Reference Manual

IBM 1401 Data Processing System
Reference Manual

IBM 1401 Data Processing System
This edition, A24-1403-5, obsoletes A24-1403-4 and all earlier editions. Significant changes have been made throughout the manual, and this new edition should be reviewed in its entirety.

This edition also obsoletes these publications:

- G24-1438 IBM 1401 Data Processing System Bulletin
  Early Card Read Special Feature for IBM 1402
  Card Read-Punch
- G24-1458 IBM 1401 Data Processing System Bulletin
  Processing Overlap
- G24-1461 IBM 1401 Data Processing System Bulletin
  Processing Overlap: Processing Time During Magnetic Tape Operations

- N24-0001 No. 1 Oct. 23, 1961
- N24-0004 No. 2 Oct. 30, 1961
- N24-0005 No. 3 Oct. 31, 1961
- N24-0009 No. 4 Dec. 18, 1961
- N24-0010 No. 5 Dec. 18, 1961
- N24-0018 No. 6 Feb. 28, 1962
This manual is a reference text for the IBM 1401 Data Processing System. It provides a detailed explanation of operation codes and the function of the system's components. The reader should have a knowledge of the 1401 system and programming techniques. The reader should be familiar with the General Information Manual, IBM 1401 Data Processing System, form D24-1401, and the various publications on applied programming material, such as Symbolic Programming System (SPS) and Autocoder.

The manual is divided into these sections:

- operation codes
- operating features
- systems timing
- appendix

The sections are independent and need not be used in the order in which they appear.

This manual is intended for the use of programmers and systems personnel who have a general knowledge of the IBM 1401 Data Processing System and who require a reference text for detailed information. The manual can also be used as a training aid in the instruction of programmers and operators.

It should be noted that other publications referenced in the manual are, in most cases, prerequisites for a complete understanding of the material presented in this reference manual.
# Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Introduction</td>
<td>7</td>
</tr>
<tr>
<td>The Stored Program</td>
<td>7</td>
</tr>
<tr>
<td>Other Input-Output Units for the IBM 1401 System</td>
<td>7</td>
</tr>
<tr>
<td>The IBM 1401 Used with Other IBM Systems</td>
<td>8</td>
</tr>
<tr>
<td><strong>Physical Features</strong></td>
<td></td>
</tr>
<tr>
<td>Solid-State Circuitry</td>
<td>9</td>
</tr>
<tr>
<td>Advanced Systems Design</td>
<td>9</td>
</tr>
<tr>
<td>IBM 1401 Processing Unit</td>
<td>9</td>
</tr>
<tr>
<td>IBM 1402 Card Read-Punch</td>
<td>10</td>
</tr>
<tr>
<td>IBM 1403 Printer</td>
<td>11</td>
</tr>
<tr>
<td>Magnetic-Core Storage</td>
<td>11</td>
</tr>
<tr>
<td>Magnetic-Tape Storage</td>
<td>12</td>
</tr>
<tr>
<td>Magnetic-Disk Storage</td>
<td>13</td>
</tr>
<tr>
<td>Language</td>
<td></td>
</tr>
<tr>
<td><strong>Processing</strong></td>
<td>14</td>
</tr>
<tr>
<td>Stored Program Instructions</td>
<td>15</td>
</tr>
<tr>
<td>Addressing</td>
<td>18</td>
</tr>
<tr>
<td>Operation of IBM 1401 Registers</td>
<td>19</td>
</tr>
<tr>
<td><strong>IBM 1401 Programming Systems</strong></td>
<td>23</td>
</tr>
<tr>
<td>IBM 1401 Programming Systems</td>
<td>23</td>
</tr>
<tr>
<td>Symbolic Languages</td>
<td>23</td>
</tr>
<tr>
<td>IBM 1401 Symbolic Programming System (SPS)</td>
<td>23</td>
</tr>
<tr>
<td>IBM 1401 Autocoder</td>
<td>23</td>
</tr>
<tr>
<td>Input-Output Control System (IOCS)</td>
<td>27</td>
</tr>
<tr>
<td>IBM 1401 Report Program Generator (RPG)</td>
<td>27</td>
</tr>
<tr>
<td>Operation Codes</td>
<td>28</td>
</tr>
<tr>
<td>Arithmetic Operations</td>
<td>28</td>
</tr>
<tr>
<td>Logic Operations</td>
<td>32</td>
</tr>
<tr>
<td>Clear, Move, Load, and Word Mark Operations</td>
<td>35</td>
</tr>
<tr>
<td><strong>Editing</strong></td>
<td>41</td>
</tr>
<tr>
<td>Input-Output Operations</td>
<td>44</td>
</tr>
<tr>
<td>Card Read Instructions</td>
<td>44</td>
</tr>
<tr>
<td>Punch Instructions</td>
<td>45</td>
</tr>
<tr>
<td>Print Instructions</td>
<td>46</td>
</tr>
<tr>
<td>Combination Instructions</td>
<td>48</td>
</tr>
<tr>
<td>Document Control Instructions</td>
<td>51</td>
</tr>
<tr>
<td><strong>Magnetic Tape</strong></td>
<td>53</td>
</tr>
<tr>
<td>Data Flow</td>
<td>53</td>
</tr>
<tr>
<td>Magnetic Tape Characteristics</td>
<td>53</td>
</tr>
<tr>
<td>Tape Units</td>
<td>55</td>
</tr>
<tr>
<td>IBM 729 Magnetic Tape Unit</td>
<td>55</td>
</tr>
<tr>
<td>IBM 7330 Magnetic Tape Unit</td>
<td>55</td>
</tr>
<tr>
<td>Tape Internals</td>
<td>55</td>
</tr>
<tr>
<td>Tape Checking</td>
<td>50</td>
</tr>
<tr>
<td>Magnetic-Tape Operations</td>
<td>57</td>
</tr>
<tr>
<td>Tape Instructions</td>
<td>57</td>
</tr>
<tr>
<td><strong>IBM RAMAC® 1401 System</strong></td>
<td>63</td>
</tr>
<tr>
<td>IBM 1405 Disk Storage Instructions</td>
<td>65</td>
</tr>
<tr>
<td><strong>IBM 1407 Console Inquiry Station</strong></td>
<td>70</td>
</tr>
<tr>
<td>IBM 1407 Console Inquiry Station Instructions</td>
<td>71</td>
</tr>
<tr>
<td><strong>Special Features</strong></td>
<td>74</td>
</tr>
<tr>
<td>Multiply-Multiply Feature</td>
<td>74</td>
</tr>
<tr>
<td>Increased Core Storage</td>
<td>76</td>
</tr>
<tr>
<td>Read Release and Punch Release Feature</td>
<td>78</td>
</tr>
<tr>
<td>Punch Feed Read Feature</td>
<td>80</td>
</tr>
<tr>
<td>Print Storage Feature</td>
<td>82</td>
</tr>
<tr>
<td>Additional Print Control Feature</td>
<td>82</td>
</tr>
<tr>
<td>Expanded Print Edit Feature</td>
<td>82</td>
</tr>
<tr>
<td>Indexing Feature</td>
<td>84</td>
</tr>
<tr>
<td>Store Address Register Feature</td>
<td>85</td>
</tr>
<tr>
<td>Move Record Feature</td>
<td>88</td>
</tr>
<tr>
<td>High-Low-Equal Compare Feature</td>
<td>86</td>
</tr>
<tr>
<td>Sense Switches Feature</td>
<td>87</td>
</tr>
<tr>
<td>Compressed Tape Operations Feature</td>
<td>87</td>
</tr>
<tr>
<td>Column Bias Feature</td>
<td>89</td>
</tr>
<tr>
<td>Numerical Print Feature</td>
<td>94</td>
</tr>
<tr>
<td>Interchangeable Chain Cartridge Adapter</td>
<td>94</td>
</tr>
<tr>
<td>Interchangeable 51-Column Read Feed</td>
<td>95</td>
</tr>
<tr>
<td>Early Card Read Feature</td>
<td>97</td>
</tr>
<tr>
<td>Processing Overlap Feature</td>
<td>97</td>
</tr>
<tr>
<td><strong>Address Modification</strong></td>
<td>106</td>
</tr>
<tr>
<td>Address Modification Without Indexing Feature</td>
<td>106</td>
</tr>
<tr>
<td>Address Modification with Indexing Feature</td>
<td>108</td>
</tr>
<tr>
<td><strong>Operating Features</strong></td>
<td>109</td>
</tr>
<tr>
<td>IBM 1401 Console Keys, Lights, and Switches</td>
<td>109</td>
</tr>
<tr>
<td>Auxiliary Console</td>
<td>110</td>
</tr>
<tr>
<td>Auxiliary Console Switches</td>
<td>110</td>
</tr>
<tr>
<td>IBM 1402 Card Read-Punch Operating Keys, Lights, and Switches</td>
<td>118</td>
</tr>
<tr>
<td><strong>IBM 1403 Printer Operating Keys, Lights, and Switches</strong></td>
<td>119</td>
</tr>
<tr>
<td>Printer Controls</td>
<td>119</td>
</tr>
<tr>
<td>Carriage Controls</td>
<td>120</td>
</tr>
<tr>
<td>Tape-Controlled Carriage</td>
<td>123</td>
</tr>
<tr>
<td>IBM 1405 Disk Storage Unit Indicator Lights</td>
<td>128</td>
</tr>
<tr>
<td>IBM 1407 Console Inquiry Station Keys and Lights</td>
<td>129</td>
</tr>
<tr>
<td>Operating Pointers</td>
<td>130</td>
</tr>
<tr>
<td>IBM 729 and 7330 Magnetic Tape Units Operating Keys and Lights</td>
<td>134</td>
</tr>
<tr>
<td>Operating Pointers</td>
<td>135</td>
</tr>
<tr>
<td>Console Operation</td>
<td>138</td>
</tr>
<tr>
<td>Console Inquiry Station Operation</td>
<td>138</td>
</tr>
<tr>
<td>Restart Procedures</td>
<td></td>
</tr>
</tbody>
</table>
Figure 1. IBM RAMAC 1401 Data Processing System
As the recordkeeping requirements of business and industry continue to mount, the need for a data processing system that can be expanded to meet these requirements becomes apparent. A system is desired that can fill present needs, and that can expand as the burden of recordkeeping increases. Large-scale systems and small-scale systems are available, but a gap existed where no system seemed to do processing on an economical job-cost basis.

To fill this gap, IBM developed the IBM 1401 Data Processing System (Figure 1). The IBM 1401 is a solid-state, high-speed processing system with the program flexibility of larger systems.

The 1401 provides system configurations to meet the requirements for processing unit records, magnetic tape and magnetic disk records, and character-sensed documents.

The 1401 system provides high-speed input-output and arithmetic and logical ability, with the advantages of stored-program techniques. Various methods of programming are available. Some of the methods are: the easy-to-use actual language, the symbolic programming system (SPS) of mnemonic instructions, and IBM 1401 Autocoder.

The IBM 1401 Processing Unit handles variable-length alphabetical data and instructions. This means no space is wasted by filling in fixed-length words.

Variable word length, high internal processing speed, fast input and output, powerful editing ability, and direct accumulation in storage make the IBM 1401 Data Processing System an efficient and valuable tool to meet today's data processing requirements.

IBM 1401 Data Processing Systems can be considered in three basic concepts: card systems, tape systems, and RAMAC® systems.

IBM 1401 card systems are planned for procedures involving large volumes of card documents as source data and output, with particular advantage to applications requiring re-entry data.

IBM 1401 tape systems are for handling magnetic tape, with all the advantages of compact record handling and storage for high-speed data processing.

IBM RAMAC 1401 systems permit rapid access to large volumes of repetitive data without the necessity of processing large card volumes or sorting tape records.

The Stored Program

The IBM 1401 performs its functions by executing a series of instructions at high speed. A particular set of instructions, designed to solve a specific problem, is known as a program. Because the 1401 stores its instructions internally, it is called a stored program system.

The 1401 normally executes instructions sequentially. But sometimes it is necessary to skip over a particular group of instructions, or otherwise change the sequence of the program. Branch instructions are provided in the system to make it possible to alter the program and take the next instruction from another area of the stored program. This feature also makes it possible to repeat an instruction, or group of instructions, as often as desired.

A series of programmed tests determines the logical path of the program. These tests are made at various points in the program to control the course of program step execution for specific conditions that can arise during processing.

Other Input-Output Units for the IBM 1401 System

To meet the ever increasing needs of business for fast, economical processing of data, the IBM 1401 Data Processing System can accept data made available to it in a variety of forms. Besides the normal input in the form of punched cards and magnetic tape, other devices that greatly increase the efficiency and flexibility of the 1401 have been provided. These units make it possible to read punched paper tape, magnetic ink characters, and printed characters, and to transmit data between 1401's or a 1401 and a tape transmission terminal.

The IBM 1011 Paper Tape Reader enables the direct input of data punched in paper tape into the 1401 system. Present day accounting and data processing practices are such that in many cases the source data for a central data processing system originates at locations remote from the central system. The ease and economy in transmitting punched paper tape make it an ideal medium for this type of reporting. A concise description of the 1011 including operating features, control panel summary, and IBM 1401 operation codes is found in the General Information Manual, IBM 1011 Paper Tape Reader, form D24-1044.
The IBM 1419 Magnetic Character Reader serves as a means of input to the 1401 system. This merging of machines increases flexibility and control in banking operations. Thus, magnetically-inscribed characters are read by the magnetic character reader and sent to 1401 core storage for processing. A full description of the 1419, including operating features and IBM 1401 operation codes, is included in the General Information Manual, Form A24-1418.

Re-entry documents can be read directly into the core storage unit of the 1401 by using the IBM 1418 Optical Character Reader as an input device. This means the elimination, in many cases, of preparing data for systems use when it is returned with a remittance. In many applications, this allows all functions necessary for processing data, from source document through the final report, to be accomplished in one operation. The use and functions of the 1418 are described in the Reference Manual, Form A24-1418.

Many businesses with decentralized accounting operations have data processing installations at branch offices as well as at the headquarters location. The source data received from the branch offices enables the central office to prepare consolidated reports to meet their accounting needs. The IBM 1009 Data Transmission Unit allows high-speed, two-way communication between two IBM 1401 Data Processing Systems or between a 1401 and an IBM 7701 Magnetic Tape Transmission Terminal. The operation and functioning of the 1009 is described in the Reference Manual, IBM 1009 Data Transmission Unit, Form A24-1039.

The IBM 7701 Magnetic Tape Transmission Terminal provides communication between outlying tape installations and a central data processing system, without intermediate conversion steps. Straight-line data flow from a 1401 tape system to a large-scale system such as an IBM 700 or 7000 series data processing system is now a fact. This development results in a fast, economical collection, processing, and return of data not previously possible. The functions and operating features of the 7701 are described in the Reference Manual, IBM 7701 Magnetic Tape Transmission Terminal, Form A24-6527.

The IBM 1404 Printer is another output medium for the 1401 system. It is a combination printer capable of processing either separate card documents or continuous forms. Under control of the 1401 stored program, and the tape-controlled carriage, this unique printer can process continuous forms at a rated speed of 500 lines per minute; or it can print on card documents at a maximum rate of 800 cards per minute. The 1404 can process cards ranging in size from 51 columns to 160 columns. It can also process two cards (either 51- or 80-columns) at a time. As many as 25 lines of data, either from 1401 core storage or from the card itself, can be printed on each card. Instructions, as well as the functional and operational characteristics, are covered in the 1401 Data Processing System bulletin, IBM 1404 Printer, Form G24-1446.

Today's modern business requires fast, efficient and inexpensive intercommunication of data between branch, or field, locations and the central, or home, office. The IBM 1012 Tape Punch provides this means of communication and meets all these requirements.

The IBM 1012 Tape Punch, connected to an IBM 1401 Data Processing System, is an ideal medium for recording data processed at remote locations. Data punched in paper or Mylar* tape can be transmitted quickly and inexpensively to a central location by common carrier lines.

At the central location, the data recorded in the tape can be converted to punched cards or used as direct input to a central data processing system. A detailed description of the IBM 1012, including operating features and IBM 1401 operation codes, are found in the General Information Manual, IBM 1012 Tape Punch, Form D24-1977.

The IBM 1401 Used with Other IBM Systems

IBM 7070 Data Processing System

The IBM 1401 Data Processing System has additional flexibility when it is used with the tape-oriented configuration of the IBM 7070 Data Processing System. The 1401 can produce, edit, sort, print, punch, and further manipulate tape data used by the 7070, thus providing more time for the operations that are more efficient and practical for each system.

IBM Scientific Data Processing Systems

The column binary feature enables the 1401 to process card and tape data recorded in binary form. This ability makes the 1401 especially useful as an auxiliary system for the IBM 704, IBM 709, and IBM 7090 Data Processing Systems.

*Trademark of E. I. duPont de Nemours Company, Inc.
Solid-State Circuitry
Transistorization of IBM 1401 components is a significant design characteristic. Transistors are relatively inexpensive, are easy to maintain, and increase reliability in the system (Figure 2). Space requirements, heat dissipation, and power requirements are carefully controlled.

The components that require operator attention are conveniently located.

Advanced System Design
Advanced system design makes the IBM 1401 a complete, independent, accounting system. But it can also perform low-cost direct input and output, and auxiliary tape operations for large-scale data processing systems.

The entire system is controlled by the stored program. Timesaving features, such as the powerful editing function, and the elimination of control panels, provide increased flexibility for application development. The ability to use magnetic tape means economy in recording, transporting, and storing large volumes of information in compact form.

IBM 1401 Processing Unit
The IBM 1401 Processing Unit (Figure 3) contains the core storage and circuitry that perform the machine logic.

Storage capacity is 1400, 2000, 4000, 8000, 12,000 or 16,000 alphamerical characters of 8-bit core storage. These eight bits consist of six bits for binary-coded decimals, a check bit, and an eighth bit for field definition.
The IBM 1402 Card Read-Punch (Figure 4) provides the system with simultaneous punched-card input and output. This unit has two card feeds. The read section has a rated reading speed of 800 cards per minute. Actual card speed realized is governed by the stored program instructions. The read feed is equipped with a device for large-capacity loading, called a file feed. With the file feed device, the read feed can be loaded with as many as 3,000 cards.

The cards pass through the read side of the machine 9-edge first, face down. The feed path is from right to left, past two sets of brushes (Figure 5). The read check station reads 80 columns of the card to establish a hole count for checking purposes. The read station also reads 80 columns, proves the hole count, and directs the data into storage.

The punch section has a rated speed of 250 cards per minute. The card hopper capacity is 1,200 cards. Cards feed 12-edge first, face down. The feed path is left to right. Cards pass a blank station, a punch station, and a read station (Figure 5). The punch station consists of 80 punch magnets for recording information. The punch read station has 80 brushes that read the data punched in the card for a hole-count check.

The IBM 1402 Card Read-Punch is equipped with five radial type stackers (Figure 6), with a capacity of 1,000 cards each. Cards from each feed can be program-directed to three of the five pockets.

The cards from the read side go to the NR (normal read) pocket unless program-directed to pockets 1 or 2. The cards from the punch side go to the NP (normal punch) pocket unless program-directed to pockets 4 or 8.

The center pocket (8/2) can receive cards from either feed. However, card merging can be accomplished only under certain, very limited conditions.
The IBM 1403 Printer (Figure 7) is another output medium for the IBM 1401 Data Processing System. This unit has a rated printing speed of 600 lines per minute. The standard printing capacity is 100 positions, with an additional 32 positions available as a special feature.

Each position can print 48 different characters: 26 alphabetic; 10 numerical; and 12 special characters (&, , $, * / % # @ = ).

Vertical spacing and skipping are initiated by the stored program. Horizontal spacing is 10 characters to the inch. Vertical spacing of either six or eight lines to the inch can be manually selected by the operator. The paper transport mechanism for line spacing is a single-speed, tape-controlled carriage in the IBM 1401 card system, Model A. Other models (B, C, and D) use a dual-speed, tape-controlled carriage that permits skipping at the rate of 75 inches per second for skips of more than 8 lines, as compared to the single-speed carriage that has a constant speed of 33 inches per second for all skips.

METHOD OF PRINTING
The alphabetic, numerical, and special characters are assembled in a chain (Figure 8). As the chain travels in a horizontal plane, each character is printed as it is positioned opposite a magnet-driven hammer that presses the form against the chain.

As each hammer magnet is energized, it is checked against the corresponding position in the print area of core storage to insure that printed output is accurate. Also the machine checks to insure that the character is printed in the correct print position, that only valid characters are printed, and that over-printing does not occur.

Magnetic-Core Storage
The IBM 1401 Data Processing System uses magnetic-core storage for storing instructions and data (Figures 9 and 10). All data in core storage is instantly avail-
able, and the special design of the core-storage unit makes each position individually addressable. This means an instruction can designate the exact storage locations that contain the data needed for that step.

The physical make-up of each core-storage location makes it possible for the IBM 1401 to perform arithmetic operations directly in the storage area. (This is called add to storage logic.)

Magnetic-Tape Storage

Magnetic tape is made of plastic material, coated with a metallic oxide. It can be easily magnetized in tiny spots, so that patterns of these magnetized spots are codes for digits, alphabetic characters, and special characters.

Data can be read from a variety of sources, and written on the tape. Magnetic spots, representing information written on the tape, remain until changed by positive action.

This means that besides being used as data storage, this data itself can be part of input and output.

This makes magnetic tape an ideal storage medium for a large volume of data, because there is no limit to the amount of information that can be kept permanently. The reels of tape can be removed from the system and filed. They can also be transported from place to place and used in other systems.

Data stored on magnetic tape is read sequentially. The data processing system can search the tape to find the data to be used. Program steps can be stored on magnetic tapes, which are commonly used to set up a library or file of procedures.

Another advantage of magnetic-tape storage is that a reel of tape, produced as an output of a procedure, can be removed from the data processing system. Reports can be written using an independent unit, while the data processing system proceeds with the next program to be performed.

Magnetic-Disk Storage

Magnetic disks are thin metal disks, two feet in diameter, that are coated on both sides with a ferrous oxide recording material. These disks are mounted on a vertical shaft, and are separated from one another. As the shaft revolves, it spins the disks at 1200 rpm.

Information is recorded on disks in the form of magnetized spots located in concentric tracks on each disk surface.

At the side of the stack of disks (Figure 11), one or more access arms can move to any desired track on any disk under stored-program control. Magnetic recording heads mounted on the access arms write or read information as directed by the program. Each access arm is forked so that when it enters the stack of spinning disks, a recording head is carried to both sides of a disk. Thus, it is possible to read or write on either surface of a disk.

The magnetic disk can be used repetitively. Each time new information is stored in a track, it erases the data formerly stored there. Records can be read from disks as often as desired until they are written over or erased.

In addition to providing increased storage capacity, magnetic-disk storage units permit the processing of data on a random basis. Because any record on any track is addressable, the IBM 1401 Processing Unit has access to any record in the disk-storage unit. This random accessibility is the key to the in-line approach to data processing. It eliminates the necessity of accumulating transactions of a like kind (batching) before entering them. Transactions can be entered as they occur — regardless of sequence. In less than a second, the 1401 program can seek a record, update it, and
return the updated record to the storage unit. Also, the 1401 can process other data within core storage while the access mechanism searches for a record.

**Language**

In the punched-card area of data processing, the language of the machine consists of holes punched in a card. As data processing needs increase, the basic card language remains the same. But in the transition from unit-record systems to the IBM 1401 Data Processing System, and from there to computer systems, another faster, more flexible machine language emerges.

Just as each digit, letter in the alphabet, or special character is coded into a card as a punched hole or a combination of punched holes, it is coded into magnetic storage as a pattern of magnetized spots.

Many different code patterns can be set up. The internal code used in the IBM 1401 Data Processing System is called *binary-coded decimal*. All data and instructions are translated into this code as they are stored. No matter how information is introduced into the system (most commonly by means of punched cards or magnetic tape), the binary-coded-decimal code is used in all data flow and processing from that point on, until it is translated into printed output as reports and documents are written, or converted to punched-card code, for punched-card output. Converting input data to the 1401 internal code, and subsequently reconverting, is completely automatic.

Figure 11. Access Arm
The manipulation that data undergoes in order to achieve desired results is called processing, and the part of the 1401 system that houses these operations is called the processing unit.

Logic

The logic function of any kind of data processing system is the ability to execute program steps; but even more, the ability to evaluate conditions and select alternative program steps on the basis of those conditions.

In unit-record equipment, an example of this logic is selector-controlled operations based on an X-punch or No X-punch, or based on a positive or negative value, or perhaps based on a comparison of control numbers in a given card field.

Similarly, the logic functions of the 1401 system control comparisons, branching (alternative decisions similar in concept to selector-controlled procedures), move and load operations (transfer of data or instructions), and the general ability to perform a complicated set of program steps with necessary variations.

Arithmetic

The IBM 1401 Processing Unit can add, subtract, multiply, and divide. Multiplication and division can be accomplished in any 1401 system, by programmed subroutines. When the extent of the calculations might otherwise limit the operation, a special multiply-divide feature is available.

Editing

As the term implies, editing adds significance to output data by punctuating and inserting special characters and symbols. The IBM 1401 has a unique ability to perform this function, automatically, with simple program instructions.

Internal Checking

Advanced circuit design is built into the 1401 to assure accurate results. Self-checking within the machine consists of parity, validity, and hole count.

PARITY CHECKING

The IBM 1401 checks characters at various locations in the system for odd-bit configurations. The six-bit, binary-coded-decimal internal language used by the 1401 also has a check bit for odd-bit checking purposes, and a word mark. The check bit is added to all characters that would otherwise have an even number of bits.

Example: A character P has a binary-coded decimal equivalent of B 4 2 1. The check bit is added to give this character an odd number of bits (C B 4 2 1).

If the character has a word mark associated with it, the word mark is included in the test for odd-bit parity.

Example: If the character P has a word mark, the the check bit is not added because the bit configuration is odd (WM B 4 2 1).

Whenever a parity error occurs, a console light turns on, indicating the place where the error occurred (see Console).

VALIDITY CHECKING

A bit configuration that does not comprise a valid 1401 character causes a validity error in the 1401. For example, an invalid character passes a parity check because it contains an odd number of bits but does not pass a validity check.

A validity check is performed on each character as it is read into the 1401 by the card reader. An invalid character can get into core storage, but the validity-check circuits detect it and cause the 1401 to stop. The validity light on the card reader turns on to indicate the error.

Four types of address validity checking are performed by the 1401 processing system. The operations, and when they are performed, follow:
1. Checking for a core-storage address greater than the installed core-storage capacity. The units position of an address on from 4 to 12 thousand positions of core storage are checked for the proper A-, B-bit configuration. This check is performed when the output of the B-register goes to the storage-address register.
2. Effective address checking is divided into three tests that occur whenever core storage is addressed. The three tests are:
   a. Incorrect parity
   b. Illegal address character
   c. A check of the hundreds and units positions of
an address for various core-storage sizes. A 1400-position system is checked for addresses between 1400 and 4000. A 2000-position system is checked for addresses between 2000 and 4000. A 12000-position system is checked for addresses between 12000 and 16000. If any of these conditions are found, a validity check occurs and the system stops.

3. **Index checking** is performed during an indexing operation to check for modification to an address in excess of installed core-storage capacity.

4. **End-around check** is made at all times except for three special operations. The modification of the low-order position of core storage by -1, except during a CLEAR operation, or the modification of the high-order position of core storage by +1, except during STORAGE SCAN and STORAGE PRINT OUT operations, causes invalid operation and a system stop.

### HOLE-COUNT CHECK

Reliability is further assured in the 1401 system by the hole-count feature of the IBM 1402 Card Reader-Punch. With this feature, the total number of holes read in each column of a card at the read-check station is compared with the total number of holes read from the same column of the same card as it passes the read station. Hole-count checking is also performed in the punch-feed side. A count of the total number of holes to be punched in each column of the card at the punch station is retained internally for one punch-feed cycle. Another column-by-column hole count is taken as this same card passes the punch-check station, and the two counts are compared.

If a hole-count error (unequal comparison) occurs in either the read or punch side, the system stops and indicates the unit involved. The operator can determine where the error occurred by setting the mode switch to STORAGE SCAN and pressing the start key. The scan stops at the storage address of the column in error.

### Variable Word Length

Stored programming involves the concept of words. A 1401 word can be a single character, or a group of characters that represent a complete unit of information. Because IBM 1401 words are not limited to a specific number of storage positions, and because each position of core storage is addressable, each word occupies only that number of core-storage locations actually needed for an instruction or data field.

### Word Marks

The use of the variable-length instruction and data format requires a method of determining the instruction and data-word length. This identification is provided by a word mark. Word marks are illustrated by underlining the characters with which they are associated.

The word mark serves several functions:

1. Indicates the beginning of an instruction.
2. Defines the size of a data word.
3. Signals the end of execution of an instruction.

