF RANDOM ORGANIZATION

A variation of random file organization is the "relative address" method of organization. In this method, the address found by the randomizing formula is a file position relative to the first address allocated to the file rather than an actual disc address. Thus, a relative address of 1000 would indicate the 1000th record of the designated file area. An additional calculation is then required to obtain the actual track and sector address for the record. Synonyms are handled in the same way as for other random files.

Choosing a Method of File Organization

Time, space, and cost are the ultimate factors which determine the best method of organization for a particular file. Knowlege of those factors, however, depends on thorough knowledge of the 9200/9300 System, the 8410 DAS, the file being considered, the software supporting the different organization methods, and the use of the file.

The following questions illustrate some of the details that must be considered before a method of file organization is chosen:

- What volumes of data are involved?
- How frequent and how large are the peak volumes?
- What is the reference frequency to data and how does this vary, both among records and among particular fields within records?
- Which processing method and addressing techniques will be used for access to the various files?
- Are disc devices to be the sole storage medium, or will certain data be stored on magnetic tape or cards?
- Are the existing record keys usable, and if not, what is the cost of conversion or modification?
- What expansion or modification of file structure is foreseen?
- What are the inquiry requirements – what data is to be printed out?
- What are the total reporting requirements and what is the desired sequence for report generation?
- Will associated records be referenced individually as required or will they be consolidated?
- Will a particular file be processed in more than one processing mode?
- What will be the extent and complexity of file maintenance requirements?
- Is the total systems approach envisaged, or will each application be processed individually?
- Is the file primarily static or dynamic?

- How many different types of master records and transaction records are used in the application?
- What are the response time requirements?

The answers to these questions can only be obtained after careful study of the available system and the specific application involved.

THE DISC DISPATCHER

The Disc Dispatcher is the supervisory subroutine which provides the interface between a problem program and the 8410 DAS. Working from a "request packet" provided by the problem program, the Disc Dispatcher issues disc I/O commands, queues requests, handles error recoveries and performs the other functions of an I/O routine. The request packet is a table of parameters. It can be supplied directly by the problem program, or can be generated automatically by the problem program through the Input Output Control Systems (IOCS) software package.

INPUT/OUTPUT CONTROL SYSTEMS (IOCS)

The Input/Output Control Systems (IOCS) is a generalized software package which controls the I/O functions of the 8410 DAS. The IOCS performs all the "housekeeping" chores of file management, leaving the programmer free to work at the logical level.

The 8410 IOCS is a macro-generated system. A macro is a single instruction from the programmer which, when processed, generates several predefined instructions or subroutines. Macros are specialized for particular programs by parameter statements. For example, for a file definition macro, parameter statements such as OUTPUT = YES or OUTPUT = NO tell the macro generator whether to include or exclude the instructions necessary to write output records on a disc file. File specification through macro parameters thus provides a compact, efficient I/O routine to the program.

To provide for the three major types of file organization, there are three IOCS packages: The Sequential, Indexed Sequential, and Random Processors. Each IOCS Processor consists of three parts:

1. An I/O file control routine
2. A file definition section defined by declarative macros
3. An imperative macro processing section

The I/O file control routine checks and produces file labels, end-of-volume and end-of-file notification, automatic unit swap, and error processing and display; and it interfaces with the Disc Dispatcher. The I/O file control routine for a particular IOCS Processor is common to all files of the related file organization, and is required only once in the user's program. For example, the Sequential Processor's I/O routine would be included only once to process several sequential files used by a particular program.

The file definition declarative macro and its parameter statements are used to define and describe in detail the file to be processed. A separate declarative macro is required for each file.

Imperative macros are instructions such as GET and PUT, which tell the macro generator to supply the instructions to move data between the 8410 DAS and the 9200/9300 System's internal memory.

The three IOCS Processors are discussed individually in the paragraphs below.

The Sequential Processor

The Sequential Processor is used to create and process sequentially organized files. Through declarative macro parameters, the programmer specializes the proper IOCS subroutine for the file to be processed.

One of the factors specified through the declarative macro is whether records are to be blocked (more than one record per sector) or unblocked (one record per sector). Sector size is 160 bytes. Records between 1 and 160 bytes can be written in the unblocked mode; any unused portion of the sector will be cleared to zeros. For blocked records, the IOCS Processor computes automatically the maximum number of records which can be contained in a sector. This number is called the "blocking factor." Blocked records must have a record size equal to or less than 80 bytes. If the record size is greater than 80 bytes and blocking is requested, the blocking factor will be 1 and no blocking will take place.

The Sequential Processor responds to four imperative macros:

1. OPEN, which must be issued before a file can be accessed.
2. GET, which causes a data record to be read into the problem program work area.
3. PUT, which causes a data record to be moved from the problem program work area to the disc IOCS area.
4. CLOSE, which indicates that the problem program is finished processing a file. If additional processing of a file is required after the file is closed, another OPEN macro must be issued for that file.

The Sequential Processor handles four types of files:

1. *Input Files*. These are single or multivolume files which are read from the disc. Records may be blocked or unblocked; maximum record size is 160 bytes.

   When the Sequential Processor detects the end-of-file condition, it automatically transfers control to the end-of-file routine for that file. The address (label) of the end-of-file routine is supplied by the programmer through a declarative macro parameter statement.

2. *Input/Update Files*. These are identical to Input files, with the added feature that a record just read may be processed and rewritten. The record will be written to the same sector, or for blocked files to the same position in the block, from which it was read.

   When an Input/Update file is processed, any number of GET commands may be issued without an intervening PUT; however, two PUT's must not be issued without an intervening GET.

3. *Output Files*. These are files being written to the disc. Like Input files, they may be single or multivolume; records may be blocked or unblocked, and the maximum record size is 160 bytes.

Each sector written is automatically read back and compared to the data still resident in the 8410 DAS buffer. This verification procedure requires one extra disc revolution for each record written. The programmer may, if he wishes, exercise the option of writing without check.

4. *Work Files.* These provide a scratch area on disc, which may be used as input, output, or both during the execution of a problem program. The Work file is a single volume unblocked file with a maximum record size of 160 bytes. Like any other file, it must be defined and disc storage allocated for it.

Once a Work file has been opened for processing, the first GET or PUT macro sets the direction of the file until a CLOSE instruction is issued. For example, if the first instruction following the OPEN was a GET, then the Work file would be set for input and only GET requests would be accepted. To change the direction of a Work file, the user must close the file, reopen it, and issue either a PUT or GET, as the case may be. Once a Work file is set for either input or output, it is treated like any other Input or Output file.

The Indexed Sequential Processor

The Indexed Sequential Processor, as its name implies, is used to create and process indexed sequential files. These files may be less than one volume or as large as eight volumes; records may be blocked or unblocked. When the programmer writes the file, he must define a prime area and an overflow area. When the file is loaded, the Indexed Sequential Processor stores the records densely in ascending order in the prime area. At the same time, it creates an indexing system, consisting of two parts:

1. A track index, residing on the Fastband. This index identifies the highest record contained by each logical track assigned to the prime area. Because of the low access time of the Fastband, the track index provides fast reference to any given track of the file.

2. A sector index, which occupies the first positions of each sector. It consists of two fields: a sector key field, which contains the record key of the highest record associated with that sector, and a 5-byte link field, which provides a reference or chain to the overflow area. The chain identifies a logical extension of the sector.

At the time of loading, the overflow area of the file is blank and the link fields in each sector contain no linking information. When records are added to the file, the overflow area is used as follows: Suppose a file contains unblocked Records 4 and 6 in the prime area. If Record 5 is added to the file, Record 6 is moved to the overflow area; Record 5 is placed in the location vacated by Record 6; and the overflow address of Record 6 is put in the link field of Record 5.

If the file has blocked records, the last record in the sector is the one moved to overflow. For example, suppose a sector contains Records 4, 6, 8, and 9. If Record 5 were added to that file, Record 9 would be moved to overflow, Records 6 and 8 would be moved over one record length, and Record 5 would be inserted in the gap left between Records 4 and 6. The overflow address of Record 9 would be placed in the link field at the beginning of the prime area sector. With this filing technique, additions can be made without recreating the file, and entries in the track and sector indexes need only be updated when records are added to the end of the file when the prime area is full.

Files loaded by the Indexed Sequential Processor can be accessed sequentially or directly. When a file is accessed sequentially, four imperative macros are used: SETL, GET, PUT, and WRITE NEWKEY. The programmer may define the point at which he wants to start processing by using the SETL (set lower limit) macro immediately after the file is opened. If the SETL macro is not used, processing starts with the first record in the file. Note that SETL may be issued any number of times after the file is opened. All records, including linked ones, are accessed in record key sequence. The GET macro transfers a record from the file to the user's work area. The PUT macro will rewrite an updated record to its same position in the file. The WRITE NEWKEY macro indicates to the Indexed Sequential Processor that a file is being created or that a record is to be added to the file. Upon receipt of this command, the Indexed Sequential Processor alters the Index to include the new record and writes the record in the file.

When a file is accessed directly, two imperative macros are used: READ and WRITE. First, the programmer must specify that the file is to be accessed directly rather than sequentially. Then the READ macro using the record key will transfer a specified record from the disc to the user's work area, where the data in the record (but not the record key itself) may be updated. The WRITE macro transfers the record in the work area back to its original position in the file.

The Random Processor

The Random Processor is used to process files having the relative address form of random organization. Files as large as eight volumes (or a total of 79,568 records) can be accessed. Records are unblocked and maximum record size is 160 bytes.

Random processing is most often used to avoid the time-consuming process of stepping through a sequential file or of reading an index before accessing a record. However, because these somewhat mechanical steps are bypassed, the Random Processor requires more information from the worker program than do the Sequential or Indexed Sequential Processors. The Random Processor considers the string of records making up the file to be in logical sequence, with the first record being Record 0, the second, Record 1, and so forth. When a record is to be accessed, its file position relative to 0 must be calculated by the problem program and presented to the Random Processor with a GET or PUT macro. Thus, in effect the worker program says, "GET the record in file position 138." The Random Processor then delivers the 139th record (since the first record is in file position 0) to the worker program, without regard for the record's actual key value; the worker program, in turn, need not be concerned with the disc unit, track, or sector where the record was stored.

DIRECT PROCESSING

Although the Univac IOCS Processors will satisfy the great majority of user requirements, it is conceivable that, because of system design or application considerations, other file management techniques may become desirable. In that case, the worker program would process the file directly; that is, it would issue commands directly to the Disc Dispatcher without the aid of IOCS.

In direct processing, the physical input/output operations to the disc are performed by the 8410 Disc Dispatcher, while logical control of the file is the responsibility of the worker program. The worker program gives the Disc Dispatcher a request packet that includes the address (disc unit, track, and sector) of the desired record as well as other pertinent information. (The contents and format of a Dispatcher request packet are explained in the 8410 DAS software manuals.)

In addition to defining and controlling the request packet, the worker program must also perform other functions, such as providing error checking and handling routines, checking and maintaining file labels if used, and blocking and deblocking records if multiple records are to be stored in a sector.

2. SCHEDULING

INTRODUCTION

Careful scheduling is essential to the successful installation of any data processing system. A good schedule serves a dual purpose: It helps to organize the installation activities as efficiently as possible, and it provides a means of evaluating and controlling the progress of each activity as the installation proceeds.

Development of the installation schedule involves:

1. *Classifying and defining the work to be done.*
2. *Determining the sequence in which the tasks are to be performed.*
3. *Evaluating the experience of personnel who will be doing the work.*
4. *Estimating the time required to complete the work.*

The approach to scheduling suggested in the following pages utilizes a number of Univac-developed scheduling and control forms. Proper use of these forms, explained in detail in this section, will greatly facilitate the scheduling work.

ESTABLISHING THE DELIVERY DATE

Ideally, the delivery date is determined by establishing the work to be done and then estimating the time required to complete it. In practice, however, other factors influence this decision. For example, delivery may be wanted during a low activity period, or in time to handle a peak activity period of a major seasonal application. There might be also an urgent need to automate a particular application or planning and preparation might be telescoped to take advantage of an early equipment availability date. All these factors have a bearing on the establishment of the equipment delivery date.

Once the delivery date has been set, the workload necessary to meet it should be estimated as soon as possible and a tentative schedule prepared. If the schedule shows that the delivery date can be met, the schedule becomes firm and the work can proceed. If it does not appear that the delivery date will be met, the following alternatives can be considered:

1. *Additional personnel can be acquired.*
2. *The number of applications, or their scope, can be reduced.*
3. *The delivery date can be rescheduled.*

CLASSIFICATION OF APPLICATIONS

The selection and classification of applications to be handled by the UNIVAC 9200/9300 Disc System is a primary consideration in planning for systems installation. Above all else, scheduling must reflect the organization's ability to prepare the selected applications by the time the equipment is installed. The number and scope of applications and the application implementation dates must be reasonable in terms of the equipment delivery deadline.

Applications to be handled by the UNIVAC 9200/9300 Disc System fall into three categories:

- Category 1
  Present punched card applications which will remain essentially punched card applications in the new system.

- Category 2
  Applications which will be converted from punched card to direct access, possibly with some broadening of scope.

- Category 3
  Present nonautomated applications to be converted to the computer system.

Category 1 Applications

Applications in this category are usually minor or intermediate in scope. Depending on the condition of the present systems documentation, they can involve little or no systems investigation and design effort, and should be relatively easy to handle with the straight conversion method. Such applications provide an excellent means of acquiring experience with the Report Program Generator. Each designer-programmer should be assigned at least one of these applications as soon as possible after completing a programming course. These applications will also be used to cover slack time between efforts on intermediate and major applications. The greater the number of applications that can be assigned to this category, the more the total installation effort will be reduced. Following installation, these applications may be reevaluated and perhaps increased in scope to take full advantage of the processing capabilities of the UNIVAC 9200/9300 Disc System.

Category 2 Applications

Applications in this category are usually the ones which provide the basis for acquiring the UNIVAC 9200/9300 Disc System. Handling them will usually have constituted the greater part of the present data processing system workload. Knowledge about, and documentation pertaining to, these applications can make them the ideal instrument for bridging the gap between punched card and direct access methods. In converting these applications to direct access, they can usually be broadened to include additional organization requirements. To the extent that they are expanded, some degree of systems design effort will be required. Category 2 applications, which will usually require the major portion of the installation effort and which must be operational before any of the present equipment can be removed, should be scheduled for implementation as early as possible.

Category 3 Applications

Before being assigned to this category, applications should be studied thoroughly to determine the practicability of, and need for, direct access processing. Automation of applications in Category 3 requires greater experience and a larger amount of systems design effort. A significant characteristic of applications belonging to this category is the number of changes which occur as the personnel involved become increasingly familiar with the potential of disc processing with the UNIVAC 9200/9300 Disc System. Objectives prescribing the intent of the original implementation should be developed for all Category 3 applications to ensure adherence to the original schedule and to reduce the number of changes required. Knowledge of the application gained during the initial implementation phase should be documented so that changes and improvements can be made soon after delivery of the new system.

For planning and scheduling purposes, the time estimates for systems investigation and application design obtained from the estimating charts for Category 3 applications should be doubled. It is recommended that Category 3 applications not be scheduled until all Category 2 applications have been developed to the point where their completion by the equipment delivery date is assured.

DEFINING THE APPLICATION

In order to estimate the amount of systems investigation and systems design time which will be required by individual applications, it is necessary to define the scope and complexity of each application. On the basis of this determination, applications are rated as minor, intermediate, or major. These terms are relative rather than absolute, however, and it is necessary to establish definitions for them which are relevant to the particular installation.

Minor applications require less systems design time than intermediate applications; these, in turn, require less than major applications. Category 1 applications, since they require little systems investigation and design and provide little opportunity for the use of advanced direct access techniques, are usually minor. Although Category 2 applications may require little systems investigation outside of the data processing department, they do require additional systems design time to enable files and procedures to take advantage of disc processing. Category 3 applications require extensive efforts in both systems investigation and systems design.

In order to classify an application as minor, intermediate, or major, it is necessary to specify exactly what the application is to include.

To illustrate, let's assume that a manufacturing organization has the following requirements: order entry, billing or invoicing, inventory, sales analysis, accounts receivable, inventory control, production scheduling, payroll, manpower scheduling, cost accounting and control, purchasing, accounts payable, and fiscal accounting. For the UNIVAC 9200/9300 Disc System, a single application could include any one, or many combinations, of these tasks. The "mix" would depend on the amount of system capability required.

Several typical applications are described below.

Typical Minor Application: Sales Analysis

Input for processing this application is provided by punched cards which contain the information for each line of customer invoice as well as customer, salesman, and sales territory code numbers. The system is to produce a current month and year-to-date report of sales dollars, by customer, for each salesman within each territory, and a current month and year-to-date report by product code number.

Year-to-date figures are now kept in punched card summary files. If this is to be a Category 1 application (thus remaining essentially a punched card operation) summary files would be maintained on discs to reduce the time required to produce the updated summary record. There would be no other change to the application.

Typical Minor Application: Accounts Payable

After vendor invoices are approved, cards are punched with the vendor number, ledger account number, date due, discount date, and amount. Cards are listed in payment date sequence. When it has been determined what vendors are to be paid in a given cycle, their cards, together with a vendor name and address file, are used to prepare the checks and, subsequently, account distribution for ledger entries.

In the new system the vendor name and address file might be maintained on discs. Other than that, the application would remain the same.

Typical Intermediate Application: Payroll

There are no complicated procedures required to arrive at gross pay for this 1,000-man payroll. Daily input is provided by punched cards which contain: man number, job number, pay rate, straight time, premium time, and shop code. Cards are extended daily to produce straight, premium, shift, and gross earnings. Weekly input is provided by punched cards containing man number and attendance hours. The job number is used to prepare weekly labor distribution reports. Payroll checks and allied production registers are produced weekly. There are also a few monthly reports, the quarterly state and federal tax reports, and annual W-2's. This application could become a Category 2 application for two reasons: the need to use direct access methods for processing, and a desire to broaden the scope of the application to produce additional output parts. Satisfying these requirements necessitates greater systems design effort than was needed for the Category 1 applications described above.

Intermediate Category 2 Applications

Careful attention should be given to the extent that intermediate Category 2 applications are increased in scope. The production of additional reports from information already available is usually easy to accomplish with the additional speed and capacity of the UNIVAC 9200/9300 Disc System. If additional input data is required to produce new reports, however, the application could easily become major. New types of input data require systems design investigation to determine outside preparation and control procedures. New input formats must be devised. Design changes to master files are necessary and completely new files are required. Systems design reflecting the flow of data from run to run will be affected and additional runs will have to be designed or new functions added to existing runs. For these reasons, intermediate applications currently being processed utilizing punched cards require the most careful consideration. Placing such applications in Category 1 greatly lessens the difficulty of meeting the original schedule, particularly if there are also major applications to be considered.

ESTABLISHING APPLICATION IMPLEMENTATION PRIORITIES

To establish implementation priorities, applications are divided into two groups:

- *Group 1* – consists of those applications which *must* be completed by equipment delivery date.

- *Group 2* – consists of applications which are to be implemented after delivery date. Applications in this group should not be scheduled for implementation until the timely conversion of Group 1 applications is assured.

Group 1 is made up of those Category 1 and 2 applications which are currently being processed on equipment that will be replaced by the UNIVAC 9200/9300 Disc System. There may also be justification for including a Category 3 application in Group 1. Applications assigned to Group 1 must, of course, be given the highest application priority.

Priority within Group 1 is normally assigned on the basis of category. Category 2 applications will usually be given a higher priority than Category 3 applications. Category 1 applications are handled as personnel availability permits until, as the equipment delivery date approaches, they assume a priority which places them above Category 3.

The sequence in which applications within each category are to be implemented must also be determined. The selection of Category 1 applications for use as additional training vehicles should be based on the simplicity of each application and the programmer's familiarity with it. The scheduling of Category 2 applications would be based on the availability of individuals who have the most knowledge of, and experience with, the applications. A second factor is the occurrence of peak periods. During peaks resulting from seasonal activity, personnel outside of the data processing department may be too busy to work with the systems designer. If the peak period is in the data processing department, present system processing requirements may make heavy drains on the time available.

RATING PRESENT SYSTEM APPLICATION DOCUMENTATION

In addition to rating the application itself, it is also necessary to estimate the condition of the present system documentation. If such documentation is reasonably complete and up to date, it should be rated as "good." If it is not, it should be rated "poor." If it is somewhere in between, or if its status is mixed, it is rated as "fair." For Category 3 applications that are not presently mechanized, there is of course no documentation. The documentation rating is thus "poor."

EVALUATING PERSONNEL EXPERIENCE

In addition to classifying and rating individual applications and determining the quality of existing documentation, it is necessary to determine the experience of personnel who will be performing the tasks. Based on the extent of their systems investigation and systems design experience, individuals should be rated as inexperienced, average experience, or experienced. The first rating, of course, reflects no experience at all. Individuals who have previously investigated and designed at least several applications for direct access processing can be rated as experienced. Those whose experience falls in between the two extremes are rated as having average experience. The Systems Design Estimating Chart (Figure 1–5) and Programming Estimating Chart (Figure 1–7) provide a graduated scale within each classification as a further experience rating aid.

Individuals performing the systems design and programming work for a program must also be given an experience rating. Factors considered here should include experience with direct access programming and experience with the programming languages – RPG, Assembler, and COBOL – used by the UNIVAC 9200/9300 Disc System. Initial assignment to simple programs of minor applications will provide inexperienced individuals with the opportunity to gain experience in these languages.

EVALUATING THE PROGRAMMING EFFORT

In order to schedule the design and production of programs, it is necessary to determine the number and complexity of programs required by each application. This determination is made after the application has been designed and an application flowchart prepared.

The documentation and flowcharts for each application provide the information necessary to arrive at this evaluation. In order to determine the systems design time and programming time required for each program, programs must be rated as simple, average, or complex. As an indication of the meanings of these terms refer to the examples previously given of minor and intermediate applications. In the typical Minor Application No. 1 (Sales Analysis) the program which produced the month and year-to-date reports and new year-to-date files would be classified as "simple." The program required for Minor Application No. 2 (Accounts Payable) would also be classified as "simple." In the Intermediate Application (Payroll), the program which produced daily gross earnings would be classified a little below average, and the program which calculated from gross to net, updated the year-to-date, and produced the payroll check and register data would be classified as a little above average.

An example of a complex problem would be an invoicing program meeting the following requirements: quantities coded by unit of measure, but priced on a unit basis (for example, 2½ dozen at $1.375 each); item discount by product class and customer; taxable and nontaxable items; weight per unit used to calculate shipping charges; sales statistics by product class; back order statistics; and accounts receivable data.

THE USE OF UNIVAC SCHEDULING FORMS

Among the many advantages of the Univac method of installation planning is the availability of a number of carefully designed forms and charts which facilitate, and increase the accuracy of, the scheduling effort. The following paragraphs summarize the function of each of these forms. The remainder of Section 2 explains how the forms are prepared.

The heading information at the top of each form and chart is self-explanatory. Blocks are provided at the top of each scheduling form in which to enter the names of the months (starting with the month in which the first scheduled event occurs) during which the installation will be accomplished. (Blocks are divided so that semimonthly progress can be recorded.) Event entries are made in the column identified by the month in which the event is scheduled. Date entries should be circled when the event has been completed. Both estimated (Est.) and actual (Act.) times may be entered.

Since it is frequently more convenient to change an entry on an existing form than to start a new form, entries should be made so that they may be erased and rewritten. All changes should be explained in an appropriate entry in the related supplementary file (described below).

Master Schedule I

Master Schedule I (Figure 2–1) is a checklist and summary of all major activities which take place during the installation effort. It provides a means of recording all the information needed to plan, schedule, and review installation activities. It summarizes the detail plans and schedules from the other forms, giving an overall picture of installation plans and progress.

Master Schedule II

Master Schedule II (Figure 2–2), which may contain any number of pages, lists individually all the applications to be converted for processing by the UNIVAC 9200/9300 Disc System. Entries are made for the major phases in the preparation of each application.

The Work Assignment Chart

Extremely important to the planning, scheduling, and control effort is proper utilization of the people who are to perform the work. To help achieve this, a Work Assignment Chart (Figure 2–3) is prepared for each individual who will participate in the installation effort. This chart is a combination schedule and control form. Space is provided to list all the individual's activities – those which are directly involved with the installation effort as well as those which are not. Some individuals may be involved with the continued performance of the present system during the installation period. These activities and others, such as vacation and training, are entered first on the Work Assignment Chart. The date and duration of these activities should be easily determined.

The Application Schedule

An Application Schedule (Figure 2–4) is prepared for each of the applications entered on the Master Schedule. The first several items preprinted on the Application Schedule are those activities which have to do with the application as a whole. Following these items, space has been provided to enter the names of the programs to be designed and written for this application. The vertical columns provided for recording the estimated and actual progress can be aligned with the columns on Master Schedule II for easy transcription.

The Systems Design and Programming Estimating Charts

Because the length of time required for both systems design and programming is subject to considerable variation, these phases of the installation effort must be the object of particularly careful scheduling and control. The Systems Design Estimating Chart (Figure 2–5) and the Programming Estimating Chart (Figure 2–7) facilitate this job by taking into consideration the principal factors influencing these estimates: (1) the difficulty of individual tasks, and (2) the experience of individuals assigned to them.

By permitting a value to be assigned to each of these factors for each task, the charts help you arrive at initial estimates for the time required. These estimates are posted to the Work Assignment Chart for the individual involved and to the appropriate Program Chart for the activity. This ensures control of the work being done and allows the accuracy of the values assigned to these factors to be determined. In this way the accuracy of such estimates can be improved early in the installation period, resulting in a more reliable overall schedule.

The Program Chart

A Program Chart (Figure 2–6) is also prepared for each of the applications to be implemented. The Program Chart is both a schedule and a control sheet for the preparation of each program. Space is provided on the form to enter the name of the program and the name of the individual(s) assigned to the task. Each of the programs entered on the Application Schedule will be entered on the Program Chart for the application.

MAINTAINING SUPPLEMENTARY FILES

To explain and supplement the Master Schedule, written records pertaining to all scheduling and control activities should be kept. It is suggested that initially these records be filed and indexed in accordance with the major headings of the Master Schedule. As the installation proceeds, additional files reflecting the major headings on the Application Schedule should be created and maintained. A file should also be maintained to supplement each individual's Work Assignment Chart.

Avoid verbal communications wherever possible. Written communications reduce misunderstandings between individuals involved in the installation. Copies of such communications should be appropriately filed. It is important, too, that the reasoning behind the direction taken at all critical points in the planning and scheduling efforts be documented.

PREPARING MASTER SCHEDULE I

Master Schedule I (Figure 2–1) is prepared in accordance with the following directions. Freestanding, nonindented headings reflect major headings at the left of the schedule. "Dependent Items" identify other entries which are established with the help of the entry under which the dependent item occurs. An asterisk indicates that the entry is found on Master Schedule I. If the entry is on another form, the name of the form is added.

Delivery and Checkout

Enter in the appropriate month column the date on which delivery and checkout are scheduled. This is the week that control of the UNIVAC 9200/9300 Disc System will be assumed by the user organization.

Dependent Items:

(a) Present Equipment*

(b) Site Preparation*

(c) Training (Operator)*

(d) Order (Tapes, Discs, and Supplies)*

(e) Parallel Run and Operational (Application Schedule)

MASTER SCHEDULE I

INSTALLATION ________ PREPARED BY ________ DATE ___ PAGE ___ OF ___

ENTER MONTHS (From starting month on)																								
UPDATE SEMI-MONTHLY		1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	2	1	2
DELIVERY & CHECK OUT	EST																							
	ACT																							
PRESENT EQUIPMENT CANCELLATION	EST																							
	ACT																							
REMOVAL	EST																							
	ACT																							
SITE PREPARATIONS ESTABLISH	EST																							
REQUIREMENTS	ACT																							
REVIEW	EST																							
	ACT																							
PREPARATION	EST																							
	ACT																							
TRAINING INSTALLATION	EST																							
GUIDE	ACT																							
PROGRAMMING	EST																							
	ACT																							
OPERATOR	EST																							
	ACT																							
ORDER FORMS	EST																							
	ACT																							
TAPES/DISC	EST																							
	ACT																							
SUPPLIES	EST																							
	ACT																							
TEST CENTER AVAILABLE	EST																							
	ACT																							
ORDER TEST	EST																							
TAPES/DISC	ACT																							
TRIPS	EST																							
	ACT																							

Figure 2–1. Master Schedule I

Present Equipment

Cancellation – Enter the date on which the supplier of present equipment must be notified of its cancellation.

Dependent Items: Present Equipment (Removal)*

Removal – Enter the date on which the present equipment will no longer be needed. (Any change of this date should of course be reported to the supplier.)

Dependent Items: None

Site Preparation

Establish Requirements – As soon as the length of time required to prepare the site has been determined, enter the date on which site plans will be developed and submitted to Univac for review. If more than one week will be required, enter the number of weeks in parentheses following the date. (Refer to UNIVAC 9200/9300 Systems Installation Planning Specifications, MH-1185, for assistance in determining site requirements.)

Dependent Items: Site Preparation (Review and Preparation)*

Review – Enter the date on which Univac personnel will review site preparation plans.

Dependent Items: Site Preparation (Preparation)*

Preparation – Draw a line through the period during which the physical preparation of the site is to occur.

Dependent Items: Delivery and Checkout*

Training

These entries are made from the Work Assignment Charts as personnel are scheduled for training. Draw a line to reflect the duration of the training. If more than one training session is required, an equal number of additional lines should be drawn.

Installation Guide – This should be the earliest scheduled event.

Dependent Items: None

Programming – Color codes may be used for RPG or Assembler.

Dependent Items: All entries on the Application Schedule

Operator – This training may occur on a Test Center visit or after equipment delivery, or both.

Dependent Items: None

Order

Enter the month, day, and year of the last date (based on supplier lead time) on which materials can be ordered and still arrive when needed. This is a "last possible" date and not the date on which it is planned to enter the order, which should be earlier. Except for general supplies, ordering will be scheduled on the basis of information entered on the Application Schedule.

Forms – See preceding paragraph.

Dependent Items: Forms and Supplies (Application Schedule)

Tapes and Discs – See paragraph under "Order." (Note that a price reduction based on quantity may frequently be obtained by including all tape and disc requirements in a single order.)

Dependent Items: None

Supplies – See paragraph under "Order."

Dependent Items: None

Test Center

Available – Enter the date when the desired Test Center is available. Information for this entry is secured from the Univac Account Representative.

Dependent Items:

(a) System Test and File Conversion (Application Schedule)

(b) Program Test (Program Chart)

Order Test Tapes – Tape reels for test data and for use as scratch operating tapes are needed for Test Center visits. These tapes should be ordered, taking into account supplier lead time, so that they arrive prior to the Test Center availability date.

Dependent Items: Test Center (Trips)*

Trips – The "Ready To Test" column on Program Charts indicates the possible need for scheduling a Test Center trip. This information and the dates for which program completion are scheduled provide the input for this entry. Note that preparation for program test data does not appear as an entry on any of the charts. It is assumed that test data is available for the system. If this is not the case, however, a program should not be indicated as "Ready To Test" until test data has been prepared. The need for special programs to convert test data from present system media and format to those of the new system should also be considered. If possible, such special programs should be written in time for use during file conversion for the new system.

Dependent Items: None

PREPARING MASTER SCHEDULE II

Master Schedule II (Figure 2–2) presents information pertaining to individual applications to be converted for processing by the UNIVAC 9200/9300 Disc System. Although the sequence in which applications are entered is largely a matter of convenience, there are several considerations worth mentioning.

Entering first the Category 1 applications used to reinforce classroon program training allows the relationship of activities occurring early in the planning, scheduling, and installation phases to be seen at a glance. Alternately, the major Category 2 applications, since they represent the major portion of the installation effort, could be entered first. Or, if it is desired to establish chronological order down and accross the schedule, applications could be entered in priority sequence. Information posted to these items is obtained from the individual Application Schedule.

MASTER SCHEDULE II

INSTALLATION ______________ PREPARED BY ______________ DATE ____ PAGE __ OF ____

ENTER MONTHS (From starting month on)													
UPDATE SEMI-MONTHLY		1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2	1 2
APPL. NAME													
INVESTIGATION	EST												
	ACT												
SYSTEM DESIGN	EST												
	ACT												
PROGRAMMING	EST												
	ACT												
OPERATIONAL	EST												
	ACT												
APPL. NAME													
INVESTIGATION	EST												
	ACT												
SYSTEM DESIGN	EST												
	ACT												
PROGRAMMING	EST												
	ACT												
OPERATIONAL	EST												
	ACT												
APPL. NAME													
INVESTIGATION	EST												
	ACT												
SYSTEM DESIGN	EST												
	ACT												
PROGRAMMING	EST												
	ACT												
OPERATIONAL	EST												
	ACT												
APPL. NAME													
INVESTIGATION	EST												
	ACT												
SYSTEM DESIGN	EST												
	ACT												
PROGRAMMING	EST												
	ACT												
OPERATIONAL	EST												
	ACT												
APPL. NAME													
INVESTIGATION	EST												
	ACT												
SYSTEM DESIGN	EST												
	ACT												
PROGRAMMING	EST												
	ACT												
OPERATIONAL	EST												
	ACT												

Figure 2–2. Master Schedule II

SCHEDULING THE INSTALLATION

Although there may be several ways to begin scheduling the installation effort, the methods described below are recommended for the majority of installations. Regardless of how this schedule is derived, however, it must include margin for error or for unforeseen difficulties.

Applications, identified by group, category, and priority sequence, should already be listed on Master Schedule II at this point, as should some of the other events. The next step, then, is to relate work to be done to personnel available.

Determining Personnel Requirements and Availability

Prepare a Work Assignment Chart (Figure 2–3) for each individual participating in the installation effort. If such determinations can be made at this time, enter a figure representing the percentage of the individual's time which can be devoted to each activity under the column headed "Work %." If the individual's availability is apt to vary during the course of the installation effort, enter this figure as the individual is scheduled for each activity.

In conjunction with personnel availability, it is also necessary to determine what personnel are required to operate the present system. If certain individuals must be allocated to present system duties, enter "Required for Duty" as the first activity on their Work Assignment Charts.

WORK ASSIGNMENT CHART

NAME ________________

ACTIVITY		SCHEDULE FACTORS											SCHEDULE MONTHS													
		Exper. Rating	Date Rated	Chart Days	Work %	Work Days	Accum. Work Days	% Comp.	Start Date	End Date	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									
	EST																									
	ACT																									

Figure 2–3. Work Assignment Chart

Scheduling Personnel Training

The next step is to schedule personnel training and, depending on the time of year, vacations. A schedule of courses offered by Univac may be obtained from the Univac Account Representative. In determining when personnel are to attend such courses, consider the advantages of having them attend after, rather than before, their vacation. It is suggested that the more experienced systems design and programming people be the first to attend Univac classes. This arrangement will enable the experienced staff to begin the preparation of applications to be programmed by the less experienced people after the latter have attended the class.

To schedule each individual, enter the Activity Name, the Work Days, the Start Date and the End Date on his Work Assignment Chart, and draw a solid line through the calendar portion. These entries should all be on the "Est." (estimated) line. When the activity actually begins, enter the date started on the "Act." (actual) portion of the Start Date column. If this does not coincide with the estimated Start Date, make a vertical mark (equivalent to one day) in the calendar portion of the "Act." line. Determine immediately what effect, if any, this change in start time will have on later activities scheduled for this individual. Also note if this change will have an effect on any other dependent activities.

When these entries have been made on each individual's Work Assignment Chart, post the composite schedule to Master Schedule I by placing a line in the calendar portion of the "Est." line. Also, prepare a memo describing the training plans and file it, together with copies of the enrollment requests transmitted to Univac, in the supplementary folder.

Scheduling Systems Investigation

Based on the estimated completion dates for training, assign individuals to the systems investigation of minor applications which are to be used as training reinforcement. Fill in the heading information on the Application Schedule (Figure 2–4) for each application. The following factors must be known in order to assign systems investigation tasks to individuals:

1. The rating of the applications
2. The rating of the documentation
3. Experience rating of the individual

APPLICATION SCHEDULE

APPLICATION ____________________

INSTALLATION ________________ PREPARED BY ________________ ____ PAGE ___ OF _____

ENTER MONTHS (From starting month on)																							
UPDATE SEMI-MONTHLY		1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2	1	2
INVESTIGATION	EST																						
	ACT																						
SYSTEM DESIGN	EST																						
	ACT																						
FORMS & SUPPLIES	EST																						
	ACT																						
SYSTEM TEST	EST																						
	ACT																						
FILE CONVERSION	EST																						
	ACT																						
PARALLEL RUN	EST																						
	ACT																						
OPERATIONAL	EST																						
	ACT																						
PROGRAM NAME																							
SYSTEM DESIGN	EST																						
	ACT																						
PROGRAMMING	EST																						
	ACT																						
TEST	EST																						
	ACT																						
PROGRAM NAME																							
SYSTEM DESIGN	EST																						
	ACT																						
PROGRAMMING	EST																						
	ACT																						
TEST	EST																						
	ACT																						
PROGRAM NAME																							
SYSTEM DESIGN	EST																						
	ACT																						
PROGRAMMING	EST																						
	ACT																						
TEST	EST																						
	ACT																						
PROGRAM NAME																							
SYSTEM DESIGN	EST																						
	ACT																						
PROGRAMMING	EST																						
	ACT																						
TEST	EST																						
	ACT																						

Figure 2–4. Application Schedule

Using these factors and the Systems Design Estimating Chart (Figure 2–5), determine the estimated systems investigation time as follows:

Place a ruler at the lower left-hand corner of the Systems Design Estimating Chart at an angle determined by the application rating. If the rating is the typical minor, intermediate, or major, use the line drawn on the chart. Locate the experience rating of the individual on the scale at the left side of the chart and follow this rating horizontally until it intersects the application rating. Proceed vertically from this intersection to the top of the chart and select the number of days from the proper horizontal scale which reflects the documentation rating of the application. Enter this number in the "Est." portion of the Chart Days column on the Work Assignment Chart for the individual. Using the "Work %" factor, determine and enter the Work Days estimated for the task. Keep in mind that a Work % of 50 means that the Work Days are *twice* the Chart Days; a Work % of 66 2/3 means half again longer than Chart Days.

Estimate the start date and end date, and draw a solid line in the calendar portion of the Work Assignment Chart. Examine the calendar portion of the chart above and below the line to make sure the individual is not scheduled for any other activity during this period. Indicate the fact that the event has been assigned and scheduled by drawing a solid line in the calendar portion of the Application Schedule opposite "Investigation."

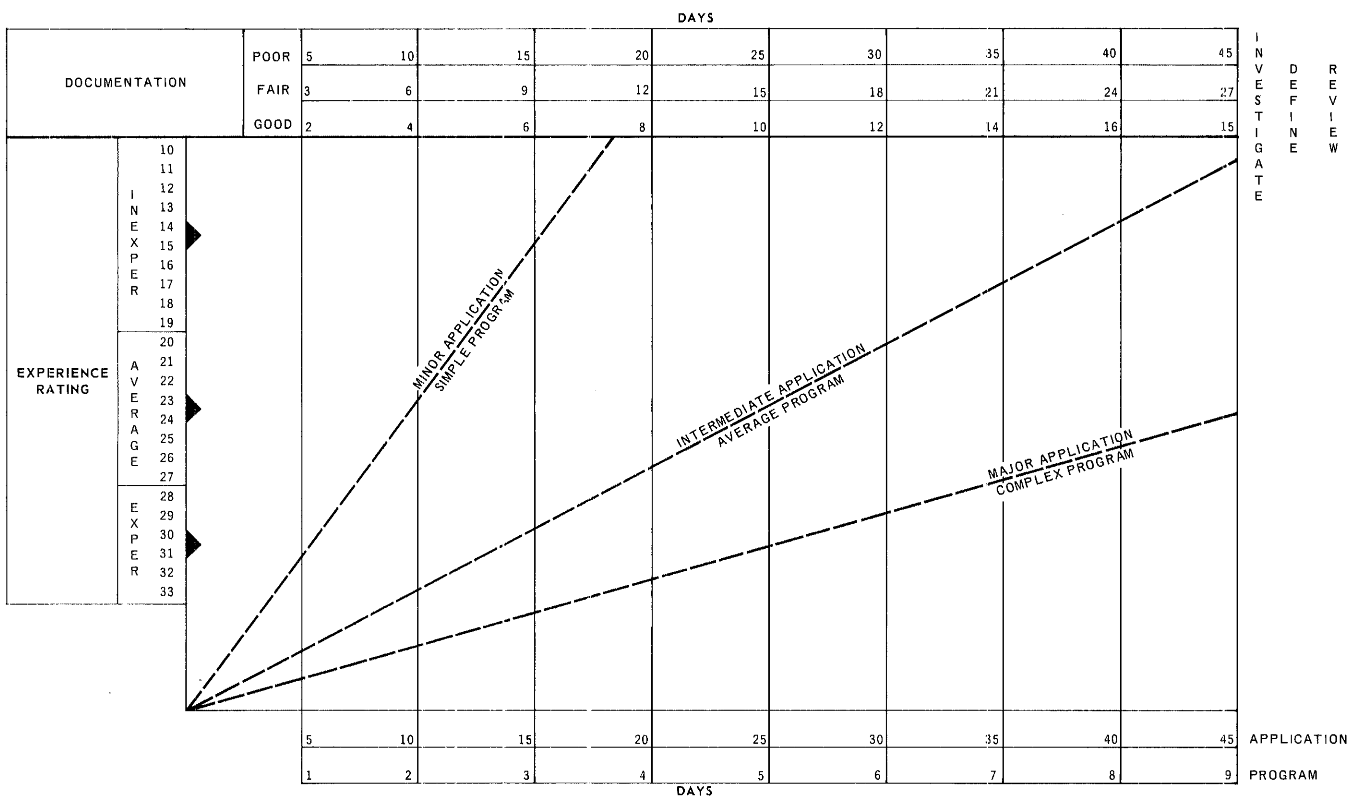

Figure 2–5. Systems Design Estimating Chart

Scheduling Systems Design

Follow this same basic procedure to estimate the systems design time for each application. This time, however, proceed from the intersection of the experience rating and the application rating *downward* to the horizontal scale of days for *Application*. (This is the upper part of the two lower scales. The lower part is used to estimate systems design for a single program.) Post the Chart Days, Work Days, etc., to the Work Assignment Chart and then to the Application Schedule. Information from the Application Schedule should now be posted to Master Schedule II.

As the work proceeds, each individual should report the actual time he spends on the activity. Reporting frequency normally depends on the length of the activity and its relationship to other items on the schedule. Weekly reporting should be sufficient for a task estimated to require two or three weeks. If a task is estimated at less than two weeks, it may be sufficient to make verbal progress checks and have the actual time reported only on completion. Or, the individual may be required to report when half of the estimated time has elapsed. Reports reflecting actual time and percentages of completion are posted to the individual's Work Assignment Chart. Add the reported time to the Accumulated Work Days and post the new sum. By referring to the Est. Start Date, Act. Start Date, Est. End Date, Accum. Work Days, and % Comp., progress on the activity itself and the effect of this progress on the entire installation schedule can be determined.

If it is determined that the actual time will exceed, or be less than, the estimated time, and that the Est. End Date will not be met, use the following procedure:

Using the actual time and percentage of completion figures, estimate the remaining time required to complete the activity. Add this estimate to the Accum. Work Days to arrive at a new total time estimate for the activity. Subtract the original Est. Work Days from the new total time and enter the difference (plus or minus) in the Work Days column on the "Act." line. (Do not change the original Est. Work Days. That figure is necessary to evaluate the ratings of the individual and of the activity.) Replace the original Est. End Date with a newly calculated Est. End Date. Make appropriate adjustments to the other schedules in which this activity appears, then determine the effect of the change in completion date of this activity on the overall schedule.

When each application's systems design work and flowchart have been completed, enter the program names and ratings on the Application Schedule. Prepare a Program Chart (Figure 2–6) for each application and enter the names and ratings of the programs.

In some cases it may be difficult to decide whether or not a sort program should be scheduled as a task. If the sorting requires very little or no "own coding," list it on the Application Schedule but not on the Program Chart. Include it as part of the assignment of the program that follows. If the "own coding" requirements of the sort program are more than minimal, list it on the Program Chart also, and rate it as simple, average, or complex. The programming language to be used for each program is also entered on the Program Chart.

PROGRAM CHART

APPLICATION ____________

			SYSTEMS DESIGN						PROGRAMMING										
PROGRAM NAME	Lang.	Rates	Last Start Date	Start Date	Days	Accum. Days	End Date	ASSIGNED TO	Last Start Date	Start Date	Days	Accum. Days	Flow Chart	Code	K.P.	Desk Check	Ready To Test	Test Time	Run Docum.
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			
EST																			
ACT																			

Figure 2-6. Program Chart

Programs should be entered on the Application Schedule and on the Program Chart in the execution sequence shown on the Application Flowchart. It may also be determined at this point that semiannual programs can be scheduled for programming after the UNIVAC 9200/9300 Disc System is operational. This will remove some of the work load in the preinstallation period. Before making this decision, however, be sure that there is sufficient time to complete it after installation and before the due date of these programs. Make certain also that the application systems design includes all of the data requirements for these reports.

The main reason for listing programs in application flowchart sequence is to determine as soon as possible whether or not an oversight was made during the application systems design. Programs should be scheduled for systems design and programming in the sequence listed. If two or more people are going to perform the programming, they should be assigned alternating programs, enabling both to acquire good overall knowledge of the application.

Scheduling Program Logical Design and Programming

Application programming can be scheduled as soon as each application's Program Chart has been prepared. Required to complete this scheduling step are:

1. The Program Chart
2. The Systems Design Estimating Chart
3. The Programming Estimating Chart
4. A Work Assignment Chart for each individual involved

The development of applications from systems investigation through programming follows a logical process. It is often desirable, therefore, to assign all phases of the work to the same persons. On the other hand, as an individual gains experience in one area, it may be preferable to take advantage of his increased experience in that area. The workload of the personnel involved and the amount of time available are the determining factors in assigning the programming work. If the programming is not done by the same person who performed the systems investigation and systems design, the documentation must be particularly clear or additional time will be required for clarification. It is recommended that the same person perform the logical design and the writing of a program.

Once the work has been assigned, the logical design for the program can be scheduled. On the individual's Work Assignment Chart enter the program name, followed by the abbreviation "Log. Des." Before the chart can be completed further, an estimate of the time required is needed. Using the Systems Design Estimating Chart, mark the complexity rating of the program (taken from the Program Chart) and the experience rating of the individual (taken from his Work Assignment Chart), then find the intersection of the two ratings. Move down on the chart to the lower scale of days, labelled "Program," to find the number of Chart Days. On the Work Assignment Chart, enter this number in the Chart Days column; then divide the Chart Days by the Work % to get the number of Work Days. From the system designer's workload, estimate the starting date and enter it on the Work Assignment Chart under "Est. Start Date." Compute and enter an Est. End Date and draw a line between the two in the calendar portion of the Work Assignment Chart.

On the Program Chart enter the system designer's name in the upper half of the "Assigned To" column. From the information on the Work Assignment Chart, enter the Start Date, Est. Days (equivalent to the work days on the Work Assignment Chart), and Est. End Date, in the columns under "Systems Design." The "Last Start Date" column is to be used only if completion of a program by a certain time is essential to the success of a major activity or of the installation effort as a whole. For example, the system test of a series of daily runs for an application may be planned for a previously scheduled trip to the Test Center. By working backwards from the last program in the series and including a cushion factor, the designer can determine the latest date on which each program in the series must be started in order to be ready at the desired time. The "Last Start Date" column serves to alert the system designer if he is not on schedule.

Programming is scheduled similarly, using the Programming Estimating Chart (Figure 2-7). From the intersection of the program rating and the experience rating, move down on the chart to the appropriate scale of the programming language to be used to find the number of days required for programming. Complete the Work Assignment Chart and the Program Chart in the same way as described for program logical design scheduling. Use the columns under the heading "Programming" on the Program Chart. Enter the programmer's name in the lower half ot the "Assigned To" column of the Program Chart if he was not the same individual who did the logical design. Enter the actual completion date for each of the activities noted.

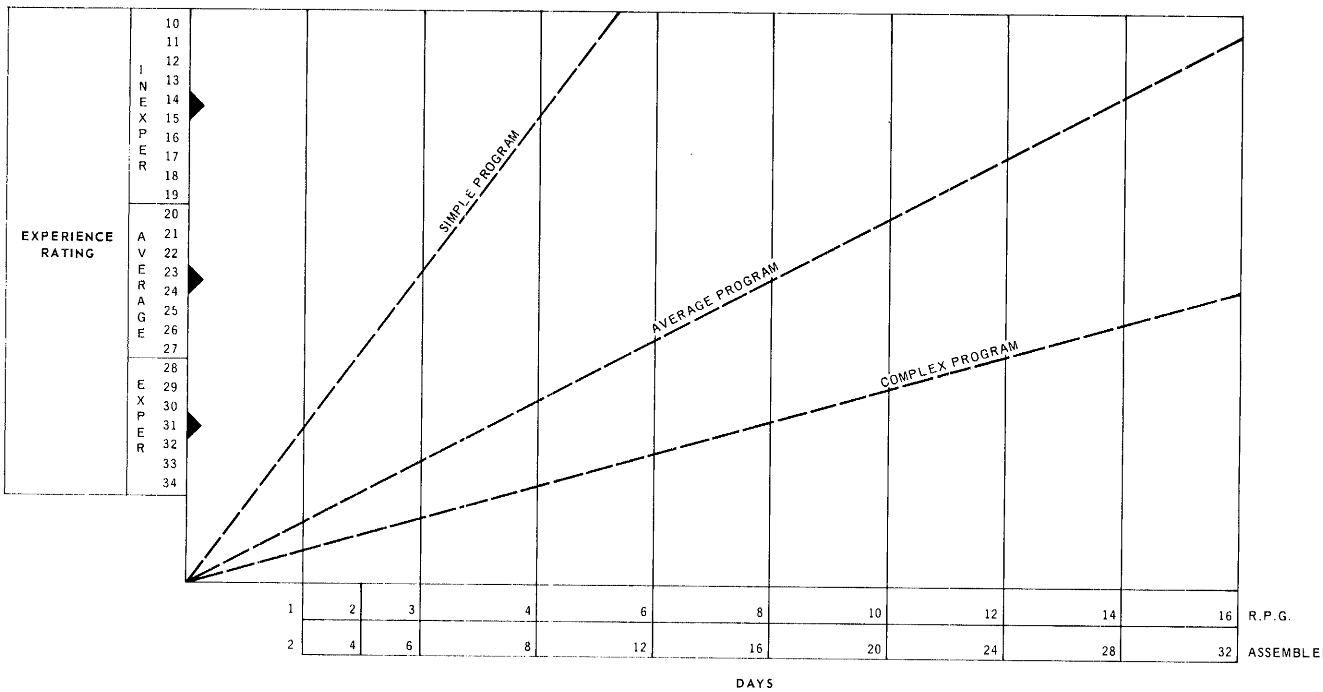

Figure 2-7. Programming Estimating Chart

The programming time entered on the Programming Estimating Chart does not include time required for keypunching of the program. If the punching of programs cannot be given a high priority, keypunch turnaround time must be added to the estimated number of Work Days (not Chart Days).

A program is not considered to be complete until it is ready for testing. If test data is not available from the present system, the time required for its preparation should be a separate assignment on a Work Assignment Chart.

If a Test Center trip is scheduled, the "Ready To Test" column (Est.) can be used to determine whether a program should be tested at that time. The "Last Start Date" mentioned above should be determined for programs that are to be included on the trip. On the other hand, the estimated and actual "Ready To Test" dates can help to determine when a Test Center trip can be scheduled. When a program is to be tested on a trip which is already planned, enter the date of the trip in the "Est." part of the "Test Time" column. When a program has been satisfactorily tested, enter the date in the "Act." part of the "Test Time" column. Next, assign and schedule "Run Documentation" on the Work Assignment Chart and enter the estimated completion date on the Program Chart.

The actual time taken to perform the logical design for a program should be reported on the same basis as described for systems investigation and systems design, using the "% Comp." column on the Work Assignment Chart for control. For long programs, the percentage of completion and the completion dates for the interior programming event columns should be entered on the Program Chart. Short-term programs can be handled on the percentage of completion basis without regard to the interior events. After an activity has been assigned and scheduled on the Work Assignment Chart and the Program Chart, the activity is posted to the calendar portion of the Application Schedule. Periodically the entries on the Application Schedule are posted to Master Schedule II. Progress on the overall application as well as on the total installation effort can thus be seen.

Scheduling Other Application Activities

The other activities listed on the Application Schedule are: Forms and Supplies, Systems Test, File Conversion, Parallel Run, and Operational.

The Forms and Supplies date depends on when the input/output requirements of the application are firm. The ordering of nonstandard forms and supplies should not present a scheduling problem unless the completion date of the programming is near the installation date. If this is the case, the programs that use the forms should be assigned as the first efforts on the application.

The systems test of an application is performed at a Test Center and consists of fitting together the individually prepared programs to determine whether they will operate consecutively, performing the processing required of the total system. The time required for preparation of test data, if this is necessary, must be considered when scheduling a Test Center trip. In the calendar portion of the Application Schedule, draw a line to coincide with the line on Master Schedule I that indicates the scheduled Test Center trip. If the time required for test data preparation is significant, draw a line in the calendar portion of the Systems Test entry on the Application Schedule to indicate this activity.

File conversion is the preparation of the magnetic tape or disc files to be used in the new system. The systems design will provide the information necessary to determine file conversion requirements.

The programs needed for file conversion, their complexity rating, and the programming language to be used should all be listed on the Application Schedule and on the Program Chart in the same manner as the processing programs. File conversion programs can be tested by being used to prepare test files for program and systems tests. File conversion is normally performed after control of the UNIVAC 9200/9300 Disc System has been assumed by the user organization. The time required depends on the volume of data and the ease of conversion. This time must be considered when scheduling a parallel run.

The parallel run technique is used to validate operation of the new system by processing the same "live" data on the old system. The period of parallel operation varies with each application and each installation. If a parallel run is planned, its estimated date and duration should be entered on the Application Schedule. From this information, the date on which the new system will be operational can be estimated. The "Est. Operational" date should then be entered on the Application Schedule and on Master Schedule II.

3. SYSTEMS DESIGN

INTRODUCTION

Systems design is the first major milestone in preparing for the new data processing system. The objectives of systems design, and suggested means of meeting these objectives, are discussed in this section.

Prerequisite to successful systems design is knowledge in two specific areas: the basic concepts of direct access operation, and the hardware and software characteristics of the UNIVAC 9200/9300 System. The following Univac reference manuals provide valuable background in these areas:

- UNIVAC 9200 System Description (UP-4086)
- UNIVAC 9300 System Description (UP-4119)
- UNIVAC 8410 Direct Access Subsystem General Description (UP-7567 Rev. 1)
- UNIVAC 8410 Disc File Component Description (UP-7594)
- UNIVAC 9200/9300 Systems Card Assembler Programmers Reference (UP-4092 Rev.2)
- UNIVAC 9200/9300 Systems Card Report Program Generator Reference (UP-4106)
- UNIVAC 9200/9300 Tape and Disc Report Program Generator Programmers Reference (UP-7620)
- UNIVAC 9200/9300 Central Processor and Peripherals Programmers Reference (UP-7546)

INVESTIGATING THE PRESENT SYSTEM

General

Essential to the successful design of a new system is a thorough understanding of the present system. Existing procedures and operations must be analyzed and all available documentation collected. If the present system is inadequately documented, sufficient new documentation must be created to completely define all aspects of the system.

Defining The Present System's Objectives

Before the requirements of an application to be implemented on the new system can be established, the general objectives met by that application in the present system must be determined.

Theoretically, the only information needed by the system designer to define system objectives is that relating to output and timing. In practice, however, the system designer must also consider such factors as input (what is available?), present equipment (what are its advantages and limitations?), system exceptions, and company policy. Definition of system objectives is thus based on a thorough understanding of all facets of the system.

Since there is usually no single individual completely familiar with all aspects of the application, the information needed to define system objectives is gathered in several ways:

1. By studying all available documentation and interviewing report users.
2. By observing activities, both clerical and mechanical, that lead to the origination of printed documents and the maintenance of files.
3. By creating new documentation where none is available.

Flowcharting The Present System

An extremely important source of information is the system flowchart, in that it depicts the whole of an application, bringing together and showing graphically all of its operations and their relationships.

If existing documentation includes flowcharts that are complete and understandable, further flowcharting may not be required. Thorough understanding of the present system, however, frequently requires the preparation of multiple levels of flowcharts for each application. Flowcharts may range from system flowcharts which show the general flow of information through the system without reflecting run details, to flowcharts showing all the details of a *single* data processing function.

Two types of flowcharts are particularly important in documenting the present system:

1. *The system flowchart* – (illustrated in Figure 3–1) shows on one document all the operations, manual and mechanical, that are performed within an application.
2. *The application (run) flowchart* – shows in detail all the operations performed within a single run.

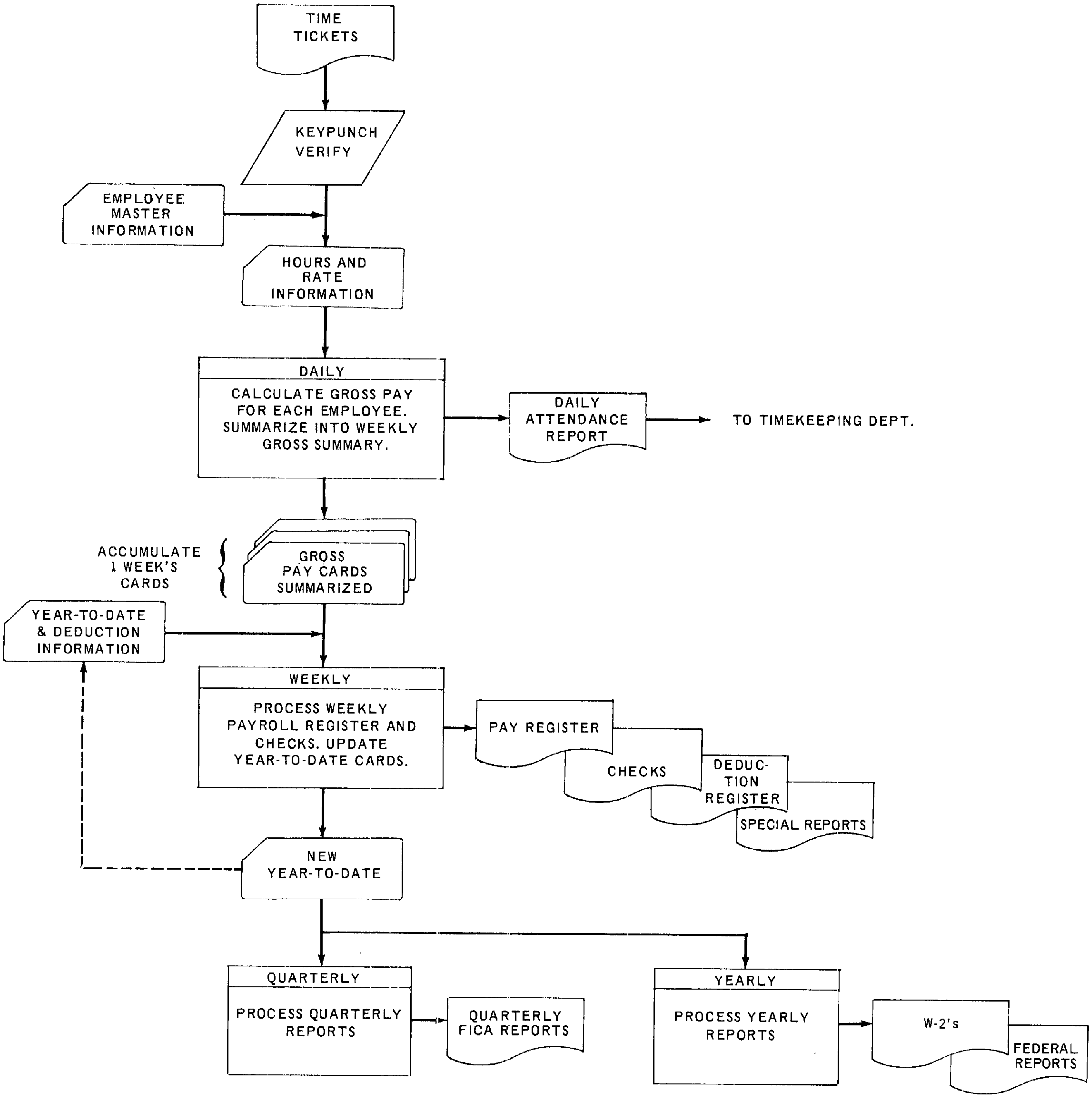

Figure 3–1. Typical System Flowchart for a Payroll Application

Figure 3–2 shows how a Univac-designed application chart is used to expedite the flowcharting of the runs of the payroll application in Figure 3–1. The chart is divided into vertical sections, each section containing symbols representing various data processing functions. The symbols used represent (1) an operation or function, (2) an input or output document, (3) input or output cards. Each operation symbol, or box, has room to enter the operation number, the number of the machine on which it is performed, and a brief description of the operation. If necessary, additional input or output symbols may be drawn to the left of the operation box. Connecting lines show the relationship and sequence of the data processing functions.

UNIVAC

APPLICATION CHART

UNIVAC 9200/9300 SYSTEM

APPLICATION: NO. 1 NAME PAYROLL (DAILY GROSS)

PAGE 1 OF 3

PREPARED BY D.P. MANAGER DATE 1/1/67

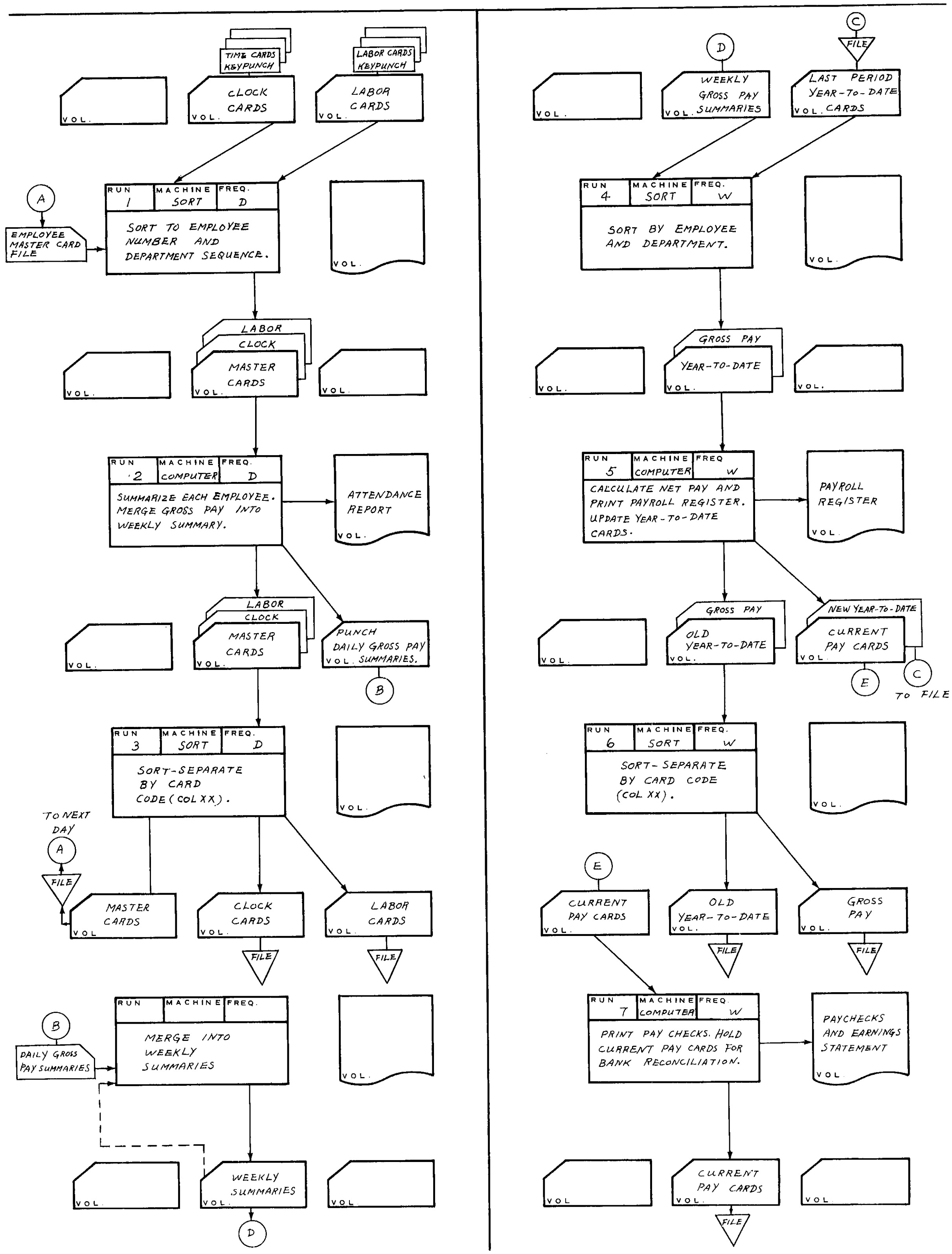

Figure 3–2. Run Flowchart: Existing Punched Card Application on Standard Form

Symbols for input and output documents should contain the report name (or brief description of the document) and the average card volume or the number of document lines.

The connector symbol is a means of representing the junction of two lines of flow or a break in a single line of flow.

A triangle may be used as a symbol for a file or storage point.

A significant benefit of thorough flowcharting is the opportunity it offers of identifying inefficiencies in the present system. Careful analysis of the completed flowcharts, for example, may uncover previously unnoted relationships and similarities among data or functions. Such information, if available to the system designer, can result in appreciable improvements in the design of runs for the new system.

Specific limitations of present equipment should also be noted during the investigation. Knowing these limitations and the special processing procedures they necessitate increases the system designer's awareness of areas in which greater processing efficiency can be realized with the UNIVAC 9200/9300 Disc System.

Documenting The Individual Run

Extra care and time spent in run documentation will be many times repaid by the speed and accuracy with which the resulting program can be written. Documentation for each run must be detailed and exact. All information pertinent to that run must be included, even though some of it (such as report forms, card and tape record layouts) may also pertain to other runs. The appearance of information in several places is less bothersome than having to search for it if it is documented once only.

Documentation for each run of an application must include, in addition to the appropriate flowcharts:

1. *Input file format*

2. *Output file format*

3. *Narrative-type documents describing operating procedures, volumes, timings, etc.*

Two charts provided by Univac facilitate the recording of run documentation for both the present and the new system:

1. *A Multiple Card Layout Chart*

2. *The Printer Format Chart*

The Multiple Card Layout Chart: To aid in the description and clarification of punched cards (such as the one illustrated in Figure 3–3), and to establish a workable reference document for systems design and programming personnel, Univac provides a Multiple Card Layout Chart (Figure 3–4) on which data for as many as six individual input and output cards can be recorded.

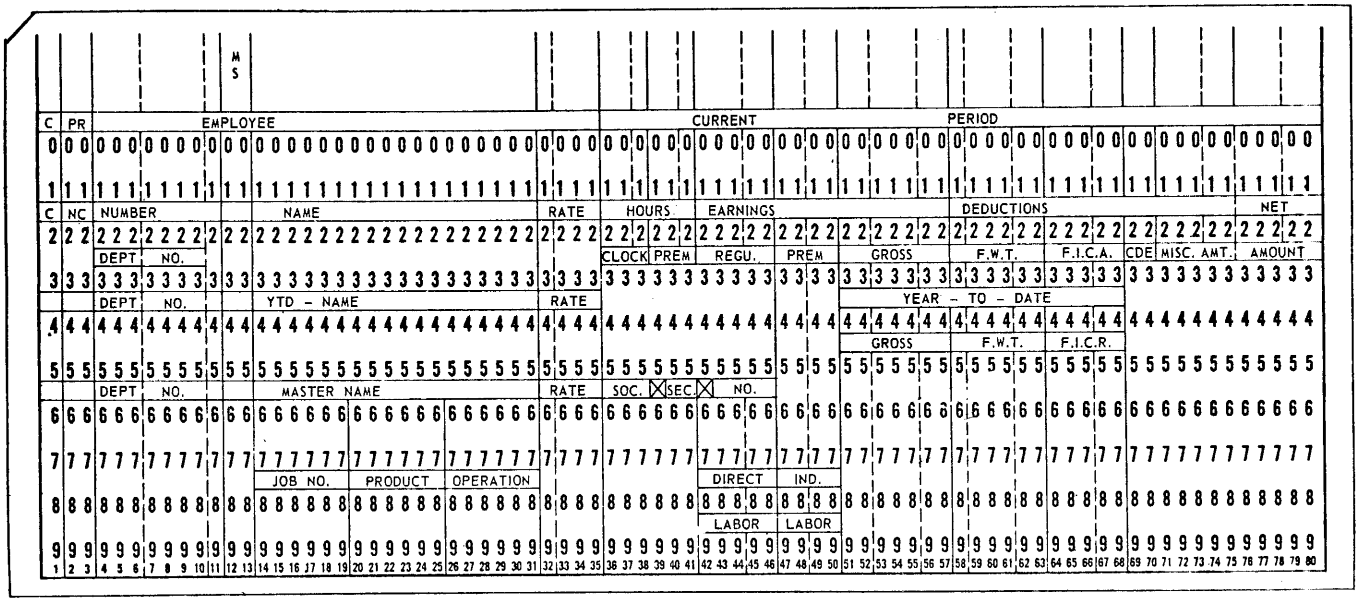

Figure 3–3. Typical Punched Card for a Payroll Application

Card names, control position, and other pertinent data are recorded in the spaces provided at the left of each field. Two types of multiple card layout charts should be kept: (1) A master chart on which all cards used for the application are recorded, but which excludes information common to all runs (such as that pertaining to total control columns, input and output, etc.); (2) The card layout chart for each individual run, which includes all information pertinent to that run, including that which is omitted from the master chart.