The rules governing the use of word marks are:

1. Predetermined locations for word marks are assigned in planning the program. These predetermined word marks are normally expected to remain in these locations throughout the complete program. The word marks are set into storage locations by a loading routine.
2. Word marks are not moved with data during processing, except when a load instruction (see Move and Load) is used.
3. For an arithmetic operation, the B-field must have a defining word mark, and the A-field must have a word mark only when it is shorter than the B-field.
4. A load instruction moves the word mark and data from the A-field to the B-field, and clears any other word marks in the designated B-field, up to the length of the A-field.
5. When moving data from one location to another, only one of the fields need have a defining word mark, because the move instruction implies that both fields are the same length.
6. A word mark must be associated with the high-order character (operation code) of every instruction.
7. A 4-character unconditional branch instruction (BXXX) is the only instruction that can be followed by a blank without a word mark; all other instructions must be followed by a word mark.
8. A word mark must be set in the storage position at the immediate right of the last character of the last instruction in the program.

Two operation codes are provided for setting and clearing word marks during program execution.

### Stored Program Instructions

All machine functions are initiated by instructions from the 1401 stored program. Because the 1401 uses...
7. The 4-character unconditional branch instruction, the 7-character set word mark, and clear storage and branch instructions are the only instructions that can be followed by a blank without a word mark. All other instructions must be followed by a word mark.
the variable-word-length concept, the length of an instruction can vary from one to eight characters, depending on the operation to be performed.

**Instruction Format**

\[
\text{Op Code} \quad A\text{- or } I\text{-address} \quad B\text{-address} \quad d\text{-character}
\]

*Op Code.* This is always a single character that defines the basic operation to be performed. A word mark is always associated with the operation code position of an instruction.

*A-Address.* This always consists of three characters. It can identify the units position of the A-field, or it can be used to select a special unit or feature (tape unit, disk storage unit, IBM 1419 Magnetic Character Reader, etc.).

*I-Address.* Instructions that can cause program branches use the I-address to specify the location of the next instruction to be executed if a branch occurs.

*B-Address.* This is a three-character storage address that identifies the B-field. It usually addresses the units position of the B-field, but in some operations (such as tape read and write) it specifies the high-order position of a record-storage area.

*d-Character.* The d-character is used to modify an operation code. It is a single alphabetic, numerical, or special character, positioned as the last character of an instruction.

Examples of the six combinations possible in variable-length instructions are shown in Figure 12.

**Instruction Descriptions**

Specific instructions have been described in a standard format:

*Title:* This is the description of the instruction.

*Instruction Format:* This is the format of the particular instruction described. The mnemonic operation code used in the IBM 1401 Symbolic Programming System (SPS) is shown and the mnemonic for IBM 1401 Autocoder (A) is given, if it is different from the SPS mnemonic.

*Function:* This is the function of the instruction.

*Word Marks:* This is the effect of word marks with regard to data fields.

*Timing:* This is the formula to be used in calculating the timing of the instruction. Key to abbreviations used in formulas is shown in Figure 240.

*Notes:* These are special notations or additional information that pertain to the operation.

*Address Registers After Operation:* The contents of the address registers are represented by the codes described in the Chaining Instructions section.

*Example:* A practical application of the instruction is described and shown as labels for Autocoder and SPS (Symbolic Programming System) languages. With the label is the actual machine address in parentheses. It is not necessary for the programmer to know the actual address of a label when writing the program. The processor program assigns the actual address during the program assembly.

*Assembled Instruction:* This is the actual machine language instruction that is assembled by the Symbolic Programming System or Autocoder processor programs from the symbolic entries shown in the example.
<table>
<thead>
<tr>
<th>NUMBER OF POSITIONS</th>
<th>OPERATION</th>
<th>INSTRUCTION FORMAT</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>READ A CARD</td>
<td>OP code 1</td>
</tr>
<tr>
<td>2</td>
<td>SELECT STACKER</td>
<td>Op code 2</td>
</tr>
<tr>
<td>4</td>
<td>BRANCH</td>
<td>Op code 4</td>
</tr>
<tr>
<td>5</td>
<td>BRANCH IF INDICATOR ON</td>
<td>Op code 5</td>
</tr>
<tr>
<td>7</td>
<td>ADD</td>
<td>Op code 7</td>
</tr>
<tr>
<td>8</td>
<td>BRANCH IF CHARACTER EQUAL</td>
<td>Op code 8</td>
</tr>
</tbody>
</table>

Figure 12. IBM 1401 Instruction Formats
Addressing

Instructions and data to be used for processing in the 1401 are kept in core storage. Each position in storage is addressable (Figure 13). A field is defined by an eighth bit (called a word mark) and can contain either an instruction or data.

A data field is defined by a word mark in the high-order position. The units or low-order position of a data field is specified in the A- or B-address of the instruction. The data field is read from right to left until a word mark in the high-order position is sensed.

An instruction is addressed by giving the high-order (operation code) position of the instruction. All operation codes must have a word mark. (This word mark is normally set by the loading routine when the instructions are loaded.) The machine reads an instruction from left to right until it senses the word mark associated with the next sequential instruction. The final instruction in the program must have a word mark set at the right of its low-order position.

Example: Instruction address 400 (Figure 14) contains the operation code for the following instruction:

<table>
<thead>
<tr>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>542</td>
<td>560</td>
</tr>
</tbody>
</table>

When this instruction is executed, the data in the A-field is added to the data in the B-field:

```
0025347
04601231
04626378
```

The result is stored in the B-field.

Input-Output Storage Assignments

Three areas of storage are reserved for input and output data. Storage positions 001 through 080, are reserved for the information from the 80 columns of the card. The second area of storage, positions 101 through 180, is reserved for assembling data to be punched. Positions 000 and 100 should not be used. Data stored in position 000 before a card read operation is replaced by CAB bits at the end of the read operation. Data stored in position 100 before a punch

---

**Figure 13. Core Storage Address Codes**

---

**Figure 14. Data and Instruction Addressing**
operation is replaced by C82 bits at the end of the punch operation. The third area of storage, positions 201 through 300 or 332, is reserved for assembling characters to be printed. Positions 81 through 99, and 181 through 200, are available for normal storage use. When the reserved areas are not being used as specified, they can be used for other storage operations (Figure 15).

**Operation of IBM 1401 Registers**

The IBM 1401 Data Processing System can operate on and process data to produce a desired result by executing a series of instructions at high speed. A series of instructions designed to solve a problem is known as a program. Because these instructions are retained in core storage, it is more properly called a stored program.

The processing unit must interpret an instruction and perform the function prescribed by the instruction. In order to do this, various types of devices that are capable of receiving information, storing it, and transferring it as directed by control circuits are used. These devices are known as registers. The 1401 has seven registers, four are address registers and three are character registers (Figure 16).

**Address Registers**

There are three address registers in the IBM 1401 Processing Unit. One controls program sequence, and the other two control the transfer of data from one storage location to another.

**I-Address Register**. The I- (Instruction) address register always contains the storage location of the next instruction character to be used by the stored program. The number in this register is increased by one as the instruction is read from left to right.

**A-Address Register**. The A-address register normally contains the storage address of the data in the A-address portion of an instruction. Normally, as the instruction is executed, the number in this register is decreased by 1 after each storage cycle that involves the A-address.
**Note:** If the A-address portion of the instruction does not contain a 1401 storage address (for example, %Ux), the contents of the A-address register are not disturbed as the instruction is executed.

**B-Address Register.** This register normally contains the storage location of the data in the B-address portion of an instruction. Normally, as a storage cycle involving the B-address is executed, the storage address in the B-address register is decreased by 1.

**Character Registers**

The A- and B-character registers and the Op-register are single-character registers used to store data during the execution of an instruction.

**Op-Register.** The Op- (Operation) register stores the operation code of the instruction in process for the duration of the operation. The operation code is stored in BCD code including the check bit, but excluding the word mark.

**B-Register.** Each character leaving 1401 core storage enters the B-register. The character is stored in 8-bit form (BCD code, check bit, and word mark). The B-register is reset and filled with a character from core storage on every storage cycle.

**A-Register.** The A-register is reset and filled with the character from the B-register during each storage cycle that involves the A-address, and during all instruction cycles except the first and last I- (Instruction) cycle of each instruction. Data is stored in 8-bit form.

**Note:** Information can be written back into core storage directly from either the A- or B-register.

Figure 17 shows the I-phase of an operation and gives a detailed schematic for loading a 7-character instruction in the operation code register, in the A- and B-registers and in the I-, A-, and B-address registers. Eight storage cycles are required to load the complete instruction in the register. Each storage cycle requires .0115 millisecond.

**Note:** The A- and B-address registers contain 3-character addresses. Actual addresses are shown in this schematic because the storage display lights on the console show 4-digit addresses.

**Chaining Instructions**

In some programs, it becomes possible to perform a series of operations on several fields that are in consecutive storage locations. Some of the basic operations, such as ADD, SUBTRACT, MOVE, and LOAD, have the ability to be chained so that less time is required to perform the operations, and space is saved in storing instructions. Here is an example of the chaining technique: assume that four 5-position fields stored in sequence are to be added to four other sequential fields. This operation could be done using four 7-character instructions:

<table>
<thead>
<tr>
<th>A</th>
<th>700</th>
<th>850</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>695</td>
<td>845</td>
</tr>
<tr>
<td>A</td>
<td>690</td>
<td>840</td>
</tr>
<tr>
<td>A</td>
<td>685</td>
<td>835</td>
</tr>
</tbody>
</table>

At the completion of the first instruction, the A-address register contains 695 and the B-address register contains 845. These are the same numbers that are in the A- and B-addresses in the second instruction. Eighty storage cycles would be required to execute these instructions, thus using up .920 millisecond. Also, 28 storage positions are required to store these instructions.

By taking advantage of the fact that the A- and B-address registers contain the necessary information to perform the next instruction, this same sequence of operations can be executed as follows:

<table>
<thead>
<tr>
<th>A</th>
<th>700</th>
<th>850</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

Connecting instructions together in this manner is called chaining. The first add instruction contains both
<table>
<thead>
<tr>
<th>CYCLE</th>
<th>OPERATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-Op</td>
<td>The operation code enters the B-register and the Op-register. Because this is the first I-cycle, the A-register is undisturbed.</td>
</tr>
<tr>
<td></td>
<td>OP Register</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1-1</td>
<td>The A-address register is reset to blanks during the first part of the cycle for all instructions. The B-address register is reset to blanks during the first part of the cycle for all operations except Move, Load, Store A- and Store B-address Register operation. During the 1-1 cycle, the second instruction character (first character of the A-address) enters the thousands and hundreds positions of the A- and B-address registers and the A-register by the way of the B-register.</td>
</tr>
<tr>
<td></td>
<td>OP Register</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1-2</td>
<td>The third character of the instruction enters the tens position of the A- and B-address registers, and the A-register through the B-register.</td>
</tr>
<tr>
<td></td>
<td>OP Register</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1-3</td>
<td>The fourth instruction character enters the units position of the A- and B-address registers, and the A-register through the B-register.</td>
</tr>
<tr>
<td></td>
<td>OP Register</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1-4</td>
<td>The B-address register is reset at the beginning of this cycle. The fifth instruction character (first character of the B-address) enters the hundreds position of the B-address register, and the A-register through the B-register.</td>
</tr>
<tr>
<td></td>
<td>OP Register</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1-5</td>
<td>The sixth instruction character goes to the tens position of the B-address register, and the A-register through the B-register.</td>
</tr>
<tr>
<td></td>
<td>OP Register</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1-6</td>
<td>The seventh character of the instruction (last character of the B-address) enters the units position of the B-address register and the A-register through the B-register.</td>
</tr>
<tr>
<td></td>
<td>OP Register</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
<tr>
<td>1-7</td>
<td>The first character of the next instruction enters the B-register only. Because this is the last I-cycle for this instruction, the A-register and the Op-register, the A- and B-address registers are undisturbed. The detection of a word mark associated with this character signals the machine that this is the Op code for the next instruction. The loading operations stop, and the instruction that was just loaded is executed. Note that the I-address register contains the address of the high-order position of the next sequential instruction.</td>
</tr>
<tr>
<td></td>
<td>OP Register</td>
</tr>
<tr>
<td></td>
<td>A</td>
</tr>
</tbody>
</table>

Figure 17. Schematic of Instruction Loading
the A- and B-addresses. The following three instructions contain only the operation code for those instructions. The A- and B-addresses are the results left in the A- and B-address registers from the previous instruction. This type of operation requires 62 storage cycles, and takes .713 ms to execute. Storage of these chained instructions requires only ten storage positions.

The ability to chain a series of instructions does not depend on the use of the same operation code. Chained instructions may have various Op codes. To be operated on, the A-fields must be in sequence, and the B-fields must be in sequence. Example:

\[
\begin{array}{c}
\text{A} & 900 & 850 \\
\text{M} & \\
\text{A} & \\
\text{M} & \\
\end{array}
\]

For example, assume that the data fields are each ten characters long:

The ten characters at location 900 were added to 850.
The ten characters at location 890 were moved to 840.
The ten characters at location 880 were added to 830.
The ten characters at location 870 were moved to 820.

The description of each instruction includes the contents of the address registers after the operation has been performed. Figure 18 shows the abbreviations that indicate the contents of these registers.

By using this information, the programmer can determine the status of the registers and decide whether chaining is practical in specific cases.

Note: Instructions that contain other than IBM 1401 storage addresses cannot be chained. For example, M %u x x R is a tape read instruction. The tape unit is signaled as the machine reads the instruction. Although the A-address register contains %4x after the operation, chaining is impossible because the machine does not select the unit from the contents of the A-address register.

Most single-address instructions (Op code and an A-address) cause the A-address to be inserted in both the A- and B-address registers (for example A xxx). However, MOVE, LOAD, and STORE A- or STORE B-ADDRESS REGISTER (Op codes M, L, Q, and H) do not disturb the B-address register, and therefore permit the programmer to use the previous contents of that register as part of the instruction.

All no-address instructions (Op code only) use the previous contents of the A- and B-address registers.

The contents of the B-address register after a branch instruction (Op code and I-address) depend on whether or not the indexing feature is installed in the 1401:

1. With the indexing feature installed, the B-address register contains the address of the next sequential instruction if a branch occurs.
2. Without the indexing feature installed, the B-address register is cleared to blanks whenever a branch occurs.

<table>
<thead>
<tr>
<th>ABBREVIATION</th>
<th>MEANING</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A-address of the instruction</td>
</tr>
<tr>
<td>B</td>
<td>B-address of the instruction</td>
</tr>
<tr>
<td>NSI</td>
<td>Address of the next sequential instruction</td>
</tr>
<tr>
<td>BI</td>
<td>Address of the next instruction if a branch occurs</td>
</tr>
<tr>
<td>La</td>
<td>The number of characters in the A-field</td>
</tr>
<tr>
<td>Lo</td>
<td>The number of characters in the B-field</td>
</tr>
<tr>
<td>Lr</td>
<td>The number of characters in a disk record</td>
</tr>
<tr>
<td>Ap</td>
<td>The previous setting of the A-address register</td>
</tr>
<tr>
<td>Bp</td>
<td>The previous setting of the B-address register</td>
</tr>
<tr>
<td>dpp</td>
<td>The d-character and the tens and units positions of the previous register setting</td>
</tr>
<tr>
<td>do</td>
<td>The d-character and the tens and units positions of the branch instruction</td>
</tr>
<tr>
<td>dbb</td>
<td>The d-character and blank in the units and tens position</td>
</tr>
</tbody>
</table>

Figure 18. Contents of Address-Register Codes
IBM 1401 Programming Systems

The 1401 Symbolic Programming System and 1401 Autocoder are the basic symbolic programming aids for IBM 1401 Data Processing Systems. Each consists of a set of language specifications and a processor program. The language is used to write the source program, and the processor program translates the symbolic language program (the source program) into the actual machine language program (object program).

Symbolic Languages

Both the Autocoder and SPS languages permit the programmer to define areas, write instructions, and exercise some control over the execution of the processor program by writing symbolic statements. These statements are written using mnemonic operation codes (Figures 19 and 20) and the symbolic names with which the programmer names data, instructions, and work areas. For example, a symbolic instruction to add the data in a field called WHTAX to the data in a field called TOTDED would be written A WHTAX TOTDED in symbolic language.

Area Definition (Declarative Operations)

The area-definition entries are used to assign sections of storage for fixed data (constants) that will be needed during processing, to set aside work areas, and to assign symbolic names to data and devices used in the program. Area-definition statements are examined by the processor program during assembly of the object program. Some produce constants (cards that are loaded with the object program), but none produce instructions to be executed in the object program.

Instructions (Imperative Operations)

The instruction entries state symbolically the procedure to be taken during execution of the object program. They are actual commands to the object machine such as ADD, SUBTRACT, READ, PUNCH, etc.

Processor Control Operations

These are special instructions given by the programmer to the processor program. They exercise such control as: where to begin assigning storage for the object program, where the program ends, how much storage is available in the object-machine, etc. Processor control statements are never executed as instructions in the object program. They are used only during object-program assembly.

IBM 1401 Symbolic Programming System (SPS)

The SPS system is essentially a one-for-one coding system in which one symbolic statement is written for each instruction that appears in the object program. Two versions of the SPS are available. SPS-1 operates on a 1401 with 1400 positions of core storage, an IBM 1402 Card Read-Punch, and an IBM 1403 Printer, but it can assemble programs for any object-machine with as many as 4,000 positions of core-storage. SPS-2 can assemble programs for any 1401 system (1400 to 16,000 positions of core storage), but must assemble the program in a 1401 system with at least 4,000 positions of core-storage, the IBM 1402 Card Read-Punch, and an IBM 1403 Printer.

An SPS source program must be coded on the 1401 Symbolic Programming System Coding Sheet (form X24-1152). This coding sheet is designed for fixed-form coding. (A special area is reserved for each item to be contained in an SPS statement.)


A sample SPS program is shown in Figure 21.

IBM 1401 Autocoder

The 1401 Autocoder is a more powerful symbolic programming system for the IBM 1401 Data Processing System. This system provides a macro facility that permits the user to call out standard sets of instructions (routines) from a library stored on magnetic tape. It also permits him to code literals (actual data to be operated on during processing) directly in the instructions that use them, thus simplifying the area-definition part of the source program.

The 1401 Autocoder system can be used to assemble programs for all IBM 1401 Data Processing Systems. However, the machine used to assemble an Autocoder program must have at least:

- 4,000 positions of core storage
- Four IBM 729 II, 729 IV, 729 V, or 7330 Magnetic Tape Units
- IBM 1403 Printer, Model 2
- IBM 1402 Card Read-Punch
- Advanced-Programming Features
- High-Low-Equal Compare Feature
- Sense Switches

An Autocoder source program, like SPS, is written on a special coding sheet (form X24-1350). Autocoder statements, however, are written in free-form. (The operand fields are not divided into special areas as
### AREA DEFINITION

<table>
<thead>
<tr>
<th>Mnemonic Operation Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DCW</td>
<td>Define Constant With Word Mark</td>
</tr>
<tr>
<td>DC</td>
<td>Define Constant (No Word Mark)</td>
</tr>
<tr>
<td>DS</td>
<td>Define Symbol</td>
</tr>
<tr>
<td>DSA</td>
<td>Define Symbol Address</td>
</tr>
</tbody>
</table>

### INSTRUCTIONS

#### Arithmetic

<table>
<thead>
<tr>
<th>Type</th>
<th>Mnemonic Operation Code</th>
<th>Description</th>
<th>Machine Language Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>A</td>
<td>Add</td>
<td>A</td>
</tr>
<tr>
<td></td>
<td>S</td>
<td>Subtract</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>*M</td>
<td>Multiply</td>
<td>@</td>
</tr>
<tr>
<td></td>
<td>*D</td>
<td>Divide</td>
<td>%</td>
</tr>
<tr>
<td></td>
<td>ZA</td>
<td>Zero and Add</td>
<td>? (Prints as &amp;)</td>
</tr>
<tr>
<td></td>
<td>ZS</td>
<td>Zero and Subtract</td>
<td>I (Prints as -)</td>
</tr>
</tbody>
</table>

#### Data Control

<table>
<thead>
<tr>
<th>Type</th>
<th>Mnemonic Operation Code</th>
<th>Description</th>
<th>Machine Language Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MCW</td>
<td>Move Characters to A or B Word Mark</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>*MCM</td>
<td>Move Characters to Record of Group Mark</td>
<td>P</td>
</tr>
<tr>
<td></td>
<td>MCS</td>
<td>Move Characters and Suppress Zeros</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td>MN</td>
<td>Move Numeric</td>
<td>D</td>
</tr>
<tr>
<td></td>
<td>MZ</td>
<td>Move Zone</td>
<td>Y</td>
</tr>
<tr>
<td></td>
<td>MCE</td>
<td>Move Characters and Edit</td>
<td>E</td>
</tr>
<tr>
<td></td>
<td>LCA</td>
<td>Load Characters to A Word Mark</td>
<td>L</td>
</tr>
<tr>
<td></td>
<td>SW</td>
<td>Set Word Mark</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>CW</td>
<td>Clear Word Mark</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>CS</td>
<td>Clear Storage</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*MIZ</td>
<td>Move and Insert Zeros (for reading 7070 Compressed Tape)</td>
<td>X</td>
</tr>
<tr>
<td></td>
<td>*MA</td>
<td>Modify Address</td>
<td>S</td>
</tr>
<tr>
<td></td>
<td>*SAR</td>
<td>Store A Address Register</td>
<td>Q</td>
</tr>
<tr>
<td></td>
<td>*SBR</td>
<td>Store B Address Register</td>
<td>H</td>
</tr>
</tbody>
</table>

#### Logic Control

<table>
<thead>
<tr>
<th>Type</th>
<th>Mnemonic Operation Code</th>
<th>Description</th>
<th>Machine Language Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>B</td>
<td>Branch</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>BWZ</td>
<td>Branch if Word Mark and/or Zone</td>
<td>V</td>
</tr>
<tr>
<td></td>
<td>C</td>
<td>Compare</td>
<td>C</td>
</tr>
<tr>
<td></td>
<td>NOP</td>
<td>No Operation</td>
<td>N</td>
</tr>
<tr>
<td></td>
<td>H</td>
<td>Halt</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*BEE</td>
<td>Branch if Bit Equal</td>
<td>W</td>
</tr>
</tbody>
</table>

#### System Control

<table>
<thead>
<tr>
<th>Type</th>
<th>Mnemonic Operation Code</th>
<th>Description</th>
<th>Machine Language Equivalent</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>R</td>
<td>Read a Card</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>W</td>
<td>Write a Line</td>
<td>2</td>
</tr>
<tr>
<td></td>
<td>WR</td>
<td>Write and Read</td>
<td>3</td>
</tr>
<tr>
<td></td>
<td>P</td>
<td>Punch a Line</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>RP</td>
<td>Read and Punch</td>
<td>5</td>
</tr>
<tr>
<td></td>
<td>WP</td>
<td>Write and Punch</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>WRP</td>
<td>Write, Read and Punch</td>
<td>7</td>
</tr>
<tr>
<td></td>
<td>*SRF</td>
<td>Start Read Feed</td>
<td>B</td>
</tr>
<tr>
<td></td>
<td>*SPF</td>
<td>Start Punch Feed</td>
<td>9</td>
</tr>
<tr>
<td></td>
<td>SS</td>
<td>Select Stacker</td>
<td>K</td>
</tr>
<tr>
<td></td>
<td>CC</td>
<td>Control Carriage</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td>CU</td>
<td>Control Unit</td>
<td>U</td>
</tr>
<tr>
<td></td>
<td>MU</td>
<td>Move Unit</td>
<td>M</td>
</tr>
<tr>
<td></td>
<td>LU</td>
<td>Load Unit</td>
<td>L</td>
</tr>
</tbody>
</table>

### PROCESSOR CONTROL OPERATIONS

<table>
<thead>
<tr>
<th>Mnemonic Operation Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTL</td>
<td>Control</td>
</tr>
<tr>
<td>ORG</td>
<td>Origin</td>
</tr>
<tr>
<td>END</td>
<td>End</td>
</tr>
<tr>
<td>EX</td>
<td>Execute</td>
</tr>
</tbody>
</table>

*Pertains to an optional feature.

---

Figure 19. IBM 1401 Symbolic Programming System Mnemonics

24
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>DA</td>
<td>Define Area</td>
<td>A</td>
</tr>
<tr>
<td>DC</td>
<td>Define Constant (No Word Mark)</td>
<td>B</td>
</tr>
<tr>
<td>DCM</td>
<td>Define Constant With Word Mark</td>
<td>C</td>
</tr>
<tr>
<td>DS</td>
<td>Define Symbol</td>
<td>D</td>
</tr>
<tr>
<td>DSA</td>
<td>Define Symbol Address</td>
<td>E</td>
</tr>
<tr>
<td>EQU</td>
<td>Equate</td>
<td>F</td>
</tr>
<tr>
<td>B</td>
<td>Branch Unconditional</td>
<td>G</td>
</tr>
<tr>
<td>BVE</td>
<td>Branch if Equal</td>
<td>H</td>
</tr>
<tr>
<td>B8B</td>
<td>Branch on Carry Channel 9</td>
<td>I</td>
</tr>
<tr>
<td>BCB</td>
<td>Branch on Carry Over (12)</td>
<td>J</td>
</tr>
<tr>
<td>BE</td>
<td>Branch on Equal Compare (B = A)</td>
<td>K</td>
</tr>
<tr>
<td>BEF</td>
<td>Branch on End of File or End of Real</td>
<td>L</td>
</tr>
<tr>
<td>BER</td>
<td>Branch on Error</td>
<td>M</td>
</tr>
<tr>
<td>BJ</td>
<td>Branch on High Compare (B &gt; A)</td>
<td>N</td>
</tr>
<tr>
<td>B1N</td>
<td>Branch on Indicator (B &lt; A)</td>
<td>O</td>
</tr>
<tr>
<td>SL</td>
<td>Branch on Low Compare</td>
<td>P</td>
</tr>
<tr>
<td>BLC</td>
<td>Branch on Last Card (Sense Switch A)</td>
<td>Q</td>
</tr>
<tr>
<td>BM</td>
<td>Branch on Minus (11 zones)</td>
<td>R</td>
</tr>
<tr>
<td>BPCB</td>
<td>Branch Printer Carriage Busy</td>
<td>S</td>
</tr>
<tr>
<td>BPN</td>
<td>Branch Printer Busy</td>
<td>T</td>
</tr>
<tr>
<td>BU</td>
<td>Branch on Unequal Compare (B = A)</td>
<td>U</td>
</tr>
<tr>
<td>BW</td>
<td>Branch on Word Mark</td>
<td>V</td>
</tr>
<tr>
<td>BWZ</td>
<td>Branch on Word Mark or Zone</td>
<td>W</td>
</tr>
<tr>
<td>BCE</td>
<td>Branch if Character Equal (B = d)</td>
<td>X</td>
</tr>
<tr>
<td>BS</td>
<td>Branch if Sense Switch</td>
<td>Y</td>
</tr>
<tr>
<td>C</td>
<td>Compare</td>
<td>Z</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>I/O Commands</th>
<th>Machine Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>I/O</td>
<td>Backspace Tape</td>
</tr>
<tr>
<td>B</td>
<td>Control Unit</td>
</tr>
<tr>
<td>C</td>
<td>Disengage Character Reader</td>
</tr>
<tr>
<td>D</td>
<td>Engage Character Reader</td>
</tr>
<tr>
<td>E</td>
<td>Load Unit</td>
</tr>
<tr>
<td>F</td>
<td>Move Unit</td>
</tr>
<tr>
<td>P</td>
<td>Punch</td>
</tr>
<tr>
<td>M</td>
<td>Punch Column Binary</td>
</tr>
<tr>
<td>B</td>
<td>Read</td>
</tr>
<tr>
<td>R</td>
<td>Read Colons Binary</td>
</tr>
<tr>
<td>T</td>
<td>Read Tape</td>
</tr>
<tr>
<td>M</td>
<td>Read Tape Single Word</td>
</tr>
<tr>
<td>R</td>
<td>Read Tape With Word Marks</td>
</tr>
<tr>
<td>L</td>
<td>Rewind Tape</td>
</tr>
<tr>
<td>U</td>
<td>Rewind and Unload Tape</td>
</tr>
<tr>
<td>S</td>
<td>Seek Disk</td>
</tr>
<tr>
<td>D</td>
<td>Start Punch Feed</td>
</tr>
<tr>
<td>F</td>
<td>Start Read Feed</td>
</tr>
<tr>
<td>W</td>
<td>Write</td>
</tr>
<tr>
<td>WO</td>
<td>Write Disk Single Record</td>
</tr>
<tr>
<td>WD</td>
<td>Write Disk Check</td>
</tr>
<tr>
<td>WDCW</td>
<td>Write Disk Check With Word Marks</td>
</tr>
<tr>
<td>WDT</td>
<td>Write Disk Full Track</td>
</tr>
<tr>
<td>WDTW</td>
<td>Write Disk Full Track With Word Marks</td>
</tr>
<tr>
<td>WDDW</td>
<td>Write Disk Single Record</td>
</tr>
<tr>
<td>WM</td>
<td>Write Word Marks</td>
</tr>
<tr>
<td>WP</td>
<td>Write and Punch</td>
</tr>
<tr>
<td>WR</td>
<td>Write and Read</td>
</tr>
<tr>
<td>WRB</td>
<td>Write and Read Punch Feed</td>
</tr>
<tr>
<td>WRP</td>
<td>Write, Read and Punch</td>
</tr>
<tr>
<td>WT</td>
<td>Write Tape</td>
</tr>
<tr>
<td>WTB</td>
<td>Write Tape Binary</td>
</tr>
<tr>
<td>WTM</td>
<td>Write Tape Mark</td>
</tr>
<tr>
<td>WTW</td>
<td>Write Tape With Word Marks</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Miscellaneous</th>
<th>Machine Language</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICC</td>
<td>Carriage Control</td>
</tr>
<tr>
<td>ICCB</td>
<td>Carriage Control and Branch</td>
</tr>
<tr>
<td>CS</td>
<td>Clear Storage</td>
</tr>
<tr>
<td>CW</td>
<td>Clear Word Mark</td>
</tr>
<tr>
<td>H</td>
<td>Halt</td>
</tr>
<tr>
<td>MA</td>
<td>Modify Address</td>
</tr>
<tr>
<td>NOP</td>
<td>No Operation</td>
</tr>
<tr>
<td>SAR</td>
<td>Store A Address Register</td>
</tr>
<tr>
<td>SBR</td>
<td>Store B Address Register</td>
</tr>
<tr>
<td>FSS</td>
<td>Select Stacker</td>
</tr>
<tr>
<td>FSB</td>
<td>Select Stacker and Branch</td>
</tr>
<tr>
<td>SW</td>
<td>Set Word Mark</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Control Operations</th>
<th>Mnemonic Description</th>
<th>Mnemonic Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>B</td>
<td>Control</td>
<td>XFR Transfer</td>
</tr>
<tr>
<td>END</td>
<td>End</td>
<td>SFX Suffix</td>
</tr>
<tr>
<td>ENT</td>
<td>Enter New</td>
<td>JOB Job</td>
</tr>
<tr>
<td>Coding Mode</td>
<td>INSER Insert</td>
<td>EBR Execute</td>
</tr>
<tr>
<td>B</td>
<td>Alter</td>
<td>ORG Origin</td>
</tr>
<tr>
<td>B</td>
<td>Delete</td>
<td>ORG Origin</td>
</tr>
</tbody>
</table>

*Note: Character must be placed in operand when coding in Autocoder.*
Figure 21. IBM 1401 Program in Symbolic Programming System Language

Figure 22. IBM 1401 Program in Autocoder
they are in SFS.) A complete description of 1401 Autocoder is contained in the IBM 1401 Data Processing System bulletin, Autocoder for the IBM 1401: Preliminary Specifications (form J24-1434).