Figure 3–4 shows a Master Multiple Card Layout Chart for a payroll application. The boundaries of each card field are shown by lines drawn between the proper column numbers. If a particular field is common to more than one card, lines are extended downward from the first to subsequent cards. Each field of each card should be identified by name. Numeric fields (those which are involved in calculations or are numerically sequenced) must also show decimal points. A small triangular symbol (known as a "carat") is inserted between columns to indicate a decimal point. Any numeric field expressed as a whole number should have a carat placed on the vertical line at the right margin of the field. Credit positions applying to individual fields must also be shown.

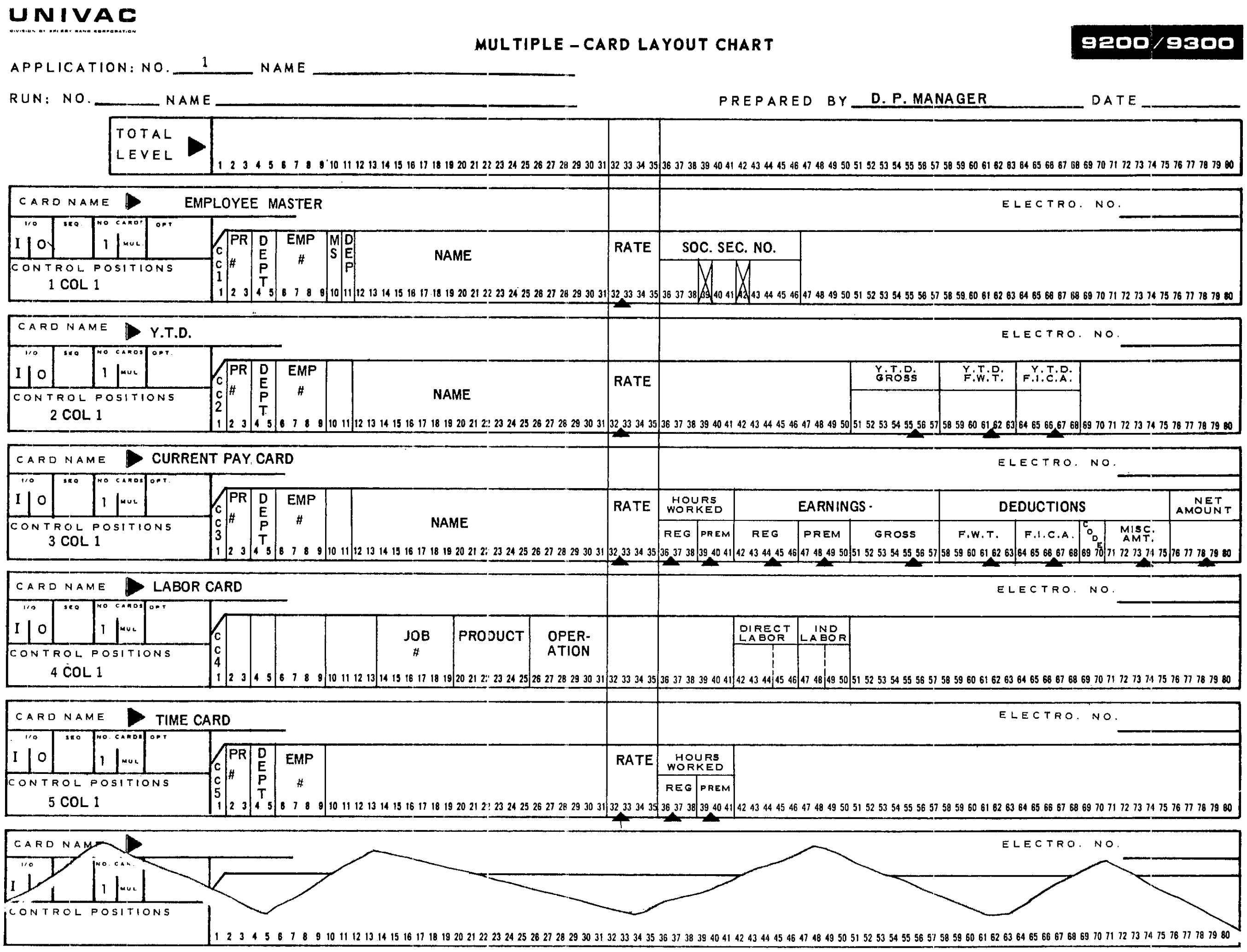

Figure 3–4. Master Multiple Card Layout Chart for a Payroll Application

Control positions that identify cards and card names should be entered in the space provided for this purpose at the left of the chart (illustrated in Figure 3–5A).

Total level control fields (those card fields that cause automatic total breaks) will vary from run to run. This information must therefore be entered separately for each run. Total level control fields are recorded by extending the vertical separator lines upward into the "total level" tier at the top of the chart (illustrated in Figure 3–5B). The class of the total (major, minor, intermediate) is then entered in the resulting box.

Sequence, number, and optional presence information also varies from run to run. It is entered in the four boxes at the left of the chart below "Card Name" (as illustrated in Figure 3–5C) under the following abbreviated headings:

1. I/O
2. SEQ
3. No. Cds.
4. OPT

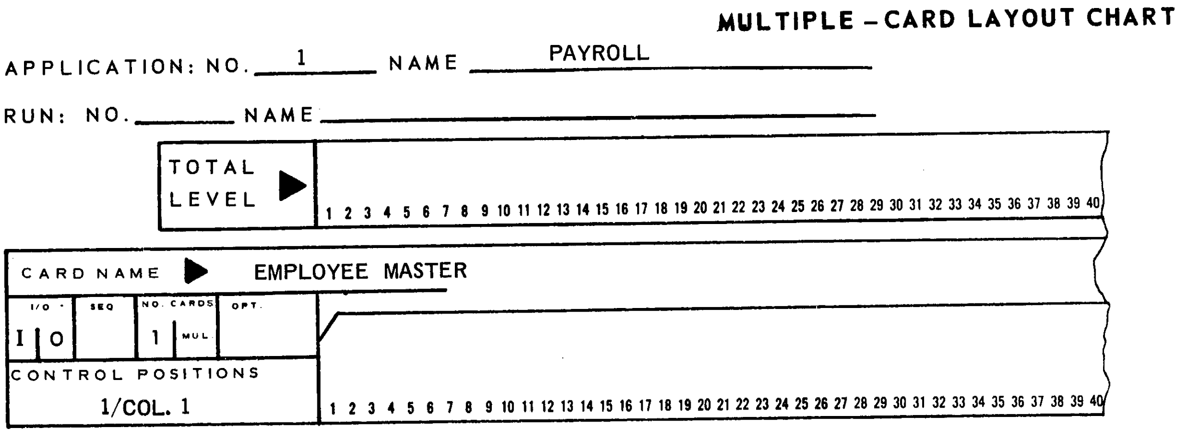

Figure 3–5A. Master Multiple Card Layout Chart: Control Position Entry

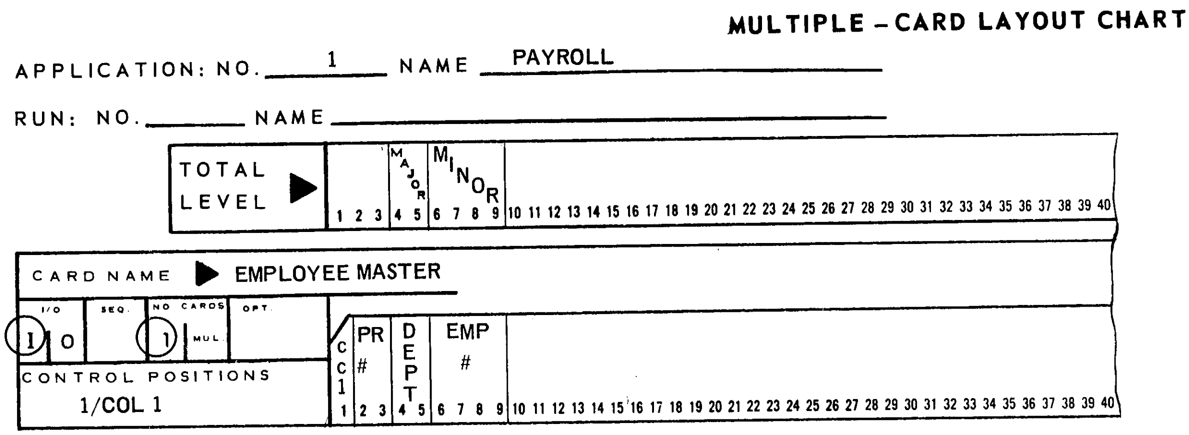

Figure 3–5B. Master Multiple Card Layout Chart: Total Level Control Fields Entries

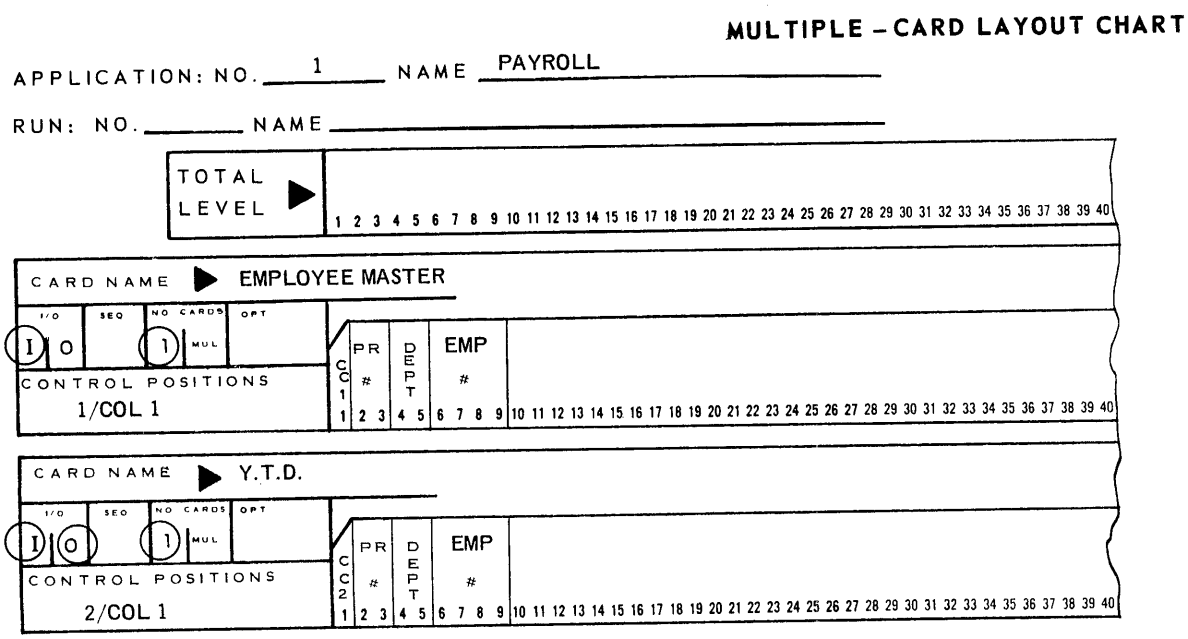

Figure 3–5C. Master Multiple Card Layout Chart: Sequence, Number, and Optional Presence Entries

The I/O column indicates whether a given card type provides input or output for a particular run. An I or an O or both (if output data is punched into the same card used for input) can be entered in this column.

The SEQ column indicates the sequence in which a particular card type is to be processed during a run. A "1" identifies the first card to be processed, "2" the second, and so forth. An "A" is entered if sequence of card types has no significance.

The No. Cds. column indicates whether one only or any number of cards of a given type may appear in a control group. The entry "1" or "MUL" (for multiple) should be made.

An "O" entered in the OPT column indicates that the presence of a given card type is optional.

The Printer Format Chart: Printing format information can usually be obtained by examining existing printouts, such as the one illustrated in Figure 3–6. This information is then entered on the printer format chart supplied by Univac. Figure 3–7 illustrates a printer format chart on which information from the sample printout in Figure 3–6 has been entered. This chart is similar to other printer format charts except for the presence of several columns at the left of the chart in which information to be used later in the development phase can be entered. The application name, run number, and other pertinent data is entered at the top of each chart. Variable characters in each field are represented by X's. Sufficient X's should be entered to fill out the entire field. Detail lines are represented by at least 2 lines of X's. Each total line is represented by a small line of X's. Blank lines indicate multiple line spacing. Special symbols such as commas, periods, etc., should be entered where required. Page and column headings must be entered in the appropriate spaces, even if they are not shown on existing forms.

Chart notations should include all vertical spacing and skipping requirements, including normal overflow to following pages. Forms that require eight-to-the-inch spacing should be identified. A summary of form construction information should be provided for each chart.

XYZ COMPANY

PAYROLL REGISTER

PAY PERIOD AUG. 15-19 PAGE NUMBER 1

EMPLOYEE NO. DEPT	EMPLOYEE NO. MAN	TAX CDE	EMPLOYEE NAME	PAY RATE	HOURS REG	HOURS PREM	EARNINGS REGULAR	EARNINGS PREMIUM	GROSS (YEAR TO DATE)	DEDUCTIONS F.M.T. (YEAR TO DATE)	DEDUCTIONS F.I.C.A. (YEAR TO DATE)	MISC. CDE	MISC. AMOUNT	NET PAY CHECK AMT.
23	1142	M4	A B ARTHUR	2 31	51 9	11 9	119 89	13 86	133 75	11 45		7	7 75	114 55
									6 687 50	572 50				
23	1236	M2	J I JONES	3 56	45 0	5 0	160 20	8 90	169 10	20 03		7	5 75	143 32
									8 455 00	1 001 50				
23	1421	M4	R B SMITHSON	3 916	45 0	5 0	176 22	9 79	185 88	16 39		15	17 75	145 61
									4 800 11	491 70				
			TOTAL CURRENT		141 9	21 9	456 21	32 55	488 73	47 87	6 13		31 25	403 48
			TOTAL Y.T.D.						19 942 61	2 065 70	831 60			
34	1196	S1	B B BROWN	2 47	40 0		98 80		98 80	12 01	3 58	11	8 25	74 96
									4 940 00	600 50	277 20			
34	1300	M2	K L LOPEZ	2 47	31 0		76 57		76 57	7 08	2 78	7	5 75	60 96
									3 828 50	354 00	138 78			
34	1577	M3	J K LIMONSON	3 41	52 0	6 0	177 32	20 46	197 78	42 05	5 44	15	12 75	137 54
									9 889 00	2 102 50	277 20			
			TOTAL CURRENT		123 0	6 0	352 69	20 46	373 15	61 14	11 80		26 75	273 46
			TOTAL Y.T.D.						18 657 50	3 057 00	693 18			
			GRAND TOTAL CURRENT		264 9	27 9	808 90	53 01	861 88	109 01	17 93		58 00	676 94
			GRAND TOTAL Y.T.D.						38 600 11	5 122 70	1 524 78			

Figure 3–6. Sample Printout of a Typical Payroll Application

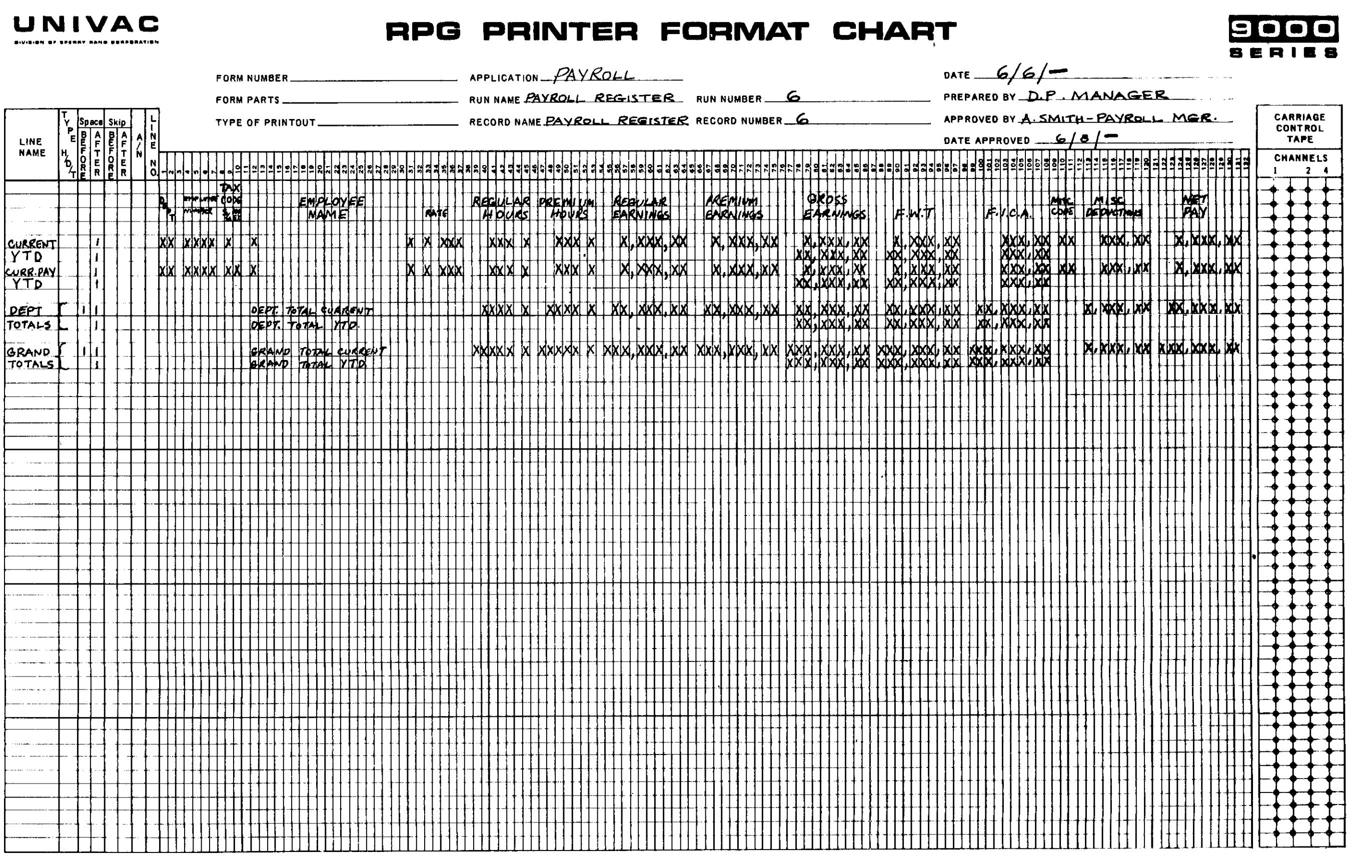

Figure 3–7. Printer Format Chart for a Typical Payroll Application

EVALUATING THE PRESENT SYSTEM

Using the information obtained during the investigation phase, the present system is evaluated to determine whether fundamental improvements can be made either in the data processing department itself or in other departments concerned with data processing operations.

Unless the Data Processing Department has had the benefit of continuing systems and procedures activity and organized program maintenance, there are likely to be known improvements which have not been made simply because other work had higher priority. Other weaknesses and potential improvements will be revealed simply because an objective investigation and evaluation is being made from a new perspective.

Inefficiencies can be located through an intensive study of system documentation, by observing day-to-day operations, and by talking with the users of the system's output. Of particular interest to the interviewer should be the answers to the following questions:

1. What report information does the recipient utilize?
2. What report information is not utilized?
3. What information is needed but not received?

Specific weaknesses to look for include:

1. *Incomplete input* – through oversight, changed conditions, or previous equipment limitations.
2. *Duplication of input* – when insufficient control permits the same transaction to be entered into the system twice.
3. *Late, incomplete, or inaccurate output.*
4. *Redundancy of file and report content.* One of the greatest savings in machine utilization involves proper consolidation and separation of file information (file design and its relationship to processing and sorting are discussed later in this section).
5. *Manual consolidation of reports.* Extreme inefficiency results when report users are forced to extract or consolidate final results from incomplete or improperly constructed reports.
6. *Insufficient accuracy control.* Additional inefficiency is introduced when operation must be halted to locate error sources. There are many techniques available to effectively control input and output accuracy, some of which are discussed later in this section.

Most of the system weaknesses identified are usually correctable. Some may not be, however, and the system designer must devise means of minimizing their effect.

When evaluating the present system it is also necessary to consider all company policy or legal restrictions influencing its design or operation, since they will also apply to the new system.

DECIDING ON A DESIGN APPROACH

With the data gathered during system investigation and evaluation, and knowledge of what management will require of the new system, the system designer is able to determine the best method of systems design. Three approaches are generally recognized: direct change-over, limited redesign, and complete redesign. Each approach has its advantages and disadvantages. In determining the one most appropriate to each application, the system designer must consider to what extent and in what ways the present system is inadequate, what new requirements will be made of the system to be installed, what improvements are possible because of new equipment characteristics, what management expects in the way of data handling capability, and what time and cost limitations management has established.

The three approaches possible and the criteria by which the selection is made are discussed in the following paragraphs.

Direct Change-Over

The principal characteristic of direct change-over is that it utilizes, with a minimum of changes, the basic features of the present system. Changes are normally limited to the conversion of master files to disc, and consolidation of master files that require multiple card records. Inputs and outputs, specific runs, and the calculations performed in them, remain largely unchanged.

Direct change-over is particularly easy to effect with the 9200/9300 Disc System because the addressable sector size is 160 bytes, just twice card capacity. With very little redesign, a 10,000 to 20,000 card master file can be transferred to a single disc surface for direct access.

Direct change-over provides the fastest, simplest, and least costly method of obtaining higher speed processing, providing that the present application is meeting all of it objectives and can be used without design changes.

Limited Redesign

Limited redesign is a moderate approach that takes advantage of direct access techniques as well as the higher speed of the UNIVAC 9200/9300 System. Principal changes are to the internal data processing procedures. Existing files are consolidated, and new files and data processing procedures are created. Inputs and outputs retain the same format but may have minor changes in content. Reports and other documents may also require modification to satisfy added phases.

By reducing the number of machine runs through an efficient direct access oriented design, limited redesign greatly increases the new system's information handling capability without altering the basic nature of the input and output documents.

Limited redesign is indicated if:

1. Present systems design was restricted by limited equipment capability,
2. The present system requires more than minor adjustments in order to meet the new output requirements, or
3. The operation of the present system is notably inefficient.

Complete Redesign

The most time-consuming approach to implementing the new system is complete redesign. A thorough study of all internal and external aspects of the data processing operation must be undertaken to determine the total requirements to be met by the new system. Complete redesign, even where indicated, is not always necessary for the entire system, however. Individual applications, and even parts of applications, may be converted to the new system without a complete redesign effort. The decision as to whether (or how much of) the system must be completely redesigned is based on many factors, including volume of data involved, run frequency, the amount of system design or programming involved, informational requirements, time and manpower available, and the effectiveness of the present system's design.

Complete system redesign is indicated where operating and reporting procedures must be changed to meet new informational requirements.

NEW SYSTEMS DESIGN

New systems design begins after the current system has been documented and analyzed. The new system must, of course, be created with the general characteristics of the 9200/9300 Disc System in mind. Usually a time factor also influences systems design to the extent that there is some degree of pressure to complete the theoretical phase of design and establish a schedule of concrete action. Certain techniques, rules, and conventions of systems design are presented in the following paragraphs.

The first step in new systems design is called hypothesis. It means that a new system is created in concept only so that it can be examined, tested, and modified on paper until agreement is reached that it is the best that can be developed within existing constraints. A flowchart is the vehicle of the hypothesis.

Development of a general flowchart illustrating graphically the flow of information, by application, through the entire system is the first stage of design. Being a master plan, the general flowchart does not show all system details. The format of input and output documents, for example, is not indicated, nor are record layouts and specific programming techniques. In addition to the general flowchart, files, input and output formats, working papers, etc. must be developed. Controls must also be established.

Although primary responsibility for application design lies with the system designer, time limitations may prevent him from knowing and understanding fully all the exceptions and restrictions imposed on the system to be. It is vital therefore that he keep in constant touch with management and with operating personnel responsible for the applications. A small error in the early stages of systems design may well necessitate an eventual major redesign effort if it is not corrected. It is essential, too, that management personnel involved be completely familiar with the proposed system as it is developed. Frequent reviews of the emerging system by both system designer and management are thus highly desirable.

Design Objectives

There are several basic objectives that systems design must satisfy if it is to produce an efficient, trouble-free processing operation.

Standardization

Standardization during the early phases of systems design is most important in the areas of document formats, machine and clerical procedures, and flowcharting conventions. To illustrate the desirability of standardization during the design phase, suppose two divisions of a company are performing similar applications independently, but one operation is automated while the other is done manually. Should the opportunity arise to add the manual operation to the automated system, the work involved would be significantly less if the original design of both operations had utilized the same formats, procedures, and conventions.

The later phases of installation planning, the development of programs and operational procedures, also lend themselves to standardization. During programming, for example, disc, tape, and card formats can be made compatible, thus permitting the use of standard subroutines for handling similar processing procedures. Standardization of conventions (labels, tape sentinels, etc.) is essential if the operating procedures themselves are to be standardized. Use of Univac software conventions is strongly recommended.

A significant benefit of standardization is the ease with which it allows personnel to adapt to new procedures. Reduction of the workload during programming, testing, and the operational phases of the installation, is a further benefit.

Streamlining

Streamlining means the elimination of functions that are of no further practical use to the company's data processing operations. This applies in particular to the generation and inclusion of superfluous information. The content of files should be kept to a minimum consistent with informational requirements. Keeping the number of system controls to a practical minimum is also part of the streamlining function.

Workload Smoothing

Workload smoothing – the evening out of the peaks and valleys in the processing cycle – is vital to the efficient utilization of the system. Although it could increase total processing time, spreading the workload helps to prevent the periodic backup of work waiting to be processed, allowing it to go through the system with fewer delays and without the necessity of costly overtime.

To avoid overloading during peak work periods, operations can be set up so that part of the work is performed ahead of time. Weekly summarizations, for example, will normally reduce the month-end workload. If several applications have the same closing period, the work volume at this time can become critical. Advancing cut-off dates, or increasing the frequency of data entry, will frequently help to lessen the severity of this problem.

Flexibility

Systems design should anticipate the need for future changes. Coding schemes should provide for expansion; runs should not be scheduled so tightly that an expansion in volume will dislocate the system.

The Importance of Documentation

It is difficult to overstress the importance of complete and accurate system documentation. The system designer must see that such documentation is created as part of the design effort. Equally important in ensuring continuing efficiency – since the system is bound to change as personnel responsible for it increase their knowledge of the UNIVAC 9200/9300 Disc System's capabilities, and as the business of which it is a part changes and grows – is the periodic review and updating of documentation to reflect current conditions.

Developing Application Flowcharts

The development of flowcharts is an essential part of systems design. Three types of flowcharts, ranging from those which present the general overview of the system to those which give specific run details, are discussed below.

The Overall Operational Flowchart

The operational flowchart is concerned primarily with the creation and flow of input data and the distribution of system output. Machine room procedures are largely ignored at this stage.

The effort required to develop the operational flowchart depends upon the nature of the application itself and the extent to which the existing application was flowcharted during systems investigation. Figure 3–8 illustrates an operational flowchart for a typical weekly payroll application.

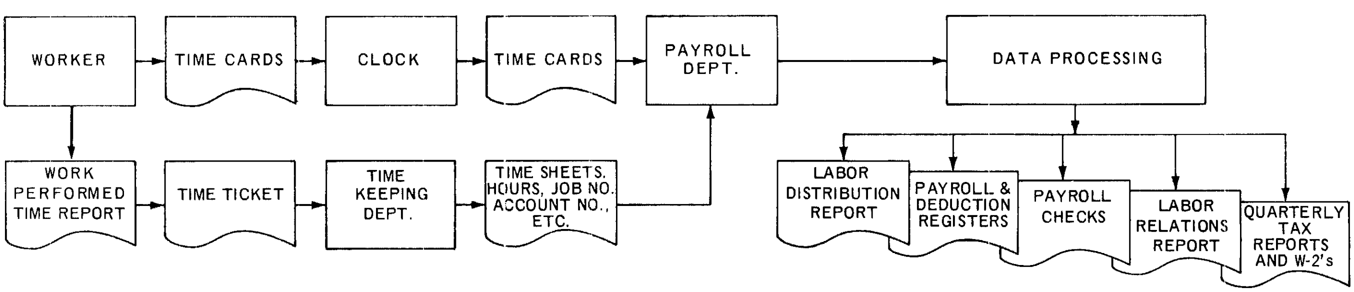

Figure 3–8. Operation Flowchart of a Payroll Application

Task Flowchart

Once the purpose and general concept of the new application has been established and approved, development of a task flowchart showing the general input, output, and run requirements can be undertaken. Information is gathered for this flowchart by determining which reports are byproducts of other reports, what timing and deadline requirements have been established, and what input information is required.

Output reports should be arranged in a logical sequence reflecting daily, weekly, monthly, etc., requirements. Figure 3–9 shows a task flowchart developed for a typical payroll application.

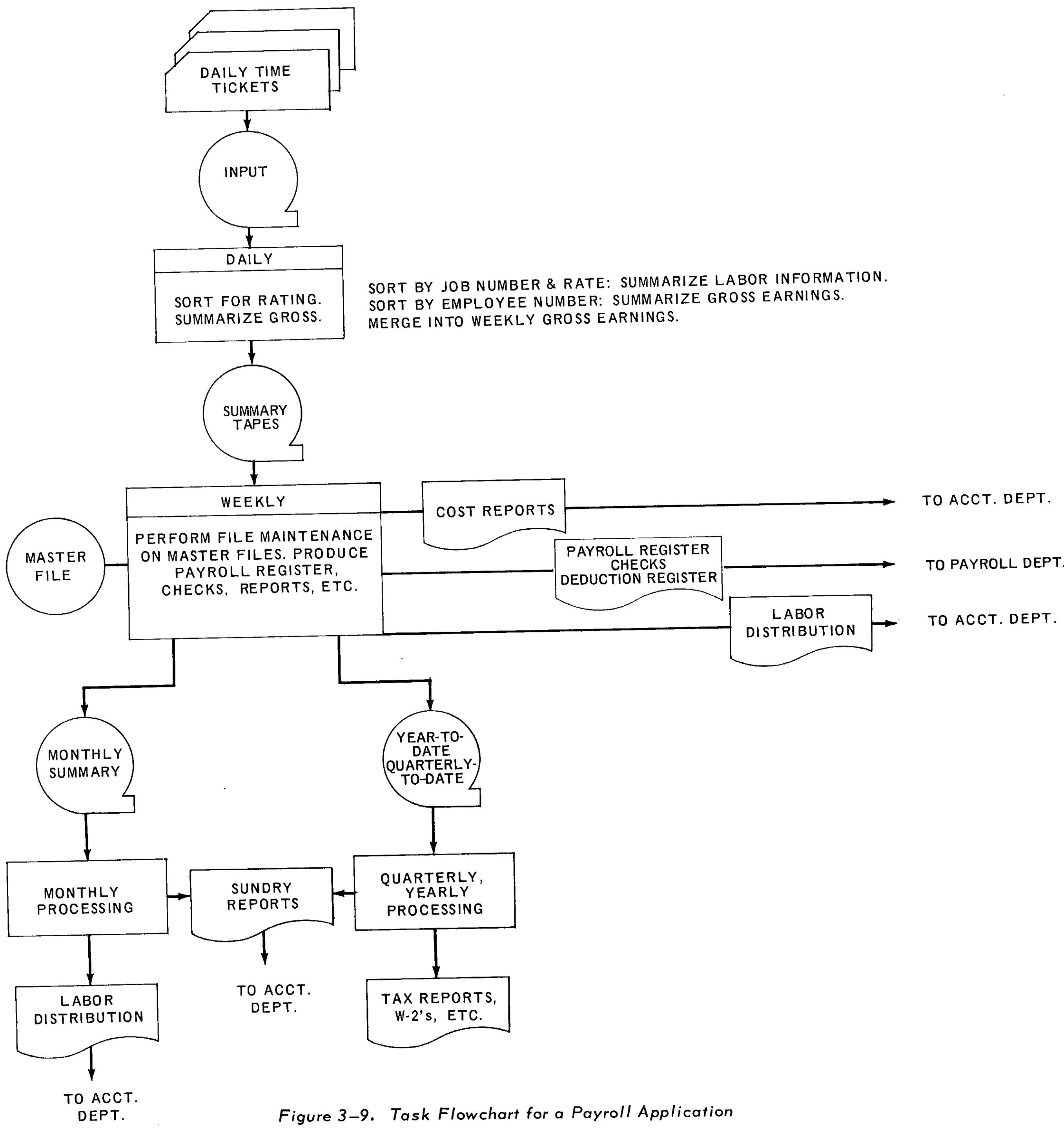

Figure 3–9. Task Flowchart for a Payroll Application

Run Flowcharts

At this point the individual runs involved in each task can be considered. Internal reports developed for control purposes may also be established. The starting point in run flowcharting is normally with the most basic input to the system. Basic input for magnetic tape systems, for example, is normally provided by card-to-tape conversion.

Relating the basic input available to the individual reports to be produced should provide answers to the following questions:

1. What output is to be produced from the input available?
2. What information must be developed as master information?
3. What outputs will eventually become inputs and undergo further processing?

The answers to these questions provide the raw material for the run flowcharts.

Figure 3–10 shows the basic daily runs for a billing application. At this level the knowledge of volumes and frequency and the simplicity or complexity of the tasks divulged by the system investigation are balanced against the anticipated computer power to allocate specific accomplishments to each computer run. Similar run flowcharts would be developed for weekly, monthly, and annual processing related to the application.

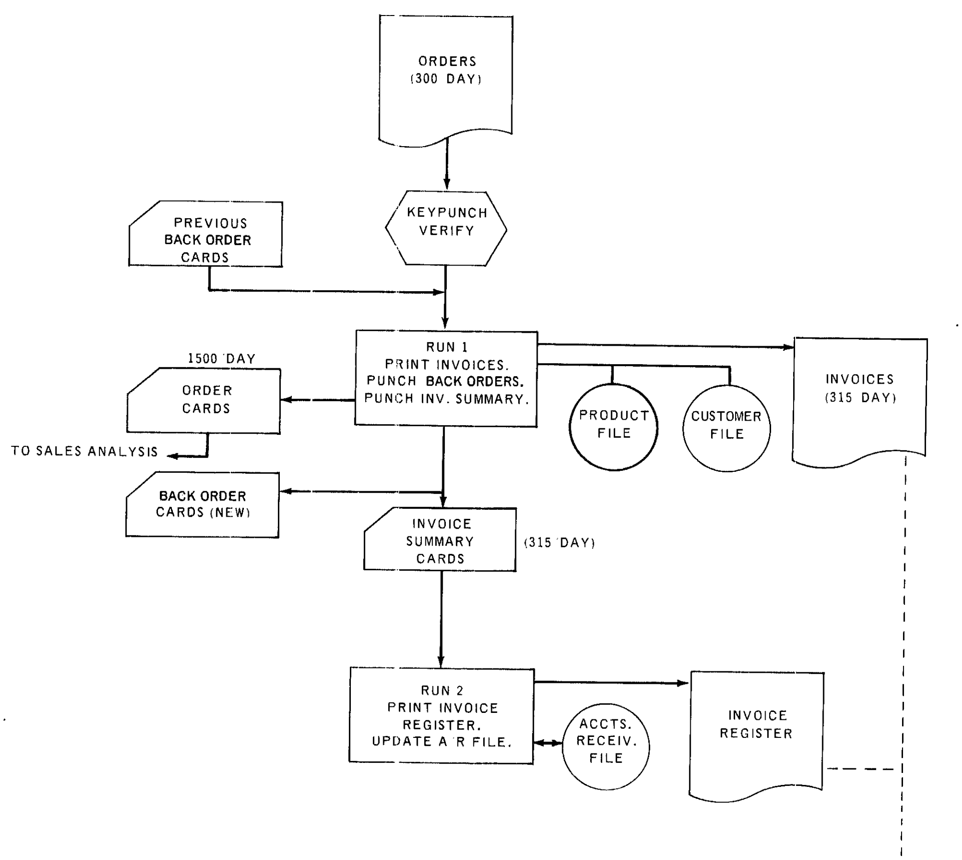

Figure 3–10. Typical Run Flowchart for a Billing Application

Individual runs generally belong to one of three categories:

1. Processing runs

2. File maintenance runs

3. Combined runs

A processing run is a run which processes activity transactions affecting current operations. Figure 3–10 shows processing runs of a typical billing application, including transactions processed against files, calculation of totals, adjustment of balances, and production of new records as output.

A file maintenance run differs from a processing run in that it contains only transactions which affect fixed or constant information maintained in a master file from one reporting cycle to the next. Examples of this type of information in a payroll file are the employee names, addresses, number of dependents. Although such information changes infrequently for individual employees, a number of such changes would probably have to be made each pay period when processing, for example, a 500-man payroll. Before the payroll could be processed, therefore, the master file would have to be updated. Figure 3–11 shows a typical file maintenance run for a master employee file.

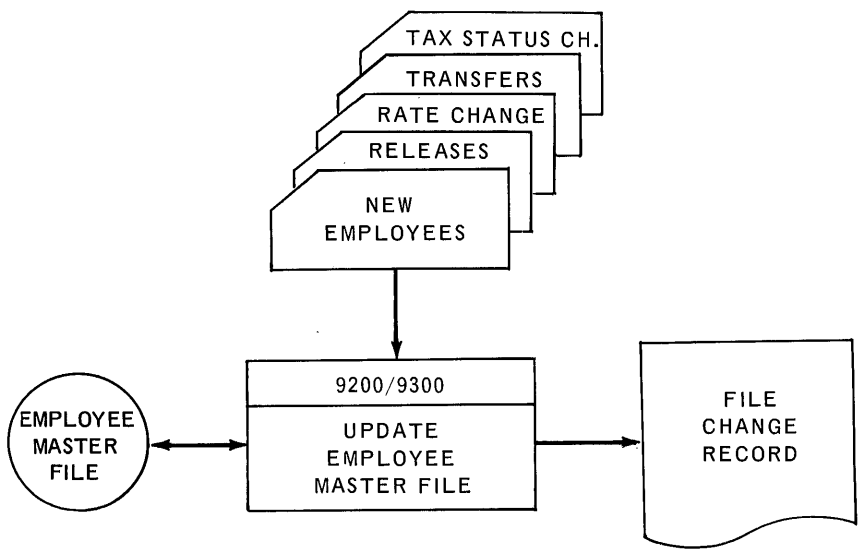

Figure 3–11. Flowchart: Master File Maintenance Run

Figure 3–12 shows a typical run flowchart for inventory file processing. Note that in Figure 3–12, input transaction cards are written onto discs and then sorted before processing against the inventory file. Alternate procedures would be to sort the transaction cards offline or to process them directly in random sequence. The procedure of choice varies from run to run and is determined by analyzing the input volume, key field size, equipment availability, and equipment time used in each technique. If the key field size is small, the card volume large, and the sequence totally unlike the reference file organization, offline card sorting may be best.

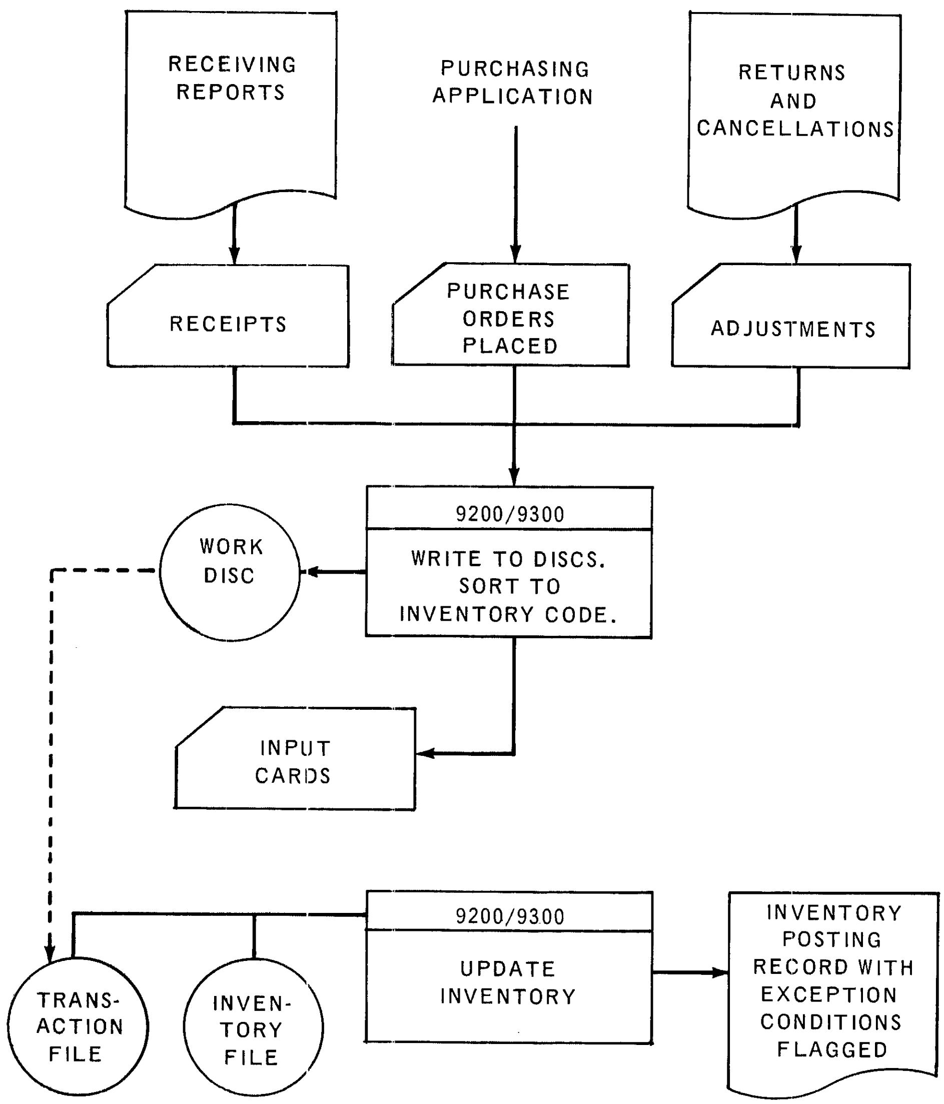

Figure 3–12. Flowchart: Inventory File Processing

A combined run that includes both data processing and file maintenance is often possible. For example, the inventory processing run in Figure 3–12 could be changed to a combined run by introducing file maintenance cards with the transaction cards. Identifying codes in the input cards would determine the action taken on the cards by the program. The advantage of a combined run is the elimination of two separate equipment set-ups and allowance for two runs in the schedule.

Validation Procedures

In the interest of simplicity, validation procedures were not shown in the preceding example. Validation is essential to effective data processing, however, and provisions for it must be included in the design of the application. The amount of validation built into the system depends on the type and history of the application. Applications which directly and immediately result in money movement to or from the business demand complete validation. Billing, Accounts Receivable, Accounts Payable, and Payroll are in this category. The systems design must provide for the correction and later processing of transactions rejected by the validation tests.

Work Files

The run flowchart states the logic involved in accomplishing the task, but does not indicate specific file content or input or output document format. In some runs a work file, either disc or magnetic tape, may be shown. Although the exact content of the work file may not be documented, the designer is aware that the information desired is available at this point. In Figure 3–12 a work file (disc) is introduced for a card-to-disc conversion and subsequent sort operation; the system designer is aware that transactions in the proper sequence and format have become available for later use. A work file is frequently created as output of a run which is logically capable of producing several output reports from the data processed during the run, but is physically limited to one output report because there is only one printer. A payroll run, for example, might easily produce paychecks and earnings statements, a payroll register, and deduction register in a single run. If, while checks and earnings statements are being printed, the lines of the payroll register and deduction register are written out to a disc or tape for later printing, that disc or tape is a work file. Later printing runs would pick only the correct lines from the work file for their output.

Developing Input/Output Specifications

The development of system I/O specifications must go hand in hand with the development of system flowcharts. Unless this is done, there will be little continuity of information between the runs.

Input Considerations

The system designer must be certain that the procedures and documents necessary to supply all the information required by the system (including that representing exceptions) are provided. This includes the design of forms which will furnish information for keypunching. Such forms are generally formatted to expedite coding and keypunching rather than computer operation, since the data can easily be reformatted by the computer when it is entered as input data.

Selection of transaction information for input to the system should not include that which is either constant or predictable. This type of information is normally contained in the master files or is generated by the system itself.

Output Considerations

The quality of system output is particularly important since the output document is the principal communication link between the data processing system and the people it serves. The primary consideration in output design is satisfaction of the recipient's needs. Content, format, timeliness, and accuracy specifications must be developed, therefore, with the user in mind.

Another consideration is the determination of the type and format of output forms. Questions such as the following must be answered: Who is to use the form? How many copies will be needed? What is the report frequency?

Although it may reduce the system's throughput rate (depending on the number of lines printed), allowing the computer itself to print column headings on stock paper will reduce forms costs. Cost and time can also be reduced occasionally by utilizing a side-by-side printing technique. This technique is commonly used for applications such as label printing, where as many as four labels can be printed simultaneously.

File Design and Maintenance

Two types of files are normally used in a direct access system: The master file containing information that is either fixed or subject to infrequent change, and the data transaction file, which contains the information reflecting normal day-to-day activity.

Master File Content

The master file is made up of permanent information pertaining to a given item, product, account, or person. The term permanent means that the information is maintained from one processing cycle to the next, but does not imply that it is never subject to change. A typical master record for a payroll application might contain the following information:

1. Employee identification number
2. Employee name
3. Employee address
4. Social Security number
5. Number of dependents
6. Deduction codes
7. Deduction amounts
8. Quarter and YTD total figures

The determination of what information should be allocated to the master file and what should be part of the normal day-to-day transaction data is based on the frequency with which it changes. Certain deductions (for example, bonds and union dues) are usually continuous from one pay period to the next, and are included in the master file. Other deductions (such as equipment purchases) are not continuing and belong in the data transaction file.

Master File Sequence

The efficiency of programs which process data against master files is affected by the organization of the master file on the file media. In an indexed sequential or random file, the degree to which both the master and transaction file are in the same sequence will govern the amount of search time and head movement required to locate each active item in the master file. The degree to which the master file is compacted will affect the total number of disc tracks over which the heads will have to move. Since master files are generally the most voluminous files in the system, every effort should be made to select an organization and access method which conserves space and minimizes search time while meeting the objectives of the run.

Master File Consolidation

The greater efficiency and larger processing capacity of the UNIVAC 9200/9300 Disc System, and the fact that records of up to 160 bytes may be used with the 8410 DAS, increases the feasibility of master file consolidation.

The desired result of consolidation is the simplification of processing procedures. Fewer files mean less file handling. More information is processed with each use of the master file. Fewer disc cartridges are required because information duplication is reduced.

There is a limit to profitable consolidation, however. Consolidation should be utilized only when the information between existing files is related. Specialized files sometimes provide greater advantage than consolidation. For example, the size of the specialized master file, because it contains only information necessary for that run, is smaller. As a result, the running time is usually less. In addition, since the consolidated file must accommodate more types of transactions, it could reach proportions that would make consolidation unfeasible. The point at which consolidation becomes practical depends on the compatibility of transaction processing and the nature of the runs involved.

ESTABLISHING DATA CONTROL

Providing the controls that assure accuracy and validity of data as it passes through the system is also an essential part of systems design. Control of input to the system is usually the most critical, since it is during this phase that data is transferred from one media to another through human action. Methods of controlling data within the computer room are more or less standard, as the intent is merely to guarantee the correctness of procedures and data output. Since fewer natural journals are produced as byproducts of direct access data processing, systems design must take into consideration the need for creating sufficient journals for auditing purposes.

Input Control and Validation

The primary purpose of input control is to assure that every transaction presented to the system is a valid one and that it is entered correctly. Once data has been entered incorrectly, recovery becomes difficult because of the speed with which it is propagated throughout the system.

Controls that aid in assuring the correctness of input include:

1. Prenumbered source documents
2. Hash totals developed at the point of origin and carried through as far as possible in the system
3. Batch totals
4. Cross-checking of manual operations
5. Automatic production of the data through the use of devices such as time clocks, counters, etc.

The use of a verifier is extremely important in obtaining accurate recording of input data from the keypunch operation. Transposition of numbers, for example, a frequent human error, can be caught by the verifier.

Validity checking of data entering the system is accomplished in a number of ways. For example, limit checks – resulting in computer rejection of any data whose value exceeds predetermined limits – is useful in many types of applications. "Reasonableness" checks are a further possibility. The advantage of using the computer for this type of checking is the speed with which the tests are performed and the certainty that it will test every item.

Internal Control

Two types of control are standard once the data has been recorded and entered into the system: (1) Operational control to assure correct run sequence and data entry, and (2) the control required to assure the accuracy of the data itself.

Label Checking

Since data on a magnetic tape or disc is not visible, files must be labeled to identify content, date produced, etc. External labeling alone, however, does not always provide absolute identification of the file's content. To guard against the possibility of an externally mislabeled file, or the inadvertent selection of the wrong file, the same labeling information is included in the first data block of each file. This allows the computer itself to make certain it's using the correct disc or tape reel.

Label checking subroutines are incorporated in all tape and disc handling software for the UNIVAC 9200/9300 Disc System, and are performed automatically unless the programmer specifies otherwise.

Item Counts

For systems including magnetic tape, item counting is a conventional device used to ensure that individual transactions are not lost during data manipulation. Transactions are counted as they are written to a magnetic tape reel, the total count being entered on the reel's final record. When the reel is subsequently used as input to another run, the transactions are again counted and the new total previously entered in the reel's final record.

Checklists

Checkoff lists of incoming and outgoing data, used in conjunction with batch proof totals, provide assurance that all data is accounted for. The use of batch totals alone as a validation procedure guarantees only that the particular batch being processed has not lost or gained any items during prior manipulations. It does not detect the loss of an entire batch – an added security that the checklist provides.

Control Totals

Control totals, although they are incorporated at input and retained through output, are used principally to detect the inadvertent loss or gain of items during processing. However, their use can be extended to include the detection of incorrect data within an item. For example, if the quantity of a particular item in which the quantity field is accumulated for a group total is destroyed or changed in prior processing, the control total of the individual field will reflect a data error. Other fields, however, will reflect correct dollar value control totals and the incorrect item can be isolated through the use of the individual field total.

The efficient use of control totals depends in part upon the determination of an optimum batch size. Too large a batch makes item searching difficult. At the same time, too small a batch requires excessive control handling because of the larger number of batches involved.

Output Control

The data processing center should maintain a continuing record of report distribution. To verify report results, such records should include dates and totals.

Error control is another consideration. Control must be established to guarantee that all errors listed and turned over to accounting personnel for correction are reentered in the system prior to the completion of the processing cycle. This can be accomplished by totaling each batch of errors for cross-checking with correction entries.

Auditing Considerations

With manual or punched card data processing systems, transactions are normally recorded frequently during processing, producing an automatic audit trail in much the same way as posting to a ledger produces a day-by-day journal of transactions. With direct access systems, however, fewer natural journals such as these are produced as run byproducts. Also, since updated record fields are written over the old data, there is no permanent record on disc of past transactions.

In order to determine the accuracy with which transactions are processed, it is necessary for the system designer to devise a means of tracing all the transactions which took place during a month's processing period. One such device is called a "change" journal, which can be produced, for example, as the byproduct of a maintenance run. As the maintenance run is made, changes to the file are recorded on a magnetic tape or printed journal to serve as a record of the transactions which took place. Transactions can then be traced at a later date through a study of the journals produced.

Another method is the creation of a special test deck containing as many different transaction types as can be identified. These transactions are then fed through the processing cycle singly, and in various combinations, against a copy of the files to determine the effect that each type and each combination of types has on the system.

4. PROGRAMMING

INTRODUCTION

After the complete system is charted, duplication eliminated, records designed, controls and auditing trails established, the job of programming can begin. Section 4 stresses the importance to program development of careful planning and common sense. Although this section describes a variety of programming practices directed at producing effective and error-free programs, it is not intended to be a programming text. It augments, but in no way replaces, the programmer's own knowledge and ability. The reference manuals listed in Section 3 of this document are the primary sources of programming information for the 9200/9300 Disc System.

It is recognized that the size of the staff and the scope of the programming effort will vary widely among UNIVAC 9200/9300 Disc System users. In some installations, the programming staff will be responsible for writing and testing programs. In other installations, programmers may have systems design and operational responsibilities in addition to programming assignments. In either case, the need for established policy and procedures cannot be overstated.

PLANNING FOR PROGRAMMING

File Conventions

Establish file conventions for all types of files to be read into, or produced by, the UNIVAC 9200/9300 Disc System. Whenever possible, the conventions for different types of files should be similar. The operating systems provided by Univac are based upon certain conventions. The 9200/9300 System programmer reference manuals should be reviewed to determine what standard conventions may be used to satisfy the installation's needs.

Programmer Training

Programmers will usually have to participate in training sessions before they can be given specific programming assignments. In addition to knowing how to program the UNIVAC 9200/9300 Disc System, they must have adequate knowledge of the user's business and of company policies. Univac is prepared to participate in training personnel to program the UNIVAC 9200/9300 Disc System. The user must provide training in company policy and procedures.

Programming Procedures

It is usually wise to provide at least a minimum number of programming procedures. Such procedures are of a control nature and provide a uniform operating guide for programmers which is independent of specific applications. Suggested general procedures are discussed below.

Coding Conventions

An example of this type of convention is the preassignment of certain ranges of the 99 different RPG indicators to specific tasks. Indicators 1 through 9, for example, might be used as record type indicators; 10 through 19 could be used as comparison result indicators. The use of such conventions makes it easier for the other people in the installation to understand and, when necessary, revise runs they did not program.

Other coding conventions might include the use of common labels for input fields used by more than one program, or the writing of statements on every other coding paper line to allow for adding correction lines (or leaving five open lines at the bottom of each page for the same purpose). One coding technique – not really a convention, but deserving of mention because it enhances the understanding of the program by others and helps the programmer in testing a run – is the use of comments. The entry of comments in the specified coding form areas aids in dividing the program into logical subroutines and assists in isolating specific subsections of the run for correction or revision.

Coding conventions are based on practicality. They should therefore not be so numerous as to cause the programmer to worry more about conventions than about the program itself, but should be devised with the idea of assisting in the communication between installation personnel.

Assembly or Compilation Procedures

Procedures to be used for program assembly or compilation must also be considered. If an open shop (one in which programmers use the machine themselves) is contemplated, the programmers must be trained to operate the computer. If a closed shop policy (allowing only operators to run the machine) is to prevail, the procedure for transmittal of source code, other needed data, and the return of assembly output to the programmer must be established. The procedure to be followed if an assembly or compilation is aborted should also be considered. Suggestions on which to build this procedure are covered later in this section.

Program Test Procedures

The method and procedures to be followed during all stages of program debugging should be considered. The degree of test and the standards to be met by each program before it becomes operational must be fixed. Procedures should be established for ensuring that adequate instructions are available for operating each program at the time it is delivered to the operations staff. Information about common test techniques and run book layouts is covered in other parts of this section. A procedure to be followed for placement of operational programs on a system tape must also be formulated.

STAGES OF PROGRAM DEVELOPMENT

All computer programs pass through many stages of development on their way to becoming fully operational. These stages are described in subsequent sections of this chapter.

Generally, implementation of a computer program comprises the following major steps:

- Reviewing systems design material.
- Preparation of a gross design; discussion and adjustment of the design during a subsequent review.
- Flowcharting the program logic. (The flowchart should clearly indicate the composition and logic of the program.)
- Writing the source code for each program.
- Assembling the source modules.
- Analyzing and correcting errors indicated by the assembly printout.
- Correcting the source code and reassembling.
- Preparing test data for use in testing the object program.
- Testing the object program, revising and reassembling the source code.
- Updating flowcharts and other technical documentation to include all changes.
- Generating a larger set of "live" test data which will provide a thorough test of the entire program.
- Retesting the program using the "live" data – correcting and reassembling the program if errors are detected.
- Removing any program testing aids embedded in the program, and retesting.
- Documenting both technical and operating instructions for inclusion in a Program Run Book when the program is error free.
- Releasing the program for acceptance testing and filing in the system's library.

CONTROLLING PROGRAM DEVELOPMENT

It can be seen from the number of stages involved in implementing a program that a method must be found for preventing production delays. For a programmer to make productive use of his time, it is often necessary that he have several programs in various stages of completion. Should delays then be experienced in obtaining computer time for assembly or test, another program is available for him to work on.

To keep the number of programs in progress to a manageable level, each programmer should maintain a production control document showing the current status of the programs assigned to him. He should give top priority to the completion of the programs that are more nearly finished. Programs will thus be completed at regular intervals, keeping to a minimum the number of unfinished programs in progress.

DESIGNING THE PROGRAM

Review of Systems Design

The design of individual programs is influenced by several factors. The first programming task after assignment of a specific job, however, should be the thorough review of the systems design material. This is particularly important if the programmer is not also the system designer. The systems design review will greatly influence the program content since it shows the interaction of individual runs within an application and points out the general processing required to produce the desired results. The systems design review should also establish the completeness of the design and determine when actual programming can begin.

Consideration of 9200/9300 Software Capabilities

As a practical matter, the design of each program must take into consideration the capabilities of the computer on which it will be run. Specific software capabilities which the UNIVAC 9200/9300 Disc System user will want to consider are discussed below. These and other capabilities are explained in detail in the operating system manuals.

Concurrent Operation

For systems which run concurrent programs (that is, more than one job sharing main storage and the computer at a time) consideration must be given to the amount of main storage used by the other program. Although it is usually difficult to determine the main storage utilization of a program in advance, an experienced programmer will have a "feel" for this kind of estimate. Knowing the approximate size of the program to be run concurrently, he can look more closely at program techniques which will save main storage in his own program.

Program Segmentation

Programs for the UNIVAC 9200/9300 Disc System can be subdivided into segments in a fashion that permits only a part of the program to be loaded into the computer at any one time. If the program is designed in this manner, the Executive Routine can be requested to search the systems disc and load any needed part of the program into a specified part of main storage. Each segment that can be so treated is called a program overlay. This technique is generally used when the program logic is complex enough to cause the program to exceed the capacity of main storage. Another technique which can be used when a program is too large is to subdivide it into two or more programs. In this case, two or more passes over the data are required to complete the processing.

Program Checkpoints

It is recommended that periodic checkpoints be included in any program expected to take a considerable amount of time to execute. Checkpoints cause the contents of main storage and other pertinent data to be recorded on an output file. The operating system uses this information to restart the program from an intermediate point rather than reprocessing the entire program.

The programmer should decide on the manner in which checkpoints are to be established. They may be established at intervals based upon the processing of a fixed number of records, at the time the end-of-file or end-of-reel is detected, or at any other point the programmer chooses.

Program Testing Aids

UNIVAC 9200/9300 Disc System software provides a main storage dump subroutine which may be incorporated in the problem program. If assembly language is being used, the main storage dump subroutine can be used to produce periodic main storage dumps on discs, tape, or printed forms under control of the problem program. The software also contains a utility tape/disc print program for use in selectively printing the contents of a dump. If the storage dump subroutine is not incorporated in the program, a storage dump routine can be loaded through the card reader. This routine causes the contents of main storage to be dumped through the printer. These main storage dumps are helpful in tracing program bugs which develop because of errors in program logic.

PROGRAM FLOWCHARTING

The importance of proper flowcharting cannot be overstated. A good flowchart is vital in both planning the program and in testing it after it has been written. One of the benefits of flowcharting is that it provides a written record of the program in capsule form. A flowchart can be used by anyone to review and understand the analysis or solution of a problem.

Several levels of flowcharts are developed during the systems design and programming phases of the installation. The first levels (discussed in Section 3) will normally be produced before the programming effort begins. The final and most detailed level is produced as part of the programming effort.

The detailed program flowchart should show the complete logic of the program as it applies to the computer. It should be detailed enough to provide the perfect document from which to code the run. An example of a final flowchart is shown in Figure 4–1.

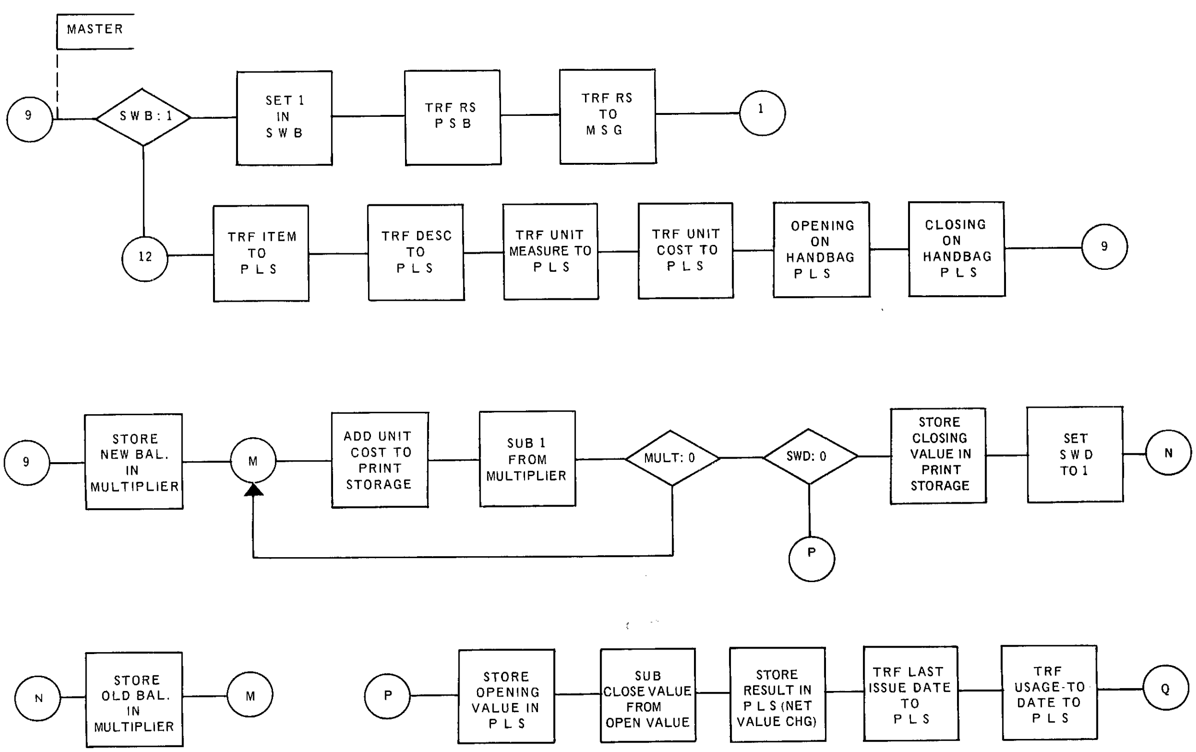

Figure 4–1. Section of a Detailed Program Flowchart for an Inventory Stock Status Report

The Univac Template (UD-1-723, Rev. 1–66) and instructions for using it are available through your Univac sales representative. The template conforms to both American and European standards. It is recommended that these standards be followed in all installations.

WRITING SOURCE CODE

A primary programming function is the translation of system requirements into a medium that can ultimately be loaded into the computer to perform the required processing. This is accomplished by writing source code statements in symbolic assembly language or by preparing Report Program Generator forms.

Detailed instructions for writing source code statements or completing Report Program Generator forms are contained in the following programmer reference manuals:

- UNIVAC 9200/9300 Systems Card Report Program Generator Reference (UP-4106 Rev. 1)
- UNIVAC 9200/9300 Systems Tape and Disc Report Program Generator Programmers Reference (UP-7620)
- UNIVAC 9200/9300 Systems Card Assembler Programmers Reference (UP-4092 Rev. 2)
- UNIVAC 9300 System Tape Assembler Programmers Reference (UP-7508 Rev. 1)

Report Program Generator (RPG)

Most business data processing installations produce a wide variety of reports. Each report usually requires a separate computer program or punched card tabulating equipment plugboard. Univac software, however, includes a Report Program Generator which automatically produces efficient and accurate programs for each type of report to be produced.

A description of the input file, the calculations to be performed (if any) and the report format is written on specially designed forms. This information is then keypunched into cards which are used as input to the Report Program Generator. The specifications are processed by the Report Program Generator to produce completely executable programs. These programs can then be filed in the library of executable programs and used repeatedly to produce the associated report.

Each generated program has a standard functional design. The specific processing of the input data and the information printed on the output forms results from the specifications furnished to the Report Program Generator. In general, the following processing steps are performed:

- The generated program obtains an input record.
- The record's type is established and the validity of the presence of this record type in the current sequence of input records is checked.
- When a control break is detected, internal indicators are automatically set for this and all lower level control breaks.

- Total-time calculations are conditionally performed and total lines are generated for output.
- According to record type, input fields are moved to the named working storage location and indicators are set according to the values of those fields.
- Detail-time calculations are conditionally performed and detail lines are generated for output.

Symbolic Assembly Language

The basic symbolic assembler provides a symbolic language for programming the UNIVAC 9200/9300 Disc System. This language provides a convenient means of writing machine instructions, designating registers, and specifying the format and addresses of storage areas, data, and constants.

Assembler output includes: (1) object modules suitable for linking into executable programs, and (2) a printout which lists source code, object code, and diagnostic messages.

The alphabetic mnemonic machine instruction codes are assigned names which suggest the function of the corresponding instruction. Assembler directives are included in the symbolic assembly language to provide for generation of data words, values, or modification of assembly control. In some instances, additional source code and certain control parameters can be supplied through control stream job commands during assembly. Procedures or macro instructions – user-coded subroutines which can be written once, placed in the software library, then incorporated in any program – can be defined and used during assembly.

SELECTING A PROGRAMMING LANGUAGE

It is sometimes difficult when a choice of programming languages is available to select the proper one for a particular program. In many cases more than one will do the job. Rather than attempt to tie a specific language to a specific application or installation, the characteristics of the languages should be weighed against other factors – such as available time, main storage availability, etc. – to determine the best language for the majority of the programming. To help you make this decision the following paragraphs discuss some of the characteristics of the four available program languages.

Assembler

The assembler, the most flexible of the programming languages, is the basis for the other types of languages. The assembler is the only type of lower level language which utilizes machine code. One of its outstanding advantages, consequently, is that it gives the user complete control of the computer's hardware operation. To use the assembler well, the programmer should have a thorough knowledge of how the computer operates and what the program is to accomplish. Using it properly will normally produce the most efficient programs in terms of main storage utilization and speed.

The assembler, however, has some drawbacks. Because it is a lower level language, more coding is necessary. Because the coding is more detailed, debugging may take somewhat longer than with other methods. As a result, the experience of the programmer is often the deciding factor in electing to use the assembler. In general, however, the more complex the program, the more likely the assembler will be used to produce optimum quality programs.

Report Program Generator (RPG)

RPG is an excellent language for the beginner to use and for jobs which do not require sophisticated input/output operations. Being problem-oriented rather than machine-oriented, it lessens the need for knowledge of how the computer operates. Since it has the characteristics of a compiler (it produces several object instructions for each source code statement), it also requires fewer lines of coding.

Designed specifically for the UNIVAC 9200/9300 System, 9200/9300 RPG takes advantage of the computer's strong points to produce efficient programs with a minimum of programming effort. RPG is relatively simple to learn, and offers the psychological advantage of resembling English rather than a programming language.

Although RPG allows the programmer only limited control over input and output functions, the incidence of runs requiring sophisticated I/O control is low in most installations.

TRANSLATING SOURCE CODE TO OBJECT CODE

Regardless of the programming language selected, the source language must be translated into machine language. This process is called "assembly" for the assembler, "generation" for the RPG, and "compilation" for COBOL or FORTRAN. For simplicity in this section, however, we will refer to the translation process in all cases as compilation.

An important function of the compilation process is checking the completeness and accuracy of the source code according to the rules of the symbolic language. While the compilation cannot check program logic, it does check the fundamental content of each source code statement and the interrelationship of these statements.

When a program is being compiled for the first time, it is usually advantageous to postpone the punching of object code card decks or the scheduling of tests until the assembly printout has been reviewed for diagnostic messages. The investigation of the diagnostic messages and subsequent correction of the indicated errors should precede the linking and testing of compiled object code.

Source code cards can be first transcribed to discs and the assembly made fully disc-oriented. Also, the compiled output can be printed or punched directly, or it can be first written on disc and then printed and punched by a disc-to-card or disc-to-print program.