Figure 22 shows the same program shown in Figure 21 coded in Autocoder language.

**Input-Output Control System (IOCS)**

The IBM 1401 input-output control system eliminates the need for detailed programming of standard input and output operations. It is included part of the 1401 Autocoder system, and provides additional control, record-definition statements, and macro-instructions. With these entries the user can specify the input and output devices and the contents of file records. The library routines for reading and writing, blocking and deblocking, file labeling, and error checking are furnished as part of the Autocoder processor and can be extracted by using the IOCS macro-instructions.

The Autocoder processor tailors the IOCS routines to fit the particular requirements, specified by the programmer, for each job. The Autocoder processor produces the minimum number of instructions needed by interpreting the detailed information the programmer supplies in the control entries.

Although IOCS is used primarily for magnetic-tape files, IOCS macro-instructions can also be used for unit-record files (input and output files) from, and to, the IBM 1402 Card Read-Punch and the IBM 1403 Printer.

The IOCS is described in detail in the IBM 1401 Data Processing System bulletin: Input-Output Control System (form J24-1462).

**IBM 1401 Report Program Generator (RPG)**

The report program generator is a special program designed to produce report-writing object programs from report specifications given by the user.

Instead of writing a specific program for a report, the user states his problem in RPG language. The RPG processor program interprets these specifications and produces an object program that uses source data from punched-card, magnetic-tape, or disk-storage files. Output from the RPG-produced programs can be printed reports, punched cards, or magnetic tape.

The reports produced by programs generated by the RPG range from a simple listing of items from the input file to complex reports that incorporate editing and calculating the input data. Included are such functions as printing various kinds of lines (heading lines, detail lines, total lines initiated by control-field changes, offset total lines, etc.); serial and page numbering; crossfooting; and summary punching. Exception records can be produced with the reports.

The RPG processor can assemble object programs on a 1401 system with at least 4,000 positions of core storage, an IBM 1402 Card Read-Punch, and an IBM 1403 Printer, Model 1 or 2. The machine requirements for executing the object program depend upon the units the user has specified.

The RPG is described in detail in the IBM 1401 Data Processing System bulletins: Report Program Generator for IBM 1401 Card and Tape Systems (form J24-0215) and Report Program Generator for IBM RAMAC 1401 Systems (form J24-1467).
**Operation Codes**

**Arithmetic Operations**

The IBM 1401 Data Processing System adds, subtracts, multiplies, and divides by applying the *add to storage* method of operation. The two factors to be combined are added within core storage without the use of special accumulators or counters. Because any storage area can be used as an accumulator field, the capacity for performing arithmetic functions is not limited by standard-size accumulators or by a predetermined number of accumulators within the system. Also, programming steps can be saved because some arithmetic operations require that only one field be transferred. In arithmetic operations, the 1401 system considers blanks and zeros the same. An unsigned field is considered positive by the system.

The design of the IBM 1401 Data Processing System provides for two types of add operations:

1. **true add**
2. **complement add**

The type of operation performed depends on the operation code and the algebraic signs of the factors. The resultant sign depends on the algebraic values of the factors (Figure 23).

A negative sign is indicated by a B-bit in the units position of a field. Any other zone-bit combination in the units position is considered a plus sign (A- and B-bits, A-no B-bit, or no A-no B-bit). Either A- and B-bits or no A-no B-bits should be used for signing a positive field.

<table>
<thead>
<tr>
<th>A-field sign</th>
<th>+</th>
<th>-</th>
<th>+</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-field sign</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Type of add</td>
<td>True</td>
<td>True</td>
<td>Comp</td>
<td>Comp</td>
</tr>
<tr>
<td>Sign of result</td>
<td>Sign of B-field</td>
<td>unless A &gt; B</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 23. Algebraic Sign Control for Addition

**Complement Add**

If a complement add is initiated (see Figure 23) the machine converts the A-field to its tens complement figure and adds it to the B-field. The machine tests to determine whether a carry occurred from the high-order position of the B-field. The presence of a carry indicates that the result in the B-field is the true answer and the operation is terminated. The sign of the B-field is the sign of the result (Figure 25).

If there is no carry from the high-order position of the B-field, it indicates that the A-field was algebraically larger and that the result is not a true figure but is stored in its tens complement form. A recomplement cycle is performed automatically to convert the answer to true form.

To do this, the machine examines each digit in the B-field from high order to low order, until it detects a B-bit. The B-bit, located in the sign (low order) position of the field initiates the actual recomplement of the result. It changes the sign of the result and converts (from low order to high order) each digit to true form (Figure 26).

The sign of the B-field (result) is always in the form of A- and B-bits if it is plus, and a B-bit if it is minus.

**Note:** If the units position of a recomplemented field is located in the high-order storage position, a wrap-around addressing error occurs when this position is addressed on the reverse scan.

![Figure 24. True Add](image)

**True Add**

In a true add operation the result always carries the sign of the B-field (Figure 24).

![Figure 26. Complement Add with Recomplementing](image)
Add

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>xxx</td>
<td>xxx</td>
</tr>
</tbody>
</table>

Function. The data in the A-field is added algebraically to the data in the B-field. The result is stored in the B-field.

Word Marks. The B-field must have a defining word mark, because it is this word mark that actually stops the add operation.

The A-field must have a word mark, only if it is shorter than the B-field. In this case, the transmission of data from the A-field stops after the A-field word mark is sensed. Zeros are then inserted in the A-register until the B-field word mark is sensed.

If the A-field is longer than the B-field, the high-order positions of the A-field that exceed the limits imposed by the B-field word mark are not processed. For overflow conditions and considerations, assume that the A-field is the same length as the B-field. (See Address Modification without Index Feature.)

Timing.
1. If the operation does not require a recocompletion cycle:
   \[ T = 0.0115 (L_A + 3 + L_A + L_B) \text{ ms.} \]
2. If a recocompletion cycle is taken:
   \[ T = 0.0115 (L_A + 3 + L_A + 4 L_B) \text{ ms.} \]

Note. Sign control: (see Figure 23).

If a recocompletion cycle is automatically taken, the sign of the B (result)-field is changed and the result is always stored in true form.

If the fields to be added contain zone bits in other than the high-order position of the B-field and the sign positions of both fields, only the digits are used in a true-add operation. B-field zone bits are removed except for the units and high-order positions in a true-add operation. If a complement add takes place, zone bits are removed from all but the units positions of the B-field.

If an overflow occurs during a true-add operation, a special overflow indicator is set, and the overflow indications are stored over the high-order digit of the B-field:

<table>
<thead>
<tr>
<th>Condition</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>First overflow</td>
<td>A-bit</td>
</tr>
<tr>
<td>Second overflow</td>
<td>B-bit</td>
</tr>
<tr>
<td>Third overflow</td>
<td>A- and B-bits</td>
</tr>
<tr>
<td>Fourth overflow</td>
<td>No A- or B-bits</td>
</tr>
</tbody>
</table>

For subsequent overflows repeat conditions 1-4.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-L_A</td>
<td>B-L_B</td>
</tr>
</tbody>
</table>

Example. Add CURERN (0506) to YTDGRO (0708), Figure 27.

Add

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
<td>xxx</td>
</tr>
</tbody>
</table>

Function. This format of the ADD instruction causes the data in the A-field to be added to itself.

Word Marks. The A-field must have a defining word mark. It is this word mark that stops the add operation. This instruction must be followed by a word mark in the position after the A-address.

Timing. \[ T = 0.0115 (L_A + 3 + 2 L_A) \text{ ms.} \]

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-L_A</td>
<td>A-L_A</td>
</tr>
</tbody>
</table>

Example. Add to itself the data at EXEMPT (0981), Figure 28.
Page 29 — Add instruction, add to last line of column one:

Overflow indication does not occur for a 1-position field.
Zero and Add

Instruction Format.

Mnemonic  Op Code  A-address  B-address
ZA  1  xxx  xxx

Function. This instruction functionally adds the A-field to a zeroed B-field. Technically, this is accomplished by moving the A-field to the B-field. The high-order positions of the B-field are set to zero if the B-field is larger than the A-field. The data from the A-field moves directly from the A-register to storage. Zone bits are stripped from all positions except the units position. Blanks in the A-field are stored as blanks in the B-field.

Word Marks. A word mark is required for definition of the B-field. It is required in the A-field, only if it is shorter than the B-field. All high-order B-field positions will contain zeros. But the transmission of data from A stops when the A-field word mark is detected.

Timing. \( T = 0.0115 (L_A + 1 + L_A + L_B) \) ms.

Note: The sign of the result always has both A- and B-bits if it is positive. If the sign is negative, it has only a B-bit.

Address Registers After Operation.

NSI  A-Lw  B-Lw

Example. Zero WHTAX area (0796-0802) and add new TAX (0749-0754) to WHTAX (Figure 29).

Subtract

Instruction Format.

Mnemonic  Op Code  A-address  B-address
S  S  xxx  xxx

Function. The numerical data in the A-field is subtracted algebraically from the numerical data in the B-field (Figure 30). The result is stored in the B-field.

<table>
<thead>
<tr>
<th>A-field sign</th>
<th>+</th>
<th>−</th>
<th>+</th>
<th>−</th>
</tr>
</thead>
<tbody>
<tr>
<td>B-field sign</td>
<td>+</td>
<td>−</td>
<td>−</td>
<td>+</td>
</tr>
<tr>
<td>Type of add</td>
<td>Comp</td>
<td>Comp</td>
<td>True</td>
<td>True</td>
</tr>
<tr>
<td>Sign of result</td>
<td>Sign of B-field unless A &gt; B</td>
<td>Sign B</td>
<td>Sign B</td>
<td></td>
</tr>
</tbody>
</table>

Figure 30. Algebraic Sign Control for Subtraction

Word Marks. A word mark is required to define the B-field. An A-field requires a word mark, only if it is shorter than the B-field. In this case, the A-field word mark stops transmission of data from the A-field.

Timing.

1. Subtract - no recomplement:
   \( T = 0.0115 (L_A + 3 + L_A + L_B) \) ms.

2. Subtract - recomplement cycle necessary:
   \( T = 0.0115 (L_A + 3 + L_A + 4 L_B) \) ms.

Note. If a recomplement cycle is automatically taken, the sign of the B (result)-field is changed, and the result is always stored in true form.

Address Registers After Operation.

NSI  A-Lw  B-Lw

Example. Subtract CUFICA (0753) from CURGRO (0896), Figure 31.
Subtract

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>S</td>
<td>xxx</td>
</tr>
</tbody>
</table>

Function. The data at the A-address is subtracted from itself. If the A-field sign is minus, the result is a minus zero. If the A-field sign is plus, the result is a plus zero.

Word Marks. The A-field must have a defining word mark. This instruction must be followed by a word mark in the position after the A-address.

Timing. \( T = 0.0115 \left( L_A + 3 + 2 L_A \right) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-LA</td>
<td>A-LA</td>
</tr>
</tbody>
</table>

Example. Subtract from itself the field labeled LIMIT (units position is 0395), Figure 32.

Zero and Subtract

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>ZS</td>
<td>1</td>
<td>xxx</td>
<td>xxx</td>
</tr>
</tbody>
</table>

Function. This instruction functionally subtracts the A-field from a zeroed B-field. Technically, this is accomplished by moving the A-field to the B-field. The high-order positions of the B-field are set to zero if the B-field is larger than the A-field. The data from the A-field is moved directly from the A-register to the B-field. Zone bits are stripped from all but the sign (units) position.

Word Marks. A word mark is required to define the B-field. If the A-field is shorter than the B-field, the A-field must have a defining word mark to stop transmission of data to B. The extra high-order B-field positions contain zeros, if A is shorter than B.

Timing. \( T = 0.0115 \left( L_A + 1 + L_A + L_B \right) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-LA</td>
<td>A-LA</td>
</tr>
</tbody>
</table>

Example. Subtract TAXEXP (0699) from ACCUMI (0755), Figure 33.
Zero and Subtract instruction, add:

**Note:** If the A-field is positive, the B-field result is negative. If the A-field is negative, the B-field result is positive.
Logic Operations

The logic operation codes provide the decision-making ability of the IBM 1401 Data Processing System. They allow the program to test for conditions that can arise during processing, and to branch to predetermined sets of instructions, or subroutines as a result of a specific condition.

For example, if an overflow occurs in an arithmetic operation, a special routine to handle this condition can be initiated by a BRANCH IF INDICATOR ON instruction.

Note: Any operation that terminates with a successful branch to another portion of core storage for the next instruction address operates as follows:
1. Without Indexing—The B-address register is reset to blanks during the next instruction operation (I-Op) cycle.
2. With Indexing — A dummy cycle occurs before the next instruction operation (I-Op) cycle and the next sequential instruction is taken. Figure 35 shows characters that are valid for the d-character and for the indicators they test. This figure also shows testing, for high, low, or equal, which is used when the high-low-equal compare special feature is installed.

The indicators tested are not turned off by this instruction except as noted by a +. When carriage tape-channels 9 or 12 are sensed, corresponding indicators are turned on. These carriage channel indicators are turned off when any other carriage tape-channel is sensed. The next compare instruction turns off the compare indicators.

Word Marks. Word marks are not affected.

Timing. $T = 0.115 (L_i + 1)$ ms.

Branch if Indicator On

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>B</td>
<td>xxx</td>
<td>x</td>
</tr>
</tbody>
</table>

Function. The d-character specifies the indicator tested. If the indicator is on, the next instruction is taken from the I-address. If the indicator is off, the next sequential instruction is taken.

Address Registers After Operation.

NSI | BI | BI

Example. Unconditionally branch to AGAIN (3498), Figure 35.

Figure 35. Branch

Word Marks. Word marks are not affected.

Timing. $T = 0.115 (L_i + 1)$ ms.

Figure 36. d-Characters for Branch Instruction
Address Registers After Operation.

NSI       B1      dbb

Example. Test for last card. If it is the last card, branch to END (0599), Figure 37.

Function. This code causes the single character at the B-address to be compared to the d-character. If it has the same bit configuration as the d-character, the program branches to the I-address, otherwise the program continues sequentially. The d-character can be any combination of the six BCD code bits (BA 8421).

Word Marks. Word marks in the location tested have no effect on the operation.

Timing. \( T = 0.0115 \) (\( L_1 + 2 \)) ms.

Address Registers After Operation.

NSI       B1      B-1

Example. This example shows how the chaining method can be used to test an entire field for blank characters. Each position in the area labeled AMOUNT (0350, 0349, 0348 and 0347) is individually tested for a blank character. If a blank is found, the program branches to BLANK (0601) for the next instruction. If the position tested contains a character, the program continues in sequence (Figure 38).

Operation Codes 33
Figure 39. Branch on Zone or Word-Mark Test

No Operation

Instruction Format.

Mnemonic  Op Code

NOP  N

Function. This code performs no operation. It can be substituted for the operation code of any instruction to make that instruction ineffective. It is commonly used in program modification to cause the machine to skip over specific instructions.

Word Marks. The program operation resumes at the next operation code identified by a word mark.

Note. If characters without word marks follow an N operation code, these characters enter the A- and B-field registers. For example:

\[
\text{N 1234 A XXXX}
\]

In this instance, the address registers after operation would be:

\[
\]

\[
\text{NSI 123 4bb}
\]

Timing. \( T = 0.0115 \) (\( L_b + 1 + 2L_w \)) ms.

Address Registers After Operation.


NSI  A  B

Example. Leave one storage position open for an operation code such as READ A CARD (1). Operation code 1 can be inserted if needed (Figure 40).

Figure 40. No Operation

Compare

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>C</td>
<td>C</td>
<td>xxx</td>
<td>xxx</td>
</tr>
</tbody>
</table>

Function. The data in the A-field is compared to an equal number of characters in the B-field. The bit configuration (BA 8421) of each character in the two fields is compared. The comparison turns on an indicator that can be tested by a subsequent BRANCH IF INDICATOR ON instruction. The indicator is reset by the next COMPARE instruction.

The same indicators set by the COMPARE instruction are also affected by a disk-unit operation (seek, read, write, and write check). The disk unit performs an address-compare operation automatically on the address in core storage, with the address on the disk record, by using the compare circuits and by setting the appropriate indicator (equal, high, or low). Therefore, careful consideration must be made in the use of a COMPARE instruction and subsequent BRANCH IF INDICATOR ON instructions for testing the results of the COMPARE instruction when disk-unit operations are to be performed.

Word Marks. The first word mark encountered stops the operation. If the A-field is longer than the B-field, extra A-field positions at the left of the B-field word mark are not compared. If the B-field is longer than the A-field, an unequal-compare results.

Timing. \( T = 0.0115 \) (\( L_b + 1 + 2L_w \)) ms.

Note. Both fields must have exactly the same bit configurations to be equal. For example, \( 001(=?0) \) compared to \( 001(=0) \) results in an unequal comparison.

All characters that can appear in storage can be compared. The ascending sequence of characters is as follows: blank, \( \# @ ? \) through \( ! / \) through \( R \# S \) through \( Z 0 \) through \( 9 \).

Address Registers After Operation.


NSI  A-L_w B-L_w

Example. Compare the department numbers punched in two cards. Department numbers are located in:

<table>
<thead>
<tr>
<th>Card</th>
<th>Label</th>
<th>Actual Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>DEPTNO</td>
<td>1098</td>
</tr>
<tr>
<td>2</td>
<td>DEPTCD</td>
<td>0004</td>
</tr>
</tbody>
</table>

Then test the result of the compare operation. If the department numbers are equal, continue the program in sequence. If they are unequal, branch to TOTAL (6435) for the next instruction (Figure 41).
Page 34 — **No Operation** instruction, add to **Function:**

Instructions that have A-addresses of %xx or @xx should have their A-address field, as well as the operation code position, changed to N's to perform a no operation function successfully.
**Halt**

**Instruction Format.**

Mnemonic: \( H \)  
Op-Code: \( - \)

**Function.** This instruction causes the machine to stop and the stop-key light to turn on. Pressing the start key causes the program to start at the next instruction in sequence.

**Word Marks.** Word marks are not affected.

**Timing.** \( T = 0.0115 \ (L_1 + 1) \) ms.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>Ap</td>
<td>Bp</td>
</tr>
</tbody>
</table>

**Example.** Figure 42 is a symbolic example of the Halt instruction.

**Halt and Branch**

**Instruction Format.**

Mnemonic: \( H \)  
Op Code: \( - \)  
I-address: \( XXX \)

**Function.** This is the same as Halt, except that the next instruction is at the I-address.

**Clear, Move, Load, and Word Mark Operations**

To organize specific storage areas for efficient processing, data must be processed, and word marks must be set in storage in the proper locations.

Because the IBM 1401 operates with variable length data and instructions, and because word marks control the definition of each, special attention must be given to the proper positioning of each word mark required. Extraneous word marks are just as damaging to program operation as the absence of a word mark when it is needed.

Before any program is loaded, all positions of storage should be cleared of data and word marks. Pressing the load key on the console causes a word mark to be set at location 001. The program itself should contain the instructions to set all other word marks needed for the operation.

Move and Load operations transfer data from one location to another. The Move operation does not affect word marks, but the Load operation causes word marks, as well as data, to be transferred.

**Operation Codes**
Clear Storage

Instruction Format.

Mnemonic | Op Code | A-address
---------|---------|---------
CS        |         | xxx

Function. As many as 100 positions of core storage can be cleared of data and word marks when this instruction is executed. Clearing starts at the A-address and continues leftward to the nearest hundreds position. The cleared area is set to blanks.

Word Marks. Word marks are not required to stop the operation.

Timing. \( T = 0.0115 \left( L + 1 + L_x \right) \) ms.

Note: During the execution of this instruction, only the B-address register is used. Therefore, when chaining is being considered, the contents of the A-address register can be ignored.

Address Registers After Operation.

-------------|-------------|-------------
NSI          | A           | x 00-1

Example. Clear WAREA5 (0500-0563), Figure 44.

Set Word Mark

Instruction Format.

Mnemonic | Op Code | A-address | B-address
---------|---------|-----------|-----------
SW        |         | xxx       | xxx

Function. A word mark is set at each address specified in the instruction. The data at each address is undisturbed.

Word Marks. Word marks are set at both the A- and B-addresses specified.

Timing. \( T = 0.0115 \left( L + 1 \right) \) ms.

Address Registers After Operation.

-------------|-------------|-------------
NSI          | A           | B-1

Example. Set word marks at locations BEGIN1 (3950) and BEGIN2 (3970), Figure 46.
Page 36 — **Clear Storage and Branch** instruction, replace **Word Marks** with:

**Word Marks.** Word marks are not required to stop the operation. It is not necessary to follow this instruction by a character with a word mark.

Page 36 — **Set Word Mark** instruction (two addresses), replace **Word Marks** with:

**Word Marks.** Word marks are set at both the A- and B-addresses specified. It is not necessary to follow this instruction by a character with a word mark.
Word Marks. A word mark is set at the A-address.

Timing. \( T = .0115 \ (L_1 + 3) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-1</td>
<td>A-1</td>
</tr>
</tbody>
</table>

Example. Set a word mark at AREA2 (2901), Figure 47.

### Clear Word Mark

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>CW</td>
<td>5</td>
<td>xxx</td>
<td>xxx</td>
</tr>
</tbody>
</table>

**Function.** This format of the CLEAR WORD MARK instruction causes the word mark to be cleared at the A-address. Data at the A-address is not disturbed.

Word Marks. Word marks are cleared at the A-address only.

Timing. \( T = .0115 \ (L_1 + 3) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-1</td>
<td>A-1</td>
</tr>
</tbody>
</table>

Example. Clear the word mark at RECN01 (3608), Figure 49.

### Move Characters to A or B Word Mark

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>M</td>
<td>xxx</td>
<td>xxx</td>
</tr>
</tbody>
</table>

**Function.** The data in the A-field is moved to the B-field.

Word Marks. If both fields are the same length, only one of the fields must have a defining word mark. The first word mark encountered stops the operation. If the word mark is sensed in the A-field, the machine takes one more B-cycle to move the high-order character from A to B. At the end of the operation, the A-address register and the B-address register contain the addresses of the storage locations immediately to the left of the A- and B-fields processed by the instruction. The data at the A-address is unaffected by the move operation. Word marks in both fields are undisturbed.

Timing. \( T = .0115 \ (L_1 + 1 + 2 L_W) \) ms.
Note. If the fields are unequal in length, chaining can produce unwanted results, because one of the fields has not been completely processed. Thus, one of the registers will not contain the address of the units position of the left-adjacent field.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-Lw</td>
<td>Bp-Lw</td>
</tr>
</tbody>
</table>

Example. Move the 5-character field NAMIN (0750) to the 5-character field NAMOUT (0850), Figure 50.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-Lw</td>
<td>Bp-Lw</td>
</tr>
</tbody>
</table>

Example. Move the following three fields (labeled EMPNO, DEPTNO and TAXCLS) and store them sequentially at RECOUT (units position at 0204), Figure 51.

Note: If the B-address register already contains the correct address, the B-label of the first instruction in the example can be eliminated.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-Lw</td>
<td>Bp-Lw</td>
</tr>
</tbody>
</table>

Example. Move and suppress the zeros in the 10-character field labeled INVBAL (0958) to the area labeled OUTPT4 (0448), Figure 52.

Move Characters to A or B Word Mark

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>M</td>
<td>xxx</td>
</tr>
<tr>
<td>MCW</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>M</td>
<td></td>
</tr>
<tr>
<td>MLC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function. This format of the move operation can be used when it is desired to move fields from the A-area and store them sequentially in the B-area. It saves program storage space and time, because the B-address is automatically taken from the B-address register, and does not have to be written or interpreted as part of the instruction.

Word Marks. A word mark is required over the high-order position of the A- or B-field. The first word mark encountered stops the move operation.

Timing. \( T = .0115 \left(L_1 + 1 + 2L_w\right) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-Lw</td>
<td>Bp-Lw</td>
</tr>
</tbody>
</table>

Example. Move and suppress the zeros in the 10-character field labeled INVBAL (0958) to the area labeled OUTPT4 (0448), Figure 52.
Note: This description of the instruction assumes a 1401 system without the expanded print edit feature. If expanded print edit is installed, a decimal does not cause zero suppression to begin again.
Move Numerical

Instruction Format.

Function. The numerical portion (8-4-2-1 bits) of the single character in the A-address is moved to the B-address. The zone portions (AB-bits) are undisturbed at both addresses.

Word Marks. Word marks are not required at either address, because the nature of the instruction always specifies that only one digit is to be transmitted.

Timing. $T = .0115 (L_1 + 3)$ ms.

Address Registers After Operation.

Example. Move the numerical portion of the units position of ONHAND (0986) to OUT5 (0789), Figure 53.

Load Characters to A Word Mark

Instruction Format.

Function. This instruction is commonly used to load data into the printer or punch areas of storage, and also to transfer data or instructions from the read-in area to another storage area. The data and word mark from the A-field are transferred to the B-field, and all other word marks in the B-field are cleared.

Operation Codes 39
Word Marks. The A-field must have a defining word mark, because the A-field word mark stops the operation. Note: If the B-field is larger than the A-field, the B-field word mark is not cleared.

Timing. \( T = 0.0115 (L_A + 1 + 2 L_A) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NI</td>
<td>A-( L_A )</td>
<td>B-( L_A )</td>
</tr>
</tbody>
</table>

Example. Transfer the data and word marks from REC4 (0950) to OUT8 (0650), Figure 55.

Timing. \( T = 0.0115 (L_A + 1 + 2 L_A) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-( L_A )</td>
<td>B-( L_A )</td>
</tr>
</tbody>
</table>

Example. Load the following three fields: EMPYNO, DEPTNO, and TAXCLS with their word marks to sequential locations, beginning at area labeled PRINT1 (0204). Assume that the B-address register is standing at 0200 (Figure 56).

**Load Characters to A Word Mark**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>LCA</td>
<td>XXX</td>
</tr>
<tr>
<td>A</td>
<td>MLCWA</td>
<td></td>
</tr>
</tbody>
</table>

Function. This format can be used when several A-fields (not necessarily in sequence) are to be loaded sequentially in the B-field. This instruction causes the A-field data and word mark to be moved to the B-field. B-field word marks are cleared, up to the A-field word marks.
Editing

Editing, in the IBM 1401 Data Processing System, means punctuating printed output data, and includes the automatic control of zero suppression, and the insertion of identifying symbols. A single edit instruction causes all desired commas, decimals, dollar signs, asterisks, credit symbols, and minus signs to be automatically inserted in a numerical field. At the same time, unwanted zeros to the left of significant digits are suppressed (Figure 57).

The control word has two parts: the body (which punctuates the A-field), and the status portion (which contains the dollar sign, sign-symbols, and class of total asterisks). The sign of the A-field determines whether or not sign-symbols will print. The sign of the A-field is removed.

To edit a field, a Load Characters to a Word Mark instruction loads the control word into the printer output area. This puts the control word where the edited information will eventually go. Then, a Move Characters and Edit instruction (with the same B-address as the previous load instruction) performs the editing function as it moves the data to the output area.

Note: A one-position field cannot be edited.