The success of compilation is predicated upon many things. Accuracy and completeness of source code input to compilation is essential. Proper operation of the computer during the process is also of prime importance. The computer programmer or operator must fully understand his responsibilities in compilation. Procedures that a programmer might follow to increase the likelihood of successful compilation include the following:

Write source code statements carefully and legibly – The keypunch and verifier operators are less likely to make mistakes if the entries on the source code form are legible. Mistakes in punched card content must be eliminated before a program can become operational.

Establish and use a card control system – To make sure that the source code is accurate and complete, it is recommended that a sequential control number be entered in each source code line and punched into the corresponding card.

Check source code forms for accuracy and completeness before punched cards are prepared – Each source code statement line should be reviewed to make sure that it is complete. Each reference to another source code statement should be checked to verify that the associated statement has been included and the name or relative address is accurate. Verify the source code logic using the flowchart.

Cross index the source code and flowchart – As part of the verification of program logic, transcribe the names of source code lines to the flowchart. This procedure will facilitate the use of the flowcharts during the program test stages.

Check source code punched cards – To ensure punching accuracy, source code cards, or a listing of them, should be checked against the original source code forms.

Correct source code deck errors prior to compilation – Prepare a new source code form for the cards to be corrected. If possible, the corrected cards should be prepared by a fully qualified keypunch operator. The card content of the newly keypunched cards should be carefully checked. The incorrect cards should be removed from the source code deck and the new cards filed in the deck at their proper place.

Provide complete instructions for the computer operator – If the programmer is to operate the computer during compilation, he should prepare a complete plan of operation before starting. This plan should cover all contingencies and should be in his possession during the assembly.

If someone other than the programmer is to operate the computer, the programmer should prepare a complete and detailed set of operator instructions indicating any special conditions which should be considered. Written instructions should be furnished for the following:

(1) The card-to-disc processing for a disc compilation.

(2) The operating instructions for the compiling program.

(3) The disposition of the input files and the compiler output.

COMPILATION PRINTOUT

The compilation printout provides a complete list of object program statements for reference during program testing. The entire content of the source code card deck is listed, together with the object code developed by the assembly. Figure 4–2 is a sample of an assembler printout; other languages offer printouts of a similar nature. Following is an explanation of the information appearing in each numbered column.

Column		*Contents*
(1)	–	contains a four-digit decimal card number indicating the number and sequence of each source code card read.
(2)	–	contains error flags identifying errors or inconsistencies detected during the assembly.
(3)	–	contains a four-digit hexadecimal number indicating the relative address of the object code resulting from the corresponding source code statement.
(4)	–	contains the resultant object code in hexadecimal notation (two characters per byte).
(5)	–	contains the contents of the source code punched cards. For instructions or assembler directives it contains the label, the operation code, the operand, and the comments (in that order). This entire area contains only comments when additional comment lines have been used.
(6)	–	contains the information punched in Columns 72–80 of the source code card. This information is included in the printout but otherwise ignored during the assembly. These columns may be used for source code control information.

```
 1    2     3     4                       5                                                    6
0001                            CONC     START 4096                                        CONC     0101
0002        1000                         USING *,0                                         CONC     0102
0003                                     USING *,1                                         CONC     0103
0004                                     USING *,2                                         CONC     0104
0005                                     USING *,3                                         CONC     0105
0006        1000 00             TEST     DC     XL1'0'             KEY_IN LOCATION                  0212
0007S       1001 C3C1D9C4E2D6E4E3  FID   DC     CL8'CARDSOUT'      FILE ID        TAPE OUT  CONC    0106
0008        1009 F1F0F2F7F6F7   CDT      DC     CL6'102767'        CREATION DATE            CONC    0107
0009        100F F0F0F0F0       GNO      DC     CL4'0000'          GENERATION NO            CONC    0108
0010        1013 F0F0           VOL      DC     CL2'00'            VOLUME NO                CONC    0109
0011        1015 F1F0F2F6F6F7   XDT      DC     CL6'102667'        EXPIRATUON DATE          CONC    0110
0012        1018                CBUF     DS     CL80               CARD BUFFER AREA         CONC    0201
0013        1068                CARD     DS     CL80               CARD WORK AREA           CONC    0204
0014        10B8                LINE     DS     12CL11             PRINTER WORK AREA        CONC    0205
0015S       113F                DIN1     DS     10CL80
0016        145F                DIN2     DS     10CL80
0017        177F                DOT1     DS     10CL80             PRIME TAPE OUT RAEA      CONC    0208
0018        1A9F                DOT2     DS     10CL80             SECONDARY TAPE OUT AREA  CONC    0209
0019        1DBF FF             RDBY     DC     XL1'FF'            READER BYPASS SWITCH     CONC    0210
0020        1DC0 FF             PTBY     DC     XL1'FF'            PRINTER BYPASS SWITCH    CONC    0211
0021        1DC1 00             CCXF     DC     XL1'0'                                      CONC    0213
0022        0008                EQ       EQU    8                  OP1 = OP2                        0214
0023        0004                LOW      EQU    4                  OP1 < OP2                        0215
0024        0002                HIW      EQU    2                  OP1 > OP2                        0216
0025        0006                UNEQ     EQU    6                  OP1   OP2  AL                    0217
0026        000F                ALL      EQU    15                 UNCONDITIONAL                    0218
0027        0000                NOOP     EQU    0                                                   0219
0028                                     ENTRY CBUF                CARD BUFFER              CONC    0303
0029                                     ENTRY ENDC                CARD EOF LABEL           CONC    0304
```

Figure 4–2. Typical Compilation Printout (Part 1 of 6)

```
0030S                                 ENTRY DIN1
0031                                  ENTRY DIN2                                 CONC   0306
0032                                  ENTRY DOT1                                 CONC   0307
0033                                  ENTRY DOT2                                 CONC   0308
0034                                  EXTRN DCI        I/O CALLS                 CONC   0309
0035                                  EXTRN DTI                                  CONC   0310
0036                                  EXTRN DTO                                  CONC   0311
0037                                  EXTRN DPO                                  CONC   0322
0038                                  ENTRY CCXF       1001  FUNC                CONC   0323
0039                                  ENTRY C2         PRINTER BUSY / RETURN TAG CONC   0324
0040                                  ENTRY C5         READER BUSYY / RETURN TAG CONC   0325
0041                                  EXTRN B?OT       PRINTER BUSY TEST         CONC   0326
0042                                  EXTRN D?OT       1001 BUSY TEST            CONC   0327
0043                                  ENTRY ENDT       TAPE EOF LABEL            CONC   0328
0044S  1DC2 4700 1D C6          STRT  BC    0,C1
0045   1DC6 45D0 1E 72          C1    BAL   13,KEY                               CONC   0402
0046   1DCA 95FF 1D C0                CLI   PTBY,X'FF'     PRINT BYPASS SWITCH   CONC   0403
0047   1DCE 4780 1E 04                BC    EQ,C2                                CONC   0404
0048   1DD2 45E0 0C 08 C0 00          GET   DTI            READ TAPE             CONC   0405
0049S  1DD8 D244 1D BB 80 C1          MVC   LINE(69),1(8)    ITEM TO PRINT  PUNCH TO 70
005C   1DDE 45E0 C0 C8 C0 00          GET   DTI
0051   1DE4 D23E 11 00 80 01          MVC   LINE+69(63),1(8)    SECOND CARD FOR SKIP PUNCH TO 64
0052   1DEA 95F1 80 00                CLI   0(8),X'F1'     FORMS TEST            CONC   0407
0053   1DEE 4780 1E 36                BC    EQ,C3                                CONC   0408
0054   1DF2 45E0 C0 CC E2 07 40 01    CNTRL DPO,SP,,1      SPACE 1               CONC   0409
0055   1DFA 45E0 00 1D 10 BB    C4    PUT   DPO,LINE       PRINT A LINE          CONC   0410
0056   1E0C 47F0 1D C6                BC    ALL,C1                               CONC   0411
0057                            *                                                CONC   0412
0058   1E04 95FF 1D BF          C2    CLI   RDBY,X'FF'     1001 BYPASS SWITCH    CONC   0413
```

Figure 4–2. Typical Compilation Printout (Part 2 of 6)

```
0059   1E08 4780 1E52                     BC    EQ,C5                             CONC   0414
0060   1E0C 9208 1DC1                     MVI   CCXF,X'08'      TRF & READ PRIMARY     CONC   04145
0061   1E10 45E0 0008 1C63                GET   DCI,CARD        READ A CARD            CONC   0415
0062   1E16 9561 1C63                     CLI   CARD,C'/'                              CONC   04155
0063   1E1A 4780 1E42                     BC    EQ,ENDC                                CONC   04156
0064S  1E1E D24F 9000 1C63          C7    MVC   0(80,9),CARD    CARD TO TAPE           CONC   0416
0065   1E24 45E0 0010 0000                PUT   DTO             WRITE TAPE             CONC   0418
0066   1E2A 95FF 1F4A                     CLI   CLSW,X'FF'                             CONC   04185
0067   1E2E 4780 1E42                     BC    EQ,ENDC                                CONC   04186
0068   1E32 47F0 1E52                     BC    ALL,C5                                 CONC   0419
0069                                *                                                  CONC   0420
0070   1E36 45E0 000C E2D2 4007     C3    CNTRL DPO,SK,,7       HOME PAPER             CONC   04201
0071   1E3E 47F0 1DFA                     BC    ALL,C4                                 CONC   0422
0072                                *                                                  CONC   0423
0073   1E42 9200 0001               ENDC  MVI   D?OT+1,X'0'     CLOSE CARD TO TAPE     CONC   0501
0074   1E46 45E0 0004                     CLOSE DCI             CLOSE I/O              CONC   0502
0075   1E4A 45E0 0004                     CLOSE DTO             CLOSE I/O              CONC   0503
0076   1E4E 92FF 1DBF                     MVI   RDBY,X'FF'      SET CD TO TP BYPASS    CONC   0504
0077   1E52 95FF 1DC0               C5    CLI   PTBY,X'FF'                             CONC   0505
0078   1E56 4780 1DC6                     BC    EQ,C1                                  CONC   0506
0079   1E5A 47F0 1DFA                     BC    ALL,C4                                 CONC   05065
0080                                *                                                  CONC   0508
0081   1E5E 45E0 0004               ENDT  CLOSE DTI             CLOSE TAPE TO PRINT    CONC   0509
0082   1E62 92FF 1DC0                     MVI   PTBY,X'FF'                             CONC   0510
0083   1E66 9200 0001               C6    MVI   B?OT+1,X'0'                            CONC   0511
0084   1E6A 45E0 0004                     CLOSE DPO                                    CONC   0512
0085   1E6E 47F0 1DC6                     BC    ALL,C1                                 CONC   0513
0086                                *                                                  CONC   0514
0087   1E72 9500 1000               KEY   CLI   TEST,X'0'       KEY IN TEST            CONC   0515
```

Figure 4–2. Typical Compilation Printout (Part 3 of 6)

```
0088   1E76 4780 00 C0                BC    EQ,0(,13)                                    CONC   0516
0089   1E7A 9501 10 00                CLI   TEST,X'01'      PAUSE CD TO TP               CONC   0517
0090   1E7E 4780 1E D6                BC    EQ,C10                                       CONC   0518
0091   1E82 9502 10 00                CLI   TEST,X'02'      PAUSE TP TO PR               CONC   0519
0092   1E86 4780 1E E2                BC    EQ,C11                                       CONC   0520
0093   1E8A 9504 10 00                CLI   TEST,X'04'      CLOSE DD TO TP               CONC   0521
0094   1E8E 4780 1E EA                BC    EQ,C12                D                      CONC   0522
0095   1E92 9508 10 00                CLI   TEST,X'08'          CLOSE TP TO PT           CONC   0601
0096   1E96 4780 1E F2                BC    EQ,C13                                       CONC   0602
0097   1E9A 9510 10 00                CLI   TEST,X'10'          OPEN CD TO TP            CONC   0603
0098   1E9E 4780 1E FA                BC    EQ,C14                                       CONC   0604
0099   1EA2 9520 10 00                CLI   TEST,X'20'          OPEN TP TO PT            CONC   0605
0100   1EA6 4780 1F 12                BC    EQ,C15                                       CONC   0606
0101   1EAA 9540 10 00                CLI   TEST,X'40'          KILL                     CONC   0607
0102   1EAE 4780 1F 26                BC    EQ,C16                                       CONC   0608
0103   1EB2 9580 10 00                CLI   TEST,X'80'          CLOSE BOTH & KILL        CONC   0609
0104   1EB6 4780 1F 2A                BC    EQ,C17                                       CONC   0610
0105   1EBA 9581 10 00                CLI   TEST,X'81'      RESTART CD TO TP             CONC   0611
0106   1EBE 4780 1F 3A                BC    EQ,C18                                       CONC   0612
0107   1EC2 9582 10 00                CLI   TEST,X'82'      RESTART TP TO PT             CONC   0613
0108   1EC6 4780 1F 42                BC    EQ,C19                                       CONC   0614
0109   1ECA 95FF 10 00                CLI   TEST,X'FF'      CLOSE TP TO PT &KILL         CONC   0616
0110   1ECE 4780 1F 32                BC    EQ,C20                                       CONC   0617
0111   1ED2 47F0 00 C0                BC    ALL,0(,13)                                   CONC   06175
0112                            *                                                        CONC   0618
0113   1ED6 92FF 10 BF          C10   MVI   RDBY,X'FF'                                   CONC   0619
0114   1EDA 9200 10 00          C50   MVI   TEST,X'00'      RESET TEST                   CONC   0620
0115   1EDE 47F0 00 00                BC    ALL,0(,13)                                   CONC   0621
0116                                                                                     CONC   0622
```

Figure 4–2. Typical Compilation Printout (Part 4 of 6)

```
0117   1EE2 92FF 1D C0        C11    MVI   PTBY,X'FF'           CONC   0701
0118   1EE6 47F0 1E DA               BC    ALL,C50              CONC   0702
0119                          *                                 CONC   0703
0120   1EEA 92FF 1F 4A        C12    MVI   CLSW,X'FF'           CONC   0704
0121   1EEE 47F0 1E DA               BC    ALL,C50              CONC   0705
0122                          *                                 CONC   0706
0123   1EF2 9200 10 00        C13    MVI   TEST,0               CONC   0707
0124   1EF6 47F0 1E 5E               BC    ALL,ENDT             CONC   0708
0125                          *                                 CONC   0709
0126   1EFA 45E0 00 00        C14    OPEN  DCI                  CONC   0710
0127   1EFE 9230 00 01               MVI   D?OT+1,X'30'         CONC   0711
0128   1F02 45E0 00 00               OPEN  DTO                  CONC   0712
0129   1F06 9200 10 B5               MVI   RDBY,0               CONC   0713
0130   1F0A 9200 1F 4A               MVI   CLSW,0               CONC   07135
0131   1F0E 47F0 1E DA               BC    ALL,C50              CONC   0714
0132                          *                                 CONC   0715
0133   1F12 45E0 00 00        C15    OPEN  DPO                  CONC   0716
0134   1F16 9230 00 01               MVI   B?OT+1,X'30'         CONC   0717
0135   1F1A 45E0 00 00               OPEN  DTI                  CONC   0718
0136   1F1E 9200 1D C0               MVI   PTBY,0               CONC   0719
0137   1F22 47F0 1E DA               BC    ALL,C50              CONC   0720
0138                          *                                 CONC   0721
0139S  1F26 A11A 00 00        C16    SRC   0,26
0140                          *                                 CONC   07235
0141   1F2A 92FF 10 00        C17    MVI   TEST,X'FF'           CONC   0801
0142   1F2E 47F0 1E 42               BC    ALL,ENDC             CONC   0802
0143                          *                                 CONC   0803
0144   1F32 9240 10 00        C20    MVI   TEST,X'40'           CONC   0804
0145   1F36 47F0 1E 5E               BC    ALL,ENDT             CONC   0805
```

Figure 4–2. Typical Compilation Printout (Part 5 of 6)

```
0146                      *                               CONC  0806
0147  1F3A 92001DBF       C18    MVI    RDBY,0            CONC  0807
0148  1F3E 47F01EDA              BC     ALL,C50           CONC  0808
0149                      *                               CONC  0809
0150  1F42 92001DCD       C19    MVI    PTBY,0            CONC  0810
0151  1F46 47FC1EDA              BC     ALL,C50           CONC  0811
0152                      *                               CONC  0812
0153  1F4A 00             CLSW   DC     X'0'              CONC  0813
0154S 1F4B                TAG    EQU    *                 CONC  0001
0155  001C                       ORG    16                CONC  0002
0156  001C 0000                  DC     Y(0)              CONC  0003
0157  0012 0126                  DC     Y(294)            CONC  0004
0158  1F4B                       ORG    TAG               CONC  0005
0159S                            ENTRY  FID               CONC  9901
0160                             ENTRY  CDT               CONC  9902
0161                             ENTRY  GNO               CONC  9903
0162                             ENTRY  VOL               CONC  9904
0163                             ENTRY  XDT               CONC  9905
0164                             END    STRT              CONC  0813
```

Figure 4–2. Typical Compilation Printout (Part 6 of 6)

CORRECTING THE SOURCE CODE DECK

It is most important that the source code deck for each program be corrected and maintained. If object code corrections are permitted, full details should be entered on the latest printout. Source code corrections should be entered on a new set of source code forms. These forms should be identified by date, program name, and the name of the programmer who made the corrections. After the source code cards have been punched and checked, these forms should be filed with the originals.

PERFORMING RECOMPILATION

Most programs will have to be compiled several times during testing to eliminate program logic errors revealed by diagnostic messages in the listing, or to make changes in the values of constants used by the program. Several methods can be used to make temporary changes in the object code, but the source code should always be corrected to reflect such changes. A final recompilation should be performed to eliminate any program testing aids used. The program should then be tested before being released to operational status.

PROGRAM TESTING

Generally speaking, each computer program written in symbolic language is a hand-crafted part of a computer system. It cannot be assumed to be correct or reliable until it has been thoroughly tested. Depending upon the complexity of the program, it is usually desirable to test each program in logical subsections. After the parts of a program are found to be individually accurate and workable, they can then be combined for a composite test. Various aspects of testing UNIVAC 9200/9300 Disc System programs are described below.

Generation of Test Data

The planning and preparation of test data for each program requires a significant amount of programmer effort. It is desirable, therefore, to institute a system for the collection of all test data into a permanent systems test library. The cost of future tests will thus be reduced and their comprehensiveness will be ensured. In a relatively short period of time, the aggregate test library will provide a higher degree of accuracy for the system than could be obtained if testing were limited to currently prepared test cases.

One method of developing test files is the piecemeal creation of test cases by coding and punching full data records. The fields in these records are assigned arbitrary values which meet the various conditions for which the program provides. Both correct and incorrect data are included in such records.

Another method is to develop a computer program which fabricates test records on a logical or random basis. Such a program, if developed, should be a generalized one. After it has been debugged, then, it can be used to produce test data for many programs.

Program Logic Test

It is recommended that each programmer develop a list of the conditions to be tested during the preparation of the flowchart. This list should be oriented to data file content in a way that permits it to be used by a clerical staff for producing the corresponding test records.

The points at which storage dumps are to be taken, and the part of the program to be dumped, should be indicated on the flowchart. The means by which the storage dumps are to be identified for printout and the significance of each dump must also be considered.

Test Practices

While the compiling program is designed to continue its processing regardless of the errors in input data, problem programs are much more easily stalled. For this reason a program test demands more preparation effort on the part of the programmer. Following are some recommendations for achieving satisfactory test results.

Establish Test Schedules Ahead of Time

The operations staff should establish and maintain a system for scheduling computer usage. This schedule should be established for periods from several days to several weeks. It should be published well in advance of its actual use. A schedule coordinator should be appointed to obtain estimates of the dates test time will be needed and the amount of time to be allocated to each test. A procedure for cancelling or exchanging test periods should be established and responsibilities fixed for taking the initiative in altering the schedule. Where possible, scheduled backup jobs and tests should be available for processing in the advent of last minute cancellations.

Have All Files Ready Before Test Time

The file containing the program, all input test files, and adequate output file media should be compiled prior to the test. The programmer should verify that everything is in readiness in advance of test time. Care should be exercised to avoid last minute changes in test objectives or tardy preparation of the executable program.

Prepare a Documented Plan of Operation

The programmer should prepare a complete plan of operation for conducting the test. This plan should be documented and contain all the significant details of the test. The name of each file and the reel or disc number containing it should be indicated. The device on which each file is to be mounted and the disposition of the output files should be recorded. A documented plan covering the alternatives in case of errors should be in the programmer's possession. A good approach to documentation of operating procedures is to design a form similar to that illustrated in Figure 4–3. This form can eventually become the basis for the operator's Run Book.

RUN I.D.	PROGRAMMER:

INPUT	FILE NAME:		DEVICE:
	FILE NAME:		DEVICE:
OUTPUT	FILE NAME:		DEVICE:
	FILE NAME:		DEVICE:
LOAD PROCEDURE	STANDARD	OTHER:	
PROGRAM STOPS	DISPLAY	REASON	ACTION
PRINTER FORM SET-UP	FORM NAME:		
	CARRIAGE LOOP NO.:		
SPECIAL INSTRUCTIONS			

Figure 4–3. Program Operating Procedures

Have Program Documentation Available For Reference

Generally, the problems encountered during program testing are unpredictable. Because of this, the programmer must be able to reference program documentation to determine the nature of a problem. An adequate work area should be placed in the immediate vicinity of the computer to facilitate this reference. The programmer should be familiar enough with the program documentation to locate rapidly the documentation corresponding to any specific point in the program. It is most helpful if the programmer provides himself with a checklist of items necessary for successful testing. The checklist might resemble the one in Figure 4–4.

I. RUNS TO BE TESTED

____
____
____
____
____
____
____
____

II. SOFTWARE REQUIRED

☐ RPG ☐ ASSEMBLER ☐ COBOL ☐ FORTRAN

OTHER ______________________

III. NEEDED ITEMS

☐ PROGRAM FLOWCHARTS

☐ SOURCE CODE FORMS AND DECK

☐ SOFTWARE OPERATING INSTRUCTIONS

☐ TEST DATA

☐ WORK TAPES

☐ WORK DISCS

☐ PRINTER FORMS AND FORMS CONTROL LOOP

☐ SAMPLES OF VALID OUTPUT

Figure 4–4. Program Test Checklist

Systems Testing

Most of the programs in the system have some input-to-output relationship to other runs in the system as well as among themselves. Each series of programs should be tested by running a reasonable volume of data through two or more complete processing cycles. It is recommended that this technique be used as part of an acceptance test for each program before the full systems test is performed. When input to a program is one of its output files from a previous cycle, the programmer should make sure that such output is fully acceptable input. Also, part of the acceptance test for each program should include test processing of its output files by the other programs that are to use these files as input.

MAINTAINING THE PROGRAM LIBRARY

Day-to-day operation of a UNIVAC 9200/9300 Disc System involves the creation and maintenance of several libraries of programs. These libraries should include at least:

1. Source Code input to the compilers

2. Object programs produced by the compilation program

UNIVAC 9200/9300 Disc System software includes a utility program for use in creating and maintaining program library files. This Library Services program can be used to selectively copy all, or parts, of one or more input library discs to an output library disc or tape.

Specified portions of input libraries can be copied or deleted. Individual lines can be deleted or replaced by new instructions.

In addition to the output library file, the Library Services program produces output which can be printed directly or written on disc or tape for subsequent printing. The amount of information in this listing can be varied according to the specific library function executed. Specific list options and the command format are described in the UNIVAC 9200/9300 Assembler Reference Manuals.

THE PROGRAMMER'S DOCUMENTATION RESPONSIBILITIES

Proper program documentation is extremely important to the success of a data processing installation. It is essential to the effective operation of all affected departments, that complete and accurate documentation be available for reference during resolution of operational problems.

There are two distinct classes of documentation in a programming assignment. One, classified as technical documentation, provides exact information about the design and content of the program. The other classification comprises the operator instructions. Both classes should be collected and maintained in a program Run Book before programs are released to operational status.

Technical Documentation

Instructions for preparation of technical documentation (such as flowcharts) have been covered to some extent in the foregoing sections. Technical documentation includes the following:

- The functional description prepared at the outset of the programming effort
- Fully updated flowcharts used in designing the program
- A fully updated listing of the source code
- Fully corrected source code
- Samples of the input and output
- Main storage dumps and other informative printouts

Operator's Instructions

The overall responsibilities of the operator during program processing should be covered in this type of documentation. This material, which can be taken from the Run Book, should provide sufficient information to enable the operator to successfully complete all phases of his responsibilities.

The Run Book

The Run Book is a collection of all documentation pertaining to each program. It should include all information that will increase understanding of the problem or of the approach chosen for its solution. The effectiveness of the Run Book can be measured by the degree to which persons not associated with the development of the program find its contents readily understandable. The information included in the Run Book should be presented in the sequence dictated by the frequency of its reference. It should be made up of several distinct sections.

Introduction

This section describes in general terms the processing to be performed by this program. It includes the functional specifications of the program and a chart illustrating the processing flow of data from the input files, through the program, to the output files.

Operational Instructions

This section provides all the procedures to be followed when operating the program. Included are program loading, file mounting, and descriptions of any exchanges of messages or signals which take place between the operator and the program. A good basis for these instructions is an operations form similar to the one illustrated in Figure 4–3.

Input/Output Files

The different types of files that can be processed by the program and the data file conventions that have been followed should be described in an introductory paragraph.

Each input and output file should then be described under a separate heading (functional name). A figure might be used to illustrate the file organization and the format of file records. The different types of records which may appear in the file should be listed.

If an input file is to be automatically carried over from another routine, this should be indicated. If input file processing is contingent upon further direction from the programmer staff, the specific conditions should be described.

If an output file is to be automatically forwarded to another routine, this fact should be indicated. If output file processing is contingent upon further direction from the programmer staff, the specific conditions should be set forth. Explain the format of any hard copy output of the program. Provide an example of such outputs. The following topics should be considered:

- *Label Records* – Indicate whether label records are placed on the file. If they are used, describe label content. Describe the method used to identify each new file to distinguish it from prior output of the same program.
- *Data Records* – Provide a general description of the data records included on the file. Indicate record size and the number of records which may be included in a data block. Describe any control words or other devices which have been included in each data block.
- *Rerun/Restart* – Explain the rerun/restart capability provided for this file.

5. FILE PREPARATION

INTRODUCTION

Installation of a new computer almost always involves file preparation. Even when files used with a previous computer are perfectly compatible with the new system, new files will usually be wanted to take advantage of the higher power of the new system (such as larger main storage or new peripherals) or to apply new work to the computer. As used in this section the term "files" includes, but is not limited to, master files, transaction files, report files, program files, and tables.

Because the entire data processing system is based on the files used, file preparation demands careful planning, strict control, and complete auditing. File preparation should be treated as a complete application in itself, receiving the same care as other major applications. The standard development steps of problem definition, information gathering, data analysis, and procedure establishment should be taken.

FILE PREPARATION CHECKLIST

Initially, a File Preparation Checklist (Figure 5–1) should be prepared to define the scope of the conversion workload, prevent oversights, and help establish file priority. For every file in the new system, the following information should be included: file name and number, related application, date required (application cutover date), present media, and required media. This information can be obtained from the Systems Design Application Flowcharts (illustrated in Figure 3–9). Later, when file classifications and conversion methods have been determined, they can be added to the File Preparation Checklist.

FILE		APPLICATION	PRESENT MEDIA	REQUIRED MEDIA	DATE REQUIRED	FILE CLASS.	CONVERSION METHOD
NAME	NO.						

Figure 5–1. File Preparation Checklist

INFORMATION COLLECTION

The systems design documents (see Section 3) must be reviewed in order to determine what data is needed in the new files. These documents will also define new file format; that is, card and report layouts, record and sector designs, run descriptions, and estimated volumes.

Existing files, regardless of their media, should then be investigated for file content. The existing files must also be discussed with the people using and maintaining them, to learn the purpose and importance of the file as well as present methods of use, control, and scheduling. The degree of validation needed for each file can be estimated at this time. Needed information which is not found by these means must be sought in the various departments within the company.

Besides file content, the information gathered must define:

- *Codes* – The meaning of currently used codes.
- *Volumes* – The quantity of cards, ledgers, blocks, records, and other media in existing files. This information will be used to estimate and schedule conversion requirements.
- *Purging* – The method of removing inactive or obsolete records from present files.
- *Error Handling* – Present procedures for the detection and correction of errors.
- *Frequency of Use* – The time period when existing files are in use. This will determine the best time to release files for conversion.
- *Updating* – The method and frequency of updating existing files.

ANALYSIS AND CLASSIFICATION OF DATA

By scanning a representative sample of all existing files and source documents, the scope of the conversion and the interrelationships of the files can be estimated. Files that are already in computer acceptable media should be examined carefully for undesirable record formats, such as split fields, field sizes that no longer meet requirements, and overpunching. A work form such as that illustrated in Figure 5–2 will help expose the interrelationship of file data.

Once all the data has been gathered, it should be examined thoroughly to ensure its accuracy and completeness. It can then be classified according to the type of media on which it is recorded. The following three classifications are used:

- *Compatible* – media such as punched cards, magnetic tape, or punched paper tape, which are acceptable as direct input to the system or can be machine converted to acceptable media.
- *Incompatible* – media such as ledger cards, catalogs, and listings, which must be transcribed into an acceptable direct entry media.
- *Combination* – media of mixed types; for example, summary data in punched cards, details in ledgers.

METHODS OF FILE PREPARATION

The three categories of media lead to three possible methods of file preparation: creation of new files, straight conversion, and consolidation.

Creation

In creation of new files, information classified as "incompatible" is converted to an intermediate or final medium usable as computer input. This method is commonly used for files of Category 3 applications, which are not presently automated. (Application categories are defined in Section 2, "Scheduling.")

The creation of new files is the most demanding of the three conversion methods and the most susceptible to error. It usually requires development of computer programs or unit record procedures, as well as design and implementation of a manual procedure for the orderly and accurate collection, coding, transcription, auditing, keypunching, key-verifying, and sorting of data to be contained in the file.

It may be desirable to prepare a worksheet as an intermediate step between source document and file, particularly when:

- The information on the source document is difficult to read.
- Withdrawal of data in the sequence desired is difficult.
- The source documents are not available for the uninterrupted period required for control, keypunching and key-verifying, validation, and correction.
- A multiple source of information is used.
- Extensive coding, condensing, or editing is necessary.

FIELD NAME	LENGTH	ALPHA OR NUMERIC	ARITHMETIC OR DESCRIPTIVE	RECORD						
				INVOICE	ACKNOWL-EDGEMENT	INVOICE REGISTER	SHIPPING NOTICE	SALES ANALYSIS	ACCOUNTS RECEIVABLE	INVENTORY
Date	6	N	D	X	X	X	X		X	
Customer No.	5	N	D	X	X		X	X	X	
Invoice No.	6	N	D	X		X	X		X	
Customer Order No.	7	A/N	D	X	X		X			
Shipping Instructions	30	A	D	X	X		X			
Item No.	6	A/N	D	X	X		X	X		X
Quantity	3	N	A	X	X		X	X		X
Price	5	N	A	X	X			X		X
Cost	5	N	A					X		X
Extension	6	N	A	X						
Invoice Total	6	N	A	X		X		X	X	
Tax	4	N	A	X		X				
Customer Name	30	A	D	X	X	X	X	X	X	

Figure 5–2. Field Interrecord Relationship Work Form

The worksheet should be designed so it can be used easily by both the clerical personnel who will transcribe the information from the source document and by the keypunch operator who will punch the data from the worksheet into cards. For efficient keypunching, the data should be arranged in the same sequence on the worksheet as on the cards. If such an arrangement is not possible, it may be advisable to create an interim card format; the final record format can then be achieved by using the computer or unit record equipment.

Since the creation of a new file is not part of the routine data processing schedule and requires extensive editing and auditing, additional man-hours may be needed. They can be arranged through one or more of the following means:

1. Extending regular working hours.
2. Temporarily reassigning personnel from another department.
3. Hiring on a temporary basis.
4. Having tasks performed by an outside service organization.

Factors to consider when determining the means of obtaining additional personnel include the estimated time required to complete the manual phases of file creation, the availability of source documents, the personnel's degree of familiarity with the task, the complexity of the task, and the frequency of change within source documents.

Straight Conversion

In straight conversion, files on "compatible" media are modified by direct processing on the computer or unit record equipment, with no manual operation involved. This conversion method is generally used for files of applications in Categories 1 and 2 (defined in Section 2, "Scheduling").

Examples of straight conversion are: rearrangement of certain data fields to a more efficient format in the new system; deletion of data; increase or decrease in the size of fields or records; changes in file label format; character translation; assignment or changing of codes; and calculation of additional data when all factors are present.

Straight conversion may be performed on single files, where one unit (record, card, item, etc.) of input is to be converted to one unit of output, or on multiple files where two or more units of input are to be merged, collated, condensed, or translated into one unit of output.

Consolidation

The consolidation method of file preparation is a combination of file creation and straight conversion. It is used when the source data for the new file is contained on a combination of "compatible" and "incompatible" media.

File preparation by the consolidation method is principally used in Category 3 applications. Some degree of consolidation may be required in Category 2 applications which are expanded to utilize the greater power of the new or expanded computer. (Application categories are defined in Section 2, "Scheduling.")

PLANNING INDIVIDUAL CONVERSIONS

Conversion Procedures

The remaining tasks of file preparation are performed separately for each file and include:

- Establishing verification/validity techniques
- Creating the file (including worksheet design, implementation, and instruction in use) or writing and testing the conversion program, or both
- Pilot testing
- Performing the actual conversion

Time Requirements

The time needed to complete the conversion of individual files should now be estimated, including time for personnel training at each step.

Note that the preparatory steps through pilot testing usually require much more time than the mechanics of actual conversion. It is also important to remember that, because most personnel involved will be unfamiliar with the file preparation tasks, optimum performance rates will not be achieved initially.

Scheduling Factors

The conversion of each file is then scheduled, based on the following factors:

- The total number of files to be prepared and the dates required, taken from the File Preparation Checklist
- The frequency of use of each file, determined during information collection
- The estimated time required for each conversion
- The equipment delivery date
- The availability of equipment for program testing, pilot testing, and conversion

By taking these factors into account and calculating back from the date the file is required, the last possible starting date for each file can be determined.

Ideally, all file completion dates should occur between the last time the files are used by the existing system and the first time they are needed by the new system. Dual file maintenance would thus be eliminated, and files entering the new system would be as accurate and current as those in the earlier system.

However, the amount of equipment and the number of personnel required to complete the conversion within the short time available usually makes this approach impractical. In practice, therefore, files are prepared in advance of their use and maintained until required by the new system.

If too many files are prepared at once or they are prepared too far ahead of use, problems may arise from the burden of dual file maintenance. By the time the files are used, reconciliation or even recreation might be necessary. For this reason, it is often desirable to prepare files only as needed for cutover of each application.

Determining Special Requirements

After the file preparation workload has been determined, requirements for additional personnel and special supplies (such as magnetic tape, discs, cards, forms, and worksheets) should also be estimated and scheduled. Supplies should be ordered sufficiently in advance to coordinate with suppliers' lead times, allowing a safety margin for contingencies.

VERIFICATION METHODS

Validation

The success of any new system depends on the validity of the information it contains. Complete and accurate files will eliminate potential problem areas such as lost time because of missing records, erroneous or incomplete information, and customer complaints.

Some common validation methods are as follows:

Complete

A printout of the entire file is verified by comparing it to the original data. This method requires extensive personnel time and yet does not necessarily ensure complete accuracy because personnel alertness is often reduced and erroneous data may be overlooked.

Sampling

Selected portions of data from the file are checked thoroughly. Sampling would point out consistent errors such as those caused by misinterpretation of instructions or of original data.

Check Digit

Permanent or semipermanent codes, such as part numbers, employee numbers, or customer codes, become self-verifying by the addition of a check digit, which is calculated from the original digits of the code. The check digit will locate any change in digit or sequence of digits within a code, such as those caused by transposition, misreading or keypunching. This validation method requires that the program calculate and assign the check digit, and expand the field size of the code to accommodate the calculated digit.

Range Testing

Codes, amounts, and figures are checked to ensure that they are within preset limits. Although this type of verification does not assure complete accuracy, it will locate the error that is entirely outside the range of possibility. Thus, if all items of an inventory cost between $1.00 and $53.00, range testing would find any item showing a cost outside these amounts.

Character

Like range testing, character validation is a test of reasonableness. A field which contains one of several known codes is compared to each of the codes possible until a match is found; if no match occurs, an error is indicated. For example, suppose that a tax code field must contain a 0, 2, 5, or 7. If it contained any other code or no code, an error condition would be indicated.

Controls

Unlike validation methods, which indicate the exact item that is in error, controls point out that a file or a portion of a file is in error. (Audit trails must then be followed to find the actual error.) Controls may be of the following types:

- *Hash Totals* – The total of a group of unlike figures, which is meaningless except for control purposes and would not be maintained for any other reason. Hash totals are used to verify that all records of a group are present.
- *Quantity and Amount Totals* – The total of a group of like figures, which may already be utilized in the control of the existing system; for example, quantities in invoicing or payroll hours.
- *Unit Count* – The total number of units (such as cards, items, records, or books) within each particular group, section, or file.

Control figures should be established at logical points within the file; for example, at changes in department, after X number of items, by classification, or at any point desired depending on the file arrangement. They should not be so numerous that they are cumbersome, nor so few that isolation of the error is an enormous task.

The control figures are created (or verified if they already exist) before the initial processing, preferably as soon as the source material is available.

To permit early detection and correction of errors, control figures should be checked at various stages of file development; for example, upon completion of the worksheet, after keypunching and verifying, after sorting, and after each pass on the UNIVAC 9200/9300 Disc System. The frequency of checking required depends, of course, on the volume of data being processed.

The control figures should also be verified at the output stage – that is, after the file has been converted or prepared, prior to use by the new system.

Audit Trails

Procedures must be established to permit, upon discovery of an error, an orderly tracing of the flow of data to determine the origin and cause of the error, the type of corrective action to be taken, and the point at which to initiate a restart procedure.

STANDARDS

By setting standards where possible, the installation manager will reduce the amount of explanation and instruction needed, and increase the efficiency of the operation. Some suggested areas in which standards should be followed are:

- Type and points of control for each file
- Character representation used on worksheets; for example, O (letter) and 0 (numeral), S (letter) and 5 (numeral), Z (letter) and 2 (numeral)
- Worksheet design and use
- Initial and pilot test data
- Acceptance test minimum

CONVERSION PROGRAM WRITING

The writing of conversion programs can only begin when all the information has been collected and analyzed, and the standard and procedures have been established.

Existing or proposed file maintenance programs can sometimes be used to perform the conversion. The use of these programs, with or without modification, could eliminate the need for specialized conversion programs and would reduce the overall programming workload.

(Refer to Section 4, "Programming," for an explanation of the actual programming function.)

PILOT TESTING

Pilot tests of representative portions of the new files are as important as systems tests of applications. They should be performed for all conversion programs in order to find and correct errors that may have been overlooked in the program testing before these necessitate complete reruns of the final conversion. In addition, from the timings obtained in pilot tests, the total run time for the complete file conversion can be projected, and the schedule adjusted if necessary.

Upon completion of a successful pilot test, the partial files that have been prepared can be used as input for the final systems test of each application and of the overall system.

CONVERSION OPERATIONS

Actual preparation of the files should be done, where possible, by members of the Data Processing Department. In this way, the people using the files after conversion will become familiar with the files and file content, and gain experience in the operation of the UNIVAC 9200/9300 Disc System. Furthermore, this procedure will greatly reduce the number of explanations needed at the time of cutover.

Where the conversion operation is performed depends on the magnitude of the files, the speed of completion desired, and the availability of personnel and equipment. If time and workload permit, the conversion may be performed at the installation site upon delivery of the UNIVAC 9200/9300 Disc System, or shortly thereafter. If file preparation is to be completed before delivery, however, a Univac Regional Test Center or other nearby site could be utilized.

It may be desirable to employ the services of a Univac Information Services Center to perform portions or all of the conversion phase of file preparation. Such outside contracting can avoid the need to hire and train temporary personnel and to secure temporary equipment and word space.

NEW FILE MAINTENANCE

Procedures must be developed to maintain the files from the time the data was originally obtained until the file is used in the new system, when the regular file maintenance programs will take over.

Without proper file maintenance, the new files may require reconversion, extensive updating, or complete file reconciliation by the cutover date. To avoid such problems, precise procedures for the handling and control of additions, corrections, changes, and deletions must be developed and used.

FILE MAINTENANCE DURING USE

The purpose of file maintenance is to permit the continued use of a data file as long as possible before having to reorganize it. File maintenance involves making additions and deletions to the file, changing static information, and, in some cases, providing for activity frequency changes. Regardless of the file maintenance techniques used, however, the gaps left by deletions and the chaining required for additions eventually lead to poor utilization of space and excessive look-up time. Because the efficiency of the file gradually decreases, periodic reorganization of the file is usually necessary.

Additions

In an indexed sequential and random file, additions should be made at the time they occur. In a basic sequential file, which does not have gaps or an overflow area, additions require reorganization of the file. The most efficient way of making additions to a sequential file is to accumulate a number of them, sort them into sequence, and merge them into the file under control of the Sequential Processor.

Deletions

A data record that becomes inactive should not be removed from the file immediately because there may be a subsequent reference to it. Instead, a record to be removed should be flagged. Examples of flags are the date on which the item became obsolete or on which it may be removed from the file; the flag could also be a single bit indicator which designates immediate or later removal. A flagged item should be removed during a subsequent file maintenance or audit run.

Before a record is deleted, it should be dumped to punched cards, magnetic tape, or a line printer. This provides a historical record of all data removed from the file. It also provides for their possible reconstruction.

After a series of deletions have been flagged, all files need to be reorganized and condensed. In indexed sequential files, reorganization is done in order to avoid unnecessary searching and to make certain that all indexes reflect any changes made in the file. In files that utilize chains, the chains affected must be adjusted to preserve their workability.

During reorganization, records flagged for deletion are either removed from the file or placed last in descending order of activity and deleted when the control date for removal has been reached.

Static Information Changes

The facility to change normally static credit fields, such as the cost field of an item inventory record, should be provided in normal processing programs. When a static field is changed, the program logic should be such that the change is made before that field is used in a processing calculation affected by the change. For example, if a pay rate is to be changed, the program logic should ensure that the change is made before a paycheck affected by the new rate is computed.

Mass changes to normally static data are best made by an independent file maintenance run that prints out both old and new contents of the changed records.

Activity Frequency Changes

Files organized so that the most used records are at the front, instead of the keys being arranged in numeric sequence, should occasionally be reviewed and reorganized to reflect current activity frequency. If the activity of items in the file is naturally changeable – seasonal, for example – automatic activity count fields can be included in the records and used to rearrange the file so that the most active items are at the front and the least active at the rear.

File Maintenance Techniques

The following file maintenance techniques should be performed to maintain file efficiency:

1. Prevent file saturation. File saturation will probably force processing to halt, resulting in loss of time, manpower, and money.

2. Dump files regularly.

3. Always reorganize before the condition of the file is such that it exceeds the critical point where processing may have to halt until reorganization is performed.

4. When using overflow areas, use an alarm system to warn of impending file exhaustion.

5. Dump indexes regularly and check their validity.

6. If changes are made to any files that use indexes, verify the changes in the appropriate indexes.

7. In random files, purge inactive items regularly.

8. Perform scheduled online reviews of item status; determine whether they are active or obsolete.

9. When an item is purged, make a hard copy printout of the purged areas.

10. Establish file retention dates so that information which is important, but inactive (for example, W–2 information), is not destroyed inadvertently.

BACK-UP PROCEDURES

As computer systems become more complex, back-up procedures are becoming more essential. When a total equipment configuration includes disc, back-up procedures are required for two major reasons:

1. If a file is destroyed or becomes inaccessible, it must be reconstructed to be reprocessed.
2. The destructive update characteristic is inherent in disc devices. Destructive update means that, when information is written to the disc, those locations within a sector which do not receive data are cleared to binary zero. Thus, when records are updated, the previous information is destroyed.

In either case, the user must establish back-up procedures to preserve information. These procedures must be fully planned, checked out, documented, and readily available to all personnel who are responsible for computer room operations. To provide better control, one person should have the responsibility for implementing and maintaining the back-up procedures. Some of the most common back-up procedures are described below.

Retaining Source Documents

Source documents should be retained as long as it is feasible (that is, while storage space is available) or necessary (as required for tax information and internal accounting). The retention time of the source document will vary from application to application.

Dumping

All files should be dumped periodically to another disc, or to tape, card, or printer. Generally, it is most practical to produce the file dump as a byproduct of reports generated on a periodic basis. At first glance, it would seem that a daily dump would simplify rebuilding the file. That is to say, yesterday's file can be loaded and today's input reprocessed in case of equipment failure or of an inadvertent file overwrite. This method is, in fact, the solution with small files. However, as the size of the files increases, daily dumping becomes less practical in terms of time and volume.

Journaling

If it is not practical to reprocess all transactions that occur between dumps, a journaling method of back-up should be used, to permit faster retrieval and reprocessing. Journaling uses magnetic tape for back-up (see Figure 5–3) and is therefore only possible when the total system configuration includes magnetic tapes.

Journaling is accomplished as follows: The first time a record is written to a disc it is also written, or journaled, to tape. Subsequently, records are journaled before and after modification. As each record is journaled to magnetic tape, additional information is written with the record to facilitate future retrieval. The journal information for each record must include the magnetic tape file and record number, the date, the type of file and its name, and the address of the record on disc. The journal tape can be considered a chronicle of activity related to the file.

When the journaling technique is used, programming changes which affect the file have no effect on the reliability of the back-up data produced. If the data must be recreated, it can be reprocessed completely or reinstated at the last entry related to the file.

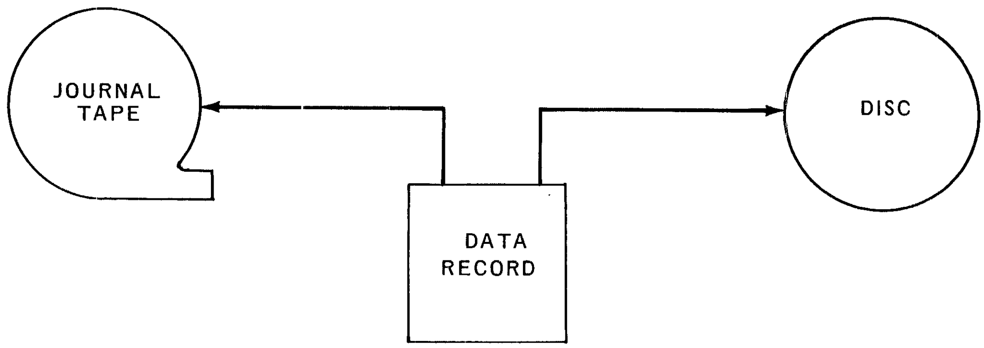

Figure 5–3. Journaling Approach to File Back-Up

Mirror-Image Recording

This technique requires duplicate recording of all records as they are written to disc. This can be accomplished in several ways. One way is to logically divide the disc into two equal sections. When a record is written to disc, it is written to both sections, in the same relative position. Another method is alternate track recording; that is, a record written on Track 0 is also written on Track 2. A third method requires a duplicate disc to record all data.

Mirror-image recording is extremely reliable, but it is costly in terms of time and storage. Moreover, if the disc becomes inoperative, the information is inaccessible.

Grandfather-Father-Son Approach

Another back-up procedure is the grandfather-father-son approach, illustrated in Figure 5–4. In this method, duplicate recording is not required. Instead, two previous generations of a file are kept as back-up to the current file. When a master disc file is updated, the new master file is called the "son," the master used as input is the "father," and the previous master, which created the "father" file, is the "grandfather." The grandfather file is held until the son file has gone through another processing cycle and has, in turn, become a father file. Thus, if information on the son disc is destroyed, the father or grandfather disc is available for reprocessing.

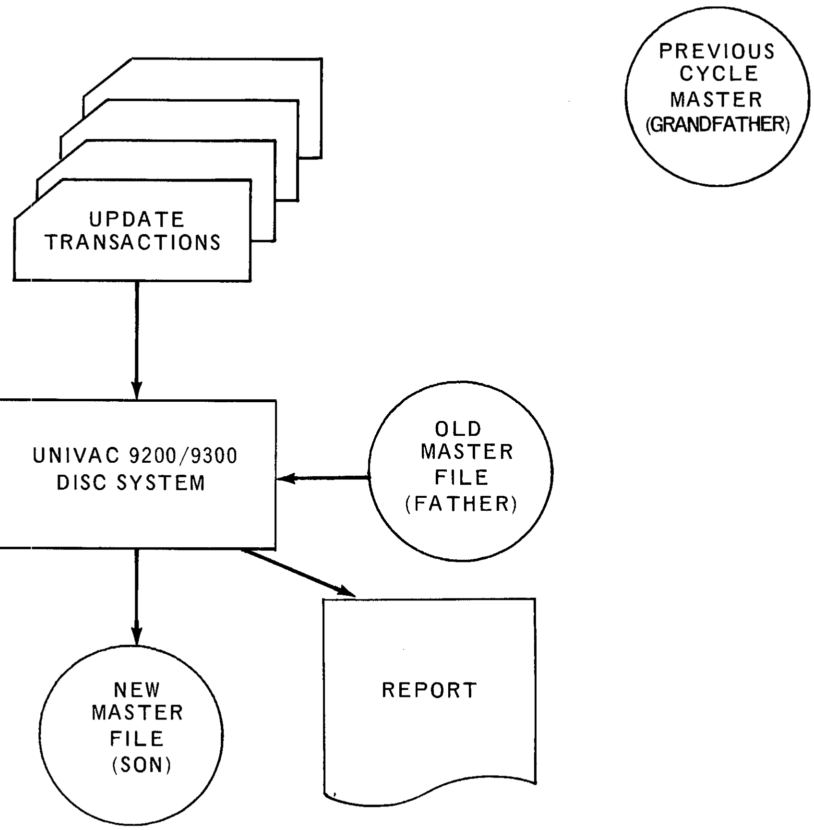

Figure 5–4. Grandfather-Father-Son Method of File Back-Up.

RESTART PROCEDURES

It is sometimes advantageous to restart processing without returning to the beginning of a run. To permit this, a series of checkpoints must be established which will preserve at selected intervals, the contents of main storage, of selected registers, and of indicators. In other words, those items which will permit the most efficient restart capabilities must be saved.

Because of the destructive nature of disc devices, other considerations must be taken into account to enable restart after failure. Main storage must be reloaded, registers must be reestablished, and previously described back-up procedures must be used to restore the file.

6. SUPPLIES

INTRODUCTION

Besides the hardware and software, other materials required for efficient operation of the Data Processing Department fall into three categories:

1. Vital Supplies
   These supplies are indispensable to the operation of the UNIVAC 9200/9300 Disc System. Included in this group are tabulating cards, discs, tapes, printer forms, carriage control tape, and other items which are directly related to the input or output of the Data Processing Department.

2. Furnishings and Handling Supplies
   This category includes desks, tables, chairs, card files, cabinets and containers, bookcases for manuals, movable carts, and other storage facilities.

3. Miscellaneous Supplies
   Included in this group would be items such as rubber stamps, buck slips, stationery and coding paper, templates, and other incidental office supplies. Report binders and log book are also necessary items. A large chalkboard is extremely helpful for group discussions or on-the-premises training.

While items of all three categories are essential to a well organized Data Processing Department, it would be difficult to advise on the amount of furniture or general office supplies needed. In installations converting from card or tape oriented systems to a direct access system, there will be little change. Section 6 therefore deals exclusively with those items vital to the operation of the equipment.

ORDERING

Supplies should be ordered far enough in advance of the system conversion to allow as smooth a transition as possible. The "vital" supplies must be given the highest priority. Knowledge of the usual quantities consumed in daily processing and of the delivery lead times involved, and use of a high quality product with establishment of an emergency stock, will substantially reduce or totally eliminate potential problems.

Supplies stock should be higher than normal during the conversion period to compensate for increased consumption due to program testing runs and to possible reruns caused by inexperience.

To minimize storage space requirements and assure fresh stock, it is usually most practical to replenish cards, forms, and ribbons on a relatively short-term, possibly monthly, basis. Often supplies can be ordered in bulk and scheduled for automatic delivery on a cyclical basis. An added benefit of this arrangement is that many vendors offer quantity discounts. If automatic delivery is planned, however, seasonal trends must be considered; the quantity delivered in each period must reflect the expected usage during that period.

Many supply vendors offer usage and inventory control systems to their customers. Use of these formalized and tested records will simplify the maintenance and control of supplies.

PUNCHED CARDS

Card Storage

The card storage area should be as near the Data Processing Department as possible so cards are easily accessible. The storage area temperature should be between 70° and 75° F, the humidity level between 40 and 60%. Ideally, temperature and humidity levels should be identical in the storage area and the data processing rooms. If they are not, the cards should be stored in the data processing room for three days before use.

Cards may be stored in the shipping cartons before use. These should be kept on steel shelving whenever possible; they should never be put against an outside wall or directly on a concrete floor since this invites moisture and warping problems. Further, they should not be stacked more than three high nor stacked on their sides. To inhibit warping, cards should be removed from partially filled 2000-card boxes and stored in trays under pressure.

Control of Card Stock

Every card type (both by number and color) should be effectively controlled to avoid shortages or excesses and to make optimum use of the available storage space.

Usage records should be maintained and reviewed. Where indicated, adjustments should be made to any automatic replenishment procedure. The usage record should show normal lead time for each card type and should note variations caused by optional features such as multicolored stock, prepunching, or any other unique consideration which might affect delivery time.

When new card formats are ordered, sufficient lead time should be allowed so that proof copies can be checked thoroughly by the people responsible for the original design.

Special Card Types

Color Coded Cards

Color coding of cards provides the advantage of easy recognition; the type or purpose of the card can be seen at a glance and a misfiled card is conspicuous by its dissimilar color. However, colored stock is more expensive than natural stock and its use must be justifiable by the resultant increase in efficiency.

Multipurpose Cards

Use of multipurpose cards (cards with several different sets of field headings) is recommended, since it will simplify inventory management. It may also result in savings, since large orders can be placed on which quantity discounts may apply.

FORMS

Stock Forms

Blank stock in single or multiple parts is necessary in every installation. Its use in as many applications as possible will reduce the number of different forms to be stocked as well as the time loss caused by frequent changing of forms in the printer. Generally, blank stock should be used for most in-house reports and for all program testing. Since stock forms are shelf items for most suppliers, they have minimal lead times.

Stock Imprints

These forms utilize the standard sizes of stock paper, but are designed so that headings and drop lines can be imprinted with minimal cost. The lead time for forms of this type is usually longer than for blank stock but shorter than for custom forms.

Custom Forms

These forms require the most careful attention, since they invariably reach the hands of management or customers, or both. They are usually multicolored and multicopy. The custom form is generally designed by the systems analyst who, at the time of application development, investigated the uses of the form. Most form suppliers are happy to assist in forms design; their representatives are professionals whose advice on factors such as color, weight, and shading should be carefully considered. The forms supplier can often provide samples of forms serving the same purpose in other installations; these may be helpful in both forms design and systems design.

The forms supplier should be advised of the computer system on which the forms will be used, to ensure that the methods employed to hold multipart forms together, the carbon quality, and the paper weight are within the tolerances of the system.

If transition is being made from a printing device whose characters are not ten-to-the-inch, most of the installation's custom forms will require redesign. Plans should be made to have the inventory of old forms at minimum level at the time of cutover.

Proof copies of all new forms should be thoroughly reviewed. Some items which require particular attention are size of columns, vertical and horizontal spacing, drop lines, and all headings. If corrections are extensive, another proof should be made. Although most forms suppliers give a cost reduction for large orders, it is advisable to sacrifice this savings on the initial order to protect against oversupply of a form containing an unforeseen problem.

DISC CARTRIDGES

The UNIVAC 8410 DAS uses disc cartridges manufactured by Univac. In ordering the initial supply of discs, the installation manager can estimate the number needed by considering the file sizes, number of files, number of antecedent copies maintained for security, and adding a sufficient number of discs for programmer use. If the system is fully disc-oriented, systems discs must also be provided. If the system configuration also contains tape, fewer discs will be required because back-up copies of discs can be kept on tape. Historically, mature disc installations have averaged about ten discs for each disc drive; five discs per drive would be a reasonable initial supply if an average rather than a calculated quantity is to be ordered.

The disc cartridge may be stored vertically or horizontally, although vertical storage is recommended.

MAGNETIC TAPE

Magnetic tape has been the most common bulk storage device since the advent of the first commercial computer, the UNIVAC I in 1951. Since that time there have been many changes in tape recording devices and in the tape itself. The improvement of recording techniques and the ability of today's computers to process more data in less time have placed increased emphasis on the quality and care of tape.

Ordering Tape

The tape selected must, of course, be of good quality. Its rated recording density should be equal to or greater than the density at which recording will take place. Tape length is commonly 2400 feet, but reels of different length may be ordered depending on the size of the files to be processed and the flexibility desired in the installation.

An adequate supply of tape should be ordered to meet normal usage and program testing, and provide safe levels of retention. A good working formula is: three working tapes per tape drive, plus three tapes per file, plus a minimum of three systems tapes. Although the actual number of tapes required will vary with each installation, this formula will give a reasonable estimate for the initial tape order.

Care of Tape

The major considerations in the storage and handling of magnetic tape are as follows:

On arrival:
1. Report visual damage to carrier.
2. Look for stacking of cartons more than five high.
3. Open sample cartons to find any hidden damage.

Storage

1. Store cartons with reels vertical.
2. Store tape in the plastic containers in which reels were shipped from vendor. If wrap-around rings are used, choose a design that does not spread the reel flanges or crimp the tape.
3. Maintain 60° to 80° F temperature and 40% to 60% relative humidity in the storage area.
4. Keep tape away from radiators and open windows.
5. Log the date received.
6. For tape stored over six months, rewind before use.
7. Keep tape away from fluorescent lights, power transformers, and telephones.

Handling

1. Handle tape (to repair, for example) only in a clean area. Otherwise, dust or other contaminants may adhere to the tape.
2. Handle reels by the hubs to avoid deformation of the reel sides.
3. Handle tape only with clean hands or white gloves.

Cleaning

1. Again, handle tape only with clean hands or white gloves.
2. Use a high quality magnetic tape cleaner on a clean cloth.
3. Don't use residue-producing cleansers such as carbon tetrachloride.

Use

1. If a tape-induced error occurs, check the tape for pin holes, dents, stretch, excess oxide, or foreign particles.
2. Check the read/write head, capstan, and tape path for dirt accumulation.
3. Check the tape wipers.
4. Don't touch head with fingers, scrape oxide with fingernail, or hold finger or any other object against moving tape.
5. Observe "No Smoking" and "No Eating" rules in computer room.

RIBBONS

An adequate supply of printer ribbons should be maintained. As for most other supplies, the number of ribbons needed will depend on the amount of use and the lead time involved. The life expectancy of a ribbon is usually determined by the number of lines printed in the installation in a given time. As a general rule, the supply should never go below three ribbons.

7. SITE PREPARATION

INTRODUCTION

An important, and too often neglected, aspect of installing data processing equipment is the preparation of the site. Certain requirements concerned with the physical installation of the equipment are mandatory, and must be fulfilled prior to equipment delivery. For a detailed description of these requirements, which pertain to such factors as power, cooling, component arrangement, and floor loading, refer to UNIVAC 9200/9300 Systems Installation Planning Specifications (MH-1185).

Other requirements, although not absolutely essential, are equally important in achieving a productive and trouble-free operation. These factors dealing with design of the environment for equipment, personnel, and storage are the subject of Section 7.

EQUIPMENT ENVIRONMENT

Temperature and Humidity Control

Since improper air conditioning is one of the primary causes of equipment failure, proper temperature and humidity control is probably the most critical aspect of site preparation.

For optimum performance, the equipment room should be kept at a humidity level of 40 to 60% and a temperature of 70° to 75° F. These ranges should be maintained 24 hours a day. Installation of a separate cooling system in the equipment room is recommended to avoid the expense of operating a central air conditioner during nonworking hours. Use of a temperature and humidity recording instrument, capable of continuous recording for at least seven 24-hour periods, is a valuable help in sensing impending trouble.

Dust Control

Equally important in obtaining optimum equipment performance is a dust-free environment. Dust is detrimental to magnetic recording devices and to photo-cell sensing; when allowed to accumulate in the system's built-in filters, it can impede cooling air flow. Floor wax should be used lightly, if at all. Daily damp mopping of the computer area is recommended.

Power Considerations

It is important that power lines feeding the data processing equipment be isolated from large cycling loads, such as those from air-conditioning units, arc welders, or large electric motors. The power line should be checked with voltage recorders for a period of one week. If stable electric power cannot be guaranteed, it is highly desirable to provide the equipment room with its own motor-generator power supply or voltage regulator.

Parallel Equipment Usage

If both old and new equipment will operate during cutover using the same site and power supply, additional air conditioning and electrical power may be required.

FIRE AND SMOKE PROTECTION

Fire fighting equipment for the data processing area must be chosen carefully, because some devices use materials which are harmful to the data media or to the processing system. Carbon dioxide is considered safe for use. Overhead water sprinklers, however, are not recommended.

Protection of the library from fire and smoke is also an important consideration, as the reliability of magnetic recording is reduced by atmospheric impurities. For this reason, smoking should be prohibited in the vicinity of system equipment or tape files. Good fire prevention practices should be followed, such as emptying waste receptacles regularly, preventing the accumulation of used forms and cards, and storing supplies in metal cabinets.

EQUIPMENT LAYOUT

Layout Considerations

In arranging the equipment, primary consideration should be given to the operation of the installation. High operator efficiency depends on good equipment placement, ample table space, and proper storage of cards, magnetic tapes, and discs.

The installation should be designed for effective work flow. Placement of punched card equipment should be functional according to the volume of cards processed, the number of each machine type to be installed, and the sequence of processing for the majority of jobs. Aisles must be provided for movement of carts and trays to and from the equipment. Since disc and tape oriented systems are largely independent of their related punched card equipment, it is generally desirable to locate the punched card peripheral equipment in a different room.

The placement of multiple unit cable-connected equipment is determined to a certain extent by cable lengths. Although there is a specific limit to the cable lengths allowed, these limits are high enough so that they seldom present a problem. (Cable lengths are discussed in Section 3 of UNIVAC 9200/9300 Systems Installation Planning Specifications, MH-1185.)

The physical appearance of the installation is sometimes a consideration in equipment layout; for instance, tape and disc drives may have to face a glass wall in a computef room viewed by the public.

The Univac Field Engineering Manager will review and give advice on all site plans.

Visual Requirements of Layout

Equipment and files in the data processing area should be arranged so that supervisory and operating personnel have ready visual access to equipment and operator controls. Equipment which requires monitoring should be within easy sight of operation personnel. The supervisor should be able to view the entire data processing area from his normal work location.

Typical arrangements of UNIVAC 9200/9300 Disc System components are shown in Figures 7–1, 7–2, and 7–3. These layouts give the operator a complete view of all control panels, card magazines,disc drives, and tape reels from one point. The floor plan should be drawn to suit the specific configuration. The completed floor plan should also show the location of desks, tables, file and storage cabinets, etc., thus allowing review of traffic patterns and access routes.

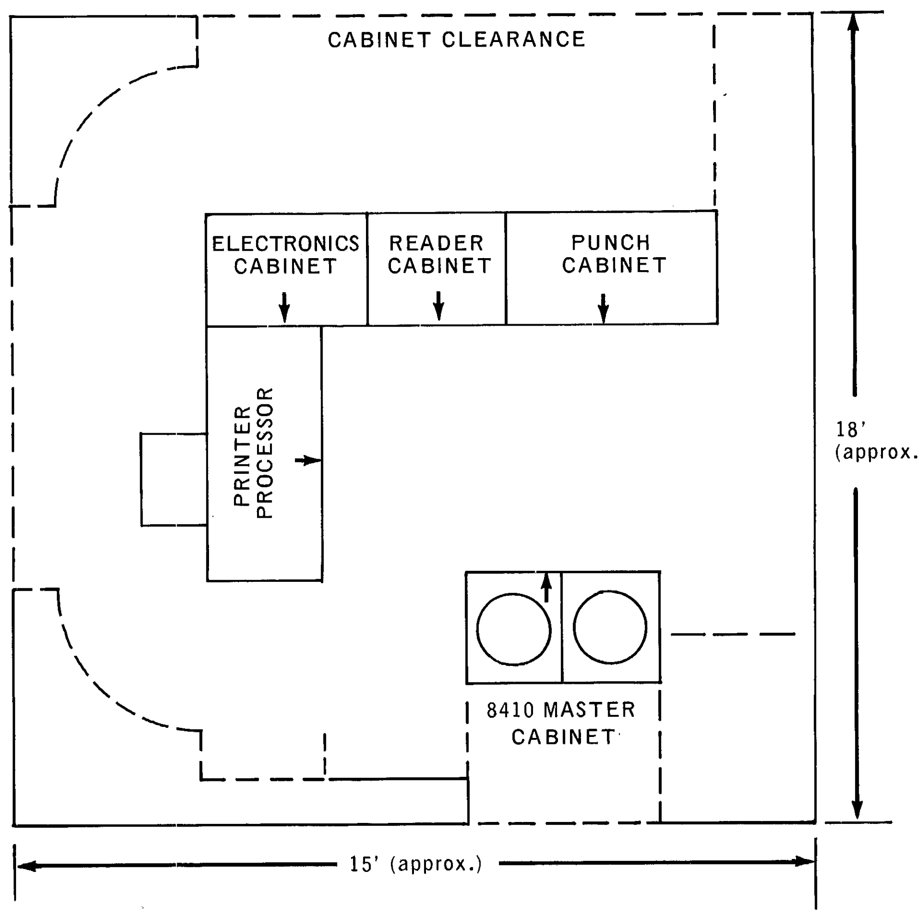

Figure 7–1. Basic UNIVAC 9200/9300 Disc System Installation

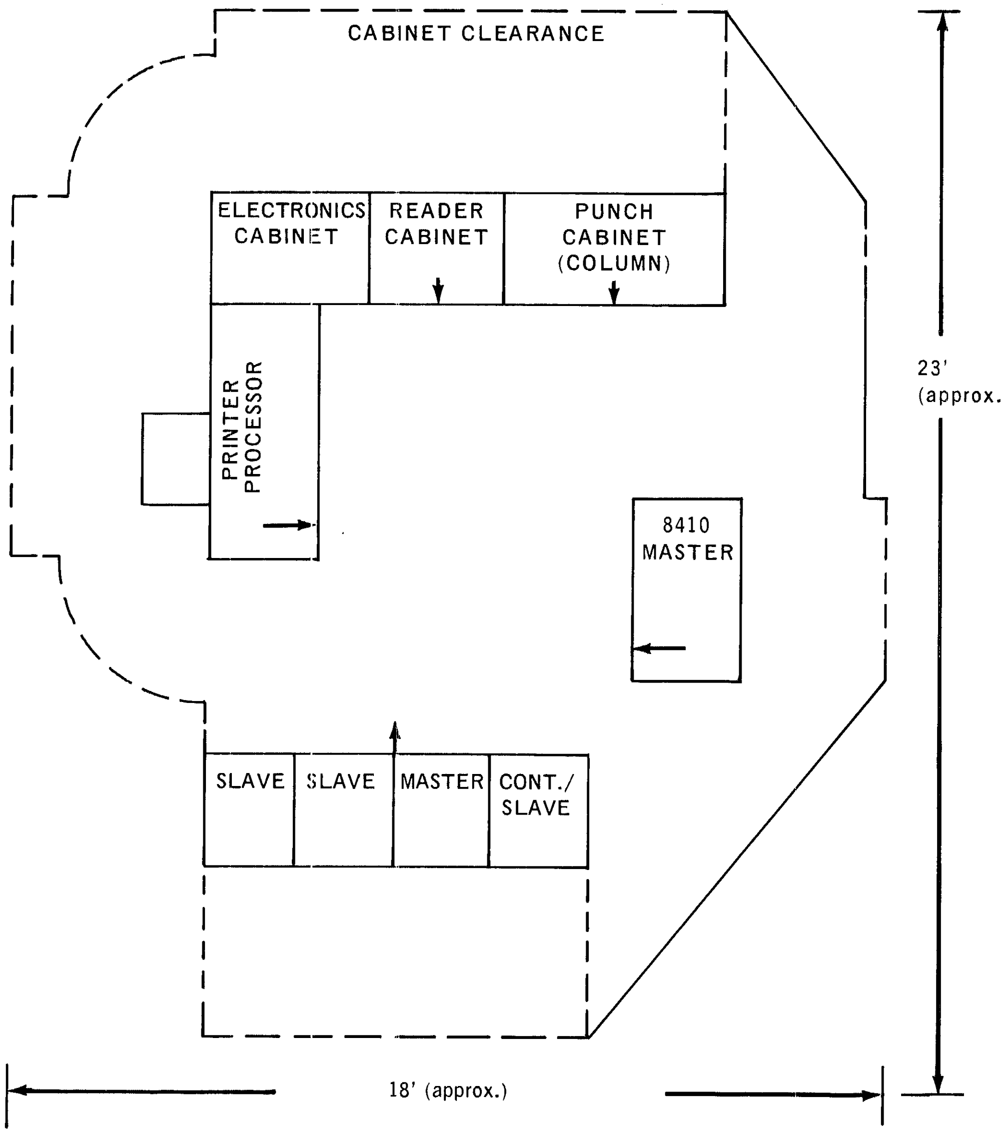

Figure 7–2. UNIVAC 9300 Installation with Four Tape Units and One Disc Cabinet

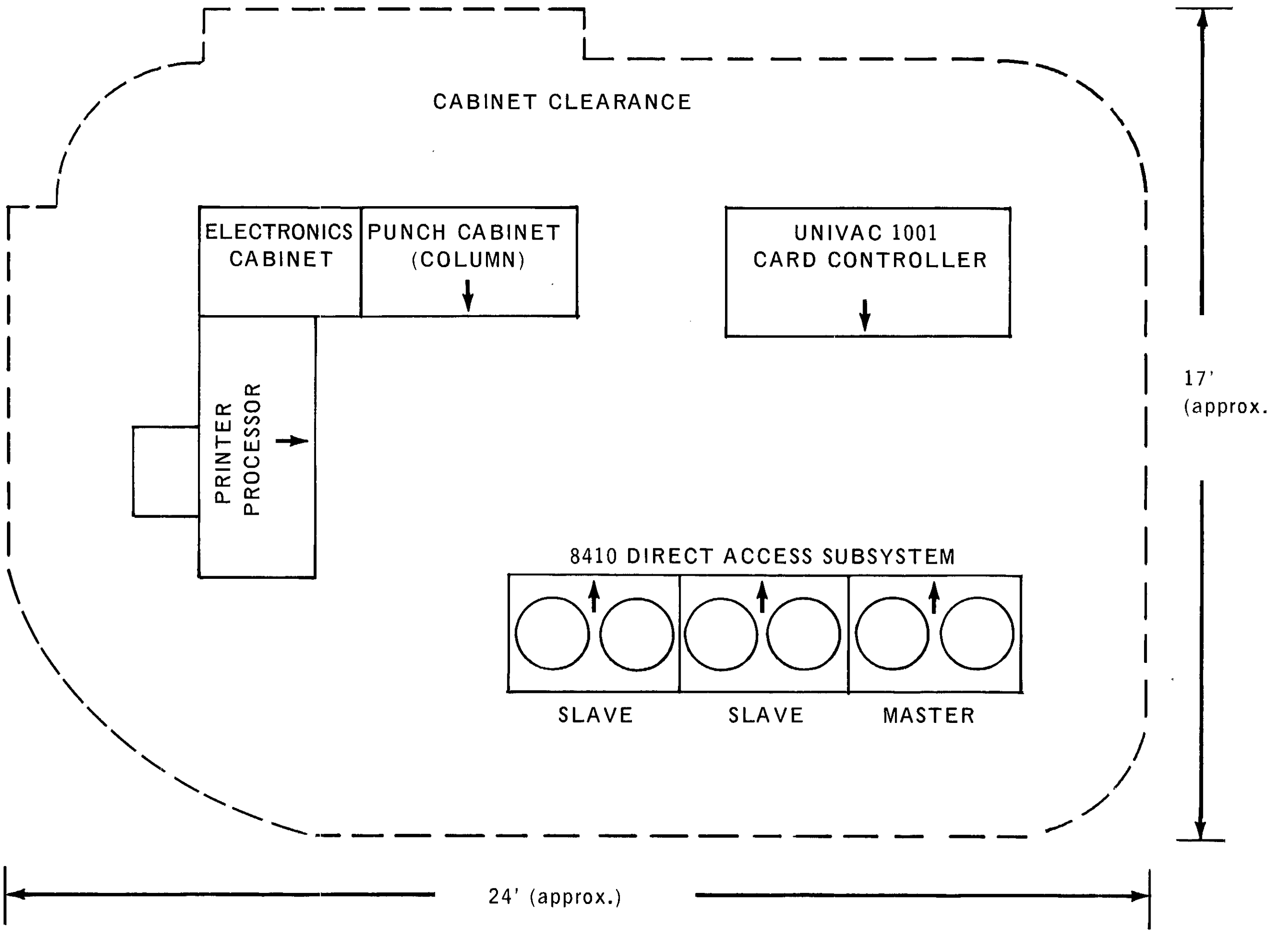

Figure 7–3. UNIVAC 9200/9300 Installation with Three Disc Cabinets and a UNIVAC 1001 Card Controller

WORKING ENVIRONMENT

Proper lighting and control of sound, as well as general appearance and neatness of the working area, are important in maintaining high morale and operating efficiency of the data processing staff.