Move Characters and Edit

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCE</td>
<td>E</td>
<td>789</td>
<td>300</td>
</tr>
</tbody>
</table>

Function. The Move Characters and Edit instruction modifies the data in the A-field by the contents of the edit-control word in the B-field, and stores the result in the B-field. The following set of rules controls the editing operation.

Rule 1. All numerical, alphabetic, and special characters can be used in the control word. However, some of these have special meanings:

<table>
<thead>
<tr>
<th>Control Character</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>b (blank)</td>
<td>This is replaced with the character from the corresponding position of the A-field.</td>
</tr>
<tr>
<td>O (zero)</td>
<td>This is used for zero suppression, and is replaced with a corresponding character from the A-field. Also the right-most &quot;O&quot; in the control word indicates the right-most limit of zero suppression.</td>
</tr>
<tr>
<td>, (comma)</td>
<td>This remains in the edited field in the position where written. It is removed during a zero-suppress operation if it is to the left of the high-order significant digit.</td>
</tr>
<tr>
<td>CR (credit)</td>
<td>This is undisturbed if the data sign is positive. It is blanked out if the data sign is negative. It can be used in body of control word without being subject to sign control.</td>
</tr>
<tr>
<td>- (minus)</td>
<td>This is the same as CR.</td>
</tr>
<tr>
<td>&amp; (ampersand)</td>
<td>This causes a space in the edited field. It can be used in multiples.</td>
</tr>
<tr>
<td>* (asterisk)</td>
<td>This can be used in singular or in multiples, usually to indicate class of total. When it is used with the expanded print edit feature, it takes on an additional function (see Expanded Print Edit).</td>
</tr>
<tr>
<td>$ (dollar sign)</td>
<td>This is undisturbed in the position where it is written. When used with the expanded print edit feature, it has an additional function (see Expanded Print Edit).</td>
</tr>
</tbody>
</table>

Rule 2. A word mark in the high-order position of the B-field controls the Move Characters and Edit operation.

Rule 3. When the A-field word mark is sensed, the remaining commas in the control field are set to blanks.

Rule 4. The body of the control word is that portion beginning with the right-most blank or zero, and continuing to the left to the control character that governs the transfer of the last position of the data field. The remaining portion of the control field is the status portion.

Rule 5. If the data field is positive, and if the CR or ~ symbols are located in the status portion of the control word, they are blanked out.

Editing 41
<table>
<thead>
<tr>
<th>Cycle</th>
<th>TYPE OF CYCLE</th>
<th>ADDRESS REGISTERS</th>
<th>REG. STORED INTO</th>
<th>&quot;B&quot; FIELD AT END OF CYCLE</th>
<th>REMARKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>t1</td>
<td>I 002 A B 0 A</td>
<td>7 7 7</td>
<td>$bbb b b b 0 b b &amp; C R  &amp; **</td>
<td>Read Instr. OP Code</td>
</tr>
<tr>
<td>2</td>
<td>t1</td>
<td>I 003 07:b:b 07:b:b 7 7</td>
<td>same</td>
<td>Load A Address Register</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>t1</td>
<td>I 004 07:b:b 07:b:b 8 8 8</td>
<td>same</td>
<td>Load A Address Register</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>t1</td>
<td>I 005 07:b:b 07:b:b 9 9 9</td>
<td>same</td>
<td>Load A Address Register</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>t1</td>
<td>I 006 07:b:b 03:b:b 3 3 3</td>
<td>same</td>
<td>Load B Address Register</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>t1</td>
<td>I 007 07:b:b 03:b:b 0 0 0</td>
<td>same</td>
<td>Load B Address Register</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>t1</td>
<td>I 008 07:b:b 02:0:0 0 0 0</td>
<td>same</td>
<td>Load B Address Register</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>t1</td>
<td>I 008 07:b:b 02:0:0 OP 0 OP 0</td>
<td>same</td>
<td>OP code of next instr.</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>A</td>
<td>008 07:b:b 02:0:0 6 6 6</td>
<td>same</td>
<td>Execute EDIT Instr.</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>B</td>
<td>008 07:b:b 02:9:9 * 6 *</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>B</td>
<td>008 07:b:b 02:9:9 6 6 6</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>B</td>
<td>008 07:b:b 02:9:9 6 6 6</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>B</td>
<td>008 07:b:b 02:9:9 6 6 6</td>
<td>same</td>
<td>Rule 1 and 5</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>B</td>
<td>008 07:b:b 02:9:9 6 6 6</td>
<td>same</td>
<td>Rule 1 and 5</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>B</td>
<td>008 07:b:b 02:9:9 6 6 6</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>B</td>
<td>008 07:b:b 02:9:9 6 6 6</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>A</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>A</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>A</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1 and 6</td>
<td></td>
</tr>
<tr>
<td>24</td>
<td>A</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1 and 6</td>
<td></td>
</tr>
<tr>
<td>25</td>
<td>A</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>26</td>
<td>A</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>27</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>28</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>A</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>A</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>33</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>34</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>36</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>37</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>38</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>39</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>40</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
<tr>
<td>41</td>
<td>B</td>
<td>008 07:b:b 02:9:9 2 2 2</td>
<td>same</td>
<td>Rule 1</td>
<td></td>
</tr>
</tbody>
</table>

Figure 58. Step-by-Step Editing Operation.
Rule 6. Zero suppression is used if unwanted zeros to the left of significant digits in a data field are to be deleted (Figure 59).

A special zero is placed (in the body of the control word) in the right-most limit of zero suppression.

To properly perform zero-suppression operations there must be at least one character to the left of the zero-suppression character in the control word.

Forward Scan:
1. The positions in the output field at the right of this special zero are replaced by the corresponding digits from the A-field.
2. The special zero is replaced by the corresponding digit from the A-field, when it is detected in the control field.
3. A word mark is automatically set in this position of the B-field (output) field.
4. The scan continues until the B-field (high order) word mark is sensed and removed.

Reverse Scan:
1. In the output field, blanks replace all zeros and punctuation, except hyphens at the left of the first significant character (up to and including the zero suppression code position).

2. When the automatically-set zero suppression word mark is sensed, it is erased and the operation ends.

Rule 7. The data field can contain fewer, but must not contain more positions than the number of blanks and zeros in the body of the control word. Dollar signs and asterisks are included in the body of the control word with the expanded print edit special feature.

Figure 59 shows the use of these rules as applied to the data in Figure 57.

Timing. \( T = 0.0115 \ (L_1 + 1 + L_A + L_B + L_N) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th>Without zero suppression</th>
<th>With zero suppression</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-address minus the length of the A-field.</td>
</tr>
<tr>
<td>NSI</td>
<td>A-address minus the length of the A-field.</td>
</tr>
</tbody>
</table>

Example. Edit the data labeled GROPAY (0985) by the edit-control word EDCONT (0325). Store the result in PRINT6 (0250), Figure 60.

Figure 60. Edit
Input-Output Operations

This section describes the timing and operation codes the 1401 uses to control card reading, card punching, and printing. Several, different, combination operation codes make it possible to perform multiple functions: card reading and punching; printing and reading; printing, reading and punching; etc. Instruction format flexibility permits automatic program branching.

Card Read Instructions

The card reader operates at a rated speed of 800 cycles per minute (one cycle every 75 milliseconds). The card reading speed depends on the timing of the READ A CARD instructions in the program. To effect continuous card-reading at the rate of 800 cards per minute, a READ A CARD instruction must be given within 10 milliseconds after the preceding card has been actually read into the IBM 1401 Processing Unit. If more than 10 milliseconds are required for processing, the card read speed drops to 400 cards per minute. This happens because of the mechanical structure of the card feed. There is only one point in the cycle during which a card can feed, and if no read impulse signals the feed at that time, the reader will be delayed for 75 milliseconds (or until the same point in the following cycle).

The read release special feature permits job time improvements by allowing more actual processing time during the read cycle.

Read a Card

Instruction Format.

Mnemonic Op Code

R 1

Function. This code causes a card to feed, and causes all 80 columns of information to be read into core-storage locations 001 through 080.

Word Marks. Word marks are undisturbed.

Timing. \[ T = 0.0115 \left( L_1 + 1 \right) \text{ ms} + \text{i/O} \]

A card read cycle requires a total of 75 milliseconds. The cycle is divided into three major operations (Figure 61):

1. Read start time is 21 ms. The read instruction must be given prior to card reading time in order to activate the card feed for that particular cycle. If the read instruction is given too late in the cycle, processing is delayed until the next card reading time occurs in the following read cycle. The processing unit is interlocked during read start time unless the read release special feature is installed.

2. Card read time is 44 ms. The actual reading of the card takes place during this part of the cycle and the data is read into core storage. The processing unit is interlocked during card reading time.

3. Processing time is 10 ms. This part of the cycle is for processing. If processing time requires more than 10 milliseconds, the reader speed drops from 800 to 400 cards per minute.

Note. The processing time allowed is increased if the read release special feature is included in the system (see Read Release).

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>Ap</td>
<td>081</td>
</tr>
</tbody>
</table>

Example. Read a card (Figure 62).

![Figure 62. Read Card](image-url)
Read and Branch

Instruction Format.

Mnemonic | Op Code | I-address
---|---|---
R | 1 | XXX

Function. This is the same as the READ A CARD instruction, except that the next instruction is taken from the I-address instead of from the next sequential instruction address. The program branch occurs after the card has been read into storage.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_1 + 1) \text{ ms} + I/O \).

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>BI</td>
<td>061</td>
</tr>
</tbody>
</table>

Example. Read a card, and branch to CALC1 (1500), Figure 63.

Punch Instructions

The card punch operates at a rated speed of 250 cycles per minute (240 ms per cycle). Actual card punching, at an optimum rate of 250 cards per minute, is controlled by punch instructions in the program.

There are four points in the cycle (occurring at 60 millisecond intervals) when the punch feeding mechanism can receive an impulse to start the punch cycle.

The punch cycle is divided into three separate functions (Figure 64):

1. Punch start time is 37 ms. After the feed mechanism has been impulsed, the time required for the card to feed and be positioned for punching is called punch start time. The IBM 1401 Processing Unit is interlocked during punch start time unless punch release special feature is employed.

2. Card punching time is 181 ms. The actual punching of the card takes place during this part of the cycle. The IBM 1401 Processing Unit is always interlocked during card-punching time.

3. Processing time is 22 ms. This is the remainder of the punch cycle that is allotted for processing in the 1401 unit.

If the punch release special feature is used in the system, a total of 59 ms are available for processing (see Start Punch Feed). In a system with the punch feed read special feature, 57 ms are available for processing.

During card-punching time, the card feeds past the punch station 12-edge first. At the appropriate digit times, the information from storage locations 101-180 is emitted in BCD code; is translated into IBM card code; and is punched in the corresponding card columns. The information in the storage punch area (101-180) is undisturbed, and can be removed by a CLEAR STORAGE instruction.

Punch a Card

Instruction Format.

Mnemonic | Op Code
---|---
P | 4

Function. The data in storage locations 101 through 180 is punched into an IBM card.
Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_1 + 1) \text{ ms} + I/O. \)

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>Ap</td>
<td>181</td>
</tr>
</tbody>
</table>

Example. Feed a card, and punch (Figure 65).

![Figure 65. Punch and Branch](image)

Autocoder

Assembled Instruction: 4 X58

Figure 65. Punch

Print Instructions

The IBM 1403 prints at a maximum rate of 600 lines per minute. When a WRITE A LINE instruction is interpreted the data in the print area (addresses 201 through 300 in the basic IBM 1403, and addresses 201-332 in a 1403 equipped with additional print positions) is transferred character-by-character to the printer. The printer spaces once after each line is printed, unless impulsed to do otherwise. The printer operates only when impulsed to print. Thus, the print cycle is started when needed and the printer immediately starts to print at the beginning of the cycle. The 100-millisecond print cycle is subdivided (Figure 67).

1. Print time is 84 ms. The line is printed during this part of the cycle. The IBM 1403 Processing Unit is interlocked during print time unless the print storage special feature is employed.

2. Process time is 16 ms. This is the normal processing time available during the print cycle (see Print Storage).

3. Form movement time is 20 ms. The normal form movement time (one space) is always overlapped by processing time. Skipping time is not overlapped and must be included in the calculation of total program time.

An internal check is performed to insure that the character set up at the print mechanism is the same as that sent from storage.

![Figure 67. Print Cycle](image)
Write a Line

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td></td>
<td>This instruction causes the data in the print area to be transferred to the printer. The program continues after printing is complete. The printer takes one automatic space after printing a line.</td>
</tr>
</tbody>
</table>

Word Marks. Word marks are not affected.

Timing. \( T = .0115 \ (L_k + 1) \text{ ms} + I/O. \)

Note. The normal 84 ms-interlock during printing can be greatly reduced, if the print storage special feature is installed (see Print Storage).

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>Ap</td>
<td>335</td>
</tr>
</tbody>
</table>

Example. Print the data in the print area (Figure 68).

Write and Branch

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>W</td>
<td></td>
<td>( \ell )-character</td>
</tr>
</tbody>
</table>

Function. This is the same as the WRITE A LINE instruction, except that the next instruction after printing is taken from the I-address.

Word Marks. Word marks are not affected.

Timing. \( T = .0115 \ (L_k + 1) \text{ ms} + I/O. \)

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>BI</td>
<td>335</td>
</tr>
</tbody>
</table>

Example. Print, and branch to STREC (0678) for the next instruction (Figure 69).

Write Word Marks

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>( \ell )-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>W</td>
<td>2</td>
</tr>
</tbody>
</table>

Function. The word marks associated with storage addresses in the print area print as the digit 1 in the corresponding print positions. After printing, the machine takes an automatic space, unless otherwise impulsive. The \( \Box \) causes the word marks to be transferred to the printer.

Word Marks. Word marks remain in their original positions in the print area.

Timing. \( T = .0115 \ (L_k + 1) \text{ ms} + I/O. \)

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>06b</td>
<td>335</td>
</tr>
</tbody>
</table>

Example. Print all word marks in the print area (Figure 70).

Figure 68. Print

Figure 69. Print and Branch

Figure 70. Print Word Marks
Write Word Marks and Branch

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS WM</td>
<td>2</td>
<td>xxx</td>
<td></td>
</tr>
</tbody>
</table>

**Function.** This is the same as WRITE WORD MARKS (2 WM) instruction, except that the next instruction is taken from the I-address instead of from the next sequential instruction.

**Word Marks.** Word marks remain in their original positions in the print area.

**Timing.** 

\[ T = 0.0115 (L_1 + 1) \text{ ms} + \text{I/O}. \]

**Address Registers After Operation.**

- I-Add. Reg: NSI
- B-Add. Reg: 335

**Example.** Print word marks, and branch to RESTAR (0890), Figure 71.

---

**Combination Instructions**

It is often practical to combine two or three input/output functions in one instruction. Several special operation codes are provided to make it possible to perform some operations simultaneously. Each combination instruction has a corresponding instruction that permits automatic branching to a predetermined instruction address after the functions are complete.

**Write and Read**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>WR</td>
<td>2</td>
<td>xxx</td>
</tr>
</tbody>
</table>

**Function.** This instruction combines the functions of READ A CARD (1) and WRITE A LINE (2). The printer takes priority, and the print cycle is completed before the actual card reading operation takes place. However, the signal to start the reader can be accepted before the end of the print cycle. Thus, read start time overlaps the print cycle.

**Word Marks.** Word marks are not affected.

**Timing.** 

\[ T = 0.0115 (L_1 + 1) \text{ ms} + \text{I/O}. \]

**Note.** If the system is equipped with the print storage special feature, the read operation can be performed as soon as the data is received in print storage.

**Address Registers After Operation.**

- I-Add. Reg: NSI
- B-Add. Reg: 081

**Example.** Print a line, and read a card Figure 72.

---

**Write, Read, and Branch**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>WR</td>
<td>2</td>
<td>xxx</td>
</tr>
</tbody>
</table>

**Function.** This is the same as the WRITE AND READ instruction, except that the next instruction is taken from the I-address.

**Word Marks.** Word marks are not affected.

**Timing.** 

\[ T = 0.0115 (L_1 + 1) \text{ ms} + \text{I/O}. \]

**Address Registers After Operation.**

- I-Add. Reg: NSI
- B-Add. Reg: 081

**Example.** Print a line, read a card, and branch to CALC2 (0759), Figure 73.
Read, Punch, and Branch

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>RP</td>
<td></td>
<td>(\frac{5}{3})</td>
</tr>
</tbody>
</table>

Function. This is the same as the READ AND PUNCH instruction except that the next instruction is located at the I-address.

Word Marks. Word marks are not affected.

Timing. \(T = .0115 \ (L_1 + 1) \ ms + I/O\).

Address Registers After Operation.

- **I-Add. Reg.**: NSI
- **A-Add. Reg.**: Ap
- **B-Add. Reg.**: 181 or 081

(See note under Read and Punch.)

Example. Read, punch, and branch to WORK5 (0596), Figure 75.

Write and Punch, and Branch

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP</td>
<td></td>
</tr>
</tbody>
</table>

Function. This code combines the WRITE A LINE (2) and PUNCH A CARD (4) instructions. The printer always operates first, but the signal to start the punch is automatically given before the end of the print operation, so that actual card punching starts soon after the print cycle is complete.

Word Marks. Word marks are not affected.

Timing. \(T = .0115 \ (L_1 + 1) \ ms + I/O\).

Note. If the print storage special feature is installed, the signal to start the punch is received shortly after the transfer of data to the print storage area.
Address Registers After Operation.

Example. Write a line, and punch a card (Figure 76).

<table>
<thead>
<tr>
<th>SP5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
</tr>
<tr>
<td>185</td>
</tr>
</tbody>
</table>

Assembled Instruction: A

Figure 76. Print and Punch

Write, Punch, and Branch

Instruction Format.

Mnemonic: WRP
Op Code: 1
I-address: XXX

Function. This is the same as WRITE AND PUNCH, except that the program branches automatically to the location in the I-address after punching is completed.

Word Marks. Word marks are not affected.

Timing. T = 0.0115 (L1 + 1) ms + I/O.

Write, Read, and Punch, and Branch

Instruction Format.

Mnemonic: WRP
Op Code: 1
I-address: XXX

Function. This is the same as WRITE, READ, AND PUNCH except that the next instruction, after punching is completed, is taken from the I-address.

Word Marks. Word marks are not affected.

Timing. T = 0.0115 (L1 + 1) ms + I/O.

Example. Write, read, and punch (Figure 78).
Example. Branch to ROUT4 (0980) after a line is printed, a card is read, and a card is punched (Figure 79).

Assembled Instruction:  

```
980
```

Figure 79. Print, Read, Punch, and Branch

Document Control Instructions

Select Stacker

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SS</td>
<td>K</td>
<td>x</td>
</tr>
</tbody>
</table>

**Function.** This instruction causes the card that was just read or punched to be selected into the stacker pocket specified by the d-character:

<table>
<thead>
<tr>
<th>d-character</th>
<th>Feed</th>
<th>Stacker Pocket</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>READ</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>READ</td>
<td>8/2</td>
</tr>
<tr>
<td>4</td>
<td>PUNCH</td>
<td>4</td>
</tr>
<tr>
<td>8</td>
<td>PUNCH</td>
<td>8/2</td>
</tr>
</tbody>
</table>

**Read Select.** A SELECT STACKER instruction must be given during the first 10 ms after actual card reading is completed. Otherwise the command is ineffective. After a card is read, it continues to the stackers without stopping. Therefore, if no SELECT STACKER signal is received within the next 10 ms the card stacks in the NORMAL stacker (NR). Read select instructions cannot be used following RP and WRP instructions because the select signal cannot be given within the prescribed 10 milliseconds.

**Punch Select.** The SELECT STACKER instruction is effective if given at any time between two PUNCH A CARD instructions. However, if a punch check occurs, the error card is directed to the NORMAL (NP) stacker.

**Word Marks.** Word marks are not affected.

**Timing.** $T = .0115 (L + 1)$ ms.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>dbb</td>
<td>dbb</td>
</tr>
</tbody>
</table>

Example. Select the last card punched, enter it in pocket 4, and branch to ROUT5 (0950) for the next instruction (Figure 81).

Assembled Instruction:  

```
950
```

Figure 81. Select Stacker and Branch
Control Carriage

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>CC</td>
<td>E</td>
<td>x</td>
</tr>
</tbody>
</table>

**Function.** This instruction causes the carriage to move as specified by the d-character. A digit causes an immediate skip to a specified channel in the carriage tape. An alphabetic character with a 12-zone causes a skip to a specified channel after the next line is printed. An alphabetic character with an 11-zone causes an immediate space. A zero-zone character causes a space after the next line is printed. The table (Figure 82) shows the function of the d-character. If the carriage is in motion when a CONTROL CARRIAGE instruction is given, the program stops until the carriage comes to rest. At this point, the new carriage action is initiated, and then the program advances to the next instruction in storage.

**Word Marks.** Word marks are not affected.

**Timing.** \( T = 0.0115 (L_d + 1) \) ms plus remaining form-movement time, if carriage is moving when this instruction is given. The form-movement time is determined by the number of spaces the form moves. Allow 20 ms for the first space, plus 5 ms for each additional space.

<table>
<thead>
<tr>
<th>d</th>
<th>Immediate skip to</th>
<th>d</th>
<th>Skip after print to</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Channel 1</td>
<td>A</td>
<td>Channel 1</td>
</tr>
<tr>
<td>2</td>
<td>Channel 2</td>
<td>B</td>
<td>Channel 2</td>
</tr>
<tr>
<td>3</td>
<td>Channel 3</td>
<td>C</td>
<td>Channel 3</td>
</tr>
<tr>
<td>4</td>
<td>Channel 4</td>
<td>D</td>
<td>Channel 4</td>
</tr>
<tr>
<td>5</td>
<td>Channel 5</td>
<td>E</td>
<td>Channel 5</td>
</tr>
<tr>
<td>6</td>
<td>Channel 6</td>
<td>F</td>
<td>Channel 6</td>
</tr>
<tr>
<td>7</td>
<td>Channel 7</td>
<td>G</td>
<td>Channel 7</td>
</tr>
<tr>
<td>8</td>
<td>Channel 8</td>
<td>H</td>
<td>Channel 8</td>
</tr>
<tr>
<td>9</td>
<td>Channel 9</td>
<td>I</td>
<td>Channel 9</td>
</tr>
<tr>
<td>D</td>
<td>Channel 10</td>
<td>@</td>
<td>Channel 12</td>
</tr>
<tr>
<td>@</td>
<td>Channel 11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp;</td>
<td>Channel 12</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>d</th>
<th>Immediate space</th>
<th>d</th>
<th>After print-space</th>
</tr>
</thead>
<tbody>
<tr>
<td>J</td>
<td>1 space</td>
<td>J</td>
<td>1 space</td>
</tr>
<tr>
<td>K</td>
<td>2 spaces</td>
<td>S</td>
<td>2 spaces</td>
</tr>
<tr>
<td>L</td>
<td>3 spaces</td>
<td>T</td>
<td>3 spaces</td>
</tr>
</tbody>
</table>

Figure 82. d-Characters for Forms Control

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>?</td>
<td>dbb</td>
</tr>
</tbody>
</table>

**Example.** Skip to channel 1 after print (Figure 83).

**Control Carriage and Branch**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>CC</td>
<td>E</td>
<td>xxx</td>
</tr>
<tr>
<td></td>
<td>A</td>
<td>CCB</td>
<td></td>
</tr>
</tbody>
</table>

**Function.** This format of the CONTROL CARRIAGE instruction causes a program branch to the location specified by the I-address for the next instruction after interpretation of the d-character.

**Word Marks.** Word marks are not affected.

**Timing.** \( T = 0.0115 (L_d + 1) \) ms plus remaining form-movement time, if carriage is moving when this instruction is given. The form-movement time is determined by the number of spaces the form moves. Allow 20 ms for the first space, plus 5 ms for each additional space.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>BI</td>
<td>dbb</td>
</tr>
</tbody>
</table>

**Example.** Skip to channel 1 immediately, and branch to START3 (498) for the next instruction (Figure 84).

Figure 84. Forms Control and Branch
Note: Care should be taken when punching channels 9 and 12 of the carriage control tape on a system with the print storage feature to insure that these punches will not be ignored by line spacing (single, double, or triple). This is necessary because the next tape channel sensed resets the channel 9 and the channel 12 indicators. The next hole punched in the carriage tape following a 9- or 12-punch should be at least $1 + \frac{S}{8}$ spaces from the 9- or 12-punch ($S$ equals number of lines being spaced).

When an immediate skip or immediate space instruction is used, an additional space caused by the automatic carriage space taken after printing results. When a skip after print or space after print instruction is used, the automatic space is not taken.
An important feature in economical processing of business data is compact storage. A magnetic tape reel (10½ inches in diameter) contains 2,400 feet - enough tape to record as many as 14,000,000 characters. The tape itself is a ribbon, ½-inch wide, coated with a magnetic oxide material. Tape reels can be easily stored or transported from one installation to another. Also, magnetic tape records have gained wide acceptance as legal documents.

**Data Flow**

The IBM 1401 system accepts data from punched cards, magnetic tape, or both. Information is read into the system, rearranged, calculated, and edited by the stored program. Output can be in the form of punched cards, magnetic tape, or printed reports.

The 1401 Model D is not equipped with an IBM 1402 Card Read-Punch (Figure 85).

All data passes through 1401 core storage, where a series of validity checks insure accuracy and reliability.

**Magnetic-Tape Characteristics**

The magnetic-tape recording code used with the IBM 1401, is the same binary-coded-decimal code used with other IBM data processing systems. This compatibility permits interchanging tapes between installations that employ different IBM systems.

Data is recorded in a seven-bit code, in seven parallel channels along the tape. Figure 86 shows tape characters and their corresponding codes.

Records are separated from each other by approximately ¾ inch of blank (unrecorded) tape, called an inter-record gap.

Each tape character is composed of an even number of magnetic bits. A check bit (labeled C in Figure 87) is written if the number of bits in the other six positions is odd. An even-parity check on each character insures accuracy for tape-read and tape-write operations.

In addition to this vertical parity check, a horizontal check (HC in Figure 87) is made on each record. The bits in each horizontal row are automatically counted
when the record is written, and a bit (similar in function to the vertical-check bit) is written at the end of each odd-count row. The vertical combination of these horizontal-check bits makes up the horizontal-check character. Thus, the coding of this character can change from record to record. When the tape is read, the same automatic count is made, but now each row in the complete record should have an even number of bits, or an error exists. The horizontal-check character is used for checking only, and is never read into 1401 core storage.

To process even-bit parity tapes a U is used in the tens position of the A-address (%Ux). Odd-bit parity tapes can be processed by the 1401 by using a B in the tens position of the A-address (%Bx). The column binary special feature is not required to perform this operation.

When processing even-bit parity tapes, the A-bit character may have its bit structure changed to that of the blank character (C-bit) (Figure 88). That is, the A-bit is written on even-parity tape as CA bits; thus, when the tape is read, the character is translated as a C-bit only. Therefore, in order to retain the A-bit character, it should be written on odd-parity tape. The A-bit is then written on tape as an A-bit, and when read from tape, it returns to core-storage as an A-bit.

**Processing Magnetic Tape for Use with the IBM 7070 Data Processing System**

Because the IBM 7070 Data Processing System is a numerical machine (alphabetic characters are represented by two numbers), a special delta (\(\Delta\)) character is used to signal a mode change (from alphabetic to numerical and vice versa in a tape record).

When IBM 7070 tapes containing the delta character are read into 1401 core storage, the delta character is transferred just as any other tape character. Records can also be written from 1401 core storage with the delta character for subsequent use by a 7070. However, there is no automatic insertion of a delta character in 1401 tape-write operations on a mode change. The deltas must be inserted between numerical and alphabetic fields by the 1401 program if the records are not in core storage as a result of reading a 7070 tape.

The IBM 7070 has a tape file searching feature which permits the user to space tape forward or backward over a specified number of tape segments. Each tape segment is defined by a special character called a tape-segment mark (1401 character A8421). Although the 1401 does not have this automatic feature, the tape segment can be detected in 1401 core storage by using a COMPARE instruction. The 1401 IOCS treats a tape segment mark as a noise record, therefore, it cannot be used with programs using IOCS.

The 1401 word mark, which is written as a word separator character when 1401 WRITE TAPE WITH WORD MARKS instructions are executed, has no special significance to the 7070. Thus, WRITE TAPE instructions should be used in programs that prepare tapes to be used by the 7070. With this operation, word separators are not written on the tape.

<table>
<thead>
<tr>
<th>WRITE</th>
<th>READ</th>
<th>WRITE</th>
<th>READ</th>
</tr>
</thead>
<tbody>
<tr>
<td>1401 Core Storage</td>
<td>Even Parity Tape (%Ux)</td>
<td>1401 Core Storage</td>
<td>Odd Parity Tape (%Bx)</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>A</td>
<td>C</td>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>A</td>
<td>A</td>
<td>A</td>
<td>A</td>
</tr>
</tbody>
</table>

Figure 88. C- and A-Bit Tape Characteristics
**Tape Units**

**IBM 729 Magnetic Tape Unit**
The IBM 1401 can use either of three models of the IBM 729 Magnetic Tape Unit (Model II, Model IV, and Model V). The tape system can accommodate as many as six IBM 729 tape units (Figure 89) which are attached to the tape adapter. The IBM 729 dual-density tape switch makes it possible for the IBM 729 tape unit to operate with magnetic tapes recorded at either 200, 556, or 800 characters-per-inch.

**IBM 7330 Magnetic Tape Unit**
The IBM 7330 Magnetic Tape Unit (Figure 90) provides the advantages of magnetic tape to IBM 1401 system users whose operations do not demand the high-speed operation of the IBM 729 Magnetic Tape Unit. The 7330 tape unit has dual density. That is, it can read or write tape at character rates of 200 or 556 characters-per-inch. The dual-density feature, the same length inter-record gap (¾ inch), and the same BCD recording make the tapes prepared by the IBM 7330 Magnetic Tape Unit compatible with tapes prepared by the IBM 727 and 729 Magnetic Tape Units and the IBM 7701 and 7702 Magnetic Tape Transmission Terminals. This means magnetic tapes, prepared in an installation using the IBM 7330, can be read in another installation using IBM 729 tape units.

The 7330 tape unit is housed in two SMS cubes. The tape reels, read-write heads, and tape circuits are in the upper cube. The lower cube contains the tape unit entry and exit connectors.

The processing speed is the primary difference between the IBM 729 and IBM 7330 tape units. Figure 91 shows a comparison of the four tape units available for use with the IBM 1401 Data Processing System. Where speed of tape processing is not a prime concern, but the volume of repetitive information is too cumbersome for efficient unit-record processing, the 7330 tape unit can be used to advantage.

Six 7330 tape units can be accommodated by the tape control unit. The IBM 1401 stored-program addresses for these units is %U1 through %U6.

**Tape Intermix**
This special feature makes it possible to have 729 II, IV, V and 7330 magnetic tape units in any combination, connected to the same 1401 system at one time. Thus, the tape units best suited for the job to be done can be used. Some jobs may require a pair of high-speed tape units (729 IV's) while the balance of the operation can be performed on a slower unit (7330).

Now, any combination of tape units can be provided to best suit the need of the application. The intermix
feature also allows a 1401 with IBM 7330 Magnetic
Tape Units to borrow IBM 729 Magnetic Tape Units
(IV or V) from another system (such as an IBM 7090
Data Processing System) when a high-speed run is
required. The intermix feature greatly increases the
flexibility of the IBM 1401 Data Processing System to
meet the requirements of any application.

Tape Checking
The IBM 729 and 7330 tape units maintain a high
reliability in originating data storage on tape.

The two-gap head, makes it possible to verify automa­
tically the validity of recorded information at the
time it is written. The relative positions of the read
and write gaps (Figure 92) are such that a character
recorded by the write gap passes the corresponding
read gap almost instantaneously. Thus, as each char­
acter of a record is written, it is read, and a parity
check is applied.

If an error is detected, the stored program receives
a signal, and corrective action can be taken following
the completion of the write operation.

A part of the tape read circuits is used for read
checking purposes for both a read and a write opera­
tion. There are two, seven-position registers, each of
which is sensitized to accept a specified minimum
tape output signal.

Figure 93 shows the sensitivity levels, and the rela­
tive strength of pulses that are acceptable or not
acceptable in read and write operations. The high
and low registers accept tape signals equal to, or greater
than, the percentages shown in the figure.

In a write operation, the high register is checked
for correct parity, to insure adequate signal strength,
and then compared, bit by bit, to the low register. If
either of these checks detects an error, the tape error
indicator is turned on.

During a read or write operation, the output of
each of the high-register bit positions (low register, if
the high register is in error) is sent to the longitudinal-
redundancy check register (LRCR). This insures that
an even number of bits, including check bit, are read
in each of the seven tracks on tape. An LRCR error
turns on the tape error indicator.

During a tape read operation, the high register is
checked for correct parity. If it is correct, its contents
are transferred to the read-write register and to the
LRCR. If parity is not correct, the contents of the
low register are unconditionally transferred to the
read-write register and to the LRCR. In this case, the
error in the high register does not turn the tape error
indicator on. Thus, a bit which has a weak signal can
be read from tape.

The read-write register is also checked for correct
parity. If incorrect parity is found, the tape error indi­
cator is turned on.

If correct, the contents of the read-write register is
sent to core storage. If a tape error is suspected, the
tape unit can be backspaced by programming, and
the record re-read. If the error persists, the operator
can intervene, or the program can branch to an error
routine.

Dust or damage to the magnetic tape is the most
frequent cause of errors detected during write opera­
tions. Such imperfections are usually isolated, so, in
order to skip the defective section, the 1401 has been
provided with an instruction that causes the tape to
space forward approximately 3.5 inches when the next
write operation is initiated. While the tape is passed,
this short length is erased so that extraneous data are
not sensed when the tape is read. The tape-write opera­
tion continues after the skip is completed.

Another feature for file protection is a plastic ring
(Figure 94) that fits into a groove in the tape reel.
The tape can be read with or without this file-protection
ring in place, but no writing can be done
without it.

The file-protection ring should be removed from a
tape reel when writing is completed, thus protecting
tape records from accidental writing.

<table>
<thead>
<tr>
<th>OPERATING CHARACTERISTICS</th>
<th>729H1</th>
<th>729H1V</th>
<th>729V</th>
<th>7230</th>
</tr>
</thead>
<tbody>
<tr>
<td>Density, Characters Per Inch</td>
<td>200 or 200</td>
<td>200 or 556</td>
<td>200 or 800</td>
<td>200 556</td>
</tr>
<tr>
<td>Tape Speed, Inches Per Second</td>
<td>75</td>
<td>112.5</td>
<td>75</td>
<td>36</td>
</tr>
<tr>
<td>Inter Record Gap Size, Inches</td>
<td>¾</td>
<td>¾</td>
<td>¾</td>
<td>¼</td>
</tr>
<tr>
<td>Character Rate, Characters Per Second</td>
<td>15,000 or 41,667</td>
<td>22,500 or 62,500</td>
<td>15,000 or 41,667</td>
<td>7,200 or 20,016</td>
</tr>
<tr>
<td>High Speed Rewind, Minutes</td>
<td>1.2</td>
<td>.9</td>
<td>1.2</td>
<td>2.2</td>
</tr>
<tr>
<td>Regular Rewind, Inches Per Second</td>
<td>75</td>
<td>112.5</td>
<td>75</td>
<td>36</td>
</tr>
</tbody>
</table>

Figure 91. Tape-Unit Characteristics

Figure 92. Read and Write Gap
Magnetic-Tape Operations

IBM 729 Magnetic Tape Unit
If the 729 is in write status, to change to read status the program must backspace over those records that are to be read. The tape unit must then be changed back to write status (W₁, W₂, W₃, B₂, R₂, R₃, W₄, W ... ). This results in unchecked tape on the first record written after backspace.

The 729 cannot be switched directly from write to read status (W₁, W₂, R₃, R₄).

If the 729 is in read status, the tape unit can be changed directly from read to write status (R₁, R₂, W₃, W₄).

IBM 7330 Magnetic Tape Unit
If the 7330 is in write status, to change to read status the program must backspace over those records that are to be read. The tape unit must then be changed back to write status (W₁, W₂, W₃, B₂, R₂, R₃, W₄, W ... ). This results in unchecked tape on the first record written after backspace.

The 7330 cannot be switched directly from write to read status (W₁, W₂, R₃, R₄).

If the 7330 is in read status, to change to write status the program must backspace over the last record read and then rewrite that record. The 7330 then continues in write status (R₁, R₂, B₂, W₂, W₃, W ... ).

The 7330 cannot be switched directly from read to write status (R₁, R₂, W₃, W₄).

Figure 95 is a summary of 1401 magnetic-tape operations.

For detailed information concerning magnetic tape and IBM magnetic tape units, refer to the IBM Reference Manual, Magnetic Tape Units, form A22-6589.

Tape Instructions

Read Tape

Instruction Format.

Mnemonic  Op Code  A-address  B-address  d-character
SPS MU M %UX xxx R
A  RT

Function. The tape unit specified in the A-address is started. The d-character specifies a tape read operation. The B-address specifies the high-order position of the tape read-in area of storage. The machine begins to read magnetic tape, and continues to read until either an inter-record gap in the tape record...
or a group-mark with a word-mark in core storage is sensed. The inter-record gap indicates the end of the tape record and a group-mark (code CBA 8421) is inserted in 1401 core storage at this point.

If the group-mark with a word-mark occurs before the inter-record gap is sensed, the transfer of data from tape stops but tape movement continues until the inter-record gap is sensed.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_1 + 1) \) ms + \( T_M \). Time varies for type of tape unit and tape density used (see Timing section).

Address Registers After Operation.


Example. Read the record from tape unit 2 (labeled 2) to 1401 core storage. The high-order tape-record character is moved to INPUT (0419), the next character is moved to the next higher position (0420), etc., until transfer of data is stopped by an inter-record gap in the tape record, or a group-mark with a word-mark in 1401 core storage (Figure 96).

---

### Table: Summary of IBM 1401 Magnetic-Tape Operations

<table>
<thead>
<tr>
<th>STATUS</th>
<th>OPERATION</th>
<th>729</th>
<th>REMARKS</th>
<th>730</th>
<th>REMARKS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Read</td>
<td>R1 R2 W1 W2 W3 W4-----</td>
<td>Yes</td>
<td>Updating tape label</td>
<td>Yes</td>
<td>Updating tape label</td>
</tr>
<tr>
<td></td>
<td>R1 R2 Skip W3 W4-----</td>
<td>Yes</td>
<td>Results in unchecked tape</td>
<td>Yes</td>
<td>Results in unchecked tape</td>
</tr>
<tr>
<td></td>
<td>R1 R2 W3 W4-----</td>
<td>Yes</td>
<td>Unchecked tape in record W3</td>
<td>No</td>
<td>Write head is over first part of next record (W4)</td>
</tr>
<tr>
<td></td>
<td>R1 R2 W3 R3</td>
<td>No</td>
<td>Changing from W to R causes bits in the inter-record gap</td>
<td>No</td>
<td>Changing from W to R causes bits in the inter-record gap</td>
</tr>
<tr>
<td>Write</td>
<td>W2 R3 R4 W5 W6-----</td>
<td>Yes</td>
<td>Unchecked tape on record W2</td>
<td>Yes</td>
<td>Unchecked tape on record W2</td>
</tr>
<tr>
<td></td>
<td>W2 W3 Blank Area R4 R5---</td>
<td>Not Recom.</td>
<td>Results in bits in the inter-record gap and possible error on R5</td>
<td>Not Recom.</td>
<td>Results in bits in the inter-record gap and possible error on R5</td>
</tr>
<tr>
<td></td>
<td>W2 W3 R3 R4</td>
<td>No</td>
<td>Changing from W to R causes bits in the inter-record gap</td>
<td>No</td>
<td>Changing from W to R causes bits in the inter-record gap</td>
</tr>
<tr>
<td></td>
<td>W2 R3 R4</td>
<td>No</td>
<td>Changing from W to R causes bits in the inter-record gap</td>
<td>No</td>
<td>Changing from W to R causes bits in the inter-record gap</td>
</tr>
<tr>
<td>Rewind</td>
<td>W2 R2 W3 R4</td>
<td>Yes</td>
<td>Causes extraneous bits after W5 (label)</td>
<td>Yes</td>
<td>Causes extraneous bits after W5 (label)</td>
</tr>
</tbody>
</table>

R — read  
W — write  
B — backspace  
Rw — rewind

Figure 95. Summary of IBM 1401 Magnetic-Tape Operations
a word-mark in 1401 core storage stops the operation. The group-mark with a word-mark causes an inter-record gap to be created.

**Word Marks.** Word marks are not affected.

**Timing.** \( T = 0.0115 \ (L_1 + 1) \) ms + \( T_M \).

**Note.** If a group-mark with a word-mark is the first character of B-address, the tape-adapter unit and the tape unit will hang up. The condition can be reset by pressing the start-reset key if the tape-select switch (on the 1401) is in the \( N \) (normal) position.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>%4x</td>
<td>Group-mark + 1</td>
</tr>
</tbody>
</table>

**Example.** Transfer the contents of core storage to tape unit 3 (labeled 3), starting at the location labeled OUTPUT (0525) and ending at the location of the first group-mark with a word-mark (Figure 97).

**Read Tape with Word Marks**

**Instruction Format.**

```
Mnemonic Op Code A-address B-address d-character
SPS LU L %UX xxx R
A RTW
```

**Function.** This is the same as the **read tape** operation, except that word separator characters on magnetic tape (written during **write tape with word marks** instruction) are translated to word marks during transmission to 1401 core storage.

**Word Marks.** A word-separator character read from tape causes a word mark to be associated with the next tape character transferred to 1401 core storage (Figure 98).

**Timing.** \( T = 0.0115 \ (L_1 + 1) \) ms + \( T_M \).
**Word Marks.** A word mark associated with any position in 1401 storage causes a word-separator character (A841) to be written automatically on tape, one character ahead of that which contained the word mark. Thus, word marks are translated to word-separator characters for tape storage (Figure 100).

<table>
<thead>
<tr>
<th>1401 Core Storage Locations</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1401 Core Storage Code</td>
<td>C82</td>
<td>41W</td>
<td>4</td>
</tr>
<tr>
<td>1401 Meaning</td>
<td>0</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>Tape Positions</td>
<td>A</td>
<td>B</td>
<td>C</td>
</tr>
<tr>
<td>Tape Code</td>
<td>82</td>
<td>A841</td>
<td>41</td>
</tr>
</tbody>
</table>

**Figure 100. Word-Separator Character (Tape Write)**

**Timing.** \( T = 0.0115 \times (L_1 + 1) \times ms + T_M \).

**Note.** Load operations must be used when word marks are needed for identification in tape storage. If tape is written by a WRITE TAPE WITH WORD MARKS instruction, it must be read back by a READ TAPE WITH WORD MARKS instruction to insure proper translation between the tape and 1401 core storage.

**Address Registers After Operation.**

|-------------|-------------|-------------|------------------------|

**Example.** Transfer the contents of core storage to tape unit 6 (labeled 6). Insert a word-separator character where word marks exist in core storage, beginning at OUTREC (0696) and ending at the first group-mark with a word-mark in 1401 core storage (Figure 101).

**Backspace Tape Record**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>CU</td>
<td>%Ux</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>BSP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Function.** The tape unit specified in the A-address backspaces over one tape record. The first inter-record gap (IRG) encountered stops the backspace operation specified by the d-character, B.

**Word Marks.** Word marks are affected.

**Timing.** \( T = .0115 \times (L_1 + 1) \times ms + T_M \).

**Address Registers After Operation.**

|-------------|-------------|-------------|---------|

**Example.** Backspace tape unit 4 (labeled 4) until an IRG is sensed (Figure 102).

**Write Tape Mark**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>CU</td>
<td>%Ux</td>
<td>M</td>
</tr>
<tr>
<td>A</td>
<td>WTM</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Function.** This instruction causes a tape mark character (8421) to be recorded immediately following the last record on tape. When the tape mark is read back from a tape, the end-of-reel indicator is turned on. This signals the 1401 program that the end of a major group of records has been reached (end-of-file) or the end of utilized tape has been reached.

**Word Marks.** Word marks are not affected.

**Timing.** \( T = 0.0115 \times (L_1 + 1) \times ms + T_M \).

**Address Registers After Operation.**

|-------------|-------------|-------------|---------|

**Example.** Insert a tape mark on the tape in tape unit 3 (labeled 3), Figure 103.
**Skip and Blank Tape**

*Instruction Format.*

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>C</td>
<td>U</td>
<td>%Ux</td>
</tr>
<tr>
<td>A</td>
<td>SKP</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Function.* The tape unit, designated by the A-address, spaces forward and erases approximately 3.5 inches of tape. The actual skip occurs when the next WRITE TAPE instruction is given. This instruction makes it possible to bypass defective tape areas.

*Word Marks.* Word marks are not affected.

*Timing.* $T = 0.0115 \ (L_1 + 1)$ ms. Processing can continue immediately after this operation. However, 40.5 ms for IBM 729 II, 27 ms for IBM 729 IV and 108 ms for an IBM 7330 must be added to the next WRITE TAPE instruction time.

*Note.* The SKIP and BLANK TAPE instruction should be given immediately preceding a WRITE TAPE instruction for the tape unit specified by both instructions.

*Address Registers After Operation.*

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>%4x</td>
<td>dbb</td>
</tr>
</tbody>
</table>

*Example.* Erase tape on tape unit 5 (labeled 5) when the next write operation is ordered for that unit (Figure 104).

---

**Rewind Tape**

*Instruction Format.*

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>CU</td>
<td>U</td>
<td>%Ux</td>
</tr>
<tr>
<td>A</td>
<td>RWD</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Function.* This instruction is usually given after an end-of-reel condition, and causes the selected tape unit to rewind its tape. When the operation is initiated, the tape unit is, in effect, disconnected from the system.

*Word Marks.* Word marks are not affected.

*Timing.* $T = 0.0115 \ (L_1 + 1)$ ms. Rewind time is 1.2 minutes per 2,400-foot reel for the IBM 729 II, 0.9 minute for the IBM 729 IV, and 13.3 minutes for the IBM 7330, but it is not calculated with program time. Processing can continue approximately 10 ms after this instruction is interpreted.

*Address Registers After Operation.*

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>%4x</td>
<td>dbb</td>
</tr>
</tbody>
</table>

*Example.* Rewind the tape in tape unit 1 (labeled 1), Figure 105.

---

**Rewind Tape and Unload**

*Instruction Format.*

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>CU</td>
<td>U</td>
<td>%Ux</td>
</tr>
<tr>
<td>A</td>
<td>RWU</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Function.* This instruction causes the tape unit specified in the A-address to rewind its tape. At the end of the rewind, the tape is out of the vacuum columns, and the reading mechanism is disengaged. The unit is effectively disconnected from the system, and is not available again until the operator restores it to a ready status.

*Word Marks.* Word marks are not affected.

*Timing.* $T = 0.0115 \ (L_1 + 1)$ ms. Rewind time is 1.2 minutes per 2,400-foot reel for the IBM 729 II, 0.9 minute for the IBM 729 IV, and 2.2 minutes for the IBM 7330, but it is not calculated with program time. Processing can continue approximately 10 ms after this instruction is interpreted by a system using IBM 729 tape units, or 4.5 ms in a system using IBM 7330 tape units.

*Address Registers After Operation.*

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>%4x</td>
<td>dbb</td>
</tr>
</tbody>
</table>

Figure 104. Erase Forward

Figure 105. Rewind Tape
Example. Rewind the tape in tape unit 5 (labeled 5), and make it unavailable to the stored program, at the completion of the rewind operation (Figure 106).

Branch if End of Reel

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>B</td>
<td>xxx</td>
<td>K</td>
</tr>
<tr>
<td>A</td>
<td>BER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function. The end-of-reel indicator (EOR) turns on in the IBM 1401 Processing Unit if a tape mark is read by the 1401, or if a reflective spot is sensed during a write tape operation. This instruction tests the indicator and causes an automatic branch to the I-address if the indicator is ON. If it is OFF, the program continues with the next sequential instruction.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 \) \((L_1 + 1)\) ms.

Notes. The `BRANCH IF TAPE ERROR` instruction must be given after a tape read or write operation, because any tape operation on any tape unit causes the indicator to turn off. The information read from tape always enters core storage with odd-bit parity. Therefore, the tape read-in area need not be cleared after a tape read error has occurred. See Tape Select Switch in Operating Features section.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NS1</td>
<td>BI</td>
<td>dbb</td>
</tr>
</tbody>
</table>

Example. Test the tape unit just used for an end-of-reel condition. If there is an EOR condition, branch to TAPER1 (0685) for the next instruction. If no EOR exists, continue the program with the next sequential instruction (Figure 107).

Branch if Tape Error

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>B</td>
<td>xxx</td>
<td>L</td>
</tr>
<tr>
<td>A</td>
<td>BER</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function. If an error occurs in transmission between a tape unit and the 1401 during a tape read or tape write operation, an error indicator turns on in the 1401 and the tape light on the console glows red. This instruction tests the error indicator, and branches to the I-address for the next instruction if the indicator is ON. If it is OFF, the program continues with the next sequential instruction.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 \) \((L_1 + 1)\) ms.

Notes. The `BRANCH IF TAPE ERROR` instruction must be given after a tape read or write operation, because any tape operation on any tape unit causes the indicator to turn off. The information read from tape always enters core storage with odd-bit parity. Therefore, the tape read-in area need not be cleared after a tape read error has occurred. See Tape Select Switch in Operating Features section.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NS1</td>
<td>BI</td>
<td>dbb</td>
</tr>
</tbody>
</table>

Example. Read a tape record from the tape unit 3 (labeled 3) into core-storage area labeled TAPEIN (0629) and test for a tape error. If there is an error, branch to TAPER2 (0539) for the next instruction. If there is no error, continue processing with the next sequential instruction (Figure 108).

Figure 106. Rewind and Unload

Figure 107. Test for End of Reel

Figure 108. Test for Transmission Error
The IBM 1405 Disk Storage unit (Figure 109) combines the data processing capabilities of the IBM 1401 Data Processing System with the advantages and facility of large-capacity random access storage. The combination of the 1401 and 1405 provides an efficient and economical in-line data processing system.

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

Records in the IBM 1405 Disk Storage unit are stored on the faces of magnetic disks. The 1405 Model 1 has a storage capacity of 10 million alphameric characters of information on 25 disks. Model 2 has a storage capacity of 20 million alphameric characters on 50 disks.

The Model 2 disk storage unit is divided into two modules. Each module contains 50,000 records on 25 disks. Each disk face has 200 tracks which are subdivided into 5 sectors. Each sector can contain a 200-character record (Figure 110).

**Access Arms**

The disk storage unit can have two access arms. One is standard and the other is available as a special feature. The fork-shaped access arm has two read-write heads that read and record data in the unit. One read-write head is for the top disk-surface; the other is for the bottom disk-surface. During a seek operation, the access mechanism moves vertically to seek a disk, and horizontally to seek a track (Figure 111).

To execute a read, write, or write check command, the access arm must previously have been directed to the proper track location by a seek command.

**Speed**

The disks rotate on a vertical shaft at the rate of 1200 revolutions per minute. Data is read or recorded at the rate of 22,500 characters per second. Access time is 100 milliseconds, minimum, and 800 milliseconds for Model 2 (700 milliseconds for Model 1), maximum. Access time is the time required to locate a particular disk track. Read, write, and write check operations can be performed on a disk record without having to reseek if no other seek operation intervenes.

**Coding**

The magnetic-disk recording code is the same binary-coded-decimal used in the IBM 1401 Processing Unit. Data is recorded in seven-bit codes (serially by bit) on the disk.

To insure the accuracy of recorded data, a parity check is made while data is transferred from disk storage to core storage during a read or write operation. A programmed verify disk check is made to compare data written on the disk with the data in core storage.

**Disk Storage Addressing**

Each sector has an indelible 7-digit record address preceding the 200-character record area (Figure 112). The disk records are arranged sequentially in ascending order from bottom to top of the disk storage unit. The record address of the first record in the outside track of the bottom disk is x0000000. The address of the last record in the inside track of the top disk is...
The magnetic-disk recording code is the same binary-coded-decimal used in the IBM 1401 Processing Unit. Data is recorded serially by bit in 7-bit code (CBA 8421) when operating in the Move mode, and in 8-bit code (CBA 8421 M) when operating in the Load mode. The eighth bit, which is used when operating in the Load mode is for a possible word mark (M) with the character being written on, or read from, the disk record.
Seven Position Indelible Address

Figure 110. Disk Storage

Figure 111. Access Arm

<table>
<thead>
<tr>
<th>IN 1401 CORE STORAGE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Arm</td>
</tr>
<tr>
<td>X</td>
</tr>
</tbody>
</table>

ON 1405 DISK STORAGE

<table>
<thead>
<tr>
<th>Access Arm</th>
<th>Unit</th>
<th>Disk Face</th>
<th>Track</th>
<th>Sector</th>
<th>Constant</th>
<th>GM/WM</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>X</td>
<td>XX</td>
<td>XX</td>
<td>X</td>
<td>X</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 112. Record Address Format

x0999990. The "x" in the record address refers to the access arm to be used and is not part of the 7-digit indelible address.

Instruction Format

Mnemonic | Op Code | A-address | B-address | d-character
X         | X       | %Fx       | XXX       | X

OP CODE

This is always a single character that defines the basic operation to be performed.

A-ADDRESS

%Fx always appears in the (A) portion of a 1401 disk storage instruction. The %F signals that the disk unit is to be selected and the x represents the digit used to perform various operations:
### X-Position

#### Operation

<table>
<thead>
<tr>
<th>X-Position</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Seek a disk record.</td>
</tr>
<tr>
<td>1</td>
<td>Single record — Reading or writing of 200 characters is stopped when a group-mark with a word-mark, or the end of a sector, is sensed. If a group-mark with a word-mark is sensed before completing the reading of the sector of the track, reading stops and the wrong-length record indicator turns on.</td>
</tr>
<tr>
<td>2</td>
<td>Full track — An entire track is read or written (5 sectors of 200 characters each). Reading or writing of the full track begins at the sector addressed and continues for four additional sectors. If a group-mark with a word-mark is sensed before completing the reading of the last sector of the track, reading stops and the wrong-length record indicator turns on.</td>
</tr>
<tr>
<td>3</td>
<td>Write Check — Data written on a disk in a preceding write operation is read from the disk and compared, character-by-character, with the data in core storage. A write check can be given following a single record or full-track operation.</td>
</tr>
</tbody>
</table>

#### B-ADDRESS

The B-address specifies the high-order position in core storage of the eight-digit record address. The record address must be followed by a group-mark with a word-mark and the area of core storage from which data is to be read into or out of by the disk storage unit. The data area must be followed by a group-mark with a word-mark.

#### d-CHARACTER

The d-character is used to specify the operation to be performed.

### IBM 1405 Disk Storage Instructions

#### Seek Disk

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>MU or LU</td>
<td>%F0</td>
<td>xxx</td>
<td>R</td>
</tr>
<tr>
<td>A</td>
<td>SD</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Function.** The A-address specifies that a seek operation is to be performed by the access arm. The B-address specifies the high-order position in core storage of the disk record address which is followed by a group-mark with a word-mark.

The selected access arm seeks the disk and track specified in the disk record address. Processing can continue while the access arm is in motion.

#### Word Marks.

Word marks are not affected.

#### Timing.

\[ T = 0.0115 (L_1 + 9) \text{ ms} + \text{access time.} \]

**Note.** If the access arm is already at the disk track that is to be used, a seek disk instruction need not be given.

#### Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>B + 1</td>
<td>B + 8</td>
</tr>
</tbody>
</table>

**Example.** Seek record 050090 with access arm 1. Storage locations 0590-0597 (labeled INPUTA) contain 10500900 (Figure 113).

### Read Disk Single-Record

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>M</td>
<td>%F1</td>
<td>xxx</td>
<td>R</td>
</tr>
<tr>
<td>A</td>
<td>RD (single record)</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Function.** This instruction causes data to be read from disk storage into core storage. The digit 1 in the A-address (%F 1) specifies that a single record is to be read. The reading of the disk is stopped by a group-mark with a word-mark in core storage and the end of the sector. If the digit 2 is present in the A-address (%F 2) a full-track read occurs. That is, five 200-character records are read from disk storage into core storage. Reading stops at the end of the fifth sector.

The B-address specifies the high-order position in core storage of the disk-record address which is followed by a group-mark with a word-mark, and the area in storage reserved for the data read from the disk.

The R in the d-character position signifies that this is a read operation.

**Word Marks.** A group-mark with a word-mark must appear one position to the right of the record address and one position to the right of the last position reserved in core storage for the disk record. If...
a group-mark with a word-mark is detected before reading of the record is completed, the wrong-length record indicator turns on and reading stops.

**Timing.**

\[ T = 0.0115 \left( L_1 + 9 \right) + 10 \text{ ms} + \text{disk rotation.} \]

60.196 ms is maximum time for single-record read.
10.196 ms is minimum time for a single-record read.

**Note:** Before reading starts, an automatic check of the record address in storage with the record address on the disk is made. If they are not the same, the unequal address compare indicator turns on, and the data on the disk cannot be read into storage.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>B + 1</td>
<td>B + 210</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B + 1010</td>
</tr>
</tbody>
</table>

**Example.** Read a single record from disk storage to core storage, beginning at location 0599 (area is labeled INPUTA). The high-order position of the disk address is in the first eight positions of the label (0590-0597), Figure 114.

**Write Disk Single-Record**

**Write Disk Full-Track**

**Instruction Format.**

\[
\begin{array}{cccccc}
\text{Mnemonic} & \text{Op Code} & \text{A-address} & \text{B-address} & \text{d-character} & \text{SPS} \\
\text{SFS} & \text{MU} & \%F_1 & \text{XXX} & \text{W} & \text{SPS} \\
\text{WD} & \text{(single record)} & \text{WDT} & \text{(full track)} & & \text{WDT} \\
\end{array}
\]

**Function.** This instruction causes a single record (or full-track characters) in core storage to be written on a disk record. The digit 1 in the A-address (\%F 1) specifies that a single record is to be written. If a 2 is present in the A-address (\%F 2), five 200-character records are written on a disk track. Writing stops at the end of the fifth sector.