Sound Control

Soundproofing of the equipment area is important to the operating personnel. The most desirable sound absorbing materials for use in the ceiling are both fireproof and nondusting. Such materials pay for themselves through increased safety as well as efficiency and reliability. Draperies and carpeting also help to reduce the sound level.

Lighting

All areas should be adequately lighted to avoid unnecessary interruptions of normal work flow. In addition, mistakes caused by the staff's inability to see wording, forms, and designations can be costly.

ALLOCATING AUXILIARY WORKING AND STORAGE AREAS

Programming Room

Convenience and quiet are important considerations in selecting the programming room. The area set aside for the programming staff should be planned so that noise level and interruptions are kept to a minimum. It should also have direct access to the computer room or be as close as possible.

Field Engineer's Work Area

The Field Engineer performing preventive or emergency maintenance on the system will need a table surface near the equipment to hold circuit diagrams, tools, and parts while he is working on the equipment. Good site planning will provide adequate work surface for both the Field Engineer and the installation personnel, such as programmers, who require a place to spread out reference documents.

Supply Storage Area

The desirability of a supply storage area adjacent to the data processing area was explained in Section 6, Supplies.

Input and Output Handling Areas

Areas in which input or output data will be accumulated, assembled, or distributed to other departments require special consideration. Since these areas normally provide access to the Data Processing Department for outside personnel, they should be located near the entrance to the department. Because of dust generated by forms bursting and card handling, and also because of outside personnel traffic, this area should be separated from the immediate computer area.

PLANNING FOR FUTURE GROWTH

When laying out the floor plan and allocating space for the equipment room, it is wise to keep in mind the probable need for future expansion. If expansion room cannot be preallocated, at least a plan for the future displacement of an adjacent department is important. Then, when company growth leads to expansion, the installation will not find itself with such inflexible barriers that the only way to go is to the next floor.

Major facility revisions can be avoided if requirements such as space, air conditioning, and power supply are considered at this time.

8. SYSTEM TRANSITION

INTRODUCTION

The term "transition" (or cutover) refers to the transfer of operations from one system or equipment to another. In many cases, the transition involves both new equipment and new procedures. The interdependence of equipment and procedure results from the acquisition of new equipment to satisfy the requirements of new applications or to handle revisions designed to enlarge the scope of existing applications. Procedural changes also come about as systems are revised to make use of particular features of the data processing system. Such changes are quite common when a new storage medium, such as that represented by a direct access system, is being introduced.

The duration of the transition phase varies from one installation to another, and depends upon factors such as the nature of the application and the types of equipment involved. When the installation of a data processing system is to be effected with only minor procedural changes, the transition effort may involve only the substitution of equipment and new programs. The transfer of operations from one system to another in such cases can be accomplished in a very short period of time. If the installation of a new system is associated with extensive procedural changes, however, the transition period may extend over several months.

Three basic considerations are involved in planning the transition:
1. Selection of the cutover method
2. Conversion controls
3. Conversion scheduling

SELECTING THE TRANSITION METHOD

The transitions for unrelated applications should be planned and executed separately. The transition of related applications may or may not be accomplished separately, depending upon the specific nature of the applications involved. The basic division of methods lies between the "immediate" and "parallel" approaches.

Parallel Transition

When this method is used, the old system is operated for a time in *parallel* with the new system, with both systems receiving transactions and producing reports and documents. The outputs of the new system are checked against those of the old system by individual comparison or by control totals. Distribution of the reports and documents produced by the old system continues in the normal manner.

Once the reports from the new system have been examined, they are destroyed. If the new system includes revised clerical procedures, the documents produced are first passed through the clerical operations as part of the examination. This phase of the test provides assurance that the content and format of the new documents reflect efficient clerical operations. It will also enable clerical personnel to become familiar with the new forms and procedures. Documents produced by the new system must be carefully controlled and, if possible, marked to prevent inadvertent distribution.

Undesirable conditions uncovered by the checking operation are corrected by adjustments to programs or procedures. While the alterations are being made to the new system, parallel operations and the search for new discrepancies continue. As the procedures and programs of the new system are refined, the quality of the outputs will improve to a near perfect state. At this time there may be a temptation to cease the operation of the old system on the assumption that the next adjustments made to the new system will be final. This temptation should be avoided, because corrections to programs and procedures can cause additional errors. The new system should not become operational until at least one error-free cycle has been completed. When perfect output is achieved, but not before, operation of the old system is terminated and the new system carries on alone.

Although parallel operation may appear to duplicate the functions of a systems test, it should not be thought of as a substitute for a thorough systems test.

The forms and documents produced by the new system are checked by comparison to those produced by the old system or by a proof system which involves specially developed proof totals. If the outputs produced by both systems are similar in content, verification can be performed by comparison. This comparison of card outputs can be performed manually if volume permits. Whenever possible, however, the use of equipment to perform the comparison will reduce the clerical effort, increase accuracy, and make the results of the comparison available sooner. The use of the equipment as a verification tool is not restricted to card outputs which are identical.

The verification of printed forms will often require visual comparison of documents produced by both systems. Instead of directly producing reports and documents, however, a computer run may write the data onto a disc, to be printed out at a later time. When design permits this type of operation, the equipment can again be used for the comparison. The report disc can be compared to the cards used to prepare the report under the old system.

Proof totals are used alone or in conjunction with comparisons to establish the validity of the products of the new system. These totals are either part of the normal control system, or special totals produced either by routines added to the programs or by special runs. They are amounts which are natural to the application; for example, monetary amounts, quantities, or hash totals (sums of data fields which do not represent quantity). Hash totals of key numbers are often used as a means of providing assurance that records or transactions have not been lost.

Because it provides the best means of ensuring accurate results, parallel operation is a commonly used method of accomplishing the transition. It also provides more time to make and test corrections, eliminating additional errors which can result from hurried efforts. The period of parallel operation is an excellent time to train clerical and operations personnel.

Parallel operation, however, places added burdens on the operating staff, since two systems must be operated at the same time. The increased workload can be partially relieved by the use of temporary personnel.

Immediate Transition

The transition may also be effected by terminating the operation of the old system at the time the new system is activated. Since there is no check of new system against old, the systems test, specialized inspections, and the normal controls are relied upon to guarantee the accuracy of the new system.

Because only one procedure is to be followed, the use of this method reduces the effort required by the operating staff. At the same time, however, there is an increase in pressure caused by the need to produce perfect outputs without complete experience. Although immediate transition eliminates the need to have both systems on site for an extended period, it is advisable to have the old system available until the new system has been proven.

Switching to the new system on an immediate basis makes the systems test a crucial phase of the entire conversion effort. Since there is no "second chance," it must be conducted with a well selected random sample of "live" data, and the results must be carefully evaluated.

While this method reduces the effort required of the operating staff, it places increased dependence on the systems and programming staffs. These personnel are required to guarantee a near perfect system on the very first "live" run. It is highly probable that the first operations of the new system will produce forms and documents that contain errors, although they may be of slight impact. Corrections to systems, procedures, and particularly to programs, can cause other errors,thereby setting in motion a chain of error and correction that may continue through several cycles before perfect quality is achieved.

Immediate transition should be restricted, whenever possible, to cases in which the application has been revised to such a degree that there is little or no way to tie the outputs of the new system to those of the old. Such an occurrence is likely when the scope of the application is significantly enlarged.

Gradual Transition

The foregoing discussion of the parallel and immediate approaches makes no specific reference to either a single application or all applications because the methods are applicable to either situation. The transition effort may concern one application at a time, groups of related applications converted in serial fashion, or all applications at one time.

If all applications are not converted as a group, better distribution of the workload on systems and programming staffs can be achieved. As is the case with the parallel approach, however, two sets of equipment must be on hand for the entirety of the transitional period.

When a particular application or a group of interrelated applications involves high volume, the demands on personnel may become excessive in light of the added activities of systems design, programming, file preparation, and output verification. When the conversion schedule indicates that this will be the case, consideration should be given to executing the conversion in segments. Segmentation will result in a series of transitions – either immediate or parallel – each of smaller volume. The segments may all be of equal size, or they may be increased in size as the new system begins to run more smoothly. Best results are achieved when segmentation is based upon a division which is natural to the application (for example, blocked by account number or groups of sales offices).

CONVERSION CONTROLS

The most important controls to be used during the transition period are those designed for normal operation. They include both the overall systems controls, which relate to functions outside the data processing activity, and special controls used to monitor the internal activities of the data processing group. If these controls are redesigned to be compatible with the new system, the new controls should be instituted during the transition period so that their validity may be determined.

In order to bridge the gap between the old and new controls, and to provide added assurance of quality during the transition period, temporary controls may be added to the system. These controls may consist of additional proof runs, special subroutines added to programs, and visual inspection of documents. They should provide balancing points which occur more frequently than will be necessary when the new system is fully operational. In addition to raising the level of confidence in the new system, temporary controls will make earlier detection of system faults possible.

Recovery techniques are the procedures used to correct the errors revealed by the controls. Recovery techniques used during the transition period may differ from those used for normal system operations. The parallel method of conversion provides a complete check on every run, and recovery means continuous adjustment and rerunning until old and new outputs balance out. If normal recovery techniques fail during an "immediate" transition, the old system can be put back into operation with only minor consequences. This is a last resort and should be avoided whenever possible, however. The "gradual" method of transition presents a more complex problem since normal recovery is used for the portion of the run still under the old system, while new recovery techniques are used for the new system. Any errors in one system must be analyzed to discover what effects they may have on the other system.

Recovery procedures should be simple enough so that only the affected portion of the run need be done over. An example would be the discovery, after running a large payroll register, that errors were made in the third and sixth of 25 departments. Only the rerunning of these two departments, and manual correction of overall run totals, should be required. Large jobs should thus be subdivided to permit simple recovery.

Special steps should be taken for those runs in which an updated file is produced. The old file should be held for two cycles to facilitate its reconstruction in the event of anything happening to the new files. If possible, the input records that caused the changes should be kept in their original sequence.

CONVERSION SCHEDULING

System transition should be scheduled as carefully as are the systems design, systems programming, and file conversion efforts. The detailed schedule should include each operation to be performed and the sequence and the duration of each operation.

The major tasks involved in establishing the conversion schedule are:

1. Itemization of the conversion operations to be performed
2. Arrangement of the list of operations into the sequence in which they should occur
3. Selection of the time period during which the transition activities are to occur
4. Establishment of the conversion schedule

Itemizing Transition Operations

Using the Conversion Sequence Work Sheet (Figure 8–1), list all operations occurring during the transition period that will affect the new system. These operations include individual application runs, special purpose runs used only for the transition activity, and runs to be used for more than one application. Any other functions, such as special reviews, should also be included.

OPERATION	TYPE	INPUT TO	OUTPUT FROM	FREQUENCY	REMARKS

Figure 8–1. Conversion Sequence Work Sheet

As each operation is started, additional information should be entered under the appropriate heading of the Conversion Sequence Work Sheet. A description of these headings follows.

Column Heading	*Explanation of Entry*
TYPE	This entry is used to indicate whether the operation is a regular part of the system, a temporary function, or a one-time operation.
INPUT TO	The entries in this column should reveal all later operations which are directly dependent upon the current operation. In the payroll example the entry in this column might be "Prepare Checks."
OUTPUT FROM	The entries in this column should reveal all prior operations upon which the current operation is dependent. On a payroll application, for example, the operation Net Pay Calculation Run might be followed by the entry "Calculate Gross Pay" in the OUTPUT FROM column.
FREQUENCY	The entry in this column will record whether the run is daily, weekly, monthly, etc.

Sequencing of Operations

Next, arrange the operations in order of priority, using the data on the Conversion Sequence Work Sheet as determinants. The operations should be listed again on another work sheet in proper sequence. The purpose of this step is to make certain that first things occur first, so that continuity of operations under the new system is maintained.

Selecting the Time Period

The time at which the conversion is to begin is generally determined on the basis of the preceding programming and systems design phases and the requirement for the products of the new system. There are, however, certain other factors that may affect the final schedule. A checklist of these factors and other considerations follows:

1. *Anticipated workload*

   The effect of holiday schedules, vacation schedules, or periods of high report preparation activity (for example, year-end operations) must be considered.

2. *Personnel Training*

   The training of operating personnel must be completed by the time the transition begins.

3. *Equipment*

   The utilization of equipment, particularly peripheral equipment, will be higher during the transitional period. There must be sufficient equipment time available to absorb the added workload.

4. *New Equipment Availability*

   The new system should be on site prior to conversion so that operators have an opportunity to become familiar with it.

5. *Site Preparation*

The computer room should be ready for full operation. All work and storage areas must be ready for use, so that operations are able to start with maximum efficiency.

Establishing the Conversion Schedules

The conversion schedule may now be completed using the sequence established on the Conversion Sequence Work Sheet. Each operation should be listed on the ''Detail Conversion Schedule'' (Figure 8–2), which provides for both scheduled date and actual date entries.

Once the Detail Conversion Schedule is completed, its information should be transferred to the master schedule to provide a complete picture of all events.

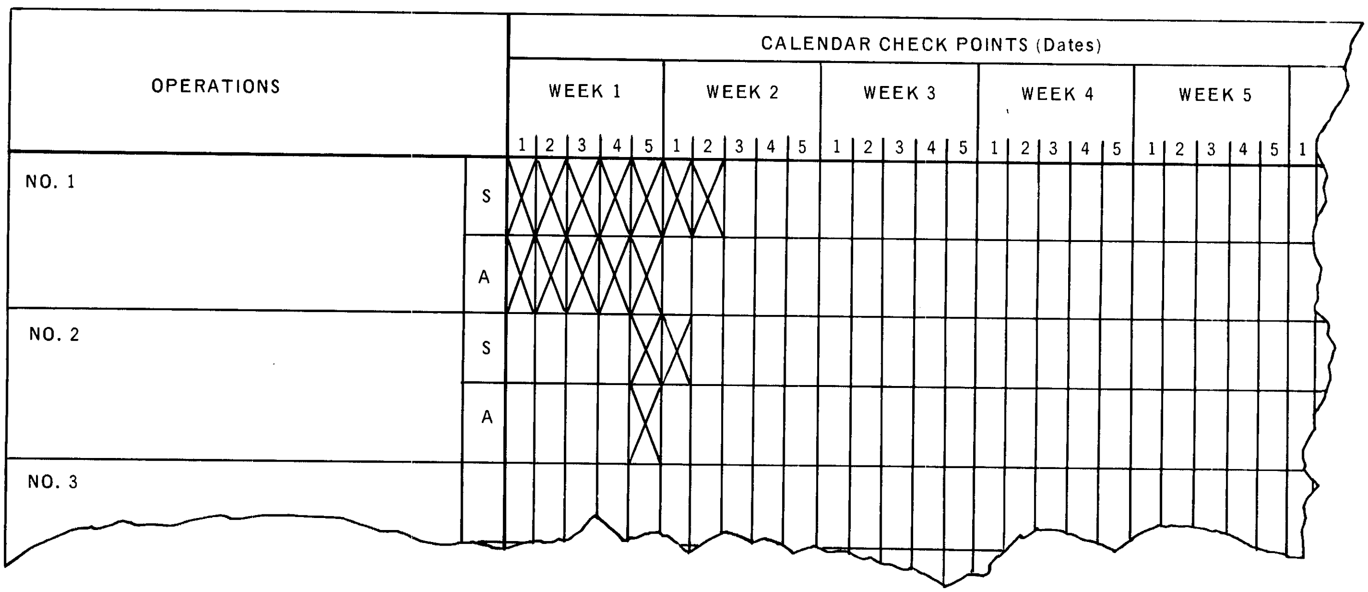

Figure 8–2. Typical Detail Conversion Schedule

9. OPERATIONS

INTRODUCTION

To gain optimum benefit from a new installation, the daily task assigned to the computer must be performed accurately and on time. Controlling the actual operation of the UNIVAC 9200/9300 Disc System is thus the subject of Section 9.

The supervisory, control, operator, and clerical functions of the Operations Department are investigated and explained. To implement planning and control activities, forms are provided which have proved effective in many installations. Not included in the discussion, however, are the auxiliary operations – those functions concerned with data processing equipment which is not used online to the UNIVAC 9200/9300 Disc System, such as keypunches, sorters, collators – since these operations have usually been established in previous installations.

DETERMINING PERSONNEL REQUIREMENTS

The functional groups of the Operations Department are shown in Figure 9–1. While these functions are needed in every installation, the personnel requirements for each function will vary: Large, complex installations may require that a number of people be assigned to a single function; in small operations one person may be able to handle several functions. The installation manager must estimate the workload each function represents and assign personnel accordingly.

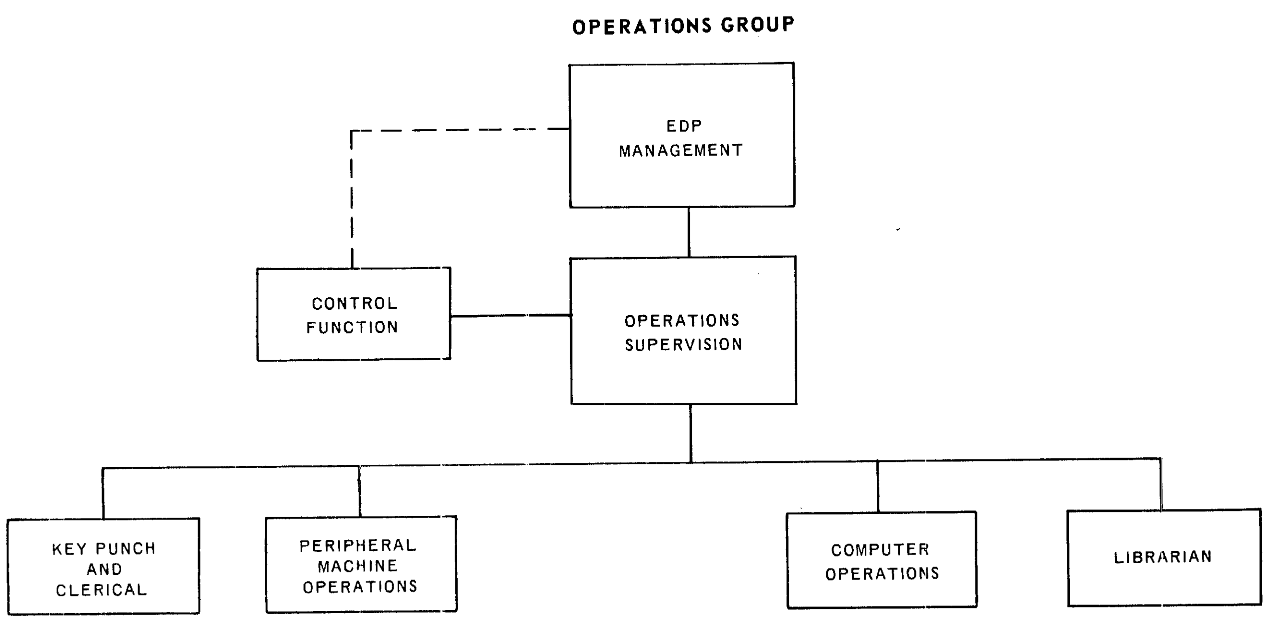

Figure 9–1. Functional Chart of the Operations Group

SUPERVISING THE OPERATIONS DEPARTMENT

In addition to the obvious responsibility for overseeing and directing department personnel, the supervisory function includes reviewing the work produced by the department, maintaining liaison with other departments, and analyzing department costs.

Supervising Department Personnel

Effective supervision of department personnel demands that a schedule of projected activities be established and maintained. With the increased speed and capacity and the new concepts inherent in the UNIVAC 9200/9300 Disc System, the supervisor can no longer afford to rely on memory – a written schedule is essential. Suggested scheduling procedures are described in detail later in this section.

It is the supervisor's responsibility to produce the quantity and quality of work of which his equipment is capable. There are many textbook treatments of the methods of measuring Data Processing Department efficiency. Even the gross quantity gauges, such as cards processed or lines printed per day, are useful. The supervisor must remember that although data processing supervision is a technical and specialized function, his ability to manage personnel will determine the success of the installation.

Reviewing Department Work

Perhaps the most critical supervisory responsibility is to review the work produced by the Operations Department. By examining the quantity and quality of results achieved, the supervisor can detect operating deficiencies as soon as they occur and institute corrective action before a major problem develops.

Maintaining Liaison

Another important duty is that of liaison. This function is three-fold:

1. *With upper management*
   The next level of management must be kept informed of the department's activities and progress. Management, as the sponsoring authority, has a need to know the effectiveness of the Operations Department. In turn, management skills and authority can prove valuable in some department problem situations. It may also provide information about plans for, or demands on, the department.

2. *With the systems design and programming groups*
   Constructive change and growth in a data processing operation is healthy. Free exchange of ideas with these groups will help to assure that changes in programs, systems design and procedures will be directed towards the improvement of the overall data processing operation.

3. *With other departments*
   Liaison with departments that supply input data and receive output information, and with those for which applications are under development, will help to anticipate problems, alleviate areas of possible friction, and correct deficiencies originating outside the Operations Department (late input delivery, for example).

Analyzing Department Costs

Since the real value of the installation cannot be estimated until the cost is known, cost analysis constitutes a fourth major supervisory function. Cost information can be obtained from corporate accounting reports of Accounts Payable, Payroll, Depreciation, Overhead, and other items charged to the Operations Department. From the departmental records of equipment utilization, personnel activity, and supplies consumption, the supervisor can determine the costs in relation to time periods, equipment, and applications. Where records reveal imbalance in peripheral usage, or overtime personnel costs caused by uneven workload distribution, or reruns necessitated by insufficiently trained personnel or inaccurate input data, improvement and correction measures should be instituted.

Some improper situations can be corrected within the department. At other times, the liaison function must be exercised to correct inefficiencies originating in other departments.

SCHEDULING SYSTEM OPERATION

The most effective tool in achieving maximum computer utilization is a realistic schedule. A good schedule will be tight enough to minimize loss of valuable computer time, yet loose enough to allow for delayed input, error recoveries, or other problems. The principal effect of a schedule is to coordinate several activities to culminate in a processing run. A realistic appraisal will reveal that sometimes not all activities will coordinate, thus maximum efficiency cannot always be attained.

General Considerations

A primary scheduling consideration is the availability of input data. Since a computer run often depends on the output of an auxiliary operation (such as keypunching), the UNIVAC 9200/9300 schedule will be closely related to the schedule of auxiliary operations.

Also important in scheduling the UNIVAC 9200/9300 Disc System is proper utilization of Control Stream, if the system configuration permits its use. This feature permits immediate call-in of a program at the conclusion of the preceding program. If good liaison has been maintained with the systems design and programming groups, programs will have been prepared to minimize file and paper changes. The supervisor can then plan the daily sequence of programs so that unavoidable operator duties can take place while the current program is operating. Thus, by reducing operator intervention, Control Stream increases productive time and lessens the possibility of introducing errors in the operation.

Note that initial efforts rarely produce an optimum schedule. Needed amendments will become obvious as the daily records of equipment utilization are examined.

The Master Schedule

By the time the conversion is completed, certain policy guidelines will have been set; priority standings of various reports will be established; and the basic cycle of operations and the runs required will be known. The run timings can be estimated either from experience or, for new programs, from pilot runs and systems tests.

To create the master operating schedule, a large daily calendar based on the work cycle (usually the calendar month or a four-week period) is used (see Figure 9–2). It could be a paper calendar, a large chalkboard, or a visual display board, depending on the user's preference.

The daily runs are scheduled first, according to priority, taking into consideration the volume fluctuations caused by business cycles. Next, the weekly runs are added, then the monthly ones, and finally any remaining ones.

The master schedule must allow for unexpected changes such as those caused by emergency reporting demands, faulty input or necessary reruns. To achieve this flexibility, periodic runs should be scheduled for execution as soon as input is available; they should also be spaced throughout the time available to provide an even workload, leaving a little spare time each day.

Personnel involved in auxiliary operations should be notified of any change in the master operating schedule so they can make a corresponding change in their schedules if necessary.

DAILY CALENDAR – SHIFT 1 – FOUR WEEKS BEGINNING______________

MONDAY	TUESDAY	WEDNESDAY	THURSDAY	FRIDAY	SATURDAY
DAILY RUN #6 WEEKLY RUN #51 MONTHLY RUN #8 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #17	DAILY RUN #6 MONTHLY RUN #102 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #2	DAILY RUN #6 MONTHLY RUN #34 WEEKLY RUN #3 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #8	WEEKLY RUN #7 DAILY RUN #6 WEEKLY RUN #9 DAILY RUN #4 DAILY RUN #9	DAILY RUN #6 WEEKLY RUN #12 DAILY RUN #4 DAILY RUN #9 MONTHLY RUN #11	
MONTHLY RUN #3 MONTHLY RUN #19 MONTHLY RUN #2 DAILY RUN #6 DAILY RUN #4 DAILY RUN #9	DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 MONTHLY RUN #4 WEEKLY RUN #51	HOLIDAY	WEEKLY RUN #17 WEEKLY RUN #2 DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #3	WEEKLY RUN #8 WEEKLY RUN #12 DAILY RUN #6 DAILY RUN #4 DAILY RUN #9	WEEKLY RUN #7 MONTHLY RUN #5
DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #51 MONTHLY RUN #14	DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #17 MONTHLY RUN #16	DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #2 WEEKLY RUN #3	DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #8 WEEKLY RUN #7 WEEKLY RUN #9	DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #12 MONTHLY RUN #6	
DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #51 WEEKLY RUN #17	DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #2 WEEKLY RUN #3	DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #8 WEEKLY RUN #12	DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 WEEKLY RUN #7 MONTHLY RUN #9	DAILY RUN #6 DAILY RUN #4 DAILY RUN #9 MONTHLY RUN #10	

Figure 9–2. A Four-Week Daily Calendar

The Daily Operations Worksheet

The Daily Operations Worksheet is a working paper used to arrive at a Daily Operations Schedule. On it, the manager lists the runs which must be performed to meet his schedules. Then, considering the units within his configuration used by the various runs, the file and forms changes which must be made between runs, the time estimate of each run, and the priority of the runs, he arranges them in optimum sequence. The Daily Operations Worksheet should be prepared as far in advance as possible so that the effect of deviations from the current schedule can be plotted into the next period.

The Daily Computer Operations Schedule

In an installation with few people, the worksheet may suffice as a daily schedule. Larger installations, however, usually need a more formal one. Figure 9–3 illustrates a simple Daily Computer Operations Schedule based on a twenty-four hour day, the smallest time increment being fifteen minutes. This schedule could be amended to represent any work-day length and any time increment desired; it could also be expanded to indicate the components scheduled for use.

DAILY COMPUTER OPERATIONS SCHEDULE

DATE ______________

TIME	OPERATION OR PROGRAMMER	TIME	OPERATION OR PROGRAMMER	TIME	OPERATION OR PROGRAMMER
0000		0800		1600	
15		15		15	
30		30		30	
45		45		45	
0100		0900		1700	
15		15		15	
30		30		30	
45		45		45	
0200		1000		1800	
15		15		15	
30		30		30	
45		45		45	
0300		1100		1900	
15		15		15	
30		30		30	
45		45		45	
0400		1200		2000	
15		15		15	
30		30		30	
45		45		45	
0500		1300		2100	
15		15		15	
30		30		30	
45		45		45	
0600		1400		2200	
15		15		15	
30		30		30	
45		45		45	
0700		1500		2300	
15		15		15	
30		30		30	
45		45		45	

Figure 9–3. Sample Daily Computer Operations Schedule

CONTROLLING SYSTEM OPERATION

A control group must be established to maintain records of documents entering and leaving the department and to establish controls on data received. The various types of input must be routed to their proper destination within the department. In addition, input documents often require editing preparatory to keypunching; for punched cards, it may be necessary to perform card counts or validation of control hole punching.

These records can then be used to determine the operating efficiency of the Data Processing Department and, most important, to assure validity of the output.

The control function may constitute a separate group or be combined with other clerical functions, depending on the size and organization of the department and on the volume of work.

Recording and Controlling System Input/Output

It is important to record the arrival of all incoming documents – whether paper, punched cards, magnetic tapes, or discs – which will be used as system input by the Data Processing Department. This is particularly true for documents which are routed through the Data Processing Department destined for other departments.

The Input Document Log, shown in Figure 9–4, has been devised for this purpose. The date and time received, source, and description of the documents should be entered, as well as any control information accompanying the documents. When the documents are released for processing, the date and time, and the name of the person taking it should be noted. Any abnormalities in the data, and any action taken to correct them, should be noted in the "Comments" column.

INPUT DOCUMENT LOG

RECEIVED		DOCUMENT	CONTROLS	RELEASED		COMMENTS
TIME	FROM			TIME	TO	
DATE ONCE EACH DAY						

Figure 9–4. Sample Input Document Log

The Use of Control Totals

The use of control totals to confirm accuracy is essential in obtaining reliable system output. If possible, control totals should be established by the originator of the input data. If they are not, control information should be developed by the Operations Department itself.

Control totals are most effectively maintained on data fields whose sums have significance to the finished reports; for example, those representing money, unit count, weight, etc. When meaningful totals are not possible, ''hash'' totals (that is, totals which have no significance other than for control) can be established for fields containing information such as account or invoice numbers.

Control totals should be checked by the Data Processing Department when recording the receipt of data. Control totals should also be used to verify the reliability of the processing that follows. They are particularly effective in localizing errors when batch identification processing techniques are used.

Control of Input Errors

The Operations Department must be prepared to handle any errors detected in the input data, or any occurrences which cause the normal work or data flow to be interrupted. (Errors or interruptions, as well as action taken to correct or alleviate them, must be noted and reported to the data processing supervisor. While events of only incidental importance do not necessarily have to be reported immediately, continual improvement of the data processing operation requires that all types, sources, and frequencies of error be analyzed so that corrective measures will eventually minimize the incidence of error.)

A common practice in computer installations is the use of the computer itself to validate input data. Examples of computer validation include tests for reasonableness and cross-footing built into the programs. Input which does not qualify for processing is ignored in the normal program flow and printed out in an error listing at the end of the run. Control totals must then be adjusted and an analysis of the error listing made.

The means of correcting data and making adjustments to operating procedures should be devised and agreed upon ahead of time. Criteria should be established for handling faulty input data. For example, under what conditions should input be corrected by Data Processing Department personnel? When should it be returned intact to its originator? When should error items only be rejected?

Another frequently overlooked problem is the reentry of previously rejected data. If the balance of an application has been run, and error items are to be run at a later time after correction, control total adjustments will have to be made for both the main part of the run and for the subsequently corrected error items.

Validation of Computer Output

Once beyond the program testing stage, valid input should result in valid output. Verification procedures, therefore, are usually confined to visual spot checks.

The computer output must be examined for any omissions of data or operations, for the entry of erroneous data, and for any signs of machine failure. An examiner who is familiar with report requirements should make a visual check to find major errors in format and content. Comparison of control totals should reveal data omissions. Any errors detected should be noted and an established procedure followed.

Handling of System Output

A function which is frequently underestimated is the physical preparation and the distribution of system output. With the amount of documentation that the 9200/9300 printer is capable of producing in a single eight-hour shift, the removal of perforated margins, decollating of multiple parts, bursting, binding, logging, and distribution can be a sizeable job in itself. Time, personnel, and equipment must be allocated to the job. Procedures must be established to get the work done quickly and efficiently. The Output Document Log, shown in Figure 9–5, is an excellent means of keeping track of this work.

OUTPUT DOCUMENT LOG

RECEIVED		DOCUMENT	COMPLETE?	NO. OF PARTS	ACTION TAKEN				RELEASED			COMMENT
TIME	FROM				BAL.	BURST	BIND	OTHER	TIME	COPIES	TO	
DATE ONCE EACH DAY	COULD BE OTHER THAN 9300		YES OR NO		BALANCED? YES OR NO	X	X					

Figure 9–5. Sample Output Document Log

Control of the distribution of confidential or sensitive information is a further requirement. The master and daily schedules will show when this type of information is to be produced. A supervisor or other responsible person might be assigned to operate the system when such output is being produced, or to distribute it outside routine channels. Such material is sometimes printed after normal working hours.

Establishing a Manual of Procedures

An invaluable tool in every data processing organization is the Manual of Procedures. This document should describe with absolute accuracy every aspect of the data processing operation. Insofar as possible, every requirement, procedure, condition, situation or result, whether desirable or undesirable, should be anticipated and planned for.

The input expected should be described and illustrated by examples. Each application should be clearly and completely defined, and the desired results described. Document routing should be described. System output should be defined, described, and illustrated with examples. Distribution of system output should be indicated. Special requirements for controlling confidential or sensitive documents should be explained. Tape and disc control procedures to be followed by the computer operator and the librarian should be specified.

Extraordinary situations – for example, the appearance of erroneous input or output data, and unusual operation or documentation requirements – should be anticipated and the action to be taken in each case explained.

The Manual of Procedures should be complete and accurate to the point that the times when the department supervisor's assistance is needed are kept to an absolute minimum.

OPERATING THE 9200/9300 DISC SYSTEM

Efficient computer operation requires as much training, intelligence, and planning as do programming and systems design. To familiarize the operator with the UNIVAC 9200/9300 System, Univac provides formal and informal training in equipment handling. Besides having expert knowledge of computer operation, however, the operator must also be efficient in minimizing the time required for set-up and handling of supplies.

Monitoring System Operation

The operator's major duty is to monitor the entire computer operation. This includes setting up the system,initiating operation, and providing necessary supplies such as forms, discs, tapes, and cards to the system. The operator should also make at least a cursory examination of input and printed output.

Should an error occur, he must recognize it, institute the proper recovery procedure, and report the occurrence. A special form should be available for reporting the occurrence of errors and the action taken. This form should provide a method of distinguishing among data, program, operator, and equipment errors.

Obviously, the operator cannot be expected to remember all the possible operating conditions of every program in the system. Consequently, he will need a Run Book giving detailed instructions for set-up, error recognition, and error recovery for each program.

The operator is also responsible for instituting the proper close-out procedures. Again he needs detailed instructions – from the Run Book for the close of the program, and from the Manual of Procedures for distribution of output and return of files to the library.

Keeping a Daily Machine Utilization Record

One of the records which the department supervisor will require, if he is to properly analyze the operation of his department, is a Daily Machine Utilization Record. On this form, the operator should record start and stop times of each run, calculate elapsed times and, if desired, note the equipment used. A sample form is shown in Figure 9–6.

The operator should be impressed with the importance of recording both start and stop times promptly and accurately. Start times are usually correct because the operator must be at the console to start the job. He may not be there when the job terminates, however, leading to inaccurate stop times, with consequent inflation of run times and lowering of set-up times. On the other hand, if accurate records are kept, set-up times can often be reduced, and the efficiency of the operation thereby increased.

PERFORMING PREVENTIVE MAINTENANCE

To assure maximum continuity of machine operation and avoid time loss due to reruns or service calls, the operating personnel should assume the following preventive maintenance responsibilities. These responsibilities are stated here in general terms. Where more specific instructions are necessary, they are covered in operating instruction manuals.

All machine surfaces should be kept clean and free of loose objects such as paper clips and pencils.

Any unusual noise or abnormal condition of a part, component, or control should be reported to Univac Field Engineering.

Punch

- Empty the Chip Receiver at regular intervals; clean the inside area around the Receiver.
- Remove card jams.
- Clean Read Photo-Diodes at regular intervals.
- Clean lint from card transport.

Reader

- Remove card jams.
- Clean Read Photo-Diodes at regular intervals.
- Clean lint from card transport.

Console Cabinet

- Make Indicator Light check each time CLEAR switch is pressed.
- Observe any malfunction in switch operation.
- Clean air filter.
- Observe level of oil gauge.

UNIVAC 9200/9300 SYSTEM

DAILY MACHINE UTILIZATION RECORD

DATE ______________

PAGE ______________

RUN IDENTIFICATION	START TIME	STOP TIME	ELAPSED TIME	USAGE CODE	OPER-ATOR	COMMENTS*	COMPONENTS IN USE								
							RDR.	PTR	PUN			SER-VO		DISC	
												0 2 4 6	1 3 5 7	0 2 4 6	1 3 5 7
												0 2 4 6	1 3 5 7	0 2 4 6	1 3 5 7
												0 2 4 6	1 3 5 7	0 2 4 6	1 3 5 7
												0 2 4 6	1 3 5 7	0 2 4 6	1 3 5 7
												0 2 4 6	1 3 5 7	0 2 4 6	1 3 5 7
												0 2 4 6	1 3 5 7	0 2 4 6	1 3 5 7
												0 2 4 6	1 3 5 7	0 2 4 6	1 3 5 7
												0 2 4 6	1 3 5 7	0 2 4 6	1 3 5 7

INSTRUCTIONS FOR USE:

Between horizontal double lines enter main program I.D., Start and Stop times, etc. Indicate components used with "X".

In same horizontal block on succeeding lines show symbiont usage. Indicate symbiont-assigned components with "O".

Fill out completely.

Be accurate with times.

Enter events as they occur.

Comment on any difficulty encountered.

USAGE CODES:

P – Productive Time
O – Operator Error
R – Rerun Time
T – Program Test Time
A – Run Diagnostic Time

I – Idle Time
M – Preventive Maintenance
D – Down Time
E – Emergency Maintenance

* Under "comments" explain any entry other than P, T, M.

Figure 9–6. Sample Daily Machine Utilization Record

Printer

- Change ribbon.
- Install forms.
- Check quality of printing.
- Check functioning of controls.

Tape Units

Perform 2- and 8-hour cleaning as described in the UNIVAC 9300 System UNISERVO VI C Subsystem Programmer/Operator Reference Manual (UP-7583). To minimize damage to the tape, keep the tape free of dirt and dust and observe proper tape handling techniques.

Card Controller

- Remove card jams.
- Clean Read Photo-Diodes at regular intervals.
- Clean lint from card transport.

8410 DAS Head Cleaning Procedure

Should loss of output or excessive error rates occur during operation of the 8410 DAS, head cleaning should be done in the following manner:

1. Stop the handler.
2. Replace the recording disc cartridge with the cleaning disc.
3. Allow the top cover to remain open.
4. Rotate the cleaning disc slowly approximately twenty full revolutions, CLOCKWISE ONLY BY HAND.

   DO NOT TURN MOTOR POWER ON.
5. Remove cleaning disc.
6. Replace recording disc on handler and resume operation.
7. If trouble persists, repeat the procedure one more time.
8. If trouble still persists, call Univac Field Engineering.

MAINTAINING A LIBRARY

One of the most important functions in a computer installation is maintenance of the library. Each disc or reel of tape must be classified and labeled for identification. Since discs and tapes can be reused and the information previously contained thus destroyed, their issuance must be carefully controlled. They must be properly housed when not in use. Damaged tapes must be stripped and worn tapes retired. Discs which have developed bad spots must be reprepped or limited in use to files which do not need the unusable sectors. Finally, records must be kept. Complete library control is essential, since a ruined file can upset schedules or cause other serious problems – at best it will require machine and personnel time.

For proper functioning of the library, the following two rules should be followed:

1. Each instance of file handling must be recorded as it occurs.
2. No one but the librarian in charge should remove files from or replace them in the library.

Classifying Discs and Tapes

By Application: A common method of classifying tape reels or discs is to color code them by application. In this method, each active reel or disc pertaining to an application bears a correspondingly colored identification label. When a reel or disc (both sides) no longer contains useful information, the application label is removed to indicate availability.

By Inventory Numbers: Another popular method of organizing libraries is to store reels and discs by inventory numbers and to maintain accurate cross-referenced records by file number and reel or disc number. Thus, the File Number Record would show the reel or disc on which each generation of the file was recorded; from this record the librarian can determine when a reel or disc has become available for use. The Reel or Disc Number Record would show the file currently recorded on the reel, the date of recording, and the history of the unit.

Although reel and disc filing are essentially the same, extra care must be taken to educate the personnel who handle discs. Discs are controlled by disc surface, "A" and "B," rather than by housing. Each surface can contain one or more files, unrelated to the file on the other surface. Also, it is important to remember that, when the disc is mounted, its under side is online. Therefore, so that the file being processed may be identified, the external label to that file should be placed on the side of the cartridge opposite the file itself.

Maintaining Back-Up Files

Good organization in tape installations incorporates the "grandfather-father-son" concept, which saves the two previous versions of each file as back-up to the current version. Using this concept ensures that if a current file is lost, damaged, or mishandled, it can be recreated without excessive effort. Similar protective measures should be taken in direct access oriented installations. Variance from the three generation concept may be made when another machine-readable version of the file, such as punched cards or an identical file in a different sort sequence, exists and can serve as back-up. When processing accomplishes destructive updating of files, there is no natural "new generation" file created, so protection must take the form of duplicate recording or prior file copying with retention of transaction records. The Write and Check function should be used to assure future readability of newly written records.

Maintaining Records

The Job Control Record: The librarian should have a Job Control Record (Figure 9–7) for every job to be run on the system. The name of the file which is to be run on each disc or tape drive and the number of days in the retention cycle should be entered in the lower portion of the form. As discs or tape reels are readied for release each day, the unit numbers should be entered on the appropriate Job Control Records, in the column related to the disc or tape drive on which the file will be used. This record will provide a history of the discs or reels used for a specific job; it will also act as a signal should consistent problems occur in any drive.

LIBRARY JOB CONTROL RECORD

JOB NAME: ____________________ JOB NUMBER: ________

REEL NUMBER	TAPE SERVO								DATE RUN	DISC DRIVES								DISC VOLUME NUMBER
	0	1	2	3	4	5	6	7		0	1	2	3	4	5	6	7	

SER-VO	FILE NAME	RET. CYC.
0		
1		
2		
3		
4		
5		
6		
7		

DISC	FILE NAME	RET. CYC.
0		
1		
2		
3		
4		
5		
6		
7		

Figure 9–7. Sample Job Control Record

Disc and Reel Labels: When discs or tape reels are received, a permanent label should be affixed by the librarian. The label should show the manufacturer's serial number, the date of receipt, and the installation-assigned serial number.

Working Labels: Besides the permanent label bearing the assigned reel or disc number, each reel and disc surface should have a temporary or "working" label. This label, usually printed on pressure-sensitive adhesive-backed stock, identifies the information recorded on the unit. Figure 9–8 shows two commonly used working labels; Figure 9–9 illustrates a number of special purpose temporary labels. All these and others are available from various forms and supply item vendors.

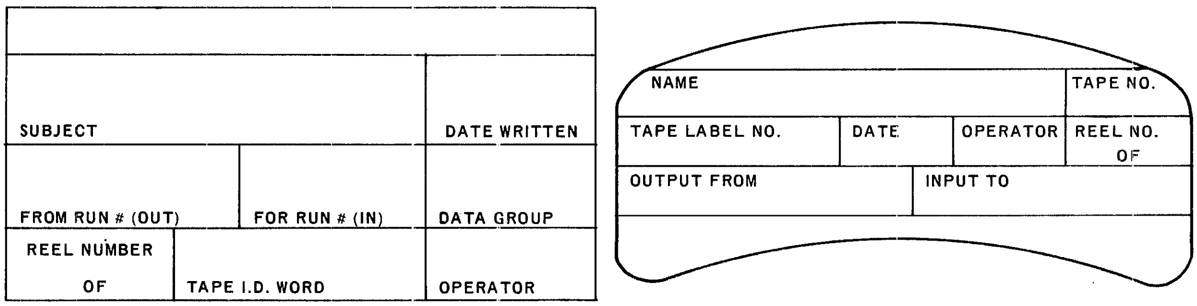

Figure 9–8. Two Commonly Used Working Labels

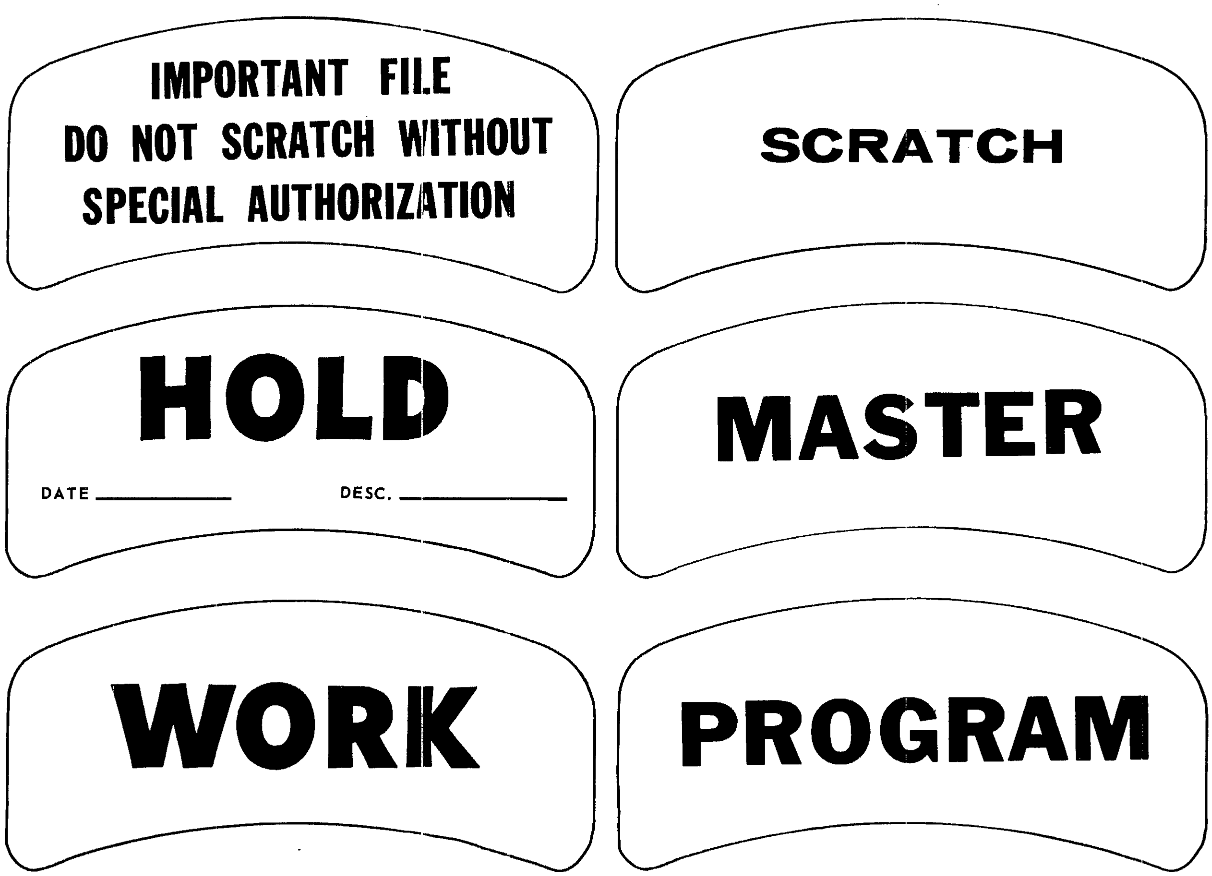

Figure 9–9. Some Special Purpose Temporary Labels

Reel and Disc Control Records:

When new disc cartridges or tape reels are received into the installation, a Control Record should be started. This record should be headed by the disc or reel serial number (assigned by the manufacturer or at the installation). It should show the date of acquisition and record each instance of use. The name of the file or files currently recorded on the unit should be shown along with the name of the run which created them, the date created, and the expiration date of the files. As tapes are shortened or discs reprepped, the control record should be changed to reflect the condition of the unit. A sample Disc/Reel Control Record is shown in Figure 9–10.

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
FILE RETENTION DATE																														

DISC/TAPE CONTROL RECORD

☐ DISC ☐ TAPE

DISC REEL SERIAL NO. ________________ REEL LENGTH ____________

DATE RECEIVED ____________ DISC PREP DATE ____________

MANUFACTURER ______________________________

FILE NAME	DATE RUN	RETEN. CYCLE	DISC SIDE		RELEASED	
			1	2	DATE	BY
CONDITION:						

Figure 9–10. Sample Disc/Reel Control Record

Issuing Tapes

Normal Issuance: The librarian assembles the discs and tapes to be run during a given period, based on the runs specified in the Daily Run Schedule. A copy of the Run Book should be available from which the librarian can get the necessary information to complete the Disc and Reel Control Records, the Job Control Record, and the working label for each unit. As these records are completed and the labels affixed to the units, the Write Enable Rings or plugs should be inserted as required. Once the librarian has checked to make sure that no error was made, the tapes and discs can be released to the operator.

Special Allocations: In some installations, certain persons (programmers, engineers) are permitted to request that tapes and discs be removed from the normal cycling of the tape library and allocated to their use. In such cases, the librarian should insist on a written request and should prepare an Allocation Record (Figure 9–11) indicating which discs or tape reels have been assigned. These units should be labeled and housed in a separate part of the library; their Control Records should be marked to indicate that the unit is not available. The Allocation Record should be reviewed periodically to avoid withholding discs or tapes unnecessarily.

DISC ALLOCATION RECORD			
NAME			
DISC NUMBER	DATE		PURPOSE
	ASSIGNED	RELEASED	

Figure 9–11. Sample Allocation Record

Availability for Unscheduled Use: Provision must be made in the library routine to have tapes and discs available for programmers or operators who may have unscheduled needs. If a tape or disc is required immediately, the librarian can forego formal checkout procedures and note only the disc or reel number and to whom and when it was released. Such units will be identified only by the disc or tape reel label. Once released, they are the responsibility of the recipient. If any data is recorded which must be protected, the person using the unit must give the librarian all the information necessary for proper labeling and updating.

Return of Discs and Tapes to the Library

The tapes and discs must be returned to the library at the end of the workday. The operator should have initialed each working label to confirm that the scheduled run was completed. The librarian must check the returned units against her records to ensure that each tape and disc is accounted for. The Write Enable Rings and plugs should be removed. The tapes, discs, and records involved can then be filed.

As the tapes and discs are filed, the associated grandfather tape and back-up disc should be examined for expiration of the retention date. If all runs are on schedule, the reserve files will have reached their retention dates and can be designated as available for use.

Tape Maintenance

The first few feet of the tape are handled to thread the tape on the tape handler. Should bad or excessive handling damage this area to the point where it becomes unusable, the tape librarian must ''strip'' the tape – that is, cut off the damaged area and move the ''Beginning of Tape'' reflective strip. When a tape has been stripped, the date should be noted and the reel length amended on the Reel Control Record. Further, the tape reel should be marked ''short'' and a flag placed on its Reel Control Record. When subsequently issuing the tape, the librarian must make sure that the shorter length will not affect the intended use of the tape.

Disc Maintenance

Disc cartridges need no maintenance except normal care in handling and protection from excessive airborne contamination. Occasionally, the disc surfaces may need cleaning with a Freon solvent. This cleaning should not be attempted by installation personnel until after instruction in the technique by a Univac Field Engineering representative.

GLOSSARY

ACCESS TIME: See *Time, Access.*

ADDITION, RECORD: (*n.*) 1. The allocation for and storage of a new record in a file. 2. A transaction containing information that is to initiate a new record.

ADDRESS: (*n.*) 1. A label, name, or number which designates a register, a location, or a device where information is stored. 2. That part of an instruction which specifies the location of an operand.

ADDRESS, LOGICAL: (*n.*) The address of a unit of storage within a system, such as the address of a record within a file, without regard to the actual physical location on direct access storage. *Syn.* – File relative address.

ADDRESS, PHYSICAL: (*n.*) The actual physical location or hardware address of a particular item, usually a record, in direct access storage.

ADDRESS, RECORD: (*n.*) The address in storage where a particular record may be found.

AFTERLOOK: (*n.*) The copy on tape or disc of a record, reflecting changes made after a transaction has been processed against it.

ALPHANUMERIC: (*adj.*) Containing both numerals and letters.

AMERICAN STANDARD CODE FOR INFORMATION INTERCHANGE: (*n.*) An eight-bit code used for data notation. Usually referred to as ASCII.

APPLICATION: (*n.*) The business system or problem to which a computer is applied.

ASCII: (*n.*) Acronym for American Standard Code for Information Interchange. See *American Standard Code for Information Interchange.*

AUDIT TRAIL: (*n.*) See *Change Tape.*

BLOCK: (*n.*) A grouping of data, in most cases, records, which remains fixed in one computer application. The size of blocks is determined by the systems analyst. When data is manipulated, it is generally considered as an integral part of its block. *Syn.* – Bucket.

BLOCK, INDEXED SEQUENTIAL: (*n.*) One of several groups of sequentially stored records in an indexed sequential file. The size of the block is predetermined for the user.

BUCKET: (*n.*) See *Block.*

CHANGE TAPE: (*n.*) A recording of all transactions or updated records (afterlooks) which is kept for a set period of time to ensure data integrity. In the event of hardware or program error, all past transactions are available on the change tape to help re-establish the files. *Syn.* – Audit Trail.

CHAIN: (*n.*) A linking together of data records or file areas by storing the address of one in the other.

CODE, HEXADECIMAL: (*n.*) A code which is based on the number system of base sixteen.

CONTROL UNIT: (*n.*) That portion of the hardware of a computer which directs a sequence of automatic operations, interprets the coded instructions, and initiates the proper signals to the computer circuits to execute the instructions.

CYLINDER OF DATA: (*n.*) See *Position Capacity.*

DASD: (*n.*) Acronym for Direct Access Storage Device(s). See *Direct Access Storage Device.*

DATA BASE, USER'S: (*n.*) 1. All of the files of a particular application which are interrelated and stored on DASD. 2. All the files of a particular customer that are stored in, or available to, the computer system.

DATA FIELD: (*n.*) The information contained within a field.

DEBUG: (*v.*) To remove the errors from a program in order to put it in operational status.

DELAY, ROTATIONAL: (*n.*) The time delay caused by the rotation of the direct access storage device. *Syn.* – Latency.

DELETION, RECORD: (*n.*) The removal of a record from a file, usually by placing zeros in its place.

DIRECT ACCESS STORAGE DEVICE (DASD): (*n.*) A device on which all information is equidistant in access time, usually as applied to electromechanical file storage.

DIRECT RELATION FILE ORGANIZATION: (*n.*) A method of file organization whereby the record identifier corresponds to the address of the record in storage.

DISC, MAGNETIC: (*n.*) A storage device in which information is recorded on the magnetizable surface of a rotating disc. A magnetic disc storage system is an array of such devices, with associated reading and writing heads mounted on movable arms.

DISTRIBUTED FILE ORGANIZATION: (*n.*) A method of file organization whereby the records are stored in buckets of several records each. A randomizing technique is used to place the records in the buckets. Each bucket must be scanned to find a desired record.

DRUM, MAGNETIC: (*n.*) A storage device in which information is recorded on the magnetizable surface of a rotating drum.

DUMP: (*v.*) To copy a DASD file, usually onto a tape.

DUMP, MEMORY: (*n.*) A listing of the contents of a storage device, or selected parts of it. *Syn.* – Storage Dump, Core Dump, Memory Printout.

EBCDIC: (*n.*) Acronym for Extended Binary-Coded Decimal Interchange Code. See *Extended Binary-Coded Decimal Interchange Code.*

EXTENDED BINARY-CODED DECIMAL INTERCHANGE CODE (EBCDIC): (*n.*) An eight-bit code for data notation which is divided into a zone group and a digit group of four bits each.

EXTENT: (*n.*) A group of contiguous tracks within a disc volume. An extent is defined by beginning and ending track addresses.

EXTRACTION: (*n.*) A randomizing technique whereby a set of random numbers is obtained by extracting several digits from each original key to produce a new set of keys.

FIELD: (*n.*) The smallest unit of data processed within the computer. It may be as small as one bit or as large as several characters.

FIELDATA: (*n.*) A six-bit coding system for data notation.

FILE: (*n.*) A collection of records; an organized collection of information.

FILE DUPLICATE: (*n.*) A copy of the master file of a system. Transactions identical to those processed against the master may also be processed against the duplicate. The duplicate need not be kept on the same device as the master.

FILE RECOVERY: (*n.*) The process of restoring a file which has been damaged in some way to the state it was in before the damage took place.

FILE RELATIVE ADDRESS: (*n.*) See *Address, Logical.*

FILE, SEQUENTIAL: (*n.*) A file that is stored in a certain order as opposed to a file which has been stored randomly.

FOLDING: (*n.*) A randomizing technique in which the original identifier is split in some way and the parts added together to produce a new identifier.

FRAME: (*n.*) A subdivision of tape, generally seven or nine bits wide, usually containing one character.

HARDWARE: (*n.*) The mechanical, magnetic, electric, and electronic devices from which a computer is constructed.

HEAD: (*n.*) An assembly of one or more coils of wire and associated polepieces, which can record information on a magnetic surface and/or read information from a magnetized surface.

HEAD, FIXED: (*n.*) A head which has a set location and reads only one track of information.

HEAD, FLYING: (*n.*) A read/write head which floats on a cushion of air close to the surface of the device.

HEAD, MOVABLE: (*n.*) A head which is moved over several tracks making it possible for less heads to be used.

IDENTIFIER: (*n.*) The field or fields of information by which a record in a file is identified, sequenced, and/or controlled. *Syn.* – Key.

INDEX (pl. INDICES, INDEXES): (*n.*) A file which is usually used to access a main record file. An index is accessed by record identifier. It contains the identifier and the storage address. It may also contain critical information.

INDEXED NONSEQUENTIAL FILE ORGANIZATION: (*n.*) A method of file organization in which the records are stored in no set pattern. The main file can be accessed only by an index and the records within the file are usually packed.

INDEXED SEQUENTIAL FILE ORGANIZATION: (*n.*) A method of file organization in which the records are stored in sequential order and are accessed through range indexes.

INDEX, RANGE: (*n.*) An index in which each entry references a block of records.

INDEX, SPECIFIC: (*n.*) An index which contains a specific reference to every record stored in the main file.

INPUT: (*n.*) Data transferred from auxiliary or external storage into the internal storage of a computer.

INQUIRY: (*n.*) A simple transaction used for retrieving stored data.

ITEM: (*n.*) See *Record.*

KEY: (*n.*) See *Identifier.*

KEY, DUPLICATE: (*n.*) 1. A key produced by a randomizing process which is the same as another key produced by the same process. *Syn.*– Synonym. 2. An identifier which appears on more than one record.

KEY, SYSTEM ENTRY: (*n.*) An identifier which is used to first enter a system.

LATENCY: (*n.*) See *Delay, Rotational.*

LOGICAL ADDRESS: (*n.*) See *Address, Logical.*

MAGNETIC CARD DEVICE: (*n.*) A direct access storage device which contains magnetic cards stored in a magazine.

MAIN MEMORY: (*n.*) See *Main Storage.*

MAIN STORAGE: (*n.*) The vital storage system of a computer in which data may be manipulated a field at a time. Usually the fastest storage device of the computer, and the one from which instructions are executed. Contrasted with Auxiliary Storage. *Syn.* – Core, Main Memory.

MAP, BIT: (*n.*) A map, containing bits in which each bit represents an area in storage. A magnetized bit indicates used storage; a nonmagnetized bit represents free storage.

OFFLINE: (*adj.*) Pertaining to the operation of I/O peripherals and other devices not under direct computer control; most commonly used to designate the transfer of information between magnetic tapes and other media.

ONLINE: (*adj.*) Pertaining to the operation of an I/O device as a component of the computer, under computer control.

OUTPUT: (*n.*) Data transferred from the internal storage of a computer to output devices or external storage.

OVERFLOW: (*n.*) 1. Records which exceed the regularly assigned storage section of a file. 2. The area in a file where records which exceed the regular storage area are placed.

OVER-INDEX: (*n.*) A range index which usually accesses another range index.

PARAMETER: (*n.*) A quantity to which arbitrary values may be assigned; used in subroutines and generators to specify item size, decimal point, block arrangement, field length, sign position, etc.

PARITY CHECK: (*n.*) A redundancy check technique based on an odd or even number of binary 1's in some grouping of binary digits. For instance, in the binary representation of a character, a parity bit is made either 0 or 1, whichever is required to make the number of 0's in the character an even number (even parity) or an odd number (odd parity).

PARITY DIGIT: (*n.*) See *Parity Bit.*

PHYSICAL ADDRESS: (*n.*) See *Address, Physical.*

POSITION CAPACITY: (*n.*) The amount of data which is available on a device without moving the read/write heads. *Syn.* – Cylinder Concept, Cylinder of Data.

PRIME NUMBER: (*n.*) A number that cannot be evenly divided by another number except itself or 1.

PROCESSING: (*n.*) The working with and manipulation of data to achieve a particular result, such as file updating and computations.

PROCESSING, BATCH: (*n.*) The procedure of processing transactions in a group against a program.

PROCESSING, INDIVIDUAL RANDOM: (*n.*) The working with and manipulation of transactions in a nonsequential mode, as they are read into the system.

PROCESSING, RANDOM BATCH: (*n.*) Processing whereby the individual records are collected until there is a certain number before processing begins. The records may or may not be stored before the run.

PROCESSING, REAL-TIME: (*n.*) The processing of current transactions on an individual basis with immediate response.

PROCESSING, SEQUENTIAL: (*n.*) Processing in an established order, one after another, as determined by identifiers.

PURIFICATION: (*n.*) The process of reducing data errors and inconsistencies as much as possible before file creation begins.

QUEUE: (*v.*) To place messages waiting to be processed in a certain order, usually with regard to priority.

QUEUE: (*n.*) A line-up in a certain order, of messages, transactions, activities, etc., waiting to be processed.

RANDOM ACCESS STORAGE: (*n.*) See *Direct Access Storage Device.*

RANDOMIZING: (*n.*) The act of performing any of several mathematical processes on a set of numbers to produce a new set of randomly distributed numbers.

REAL-TIME: (*adj.*) Used to describe a problem in which the response requirements are particularly stringent. The term is derived from the process control field and from military applications in which the data processing must "keep up" with a physical process. Used in business applications to mean "keep up" with the operations of the business.

RECORD: (*n.*) A series of fields related to a common subject. *Syn.* – Item.

RECORD, FILE DESCRIPTION: (*n.*) A separate record in a system which describes a particular file.

REDUNDANCY CHECK: (*n.*) A checking technique based on the presence of extra (redundant) information which is used only for checking purposes. Parity checking, check digits, control totals, and hash totals are all examples or redundancy checks.

SCAN: (*v.*) To examine every entry in a file and compare this against certain information in order to retrieve a desired item.

SEARCH: (*v.*) To locate a desired word or record in a set of words or records. The set searched may be located in any type of storage: internal, auxiliary, or even (in some cases) external.

SECTOR: (*n.*) A minimal addressable unit predetermined by hardware.

SECTOR, FIXED: (*n.*) A sector whose size is determined by the hardware.

SECTOR, VARIABLE: (*n.*) A sector whose size may vary according to the way the programs are written and according to the data.

SEEK: (*v.*) A hardware function to move read/write heads to the correct position.

SOFTWARE: (*n.*) A term applied to the general purpose class of routines for a computer including compilers, assemblers, executive routines, input and output libraries.

SORT: (*v.*) To sequence records according to a key contained in the records.

SPLITTING: (*n.*) The process of dividing a full block of records into two parts, each part forming a new block which is half full.

SQUARING: (*n.*) A randomizing technique whereby the original identifier is multiplied by itself to produce a new key.

STORAGE: (*n.*) Any device into which units of information can be transferred, which will hold information, and from which the information can be obtained at a later time.

STORAGE, AUXILIARY: (*n.*) A storage device which is capable of holding larger amounts of information than the main storage of the computer, although with slower access time.

STORAGE, CORE: (*n.*) A form of high-speed storage in which information is represented by the magnetization of ferromagnetic cores.

STORAGE, MAIN: (*n.*) See *Main Storage.*

STORAGE UNIT: (*n.*) 1. An online part of the hardware which provides extra internal storage for the central processing unit. 2. A module of direct access storage.

STORAGE, WORKING: (*n.*) A portion of the internal storage reserved for specific functions such as input and output areas.

STRING: (*n.*) A set of records usually in ascending (or descending) sequence according to a key contained in the records.

SYMBIONT: (*n.*) A special software program which handles I/O buffering of slow speed peripherals via DASD.

SYNONYM: (*n.*) See *Key, Duplicate.*

UPDATE: (*v.*) To modify a master file according to current information, often that contained in a transaction file, according to a procedure specified as part of a data processing activity.

UPDATE, TRANSACTION: (*n.*) A transaction that effects a change to one or more fields in an existing record in a file.

VALIDITY CHECK (*n.*) A checking technique based on known reasonable limits on data or computed results. For instance: a man cannot work 400 hours in one week; there is no day 32 in a month; a man on an hourly classification very seldom has a net week's pay greater than $2000.00, etc. Also called a reasonableness check.

VERIFICATION: (*n.*) 1. A check to see if a head is in correct position. 2. The process of comparing the data content of new files with the original documents in some manner before the operation begins.

VOLUME: (*n.*) A disc surface. The Volume may contain one file, part of a file, or more than one file.

VTOC: (*n.*) An acronym for Volume Table of Contents. A directory stored on a disc unit which contains the labels, limits, and identification of each file contained on the disc surface.

UNIVAC

SPERRY RAND

UP-7673