The B-address specifies the high-order position of the disk-record address and is followed by the data to be written on the disk.

The W in the d-character position signifies that this is a write operation.

**Word Marks.** The writing of data stops when the end of a record is reached on the disk and a group-mark with a word-mark is sensed in core storage. If the group-mark with a word-mark is sensed before the end of a record, the remainder of the disk record is filled with data from core storage and the wrong-length record indicator turns on. A group-mark with word-mark must be one position to the right of the record address.

**Timing.**

\[ T = 0.0115 \left( L_1 + 9 \right) + 10 \text{ ms} + \text{rotation time.} \]

60.196 ms is maximum time for a single-record write.
10.196 ms is minimum time for a single-record write.

**Note:** Before writing starts, an automatic check of the record address in storage, with the record address on the disk, is made. If they are not the same the unequal-address compare indicator turns on, and the data in storage cannot be written on the disk.

A WRITE DISK CHECK instruction must be performed following a write disk operation. No other disk-storage operation can be performed until the check of data written on the disk is accomplished.

If the data in core storage contains characters with word marks, only the CBA 8421 portion of the character is written on the disk (the word mark is ignored).

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>B + 1</td>
<td>B + 210</td>
</tr>
<tr>
<td></td>
<td></td>
<td>B + 1010</td>
</tr>
</tbody>
</table>

**Example.** Write a disk record (single) from the data in area labeled INPUTA (first position of data is at 0599). The high-order position of the disk address is in the first eight positions of the label (0590-0597), Figure 115.
Write Disk Check

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>MU</td>
<td>M or L</td>
<td>%F3</td>
<td>W</td>
</tr>
<tr>
<td>LU</td>
<td>WDC</td>
<td>(word marks)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>WDCW</td>
<td>(word marks)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function. The function of this instruction is to cause a comparison, character-by-character, of the data in core storage with the data just written on the disk. The system automatically reads the disk record that was the last record to be addressed by the 1401 program. This instruction must follow a write operation.

The digit 3 in the A-address specifies that a WRITE DISK CHECK is to be performed. Either a single record or a full track is checked, depending on how the data was recorded in disk storage.

The B-address specifies the area in core storage where the record address and data recorded on the disk are located.

Word Marks. A group-mark with a word-mark must appear one position to the right of the disk record address and of the disk data in core storage.

Timing. \( T = 0.0115 (L_1 + 9) \text{ ms} + 50 \text{ ms} \).

Note. If the disk address in core storage is not the same as the address on the record, the unequal-address compare indicator turns on. If any of the characters on the disk record do not agree with the characters in core storage, the read-back check-error indicator turns on.

A WRITE DISK CHECK instruction can also follow a READ DISK SINGLE-RECORD instruction to verify data read from the disk.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>B + 1</td>
<td>B + 210</td>
</tr>
<tr>
<td>B + 1010</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example. Compare the disk record with the record in the area labeled OUTPUT (first position of data is 0699). The high-order position of the disk address is in the first eight positions of the label (0690-0697), Figure 116.

Read Disk Single-Record with Word Marks

Read Disk Full-Track with Word Marks

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>LU</td>
<td>L</td>
<td>%Fx</td>
<td>R</td>
</tr>
<tr>
<td>A</td>
<td>RDW (single record)</td>
<td>RDTW (full track)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function. These instructions are similar to the READ DISK SINGLE-RECORD and READ DISK FULL-TRACK instructions except that word marks in the record area of core storage are removed, and word marks from the disk records are written in core storage. The length of the record read into core storage from disk storage is 176 positions for a single record, and 880 positions for a full track.

Word Marks. A group-mark with a word-mark in core storage terminates the read operation. If the group-mark with a word-mark is not in the position to the right of the last character read from the disk into core storage, the wrong-length record indicator turns on. A group-mark with a word-mark must be one position to the right of the record address.

Timing. \( T = 0.0115 (L_1 + 9) \text{ ms} + 10 \text{ ms} + \text{ disk rotation} \).

Note. If a disk is read in a mode different from the one in which it was written (M or L operation code) a parity error occurs. The read-parity check indicator turns on.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>B + 1</td>
<td>B + 186</td>
</tr>
<tr>
<td>B + 890</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Example. Read a record from disk storage, with its associated word marks, into the area labeled INPUT (first position of data is 0599). The high-order position of the disk address is in the first eight positions of the label (0590-0597), Figure 117.

Figure 116. Write Check

Figure 117. Read a Record with Word Marks
Write Disk Single-Record with Word Marks
Write Disk Full-Track with Word Marks

Instruction Format.

Mnemonic Op Code A-address B-address d-character

SPS LU A WDW (single record)

WDTW (full track)

Function. This instruction is similar to the write disk operation, except that word marks present in the data in core storage are recorded on the disk record. The mode of operation permits the writing of programs on disk records for systems use. One hundred and seventy-six positions of data with word marks are recorded on the disk during a write single-record operation, and 880 positions are recorded during a write full-track operation.

Word Marks. A group-mark with a word-mark one position to the right of the last character of the record in core storage terminates the write operation. If the group-mark with a word-mark is not in the correct position, the wrong-length record indicator turns ON. A group-mark with a word-mark must be one position to the right of the record address.

Timing. \( T = 0.0115 (L_i + 9) \text{ ms} + 10 \text{ ms} + \text{disk rotation}. \)

Note. The programmer should be certain that all records on a specific track are written in the same mode (either by a MOVE or by a LOAD instruction), otherwise, full-track operations are not possible. A write-disk check operation must be performed following this instruction.

Before writing starts, an automatic check of the record address in storage, with the record address on the disk, is made. If they are not the same, the unequal-address compare indicator turns on, and the data in storage cannot be written on the disk.

Address Registers After Operation.

\[
\begin{array}{c|c|c|c}
\text{I-Add. Reg.} & \text{A-Add. Reg.} & \text{B-Add. Reg.} \\
\hline
\text{NSI} & B + 1 & B + 186 \\
& & B + 800
\end{array}
\]

Example. Write a disk record with word marks from the area labeled OUTPUT (the first position of data is 0599). The high-order position of the disk address is in the first eight positions of the label (0590-0597), Figure 118.

Branch if Indicator On

Instruction Format.

Mnemonic Op Code I-address d-character

SPS H A BIN

Function. The d-character specifies the indicator tested. If the indicator is ON, the next instruction is taken from the I-address. If the indicator is OFF, the next sequential instruction is taken. Figure 119 shows symbols that are valid d-characters and the indicators they test.

The next disk storage operation turns off the access-inoperable indicator. The other disk unit indicators are turned off by any disk operation except SEEK DISK.

<table>
<thead>
<tr>
<th>d-CHARACTER</th>
<th>INDICATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>Access Inoperable</td>
</tr>
<tr>
<td>V</td>
<td>Read- or Write Parity Check or Read-Back Check Error</td>
</tr>
<tr>
<td>W</td>
<td>Wrong-Length Record</td>
</tr>
<tr>
<td>X</td>
<td>Unequal-Address Compare</td>
</tr>
<tr>
<td>Y</td>
<td>Any Disk-Unit Error Condition</td>
</tr>
</tbody>
</table>

Figure 119. IBM 1405 Branch Instruction d-Character

Indicators. Access Inoperable—An access arm becomes inoperable if the logic safety circuit detects improper operation. A customer engineer can also render an arm inoperable. In either instance, this indicator turns on when the program addresses the inoperable arm. At the same time, the appropriate access light (1405) and the RAMAC light (1401) turn on.

The indicator also turns on if an invalid (not installed) arm or disk storage unit is addressed. Because the program continues in sequence, even when an inoperable arm is addressed, a BRANCH IF INDICATOR ON instruction should follow a seek instruction.

Read- or Write-Parity Check or Read-Back Check Error—This indicator turns on if even-bit parity occurs when reading the record address and information from, or writing information on a disk. Another condition that turns the indicator on is an unequal compare during a write check operation.

Figure 116. Write a Record with Word Marks
Wrong-Length Record—This indicator turns on if the number of characters read to, or written on, the disk record is not equal to 200 or 1,000 characters (for M operation code) or 176 or 880 characters (for L operation code).

Unequal-Address Compare—This indicator turns on if an unequal condition occurs during the automatic checking of the record address in storage with the record address on the disk. This is an automatic check and does not have to be programmed.

This is the same internal circuitry that is used by the compare instruction. Care should be taken in programming that a normal-compare operation and the address-compare operation do not interfere with the setting of the equal, low, and high compare indicators set by a previous instruction.

Any Disk-Unit Error Condition—This indicator turns on if any of the other disk storage indicators are on. It can be tested by the program, and, if it is off, allows the program to proceed. If the indicator is on, then the other indicators should be checked to determine where corrective measures should be taken.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_i + 1) \) ms.

Note. After each disk unit, read, or write operation, the program must test for error indications to prevent processing of unusable data.

<table>
<thead>
<tr>
<th>Address Registers After Operation.</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
</tr>
</tbody>
</table>

Example. At the completion of a disk-read operation, test the any-disk-unit error condition indicator. If it is on, branch to the routine labeled DISKER (0690) to determine the type of error condition. If it is off, continue in the main program, which tests the other disk unit indicators and branches to error routines if the respective indicator is on. The routines are labeled: UNADCL (0790), WRLENR (0890), and RWPARC (0990), Figure 120.
Branch if Indicator ON, replace Figure 120 with: punch busy operation.

Assembled Instruction:

480 B 690 Y

690 B 790 X

695 B 890 W

700 B 990 V

Figure 120. Indicator Testing Routine
IBM 1407 Console Inquiry Station

The IBM 1407 Console Inquiry Station provides a direct and immediate means of communication between the station operator and IBM 1401 Data Processing System. Because the scope of applications processed by 1401 systems has increased, there is a need for some means of communication with the system. The IBM 1407 Console Inquiry Station (Figure 121) is designed to fill this need.

Located adjacent to the IBM 1401 Processing Unit console, the inquiry station can be used for reading data from storage or for inserting data or program steps, if necessary, into 1401 core storage.

When an inquiry is to be made, the operator presses the request enter key-light. As soon as the system is free to act on the request, the enter light comes on and the operator can type the message and enter it into 1401 core storage.

When the system completes the processing of the inquiry, it is transferred to the inquiry station by the stored program. The message is typed, and the operator may act on the reply.

The inquiry station printer has a 64-character double-case keyboard and a pin-feed platen. It can print ten digits (0-9), twenty-six alphabetic characters (A-Z), and twenty-eight special characters (Figure 122).

The table on which the console printer is located has an area on the left of the keyboard for program notebooks and other materials to aid the operator.

Figure 122. IBM 1407 Console Keyboard

The inquiry station is especially valuable when used with a RAMAC 1401 system. It can be used to call for data stored on a disk record. An account record or stock-status record needed by management can be requested by the operator and made available in a short time. Thus, management can, at a moment's notice, request information from the 1401 system and have an answer almost instantaneously.

For the user of the 1401 Model D system, the console inquiry station provides an input-output device. This system, consisting of the IBM 1401 Processing Unit, the IBM 1403 Printer, and IBM 729 Magnetic Tape Units, uses magnetic tape for entering stored programs. Now, the IBM 1407 Console Inquiry Station
permits manual alteration of the stored program without writing the changes on magnetic tape. Exception records can be typed on the console printer while the regular reports are printed on the 1403.

The IBM 1407 Console Inquiry Station provides the 1401 system user with an efficient and economical means of altering the stored program and examining data stored in the system.

The format of an inquiry or reply is controlled by the data necessary to perform the function. This means that the format is variable in all cases. The message is only as long as is necessary to provide the information. There are no format control keys or tapes. Format is under program control and is completely flexible. Pressing the clear or respond key ends a message to the 1401. When typing, a reply transmission is stopped by a group-mark with a word-mark (when the 1401 is in the run mode) or by pressing the clear key.

Thus, the programmer can provide different format for the data to be typed solely on the basis of the 1401 stored program.

IBM 1407 Console Inquiry Station Instructions

Read from Console Printer

Instruction Format.

Mnemonic Op Code A-address B-address d-character

Function. This instruction causes the enter key-light to come on, the keyboard to unlock, and the data (to be typed on the 1407) to enter 1401 core storage. The A-address specifies an inquiry station operation. The B-address is the high-order position in 1401 core storage wherein the data is to be stored. The d-character specifies a read-in operation. The inquiry request indicator must be on to process this instruction.

Word Marks. A group-mark with a word-mark must be inserted in 1401 core storage to the right of the last character sent to the 1401 from the 1407. Another method of terminating a read-in operation is pressing the clear key (see Clear Key).

Timing. $T = 0.0115 \times (L_4 + 1) \text{ ms} + \text{typing time}$.

Note. The lower case b (special character) or space bar causes a space to be taken and a blank to enter core storage. If the b-key is pressed, a lower case b is printed. The method of entering data is discussed in the section on Console Operation.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>%30</td>
<td>B + L_o</td>
</tr>
</tbody>
</table>

Example. Transfer the data typed on the IBM 1407 to the area in 1401 core storage labeled INQIN (0785), Figure 123.

Write on Console Printer

Instruction Format.

Mnemonic Op Code A-address B-address d-character

Function. This instruction causes data from 1401 storage to be typed by the inquiry station. While the data is being typed, the typeout light is on. The B-address is the high-order position in 1401 core storage of the data to be transferred to the console printer. The d-character specifies a write operation.

Word Marks. A group-mark with a word-mark in 1401 core storage stops the transfer of data to the 1407 and causes a carriage return. Pressing the clear key also stops the transfer of data from the 1401 (see Clear Key).

Timing. $T = 0.0115 (L_4 + 1) \text{ ms} + \text{typing time}$.

Note. Characters that have incorrect parity (even-bit) are typed as a ~ if the process check stop switch is off. If the switch is on, typing stops before typing the incorrect character.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>%30</td>
<td>B + L_o</td>
</tr>
</tbody>
</table>

Figure 123. Read into Storage
Example. Type the data, beginning in the area labeled INQOUT (0785) and ending with a group-mark, with a word-mark (Figure 124).

**Autocoder**

**Assembled Instruction:** \texttt{L \%T0 785 W}

**Figure 124. Read out of Storage**

**Read from Console Printer with Word Marks**

**Instruction Format.**

\texttt{Mnemonic \ Op Code \ A-address \ B-address \ d-character}

\texttt{LU \%T0 \textit{xxx} R}

**Function.** This instruction causes the enter key light to come on, the keyboard to unlock, and the data with word mark (to be typed on the 1407) to enter 1401 core storage. The A-address specifies an inquiry station operation. The B-address is the high-order position, in 1401 core storage, in which the data is to be stored. The d-character specifies a read-in operation. Word marks are entered by first pressing the word-mark key, and then pressing the associated character key. Characters with a word mark print in red. The inquiry request indicator must be on to process this instruction.

**Word Mark.** A group-mark with a word-mark must be inserted in 1401 core storage to the right of the last character sent to the 1401 from the 1407. Another method of terminating a storage read-in operation is to press the clear key (see Clear Key).

**Timing.** \( T = .0115 \ (L_1 + 1) \ \text{ms} + \text{typing time.} \)

**Note.** The lower case \( b \) (special character) or space bar causes a space to be taken and a blank to enter core storage. If the b-key is pressed, a lower case \( b \) is printed. The method of entering data is discussed in the section on 1407 Console Operation.

**Address Registers After Operation.**


\texttt{NSI \%30 \textit{B + L}_a}

**Example.** Transfer the data with word marks located in the area labeled INQOUT (0785) to the area in 1401 core storage labeled INQIN (0785), Figure 125.

**Write on Console Printer with Word Marks**

**Instruction Format.**

\texttt{Mnemonic \ Op Code \ A-address \ B-address \ d-character}

\texttt{LU \%T0 \textit{xxx} \ W}

**Function.** This instruction causes data from 1401 storage to be typed by the inquiry station. While the data is being typed, the typeout light is on. The A-address specifies an inquiry station operation. The B-address is the high-order position in 1401 core storage of the data to be transferred to the typewriter. The d-character specifies a write operation.

**Timing.** \( T = .0115 \ (L_1 + 1) \ \text{ms} + \text{typing time}. \)

**Word Marks.** A group-mark with a word-mark stops the transfer of data to the 1407 and causes a carriage return. Pressing the clear key also stops the transfer of data from the 1401 (see Clear Key).

**Note.** Characters that have a word mark in association with them are typed in red. All other characters are typed in black. A space is printed as a lower case \( b \). Characters with incorrect parity (even-bit) are typed as a \( H_< \) if the process check stop switch is off. If the switch is on, typing stops before typing the incorrect character.

**Address Registers After Operation.**


\texttt{NSI \%30 \textit{B + L}_a}

**Example.** Type the data with word marks located in the area labeled INQOUT (0785) and ending with a group-mark with a word-mark (Figure 126).
Line Space

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>LU or MU</td>
<td>L or M</td>
<td>%TO</td>
<td>xxx</td>
<td>W</td>
</tr>
</tbody>
</table>

Function. This instruction causes the console printer to space one line. The B-address is the storage location of a group-mark with a word-mark.

**Timing.** \( T = 0.0115 \) \((L_t + 1)\) ms + space time.

Word Marks. A group-mark with a word-mark must be at the B-address.

Note. Multiple line spacing is controlled by this instruction and by the setting of the line space lever.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>%30</td>
<td>B + 1</td>
</tr>
</tbody>
</table>

Example. Space a single line on the console printer.

The storage position labeled GMWM (0895) contains a group-mark with a word-mark (Figure 127).

![Figure 127. Line Space](image)

Autocoder

![Autocoder](image)

Assembled Instruction: \( L \ %TO \ 895 \)

Figure 127. Line Space

Branch if Indicator On

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>B</td>
<td>A</td>
<td>xxx</td>
</tr>
</tbody>
</table>

Function. The d-character specifies the indicator tested. If the indicator is on, the next instruction is taken from the I-address. If the indicator is off, the next sequential instruction is taken. Figure 128 shows symbols that are valid d-character and the indicators they test.

<table>
<thead>
<tr>
<th>d-CHARACTER</th>
<th>INDICATOR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Inquiry Clear</td>
</tr>
<tr>
<td></td>
<td>Inquiry Request</td>
</tr>
</tbody>
</table>

Figure 128. 1407 Branch Instruction d-Character

Indicators. Inquiry Clear -- This indicator turns on when the clear key-light is pressed, if the 1401 is in the run mode. It turns off when the 1401 program processes a console inquiry instruction or the start reset key on the 1401 console is pressed. It must be tested before processing the next inquiry.

Inquiry Request -- This indicator turns on when the request enter key-light is pressed to signal the 1401 that an inquiry is to be processed, and the 1401 is in the run mode. It is turned off after the 1401 processes a console inquiry instruction. Pressing the start reset key on the 1401 console or the clear key on the 1407 also turns this indicator off.

Word Marks. Word marks are not affected.

**Timing.** \( T = 0.0115 \) \((L_t + 1)\) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>B</td>
<td>dbb</td>
</tr>
</tbody>
</table>

Example. Branch to the inquiry routine labeled INQ-RUT (0950) if the inquiry request indicator is on (Figure 129).

![Autocoder](image)

Assembled Instruction: \( B \ %50 \ Q \)

Figure 129. Test and Branch
The special features described in this section are available for use in the IBM 1401 Data Processing System. They offer additional flexibility in applications where special processing requirements exist.

**Multiply-Divide Feature**

This feature makes it possible to perform direct multiplication and division in the IBM 1401 Data Processing System.

**Multiply**

**Instruction Format.**

```
Mnemonic  Op Code  A-address   B-address
M       @    xxx      xxx
```

**Function.** The multiplicand (data located in the A-field) is repetitively added to itself in the B-field. The B-field contains the multiplier in the high-order positions, and enough additional positions (low order) to allow for the development of the product. At the end of the multiply operation, the units position of the product is located at the B-address. The multiplier is destroyed in the B-field as the product is developed. Therefore, if the multiplier is needed for subsequent operations, it must be retained in another storage area.

**Rules:**

1. The product is developed in the B-field. The length of the B-field is determined by adding “1” to the sum of the number of digits in the multiplicand and multiplier fields.

   Example:  
   
   \[
   \begin{array}{c}
   1246 \\
   \times \ 543 \\
   \end{array}
   \]

   4-digit multiplicand
   3-digit multiplier

   \[ + 1 \]

   8 positions must be allowed in the B-field.

2. A word mark must be associated with the high-order positions of both the multiplier and multiplicand fields.

3. A- and B-bits need not be present in the units positions of the multiplier and multiplicand fields. The absence of zone bits in these positions indicates a positive sign. At the completion of the multiply operation the B-field will have zone bits in the units position of the product only. The multiply operation uses algebraic sign control (Figure 130).

![Table of Algebraic Sign Control for Multiplication](image)

<table>
<thead>
<tr>
<th>Multiplier Sign</th>
<th>+</th>
<th>+</th>
<th>-</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multiplicand Sign</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Sign of Product</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
</tbody>
</table>

**Timing.** The average time required for a multiply operation is:

\[
T = 0.0115 (L_M + 3 + 2L_C + 5L_CL_M + 7L_M) \text{ ms.}
\]

\[L_C = \text{length of multiplicand field.}\]

\[L_M = \text{length of multiplier field.}\]

A chart of approximate timings is included in the section on **Timing**.

**Notes.** The first addition within the multiply operation inserts zeros in the product field from the storage location specified by the B-address up to the units position of the multiplier.

The A-address register and the B-address register indicate positions within the A- and B-fields on which operations are currently being performed.

**Word Marks.** A word mark must be associated with the high-order positions of the multiplier and multiplicand fields.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A minus the length of the multiplicand.</td>
<td>B minus the length of product field.</td>
</tr>
</tbody>
</table>

**Example. Multiply:**

```
Label  Location of Data Word  Contents of Data Word  Description
MULCAN  0502  1246  Multiplicand
MULIER  0085  543  Multiplier
PRODCT  0810  Product
```

The size of the product field is \(4 + 3 + 1 = 3\).

The multiplier is placed in the three high-order
positions of the PRODCT area (0603, 0604, and 0605). At the completion of the multiply operation, load the product in the area labeled OUT2 (0178). The units positions of the multiplier and multiplicand fields may be signed (Figure 131).

### SPS

<table>
<thead>
<tr>
<th>OPERAND</th>
<th>OPERAND</th>
</tr>
</thead>
<tbody>
<tr>
<td>...0...</td>
<td>...0...</td>
</tr>
<tr>
<td>...0...</td>
<td>...0...</td>
</tr>
<tr>
<td>...0...</td>
<td>...0...</td>
</tr>
</tbody>
</table>

**Autocoder**

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>...0...</td>
<td>...0...</td>
</tr>
<tr>
<td>...0...</td>
<td>...0...</td>
</tr>
<tr>
<td>...0...</td>
<td>...0...</td>
</tr>
</tbody>
</table>

**Assembled Instruction:**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>...0...</td>
<td>...0...</td>
</tr>
<tr>
<td>...0...</td>
<td>...0...</td>
</tr>
</tbody>
</table>

Figure 131. Multiply

### Divide

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>D</td>
<td>%</td>
<td>xxx</td>
<td>xxx</td>
</tr>
</tbody>
</table>

**Function.** This instruction divides the data (dividend) in the low-order positions of the B-field by the divisor located in the A-field, and develops the quotient in the high-order positions of the B-field. The remainder is left in the low-order positions of the B-field.

**Rules:**

1. The quotient is developed in the B-field. The length of the B-field is determined by adding 1 to the sum of the number of digits in the divisor and dividend fields.

   **Example:**
   
   
<table>
<thead>
<tr>
<th>543</th>
<th>1246</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 digit dividend</td>
<td>3 digit divisor</td>
</tr>
<tr>
<td>+1</td>
<td></td>
</tr>
</tbody>
</table>
   
   8 positions must be allowed in the B-field.

2. A word mark must be associated with the high-order position of the A-field.

3. In all cases A- and B-bits (plus sign) or B-bit (minus sign) must appear in the units position of the dividend field. The divisor may be either signed or unsigned. If there are no bits in the units position of the divisor, the machine assumes the divisor factor is positive. The divide operation uses algebraic sign control (Figure 132).

### Figure 132. Algebraic Sign Control for Division

<table>
<thead>
<tr>
<th>Divisor Sign</th>
<th>+</th>
<th>+</th>
<th>-</th>
<th>-</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dividend Sign</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Quotient Sign</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Remainder Sign</td>
<td>+</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
</tbody>
</table>

4. The dividend is loaded in the low-order positions of the B-field (Figure 133) by a zero and add instruction to insure that zeros are present in the high-order positions of the B-field.

### Figure 133. Dividend in B-Field

5. The B-address in the divide instruction specifies the high-order position of the dividend.

At the completion of division:

a. The quotient is in the high-order positions of the B-field. The location of the units position of the quotient is the address of the units position of the dividend, minus the length of the divisor, minus one.

b. The remainder is in low-order positions of the B-field.

c. The sign of the quotient is over the units position of the quotient field.

d. Because only one quotient digit can be developed at a time, it is important to address the high-order position of the dividend (B-address of the divide instruction). This insures that the first divide operation will result in a single high-order quotient digit. A dividend improperly addressed can cause an arithmetic overflow if the result of the first divide operation is greater than 9.

**Note:** A divide operation refers to the process of developing each quotient digit. If the quotient field is not large enough, no overflow is indicated. This is a programming error for which the machine does not check. Division by zero results in an arithmetic overflow condition. Figure 134 shows the result of a divide operation.

### Figure 134. Location of the Results of a Divide Operation

Extra zeros can be added to the dividend prior to a divide operation when a larger quotient is required. For each additional quotient digit desired,
place one zero to the right of the dividend as shown in Figure 135. Note that in this example, the units position of the quotient is not located in the position previously described in the section At The Completion of Division, a.

Word Marks. A word mark must define the high-order position of the divisor.

Timing. Average time required for the execution of a divide operation is calculated:

\[ T = 0.115 \left( \frac{L_d}{10} + 2 + 7 \frac{L_r}{10} + 8 \frac{L_q}{10} \right) \text{ ms} \]

\( L_q \) = length of the quotient field.

\( L_r \) = length of the divisor field.

A chart of approximate timings is included in the section on Timing.

Note. The quotient field is not cleared before actual division begins.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A minus the length of the divisor</td>
<td>Tens position of quotient. If divisor has all zeros, the B-address register stands at the units position of the dividend minus the length of the divisor minus the length of the dividend minus 1.</td>
</tr>
</tbody>
</table>

Example. Figure 136 is a symbolic example for DIVIDE.

<table>
<thead>
<tr>
<th>Label</th>
<th>Location of Data Word</th>
<th>Data Word</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIVEND</td>
<td>0502</td>
<td>1246</td>
<td>Dividend</td>
</tr>
<tr>
<td>DIVSOR</td>
<td>0005</td>
<td>543</td>
<td>Divisor</td>
</tr>
<tr>
<td>QUOT</td>
<td>0985</td>
<td></td>
<td>Quotient</td>
</tr>
</tbody>
</table>

Figure 136. Division

Increased Core Storage

The IBM 1406 Storage Unit (Figure 137) is an additional component of the IBM 1401 Data Processing System. It is used with IBM 1401 processing units containing 4,000 positions of core storage. The 1406 increases the core storage capacity to as much as 16,000 positions. This additional storage capacity greatly increases the range of applications that can be economically handled by the IBM 1401.

IBM 1406 Storage Unit Models

The 1406, Model 1 contains a block of 4,000 core-storage positions. It increases the capacity of the system to 8,000 positions.
Model 2 contains a block of 8,000 core-storage positions. It increases the capacity of the system to 12,000 positions.

Model 3 contains a block of 12,000 core-storage positions. It increases the capacity of the system to 16,000 positions.

**Addressing**

The additional core-storage locations are addressed by the presence of zone bits located over the units position of each storage address over 3999. These zone bits are added to the basic 4,000-character coding system.

<table>
<thead>
<tr>
<th>Storage Addresses</th>
<th>Zone Bits over Units Position of Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>4,000-7,999</td>
<td>A</td>
</tr>
<tr>
<td>8,000-11,999</td>
<td>B</td>
</tr>
<tr>
<td>12,000-15,999</td>
<td>AB</td>
</tr>
</tbody>
</table>

Storage addresses 0000-3999 have zone bits over the hundreds position, and no zone bits over the units position of the 3-character address.

Storage addresses 4000-7999 have zone bits over the hundreds position, and A-bits (0 zone) over the units position of the 3-character address.

Storage addresses 8000-11,999 have zone bits over the hundreds position, and B-bits (11 zone) over the units position of the 3-character address.

Storage addresses 12,000-15,999 have zone bits over the hundreds position, and A- and B-bits (12 zone) over the units position of the 3-character address. Figure 138 is a chart of the addressing system.

The IBM 1401 addresses core-storage locations by assigning a digit value to each bit that appears over the hundreds and units positions of the 3-character address.

<table>
<thead>
<tr>
<th>Bit and Location</th>
<th>Digit Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-bit over hundreds position</td>
<td>1</td>
</tr>
<tr>
<td>B-bit over hundreds position</td>
<td>2</td>
</tr>
<tr>
<td>A-bit over units position</td>
<td>4</td>
</tr>
<tr>
<td>B-bit over units position</td>
<td>8</td>
</tr>
</tbody>
</table>

The machine adds the assigned digit values of the hundreds and units positions to determine the thousand block of storage addressed. For example, to address core-storage locations, the IBM 1401 assigns digit values to the bits.

$$799 = \frac{A}{B} = \frac{1}{2} = 3099$$

$$15R = \frac{A}{B} = \frac{2}{8} = 11,859$$

**Address Validity**

The IBM 1401 checks each address to ensure that it is valid for the storage capacity installed. The system stops on an address validity error, if an invalid address is encountered.

<table>
<thead>
<tr>
<th>Core-Storage Capacity</th>
<th>Valid Addresses</th>
<th>Invalid Addresses</th>
</tr>
</thead>
<tbody>
<tr>
<td>1,400</td>
<td>0000-15999</td>
<td>1400-15999</td>
</tr>
<tr>
<td>2,000</td>
<td>0000-15999</td>
<td>2000-15999</td>
</tr>
<tr>
<td>4,000</td>
<td>0000-7999</td>
<td>4000-15999</td>
</tr>
<tr>
<td>8,000</td>
<td>0000-11999</td>
<td>8000-15999</td>
</tr>
<tr>
<td>12,000</td>
<td>0000-15999</td>
<td>12000-15999</td>
</tr>
<tr>
<td>16,000</td>
<td>0000-15999</td>
<td>NONE</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ACTUAL ADDRESSES</th>
<th>ZONE BITS OVER HUNDREDS POSITION</th>
<th>ZONE BITS OVER UNITS POSITION</th>
<th>3-CHARACTER ADDRESSES</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 to 9999</td>
<td>No Zone Bits</td>
<td>No Zone Bits</td>
<td>000 to 999</td>
</tr>
<tr>
<td>1000 to 1999</td>
<td>A-Bit (Zero-Zone)</td>
<td>No Zone Bits</td>
<td>+00 to 299</td>
</tr>
<tr>
<td>2000 to 2999</td>
<td>B-Bit (11-Zone)</td>
<td>No Zone Bits</td>
<td>100 to 899</td>
</tr>
<tr>
<td>3000 to 3999</td>
<td>AB-Bits (12-Zone)</td>
<td>No Zone Bits</td>
<td>700 to 199</td>
</tr>
<tr>
<td>4000 to 4999</td>
<td>No Zone Bits</td>
<td>A-Bit (Zero-Zone)</td>
<td>00# to 99Z</td>
</tr>
<tr>
<td>5000 to 5999</td>
<td>A-Bit (Zero-Zone)</td>
<td>A-Bit (Zero-Zone)</td>
<td>+0# to 79Z</td>
</tr>
<tr>
<td>6000 to 6999</td>
<td>B-Bit (11-Zone)</td>
<td>A-Bit (Zero-Zone)</td>
<td>10# to 89Z</td>
</tr>
<tr>
<td>7000 to 7999</td>
<td>AB-Bits (12-Zone)</td>
<td>AB (Zero-Zone)</td>
<td>70# to 19Z</td>
</tr>
<tr>
<td>8000 to 8999</td>
<td>No Zone Bits</td>
<td>B-Bit (11-Zone)</td>
<td>001 to 99R</td>
</tr>
<tr>
<td>9000 to 9999</td>
<td>A-Bit (Zero-Zone)</td>
<td>B-Bit (11-Zone)</td>
<td>+01 to 299</td>
</tr>
<tr>
<td>10000 to 10999</td>
<td>B-Bit (11-Zone)</td>
<td>B-Bit (11-Zone)</td>
<td>101 to 89R</td>
</tr>
<tr>
<td>11000 to 11999</td>
<td>AB-Bits (12-Zone)</td>
<td>AB (11-Zone)</td>
<td>701 to 19R</td>
</tr>
<tr>
<td>12000 to 12999</td>
<td>No Zone Bits</td>
<td>AB-Bits (12-Zone)</td>
<td>002 to 99I</td>
</tr>
<tr>
<td>13000 to 13999</td>
<td>A-Bit (Zero-Zone)</td>
<td>AB-Bits (12-Zone)</td>
<td>+02 to 29I</td>
</tr>
<tr>
<td>14000 to 14999</td>
<td>B-Bit (11-Zone)</td>
<td>AB-Bits (12-Zone)</td>
<td>102 to 89I</td>
</tr>
<tr>
<td>15000 to 15999</td>
<td>AB-Bits (12-Zone)</td>
<td>AB-Bits (12-Zone)</td>
<td>702 to 19I</td>
</tr>
</tbody>
</table>

Figure 138. Addressing System
ADDRESS ARITHMETIC

To facilitate address arithmetic, an additional operation code (MODIFY ADDRESS) is added to IBM 1401 systems equipped with more than 4,000 characters of core storage.

Modify Address

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>MA</td>
<td>#</td>
<td>xxx</td>
</tr>
</tbody>
</table>

Function. This instruction causes the 3-character field, specified by the A-address, to be added to itself. The result is stored in the A-field.

Word Marks. Word marks are not required to define the A-field. If they are present, they are ignored and remain undisturbed in the A-field.

Timing. T = .0115 (L + 9) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-3</td>
<td>A-1 or A-3</td>
</tr>
</tbody>
</table>

Example. Double the address labeled ADDC (2856), and store the result at ADDC (Figure 140).

Read Release and Punch Release Feature

With this feature it is possible to release the read-start-time and punch-start-time interlocks that normally occur during card-read and card-punch cycles, thus providing more processing time during input-output operations.

Start Read Feed

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SRF</td>
<td>8</td>
</tr>
</tbody>
</table>

Function. This instruction works in conjunction with the read release special feature. It releases the interlock that occurs during read start time, and permits a gain of 21 milliseconds of processing time between
card-read cycles. Also, it activates the card-read feed and moves the next card into reading position.

**Word Marks.** Word marks are not affected.

**Timing.** \( T = 0.0115 (L_t + 1) \) ms.

**Notes.** After the `START READ FEED` instruction is executed, a `READ A CARD` instruction must be given before the reader is ready to read the 9-row of the card. If the `READ A CARD` instruction comes too late, then the card is not read, and feeds into the NR pocket, and the machine stops. The reader light comes on, and the I-address register is at the location of the instruction following the one on which read release time was over-extended. To insure optimum processing time, a `START FEED READ` instruction should follow the `READ A CARD` instruction, within 10 ms, if continuous card reading is desired. Then the machine is ready to accept the next read instruction on the following cycle.

`START READ FEED` instructions can also be given in cases other than those that cause continuous card feeding, provided that a `READ A CARD` instruction follows within the next 21 milliseconds. For this reason subroutines that can be executed after the `START READ FEED` instruction has been given should be timed to determine if a `READ A CARD` instruction is necessary in the subroutine as well as in the main routine.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>Ap</td>
<td>Bp</td>
</tr>
</tbody>
</table>

**Example.** Release the interlock on the card reader and feed a card (Figure 141).

---

**Start Punch Feed**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPF</td>
<td>9</td>
</tr>
</tbody>
</table>

**Function.** This instruction works in conjunction with the punch release special feature. It releases the interlock that occurs during punch start time, and allows a gain of 37 milliseconds of processing time between card punch cycles. It also activates the card feed, and moves the card into punching position.

**Word Marks.** Word marks are not affected.

**Timing.** \( T = 0.0115 (L_t + 1) \) ms.

**Note.** After a `START PUNCH FEED` instruction is executed, a `PUNCH A CARD` instruction must follow before the machine is ready to punch the 12-row of the card. If no `PUNCH A CARD` instruction is interpreted, the card feeds past the punch station without being punched and the machine stops. (The card punched on the previous cycle is not checked.) The I-address register is at the location of the instruction following the one on which punch release time was over-extended. For this reason, if cards are to be punched every cycle, a `START PUNCH FEED` instruction should be given within 22 ms after the `PUNCH A CARD` instruction, so that the machine is ready to punch the next card on the following punch cycle. If cards are not to be punched every cycle, a `PUNCH A CARD` instruction should always follow a `START PUNCH FEED` instruction within 37 ms to insure proper machine operation.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>Ap</td>
<td>Bp</td>
</tr>
</tbody>
</table>

**Example.** Release the punch interlock, and feed a card (Figure 142).

---

**Autocoder**

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operand</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Assembled Instruction:** 9

**Figure 141.** Read Release

**Autocoder**

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Operand</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPF</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Assembled Instruction:** 9

**Figure 142.** Punch Release
**Punch Feed Read Feature**

In some applications it is desirable to read information into the system, calculate, and punch the results in the same card from which the input data was read. By using the punch feed read feature, the card at the punch-feed-read station can be read while the card ahead of it is being punched. To permit this type of operation, a special set of 80 reading brushes, called punch feed read, is added to the IBM 1402 Card Read-Punch feed, one station ahead of the punch station (Figure 143). A special d-character specifies that the card is to be read from the punch side of the 1402. The normal read area (storage locations 001-080) receives the information from the punch feed read in the same manner as information is read from the read feed. A validity and a columnar hole-count check is made on each card column read from the punch-feed-read brushes. MLP card codes cannot be read by the punch feed read brushes.

The punching operation for machines equipped with punch feed read is the same as in the basic 1401. Storage positions 101-180 are specified as the punch area, and a hole-count check is made at the punch brushes. The hole-count check of pre-punched data is begun at the punch-read station and is completed at the punch-check station after punching has occurred.

Note: Punching in prepunched columns is acceptable, provided that the resultant character is valid and that the punches read at the punch-feed read station are not repeated. For example an X can be punched in a card column that already contains a 2, but punching a K (X and 2 punches) at the punch station if either an X or a 2 was already in the card, results in a hole-count check.

The d-character R activates the punch-feed-read brushes. It can be used with the operation codes PUNCH (4), WHITE AND PUNCH (5), and START PUNCH FEED (9).

If the combination instruction READ PUNCH (5), or WHITE READ PUNCH (7) is given, read and punch errors occur.

When punch release is combined with a punch feed read operation, processing time is reduced from 59 ms to 56 ms between successive punch cycles.

**Read-Punch Feed Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS P RF</td>
<td>4</td>
<td>R</td>
</tr>
</tbody>
</table>

**Function.** When this instruction is used, the punch feed operates and reads the card entering the read station on the punch side. It also causes the card at the punch station to be punched. The R character modifier makes this instruction effective.

**Word Marks.** Word marks are not affected.

**Timing.** T = .0115 (L1 + 1) ms + punch start time (37 ms) and punching time of 184 ms. Punch start time can be used for processing if the punch release special feature is installed.

**Note.** An additional 3 ms is required in excess of the normal punch time of 181 ms when the punch feed read feature is used. Processing time available is 19 ms without punch release and 56 ms if the punch release special feature is employed.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI dbb</td>
<td>181</td>
<td></td>
</tr>
</tbody>
</table>

Figure 143. Punch Feed Read Schematic

80
Example. Read the card at the punch feed read station and punch a card (Figure 144).

Function. This instruction causes the printer to operate and print a line, and the punch unit to read a card, and also causes the card at the punch station to be punched. The d-character R specifies that the card at the punch feed station is to be read. The printer takes priority and operates first, but the signal to start the punch feed read is automatically given before the end of the print operation, so that actual card reading starts soon after the print cycle is complete.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 \times (L + 1) \text{ ms} + \text{the timing conditions for print and punch overlap (see Timing).} \) The print operation normally takes 94 ms. Punch start time is 37 ms and the punch reading time is 184 ms. An additional 3 ms are added to the normal punching time of 181 ms. Normal processing time available is 19 ms.

Note. If the print storage special feature is installed in the system, the automatic signal to start the punch feed read operation is given shortly after the transfer to data to the print storage area. Thus, additional processing time can be gained by using print storage.

Address Registers After Operation.

Example. Print a line, read a card, and punch a card from the punch side of the 1402 (Figure 146).
that the program branches to the I-address for the next instruction.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_1 + 1) \) ms + the timing conditions for print and punch overlap (see Timing). The print operation normally takes 84 ms. Punch start time is 37 ms and the punch reading time is 184 ms. An additional 3 ms are added to the normal punching time of 181 ms. Normal processing time available is 19 ms.

Note. If the print storage special feature is installed in the system, the automatic signal to start the punch feed read operation is given shortly after the transfer of data to the print storage area. Thus, additional processing time can be gained by using print storage.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>BI</td>
<td>181</td>
</tr>
</tbody>
</table>

Example. Print a line, read and punch a card from the punch side of the IBM 1403, and branch to START6 (0695) for the next instruction (Figure 147).

Additional Print Control Feature

The number of available print positions in the basic IBM 1403 Printer increases from 100 to 132 with this feature. The core-storage area reserved for printing when additional print control is installed includes locations 201-332.

Expanded Print Edit Feature

The basic operations of the MOVE CHARACTERS AND EDIT instruction can be increased by the expanded print edit feature. With this feature, asterisk protection, floating dollar sign, decimal control, and sign control left, operations can be performed. The zero-suppression code in the control word should be in the position immediately to the left of the decimal, except as required in Decimal Control.

Note: Floating dollar sign and asterisk protection or floating dollar sign and decimal control cannot be used in the same edit operation. When asterisk protection and decimal control are combined, and a blank data field is edited, the result is asterisks in all positions to the left of, but not including, the decimal-control position.

Asterisk Protection

When asterisks are to appear to the left of significant digits, the asterisk protection feature is used (Figure 145). The control word is written with the asterisk
immediately to the left of the zero suppression code. Zero-balances can be protected with asterisks by placing control zeros in the right-most position. In this instance, asterisks print in all positions including the decimal position.

**Forward Scan:**
1. The normal editing process proceeds until the asterisk is sensed.
2. The corresponding digit from the A-field replaces the asterisk (in the output field).
3. The editing process continues normally until the B-field word mark is sensed and removed.

**Reverse Scan:**
1. Asterisks replace zeros, blanks, and commas, to the left of the first significant digit.
2. The word mark (set during the forward scan) signals the end of editing. It is erased, and the operation is stopped.

**Floating Dollar Sign**
This feature causes the insertion of a dollar sign in the position at the left of the first significant digit in an amount field (Figure 149). The control word is written with the $ immediately to the left of the zero-suppression code.

**Note:** The control word must be larger than the A-field. Three scans are necessary to complete this editing operation.

**Special Features 83**
Decimal Control

This feature insures that decimal points print only when there are significant digits in the A-field (Figure 151).

Two scans are sufficient to complete this editing operation, unless the field contains no significant digits. Then three scans are required.

First Forward Scan:
1. When the zero suppression code (0) is sensed during editing, the corresponding digit from the A-field replaces this position.
2. A word mark is set automatically in this position in the B- (output) field.
3. Editing continues normally until the B-field word mark is sensed and removed.

Reverse Scan:
1. Blanks in the output field replace zeros and commas until the decimal point is sensed.
2. The decimal point and the digits at its right are unaltered. The automatically-set word mark is erased.
   If there are no significant digits in the field, the second forward scan is initiated. Otherwise, the edit operation stops.

Second Forward Scan:
1. Blanks replace the zeros at the right of the decimal point and the decimal point itself.
2. The operation stops at the decimal column.

<table>
<thead>
<tr>
<th>Index Location</th>
<th>Storage</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>087-089</td>
<td>080</td>
</tr>
<tr>
<td>2</td>
<td>092-094</td>
<td>094</td>
</tr>
<tr>
<td>3</td>
<td>097-099</td>
<td>099</td>
</tr>
</tbody>
</table>

Address Modification

The primary use of index locations is to modify addresses automatically by adding (absolute) the contents of an index location to an address. Both the A-address and the B-address can be modified by the contents of any index location. Only core storage addresses can be modified.

The modification of the A- and B-addresses occurs in their respective address registers. For instance, if the A-address is indexed, the indexing occurs in the A-address register. This means the original instruction in storage is in no way changed or modified.

The index locations can be used as normal storage positions when not being used as index locations.

The index factor can be placed in the index location by normal programming, such as an add or move operation. The index factor can be changed by normal arithmetic ADD and SUBTRACT instructions, following the word-mark rules for those instructions. In such cases a word mark should be set by an initialization routine in the high-order position of the index location.

Indexing Feature

Many IBM 1401 programs require that the same operations be performed repetitively, with a change only in the A- or B-address. Modifying these addresses, each time a repetitive operation is performed, requires several program steps, and additional storage locations which must be set aside for this purpose.

When the indexing feature is installed, three index locations are provided in the IBM 1401 Processing Unit to modify addresses automatically. This means that fewer instructions are needed and storage space is conserved, thus providing for faster program execution and over-all simplification of programming effort.

INDEX LOCATION ADDRESSES

Addresses assigned to the index locations are:

<table>
<thead>
<tr>
<th>Index Location</th>
<th>Storage</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>087-089</td>
<td>080</td>
</tr>
<tr>
<td>2</td>
<td>092-094</td>
<td>094</td>
</tr>
<tr>
<td>3</td>
<td>097-099</td>
<td>099</td>
</tr>
</tbody>
</table>

TAGGING

It is necessary to tag each address by an indicator so that the 1401 knows which index location is to be used to modify the instruction address.

A combination of A- and B-bits in the tens position of the A- or B-address of an instruction (AAA) (BBB), selects the index location.

<table>
<thead>
<tr>
<th>Index Location</th>
<th>Tens Positions</th>
<th>Zone Punch</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A-bit, No B-bit</td>
<td>Zero</td>
</tr>
<tr>
<td>2</td>
<td>B-bit, No A-bit</td>
<td>Eleven</td>
</tr>
<tr>
<td>3</td>
<td>A-bit, B-bit</td>
<td>Twelve</td>
</tr>
</tbody>
</table>

Address Modification

The primary use of index locations is to modify addresses automatically by adding (absolute) the contents of an index location to an address. Both the A-address and the B-address can be modified by the contents of any index location. Only core storage addresses can be modified.

The modification of the A- and B-addresses occurs in their respective address registers. For instance, if the A-address is indexed, the indexing occurs in the A-address register. This means the original instruction in storage is in no way changed or modified.

The index locations can be used as normal storage positions when not being used as index locations.

The index factor can be placed in the index location by normal programming, such as an add or move operation. The index factor can be changed by normal arithmetic ADD and SUBTRACT instructions, following the word-mark rules for those instructions. In such cases a word mark should be set by an initialization routine in the high-order position of the index location.

Note. See Address Modification section for uses and examples of the indexing feature.
Store Address Register Feature

This special feature makes it possible to store the contents of the A- and B-address registers. Thus, the A- and B-addresses of program instructions can be modified directly in cases where variable length records are being processed. This facility also makes it easier to re-enter the main program from a subroutine. Because the address of the next instruction in sequence can be retained, program re-entry is simplified.

A subroutine is a set of program instructions that are executed, if a particular condition arises during the main routine. For example, if an unequal compare occurs during processing, the program branches to a subroutine in which a special set of instructions handles the condition.

Each time a subroutine is used, some method must be employed to link it with the main program. The function of the STORE A-ADDRESS REGISTER, and STORE B-ADDRESS REGISTER instructions is to establish subroutine linkage so that upon leaving the sequence of the main program it is possible to execute the steps of the subroutine, and return to the main program where the sequence was interrupted.

Store A-Address Register

Instruction Format.

Mnemonic   Op Code   A-address
SAR         Q         xxx

Function. This instruction stores the contents of the A-address register from the previous operation, in the three-position field that has its units position defined by the A-address of the STORE A-ADDRESS REGISTER instruction.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_t + 5) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-3</td>
<td>Bp</td>
</tr>
</tbody>
</table>

Example. Store the contents of the A-address register in area labeled AADDR (0625), Figure 152.

Store B-Address Register

Instruction Format.

Mnemonic   Op Code   A-address
SBR         H         xxx

Function. This instruction stores the contents of the B-address register resulting from the previous operation, in the three-position field that has its units position defined by the A-address of the STORE B-ADDRESS REGISTER instruction.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_t + 4) \) ms.

Note. If indexing is installed in the IBM 1401, the functioning of all branch commands is altered to simplify subroutine linkage. With these alterations, each time a branch occurs as a result of one of these commands, the address of the next sequential instruction in the main routine is inserted in the B-address register. See Note under Logic Operations.

Although the subroutine may be entered from many distinct points in the main program, this use of the sbr operation makes the subroutine linkage complete.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A-3</td>
<td>Bp</td>
</tr>
</tbody>
</table>

Example. The main routine branches to a multiply subroutine labeled MULTRU (0495). This example shows the last step in the main routine and the first and last steps of the multiply routine, and illustrates subroutine linkage (Figure 153). The last instruction (labeled LAST) plus three will contain the address of the next instruction in the main routine.

Figure 152. Store Contents of A-Address Register

Figure 153. Store Contents of B-Address Register
Move Record Feature

This feature provides a special move instruction that makes it possible to move a complete record from one storage area to another, without regard to word marks within the record. This instruction is especially desirable in magnetic tape operations.

Move Characters to Record or Group Mark

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS MCM P</td>
<td>xxx</td>
<td>xxx</td>
<td></td>
</tr>
<tr>
<td>A MRCM</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Function. This operation code makes it possible to move an entire record from one core-storage area to another, regardless of the presence of word marks in either field. The A- and B-addresses specify the high-order position of the respective areas. Transmission starts from the high-order addresses, and continues until a record mark (A82 bits) or a group-mark with a word-mark (WMA8421 bits) is sensed in the A-field. The record mark or group mark transfers to the B-field.

Word Marks. Word marks within the area do not affect the MOVE CHARACTERS TO RECORD OR GROUP MARK operation. Any word marks in the B-field remain unchanged. A-field word marks are not transmitted to the B-field.

Timing. \( T = 0.0115 \times (L_A + 1 + 2 L_A) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>A + L_A</td>
<td>B + L_A</td>
</tr>
</tbody>
</table>

(The length of the A-field includes the group-mark with word-mark or record mark.)

Example. Move the tape record that has its high-order character in the location labeled TARCIN (0679) to another area of core storage beginning at the label WTAAREC (0985), Figure 154.

High-Low-Equal Compare Feature

This feature expands the compare operation to include indicators for high, low, or equal conditions. Additional d-characters are included so that the BRANCH IF INDICATOR ON instruction can test for these conditions. The basic machine permits testing for unequal conditions only.

Test High, Low, or Equal Compare

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS B</td>
<td>xxx</td>
<td>x</td>
<td></td>
</tr>
</tbody>
</table>

Function. This instruction tests the compare indicator for the result of the previous compare operation and branches to the I-address, if the condition specified by the d-character is present:

<table>
<thead>
<tr>
<th>Autocoder</th>
<th>d-character</th>
<th>Branch to I-address if:</th>
</tr>
</thead>
<tbody>
<tr>
<td>BE</td>
<td>S</td>
<td>B = A (B equals A)</td>
</tr>
<tr>
<td>BL</td>
<td>T</td>
<td>B &lt; A (B is less than A)</td>
</tr>
<tr>
<td>BH</td>
<td>U</td>
<td>B &gt; A (B is greater than A)</td>
</tr>
</tbody>
</table>

If the condition is not present, the program continues with the next sequential instruction.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 \times (L_A + 1) \) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>BI</td>
<td>dbb</td>
</tr>
</tbody>
</table>

Example. Compare the data at RECNO (0596) to the control number at CONTNO (0495). If the data at RECNO is higher than the control number at CONTNO, branch to GRTAN (0797) for the next instruction (Figure 155).

Figure 154. Move Record

Figure 155. Compare
**Sense Switches Feature**

Six sense switches make it possible to control the stored program from the console (Figure 156). The **branch if indicator on** instruction expands to allow the program to test each switch.

![Sense Switches Image](image)

Figure 156. Sense Switches

For example, a sense switch is turned on if printing only is required for a given job. If the switch is off, the data is to be printed and punched.

The program can be written to interrogate the setting of this sense switch, to determine if the data is to be printed and punched or just printed. The setting of a sense switch should not be changed while the system is operating.

**Branch if Indicator On**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>A</td>
<td>BIN + d-character</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Function.** This instruction tests the position of the toggle switch specified by the d-character. If the switch is in the on position, the next instruction is taken from the I-address. If it is off, the program continues with the next sequential instruction.

For example, a sense switch is turned on if printing only is required for a given job. If the switch is off, the data is to be printed and punched.

The program can be written to interrogate the setting of this sense switch, to determine if the data is to be printed and punched or just printed. The setting of a sense switch should not be changed while the system is operating.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>B1</td>
<td>dbb</td>
</tr>
</tbody>
</table>

**Example.** Branch to the subroutine labeled CLWARE (beginning at 0585) if sense switch G is on (Figure 157).

![Branch if Indicator On Image](image)

Figure 157. Test and Branch if Indicator On

**Compressed Tape Operations Feature**

This feature makes it possible for the IBM 1401 Data Processing System to read compressed tape prepared by the IBM 7070 Data Processing System, and to expand it within 1401 core storage for processing by the stored program.

The 7070 writes a compressed tape record under control of the WRITE WITH ZERO ELIMINATION (TWZ or TWC) instruction. By using this 7070 instruction, as many as five high-order zeros in each numerical word in storage can be eliminated while the data is recorded on magnetic tape, thus conserving tape capacity and read-write time.

The READ COMPRESSED TAPE and MOVE AND INSERT ZEROS operation codes are incorporated in the 1401 to enable it to process compressed tape records.

**Read Compressed Tape**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>MU</td>
<td>M</td>
<td>%Cx</td>
<td>xxx</td>
<td>R</td>
</tr>
</tbody>
</table>

**Example.** Branch to the subroutine labeled CLWARE (beginning at 0585) if sense switch G is on (Figure 157).

**Word Marks.** Word marks are not affected.

**Timing.** $T = 0.0115 \ (L_1 + 1)$ ms.

**Notes.** Sense switch A (last card switch) is standard in all systems equipped with an IBM 1402 Card Read-Punch. When the last card passes the second reading brushes, and switch A is in the on position, the last-card indicator is turned on and a $B \ xxx \ A$ instruction causes a branch to the I-address. It is turned off: on a run-in; if the switch is physically turned off; or if the start reset key is pressed.
presence of mode change characters (Δ) in the tape record.

**Word Marks.** Word marks are not affected.

**Timing.** T = 0.0115 (L_1 + 1) ms + T_M.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI %3x</td>
<td>Address of the inserted group mark</td>
<td></td>
</tr>
</tbody>
</table>

**Example.** Read a tape record from tape unit 2 (labeled 2) in the area of core storage labeled TP AREA (0498), Figure 158.

**Autocoder**

**Assembled Instruction:** \( M %C2 \) 498 R

Figure 158. Read Compressed Tape

**Move and Insert Zeros**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIZ</td>
<td>X</td>
<td>xxx</td>
<td>xxx</td>
</tr>
</tbody>
</table>

**Function.** The MOVE AND INSERT ZEROS instruction moves the compressed tape data that was read into core storage by a READ COMpressed TAPE instruction to another storage area, and expands each field to fill the storage locations allotted to it by the field-defining word marks. The A-address specifies the units position of the compressed tape record in core storage. *(To obtain the A-address, execute a STORE B-ADDRESS REGISTER instruction immediately following the execution of the READ COMpressed TAPE instruction. This stores the address that contains the group mark (Δ) that indicated the end of the compressed tape record.)* The B-address of the MOVE AND INSERT ZEROS instruction specifies the units position of the expanded area. The data moves from the compressed area to the expanded area, and zeros are inserted into the high-order positions of the expanded-area fields.

**Word Marks.** Word marks must be preset in the expanded area to indicate the high-order position of each field. A group-mark, with a word-mark that has also been preset by the program, must appear at the position immediately to the left of the high-order storage location of the A-field. It is this group mark that signals the end of the MOVE AND INSERT ZEROS operation.

**Timing.** T = 0.0115 (L_1 + 1 + 2 Δ L_A + Δ L_Z) ms.

**Note.** When the IBM 7070 writes a tape record, it writes each word on tape in either the alphabetic or numerical mode. Each time the mode changes from alphabetic to numerical or vice versa, a mode change character, delta (Δ), is automatically written on tape. Each time a delta is read into IBM 1401 core storage by the READ COMpressed TAPE instruction, it changes the setting of an internal switch to either the alphabetic or the numerical indication, corresponding to the mode. Thus, at the completion of the operation, the mode change switch indicates the mode setting of the last tape character read.

In the expand operation, the setting of the internal mode switch determines the method of operation. The machine works on the MOVE AND INSERT ZEROS operation from right to left, moving the data, field by field, from the compressed area to the expanded area. If the compressed area field is alphabetic, it is moved, intact, to the expanded area field, as defined by the preset word marks. To insure proper operation, the expanded alphabetic fields should be equal in length to the alphabetic fields read from tape. If the data are numerical, they are moved digit-by-digit, low order to high order, until a zone bit (indicating sign position) or Δ (delta) is encountered. If any high-order positions in the expanded field are unfilled, zeros are inserted until a word mark is sensed. During this operation, the detection of a Δ in the compressed area changes the setting of the mode switch, and the mode of operation changes from alphabetic to numerical, or vice versa.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>Address of preset group-mark with a Δ in B-field minus one word-mark -1 at immediate left of tape read-in area</td>
<td></td>
</tr>
</tbody>
</table>

**Example.** A 4-word compressed tape record is prepared by the IBM 7070:

<table>
<thead>
<tr>
<th>Field</th>
<th>Mode</th>
<th>IBM 7070 Storage Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part name</td>
<td>alpha</td>
<td>two words</td>
</tr>
<tr>
<td>Part number</td>
<td>numerical</td>
<td>one word, always plus</td>
</tr>
<tr>
<td>Unit cost</td>
<td>numerical</td>
<td>one word, always plus</td>
</tr>
</tbody>
</table>

The part number can be from two to seven digits in length. The unit cost can be from three to six
digits. The compressed tape record, written by the 7070 for a specific part, looks like this:

```
EXTbbSHANKA0475C1154E
```

The letter C is a plus sign over units digit 3. The E is a plus sign over units digit 5. The mode switch is set to alpha mode during the compressed tape operation instruction time. Therefore, it was changed to the numerical mode by the A. It is necessary to perform the expand operation before the next READ COMPRESSED TAPE instruction.

Figure 159. Read and Expand Tape Record

Figure 159 shows the three program steps that read and expand the compressed tape record for this example. A group-mark with a word-mark has been preset in position 6099.

The READ COMPRESSED TAPE instruction reads into IBM 1401 storage locations 0700-0721:

```
EXTbbSHANKA0475C1154E
```

After the operation, the B-address register contains the address of the group mark (0721). The STORE B-ADDRESS REGISTER instruction modifies the MOVE AND INSERT ZEROS instruction so that the A-address contains 721:

```
X 000 724 before
X 721 724 after
```

The maximum size of the compressed tape record is 24 positions (to accommodate a 10-character part name, a 7-digit part number, a 6-digit unit cost, and the mode-change character). Thus, the expanded area is defined as locations 716-721, to the expanded field locations, 719-724. The detection of the zone bits in the letter C of the part number indicates the units position of the next field. Because no mode change character is detected, the mode switch continues to indicate numerical. A zero is inserted in position 718. The preset word mark in that position stops the insertion of additional zeros in the unit cost expanded field. In a similar way, the 0475C part number moves from positions 711-715 to positions 713-717, and zeros are inserted in positions 712 and 711, halted by the word mark in location 711. The A in position 710 indicates the units position of the next field (part name).

The A changes the setting of the mode change switch from numerical to alphabetic. In the alphabetic mode, characters are moved without insertion of zeros.

The expanded area in core storage after the operation looks like this:

```
EXTbbSHANKA000475C01154E
```

Note: To conserve storage, the compressed area overlaps with the expanded area in this example.

### Column Binary Feature

This feature makes it possible for the IBM 1401 Data Processing System to process column-binary-coded cards and magnetic tapes used with IBM scientific data processing systems such as the IBM 704, IBM 709, and IBM 7090.

The reading, writing, and logic operational facilities of the 1401 can be used to process the binary-coded data.

The operation codes and instructions described in this section are used whenever:

1. The information to be read or written is in binary cards or binary coded tapes.
2. There are invalid multiple punches in cards containing the standard IBM card code. This means that cards, coded with several punches in one column, that were designed for other machines can be entered by using this feature.

### Read Column Binary

#### Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS R RCB</td>
<td>1</td>
<td>C</td>
</tr>
</tbody>
</table>

#### Function.

This instruction causes the card at the read station to be read into the 1401 in the binary mode. During the reading of a card, the read cycle is
divided into two parts, and three different areas in storage receive the data. Card cycle 9 through 4 time uses the normal read addresses 001-080, and column binary read area addresses 501-580. The other portion of the card cycle (3-12 time) uses addresses 001 through 080, and 401 through 480. Note that storage locations 001-080 are common read-in locations for both halves of the read cycle.

At the completion of this read operation, a BCD coded image of the card is stored in addresses 001 through 080, just as in normal card reading. The portion of the card that contains column-binary information appears as hash in the corresponding addresses 001-080, and the portion of the card that contained alphabetical characters is stored in BCD code in addresses 001 through 080. Storage addresses 401-480 and 501-580 contain the true card image. In these areas all alphabetical characters and all column-binary information appear as illustrated in Figure 160:

If, for example, the following information is recorded in a binary card and appears in 1401 storage:

Card-column 1 contains an IBM card code H which is represented by a 12-punch and an 8-punch.

```
Storage Locations      Contains
----------          -------
001                BA8
401                B
501                2
```

```
Storage
----------
Addresses            Storage
----------
001                C
401-480             B
501-580             A
001-080             1
```

```
BCD CODE          PUNCHES IN CARD COLUMN
----------          ----------
Storage
----------
Addresses         Storage
----------
001              C
401              B
501              A
```

Card-column 2 is part of a binary field and contains punches in rows 12, 0, 1, 3, 4, 5, 6, 7, and 9.

```
Storage Locations      Contains
----------          -------
002                hash
402                CB441
502                BA441
```

Word Marks. Word marks are not affected.

Timing. \( T = .0115 (L_1 + 1) \text{ ms} + I/O. \)

Notes. The READ COLUMN BINARY instruction (1C) cannot be combined with any other operation.

Read checking of input data is unchanged, except for the validity check, which is not performed because all characters read are considered valid.

Address Registers After Operation.

```
NSI   dcb  481
```

Example. Read the card at the IBM 1402 read station in the column binary mode (Figure 161).

```
Figure 161. Read Column Binary
```

Read Column Binary and Branch

Instruction Format.

```
Mnemonic   Op Code   I-address   d-character
SFS R 1  xxx     C
```

Function. This is the same as READ COLUMN BINARY, except that the next instruction is at the I-address.

Word Marks. Word marks are not affected.

Timing. \( T = .0115 (L_1 + 1) \text{ ms} + I/O. \)

Address Registers After Operation.

```
NSI   BI  481
```

Example. Read the card at the read station in column binary mode, and branch to BININ (0922) for the next instruction (Figure 162).
Punch Column Binary

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS P</td>
<td>A PCB</td>
<td></td>
</tr>
</tbody>
</table>

Function. This instruction, executed in two parts, requires that the information be stored in two different areas. Information that is to be punched in rows 12-3 (card columns 1-80) is stored in locations 401-480. Rows 4-9 of the card (columns 1-80) are punched from storage locations 501-580.

Using the same data shown in the READ COLUMN BINARY example, the card is punched:

Storage Locations
401 B 12
501 2 8

This combination causes the H to punch in card column 1.

Card column 2 is punched in rows 12, 0, 1, 3, 4, 5, 6, 7, and 9 as transferred from:

Storage Locations
402 C0841 12, 0, 1, 3
502 BA941 4, 5, 6, 7, 9

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_1 + 1) \) ms + I/O.

Note. The PUNCH COLUMN BINARY instruction cannot be combined with any other operation. The punch checking of output data is unchanged.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>dUB</td>
<td>181</td>
</tr>
</tbody>
</table>

Example. Punch a card in the column binary mode, and branch to BINCD (0986) for the next instruction (Figure 164).
Branch if Bit Equal

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>BBE</td>
<td>W</td>
<td>xxx</td>
<td>xxx</td>
<td>x</td>
</tr>
</tbody>
</table>

Function. The d-character can contain any character or any combination of bits (BA 8421) that can exist in a single position of the 1401 core storage. If the character at the B-address contains any bit that matches any bit in the d-character, the program branches to the I-address. Otherwise, the program continues normally.

Word Marks. Word marks are not affected.

Timing. T = 0.0115 (L + 2) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>Bi</td>
<td>B-1</td>
</tr>
</tbody>
</table>

Example. Examine the storage location labeled UNPOS (0759) for a match in the d-character bit configuration. The d-character is a 9 (8- and 1-bits). Therefore, if the character contains either an 8- or 1-bit, the program branches to BITEST (0985), Figure 165.

Move and Binary Decode

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>MCW</td>
<td>M</td>
<td>xxx</td>
<td>xxx</td>
</tr>
</tbody>
</table>

Function. This instruction arranges data in the correct order for tape writing. The A-address is usually 572 or 580, depending on whether the card has 72 or 80 columns of binary data. It specifies the units or low-order position of the record. The B-address specifies the low-order position of the area in 1401 storage from which the record is to be written on tape by a WRITE BINARY TAPE instruction. The d-character specifies that this is a move and binary decode operation.

At the completion of this operation, the tape-write area (B-address) contains the data from both the 401-480 and 501-580 areas in this sequence:

401, 501, 402, 502, 403, etc.

Word Marks. A word mark must be preset in the 401-480 area to signify the high-order character of the record (normally in location 401). Any wordmark encountered stops the transfer to the tape-write area.

Timing. T = 0.0115 (L + 1 + 2L) ms.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>400 address of the 400 area preset word mark + 99.</td>
<td>B-Lb</td>
</tr>
</tbody>
</table>

Example. Write the data in 401-480 (labeled CLB14A) and 501-580 (labeled CLB15A) areas in the tape-write area labeled BITPAR (2080), Figure 166.

Binary Tape Instructions

Column-binary information should be recorded on magnetic tape in its logical order. To do this it is necessary to arrange the data from storage locations 401-480 and 501-580 in the following sequence in a tape write area:

Address 401 followed by 501, followed by 402, 502, etc., until the entire area is so arranged.

This puts the 12-3 data next to the 4-9 data from the same card column, in the proper sequence for writing on tape.

This arranging can be done automatically by a MOVE AND BINARY DECODE instruction.

Conversely data read from a tape unit can be arranged as a card image with 12-3 punches in the 401-480 area and 9-4 punches in the 501-580 area. The MOVE BINARY CODE operation performs this function.

Column-binary information must be written on magnetic tape in the odd-parity mode. This means that an odd number of bits must be recorded in each position of the tape record. The WRITE BINARY TAPE and READ BINARY TAPE instructions cause the data to be recorded in this manner.

Figure 165. Branch on Bit Test

Figure 166. Move and Binary Decode
Move Binary Code

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>MCW</td>
<td>M</td>
<td>xxx</td>
<td>B</td>
</tr>
<tr>
<td>A</td>
<td>MB</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Function.** Information read from magnetic tape is arranged into a coded card image in binary form, when this instruction is used. Data to be punched in rows 12-3 is transferred to max 1401 storage area locations 401-480, and data to be punched in rows 4-9 is stored in the 501-580 area for punching. The A-address specifies the units position of the tape read-in area, and the B-address is usually 572 or 580, depending on the number of columns to be punched. The d-character (B) specifies a move binary code operation.

**Word Marks.** A word mark must be preset in the high-order position of the B-field (normally 401) to stop the operation. A word mark in the high-order position of the A-field can be set to stop the operation after the following B-cycle if desired.

**Timing.** \( T = 0.0115 \left( L_t + 1 + 2L_b \right) \) ms.

**Address Registers After Operation.**

|-------------|-------------|-------------|

**Example.** Move the data from the tape read-in area, labeled BITPAR (2080), to the column-binary punch area CLB14A (0401-0480) and CLB15A (0501-0580), and store it in the proper sequence for punching (Figure 167).

Read Binary Tape

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>A-address</th>
<th>B-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>MU</td>
<td>M</td>
<td>%Bx</td>
<td>W</td>
</tr>
<tr>
<td>A</td>
<td>WTB</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Function.** A tape record written in binary form is read into core storage, beginning at the location specified by the B-address and ending at an inter-record gap between tape records or a group-mark with a word-mark in core storage. The A-address indicates the tape unit selected, and signals the column-binary tape operation. The d-character (R) specifies a read operation.

**Word Marks.** Word marks are not affected.

**Timing.** \( T = 0.0115 \left( L_t + 1 \right) \) ms + \( T_m \).

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>%Bx</td>
<td>Group-mark + 1</td>
</tr>
</tbody>
</table>

**Example.** Read a tape record in the binary mode on the tape unit labeled 4, with the data stored in the area labeled BTPOUT (2001) and ending at the group-mark with a word-mark sensed in 1401 core storage (Figure 168).
Example. Read the binary tape record from the tape unit labeled 5 into the area of 1401 core storage labeled BTPIN (2080) and ending at the group-mark with a word-mark sensed in core storage or at the first inter-record gap encountered in the tape record (Figure 169).

**Autocoder III label**

Assembled Instruction: M %B5 !80 R

Figure 169. Read Tape Binary

**Numerical Print Feature**

The numerical print feature for the IBM 1403 Printer has been designed for those businesses having certain 1401 applications that require no alphabetic printing. For example, banks, insurance companies, and utilities prepare many reports with only numerical printing. With this feature, the time required to produce these reports can be reduced by as much as 50 per cent. The manufacturing, wholesaling, and retailing levels of other industries can also use this feature for the many applications in which reports are (or can be) numerically coded.

With this feature, the systems user can switch from the alphanumerical to the numerical mode, simply by changing the chain cartridge in the 1403. The numerical chain is composed of 15 character sets, with 16 characters (digits 0 through 9 & , * - ) in each set. In the numerical mode, the 1403 can print 1285 lines per minute — more than twice as fast as in the alphanumerical mode.

To change from one mode to another, an operator, with no special tools, removes one chain and replaces it with the other. Before locking the new cartridge in place, it is only necessary to move the chain enough to permit the chain drive to engage. When a chain cartridge is placed in the 1403, the corresponding mode is selected automatically. If the printer is in the numerical mode, characters other than the 16 specified for numerical printing cause a print check error.

**Interchangeable Chain Cartridge Adapter**

Many scientific and commercial applications require distinctive type styles for particular printing jobs. This special feature for the IBM 1403 Printer allows chain cartridges to be interchanged.

With this feature, an operator can insert an interchangeable chain cartridge with a different type font, type style, or special character arrangement.

**Operating Instructions**

1. Turn off the power in the system.
2. Raise the counterbalanced cover of the printer.
3. Unlock and swing back the print unit by using the print unit release lever.
4. Open the top ribbon cover, remove the lower ribbon roll, slide ribbon from under the ribbon correction roll and store the lower ribbon roll on the ribbon cover.
5. Grasp the chain cartridge handles and raise them to a vertical position. This unlocks the cartridge from the print unit.
6. Lift straight up on the cartridge handles until the cartridge clears the locating pins. Place the cartridge on a surface that tolerates oil and ink. (A container is provided for storing the chain cartridge when it is not in use.)
7. Install the second cartridge by grasping the handles and lifting the cartridge into position over the locating pins. Check the bottom of cartridge for foreign material before placing it on the printer.
8. Position the cartridge gently over its guide pins and release the handles. Do not force either handle down. The end opposite print hammer 132 should fully settle on the base, the end opposite hammer unit 1 is not in position at this time.
9. Rotate the chain in the normal printing direction (counterclockwise as seen from the top), and at the same time press down on the button located on the top cover between the chain cartridge position and the print-timing dial. When you can no longer rotate the chain, it is properly positioned at the end opposite hammer number one.
10. Lower the chain cartridge handles to their horizontal positions. Do not force the handles. If force is required, the chain is not properly in position and steps 7 through 9 should be repeated.
11. Replace the lower ribbon roll and make sure the ribbon is positioned under the ribbon correction roll. Close the ribbon shield, latch the print unit in place, and close the counterbalanced cover. Turn the power on and resume system operation.
Interchangeable 51-Column Read Feed

The interchangeable 51-column read feed (including file feed) permits feeding either 51-column cards or standard 80-column cards in the read feed of the IBM 1402 Card Read-Punch.

The 51-column card is commonly used for charge sales slips, postal money-order forms, installment payments, inventory cards, and many other applications. Using an interchangeable feed allows direct entry to the data processing system from the stub card. This eliminates the need for reproducing 51-column cards into standard 80-column cards.

To adapt the read feed for 51-column-card operation, the operator installs a tray and hopper side plates on the read file feed, and adjusts the stackers on the read side.

Normal operations of the IBM 1402 Card Read-Punch can be performed with 51-column cards in the read feed. For example, a file of 51-column cards can be processed in the read feed while the results are punched in 80-column cards in the punch feed. However, when the stackers are adjusted to accept 51-column cards, no cards from the read feed can be selected into stacker 8/2.

Machine Features

Modifying the read file feed and stackers readily adapts the IBM 1402 Card Read-Punch for processing 51-column cards.

Modifying the File Feed

An adapter tray (Figure 170), placed on the file-feed magazine, accommodates the 51-column cards. A modified card weight enables feeding the last cards from the hopper. Inserting two hopper side plates (Figure 170) positions the 51-column cards at the center of the feed. Thumbscrews fasten the side plates to the hopper. Joggers align the cards in the hoppers, as in standard operation.

In 51-column-card operation, the first column of the card corresponds to column 15 of an 80-column card.
and is therefore read by brush 15; the last column corresponds to column 65 and is read by brush 65. A factor of 14 relates the card column to the reading brush. A switch for regulating the storing of information from a 51-column card is physically located in the 1402. It is automatically turned on when the stacker guide is pulled forward for stacking of 51-column cards.

When the switch is on, the information from a 51-column card is read into the read-in area of storage beginning in position 015 and extending to position 065. Positions 001-013 and 066-080 are not altered during a 51-column operation. Position 014 is used for cycle timing.

This switch provides for the proper loading of instructions from 51-column cards when the load key is used. With the switch on, pressing the load key causes a word mark to be set in storage position 015 and automatically clears position 016 to 065 of word marks and places 015 in the I-address register. It should be noted that a factor of 14 must be added to the read-in area addresses.

### Adjusting the Secondary Stackers

The operator adjusts the stacker guide (Figures 171 and 172) at the rear of stackers NR and 1 to accommodate 51-column cards. A finger hole permits pulling the guide forward to reduce the depth of the stacker. A spring latch holds the guide securely in either the 51- or 80-column-card position.

A pivot-plate assembly (Figures 171 and 172) adapts the front of stackers NR and 1 for stacking either 51- or 80-column cards. The 51-column pivot plate with card-retaining levers swings down and fastens to the stacker separators. This assembly provides a lower pivot for properly stacking the 51-column cards.

For standard 80-column operation, the operator pulls each auxiliary pivot-plate assembly forward and then places it under the cover.

Modified card-deck supports (Figures 171 and 172) for stackers NR and 1 permit stacking 51-column cards, standard cards, and the scored cards processed by the machine. The capacity of each of these stackers is 800 cards.

---

**Figure 171.** 51-Column Adjustable Stackers
Left-scored, 51-column cards must not contain punches in the columns that are the equivalent to columns 28 and 29 of an 80-column card. If these columns are punched, misfeeding of cards may occur. The first punch in a 51-column card may be in the column equivalent to column 30 of an 80-column card.

When changing from 51-column to 80-column operation, a few blank 80-column cards should be run through the read feed to ensure that the card transport is clear.
To set up the IBM 1402 Card Read-Punch to feed 51-column cards in the read feed:

1. Position the side plates in the hopper, and fasten firmly by turning the knurled thumbscrews. Be careful not to interfere with the card lifters.
2. Place the 51-column-card tray over the file-feed magazine.
3. Reach into stackers 11 and 1 and, using the finger hole, pull the guide forward until it latches.
4. Raise the cover over the auxiliary pivot-plate assemblies, lower one assembly partially, and then slide the main pivot-plate to the rear until it latches.
5. Swing the auxiliary pivot-plate assembly down until it latches to the stacker separators. (Repeat steps 4 and 5 for the other pivot-plate assembly.)

Reverse this procedure to return to standard card-feeding. Note: Handle and store the adapter tray and hopper side plates carefully to avoid damaging them.

### Early Card Read Feature

The early card read feature for the IBM 1402 Card Read-Punch minimizes the decrease in card-reading speed caused by lengthy processing routines. In such routines, the card-reading mechanism can engage sooner, thus reducing the time between the reading of cards.

Normally, if processing time exceeds 10 ms in a basic card-read cycle, the rated card-reading speed decreases. This occurs because of the mechanical structure of the card-read feed. There is only one time during the read cycle when the feeding mechanism can be engaged. If a READ A CARD instruction is given too late (processing time exceeds 10 ms), a card-read cycle is skipped, thus reducing the input speed from 800 to 400 cards per minute. Similarly, if the time required for processing exceeds 85 ms, two read cycles are skipped, and the input speed is reduced to 266 cards per minute.

The early card read feature provides two additional points (clutch points) where the feeding mechanism can engage. When processing time between cards exceeds 10 ms, the feed mechanism can engage 50 ms sooner than before. The time between card feeding is reduced to 100 ms rather than 150 ms. Instead of a 50 per cent reduction in the rated speed (to 400 cpm), there is only a reduction of 25 per cent (to 600 cpm). See section on Timing.

---

**Processing Overlap Feature**

The processing overlap special feature for the IBM 1401 Data Processing System provides, for many applications, great reductions in over-all job time. The high-speed processing and input-output abilities of the 1401 now can be used with maximum efficiency. Jobs requiring extensive input-output operations and lengthy programming now can be performed at or near maximum speeds. The actual increase in efficiency and consequent saving in job time varies with the specific program, and depends on the input-output requirements of the particular application.

The processing overlap feature allows processing to be interrupted in order to take input-output cycles. A character can be read, written, or punched between processing cycles.

Serial input-output devices used with the IBM 1401 Data Processing System, such as the IBM 1011 Paper Tape Reader and the IBM 1419 Magnetic Character Reader, can have their operations performed in overlap mode. In some cases, entire input-output operations can be overlapped; in others, partial overlapping can occur.

Job time required for card input-output applications is reduced because the processing operation is not interlocked during card reading or punching. This is
also true when reading or writing magnetic tape. Tape operations and processing can occur on alternate cycles. This ability can result in appreciable increase in job time economy.

The processing overlap special feature makes the IBM 1401 Data Processing System a more powerful tool for use in the field of data processing.

**Data Flow**

The overlap special feature requires the addition of an O-register and an O (overlap)-address register to the processing unit (Figure 173). These registers function in much the same manner as the A- and B-registers and the I-, A-, and B-address registers. That is, the O-register is used for movement of data when operating with an I/O unit. The O-address register is used to keep track of the location of data that is moved by an I/O operation. These registers, when used by the program, operate on alternate cycles with the normal registers. Thus the 1401 takes processing cycles and then an input-output cycle, when required, instead of interlocking processing while an input-output operation is being performed. This means a savings in overall job time.

**IBM 1402 Card Read-Punch**

When the system is in the overlap mode, processing can occur during the card cycle in which data is read from, or punched into, a card. Processing does not occur during the time used for either read or punch scanning, but alternates with scanning.

At the beginning of each read or punch scan, the character in the A-register is sent to the a-register (Figure 174) and the a-address register is set at the column being scanned. When the read or punch scan of the character is completed, the character in the O-register is sent back to the A-register. At this point processing resumes.

**Tape Operations**

During a read operation, data from magnetic tape enters the A-register and then moves into core storage. During an overlap read operation, data necessary for processing is contained in the A-register. In order to save the data, it is sent to the O-register until recalled for the next processing cycle, before data is read in from tape to the A-register (Figure 175). The O-address register contains the storage address in which the data
being read from magnetic tape is to be stored, at the same time that the I-, A-, and B-address registers keep track of the processing operation being performed.

During a write operation, data from core storage passes through the B-register and enters the O-register (Figure 176). It is sent to magnetic tape from the O-register. Processing cycles use the A- and B-register. The O-address register contains the storage address of the data being written on magnetic tape. At the same time the I-, A-, and B-address registers keep track of the processing operation being performed.

To store the contents of the O-address register at the end of an overlapped tape operation the following operations must be performed:

\[
\text{BRANCH IF TAPE BUSY INDICATOR ON.}
\]

This instruction allows the contents of the O-address register to be transferred to the B-address register (with the advanced programming feature) if the indicator is not on.

Then

\[
\text{Store B-address register}
\]

Other Input-Output Units

Input-output units such as the IBM 1419 Magnetic Character Reader, when operating in the overlap mode, function in the same manner as a magnetic tape unit. Overlapping and processing occurs just as if the operation were a tape read or write operation.

Processing Overlap Instructions

To signal the 1401 that an operation to be performed should be done in the overlap mode, special instructions and A-address changes must be used in the program.

A-ADDRESS

The hundreds position of the A-address of a tape or input-output unit (not 1405 or 1407) instruction is changed from % to @. The @ (at) symbol signals the 1401 processing unit that the operation to be performed should be done as an overlap operation. The @ symbol can be used to signal an overlap operation with the:

- IBM 1011 Paper Tape Reader
- IBM 1419 Magnetic Character Reader
- IBM 729 II, IV, V Magnetic Tape Unit
- IBM 7330 Magnetic Tape Unit

Overlap On

Instruction Format.

\[
\text{Mnemonic} \quad \text{Op Code} \quad d-character
\]

Function. This instruction sets the 1401 processing unit in the overlap mode. All 1402 read, punch, and combination card instructions that follow this instruction in the program are performed in the overlap mode.

Word Marks. Word marks are not affected.

Timing. \(T = .0115(L_1 + 1) \text{ ms.}\)

Address Registers After Operation.

- B-Add. Reg.

Example. Set the 1401 in the overlap mode (Figure 177).

Overlap On and Branch

Instruction Format.

\[
\text{Mnemonic} \quad \text{Op Code} \quad I-address \quad d-character
\]

Function. This is the same as OVERLAP ON, except that the next instruction is taken from the I-address.

Word Marks. Word marks are not affected.

Timing. \(T = .0115(L_1 + 1) \text{ ms.}\)

Address Registers After Operation.

- B-Add. Reg.

Assorted Instructions: \(K \ S\)

Figure 177. Overlap On
Example. Set the 1401 in the overlap mode and branch to CDROUT (0950) for the next instruction (Figure 178).

Autocoder

Assembled Instruction: K 950 $

Figure 178. Overlap On and Branch

Overlap Off

Instruction Format.

Mnemonic  

Op Code  

d-character  

K

Function. This instruction returns the 1401 processing unit to normal operation. All card input-output instructions following this instruction are performed without the overlap feature.

Word Marks. Word marks are not affected.

Timing. \[ T = 0.0115 \ (L_1 + 1) \text{ ms.} \]

Address Registers After Operation.


B-Add. Reg.  

NSI  

B

Example. Place the 1401 in normal operation (Figure 179).

Autocoder

Assembled Instruction: K *

Figure 179. Overlap Off

Overlap Off and Branch

Instruction Format.

Mnemonic  

Op Code  

I-address  

d-character  

K

Function. This instruction is the same as OVERLAP OFF, except that the next instruction is taken from the I-address.

Word Marks. Word marks are not affected.

Timing. \[ T = 0.0115 \ (L_1 + 1) \text{ ms.} \]

Reset Overlap

Instruction Format.

Mnemonic  

Op Code  

d-character  

K

Function. This instruction is only operative when the serial input-output adapter special feature is installed. If the system is performing an overlapped serial I/O operation when this instruction is given, the I/O device is disconnected from the system and the following indicators are turned on: I/O error (d-character 1), and tape transmission error (d-character 0).

Word Marks. Word marks are not affected.

Timing. \[ T = 0.0115 \ (L_1 + 1) \text{ ms.} \]

Note. If an overlap tape operation is in process when this instruction is given, the tape unit is disconnected and the tape transmission error indicator is turned on.

Address Registers After Operation.


B-Add. Reg.  

NSI  

B

Example. Reset overlap mode and return 1401 to normal operation (Figure 181).

Autocoder

Assembled Instruction: K *

Figure 181. Reset Overlap
Reset Overlap and Branch

Instruction Format.

Function. This instruction is the same as RESET OVERLAP, except that the next instruction is taken from the I-address.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_I + 1) \text{ ms} \).

Address Registers After Operation.

Example. Reset overlap mode and branch to routine labeled NORMOP (1755) for the next operation (Figure 182).

Address Registers After Operation.

Example. Read the record from the tape unit labeled 2 to the 1401 core-storage area labeled OUTPAR (0419). The tape-record characters are moved into core storage until the transfer is stopped by an inter-record gap in the tape record, or a group-mark with a word-mark is sensed in 1401 core storage (Figure 183).

For autocoder coding see the IBM 1401 Autocoder Bulletin (J24-1434).

Autocoder assembles Instruction: \( X 5 5 \) [3]

Assembled Instruction: \( X 5 5 \) [3]

Figure 182. Reset Overlap and Branch

Read Tape in Overlap Mode

Instruction Format.

Function. The tape unit specified in the A-address is started. The d-character specifies a tape-read operation. The \( @ \) in the hundreds position of the A-address indicates that this operation is to be performed in the overlap mode.

The B-address specifies the high-order position of the tape read-in area of storage. The machine reads magnetic tape until either an inter-record gap in the tape record or a group-mark with a word-mark in core storage is sensed. The inter-record gap indicates the end of the tape record, and a group-mark (code CBA 8421) is inserted in 1401 core storage at this point.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_I + 1) \text{ ms} + T_M \). (See section on Tape Timing.)

Note. When reading high-density tape from an IBM 729 II, IV, or V, only start time of the tape unit can be overlapped (minimum overlapped times are 6.6 ms in 729 II; 4.4 ms in 729 IV; or 6.6 ms in 729 V).

Reading low-density tape from a 729 II, IV, or V, or a 7330 in either high- or low-density permits overlapping of start time and processing time. That is, processing cycles are taken between characters read from magnetic tape.

Address Registers After Operation.

Example. Write tape in overlap mode.

Instruction Format.

Function. The tape unit designated in the A-address is started. The d-character specifies a tape-write operation. The data from core storage is written on the tape record. The \( @ \) in the hundreds position of the A-address indicates that this operation is to be performed in the overlap mode.

The B-address specifies the high-order position of the record in storage. A group-mark with a word-mark stops the operation. The group-mark with a word-mark causes an inter-record gap.

Word Marks. Word marks are not affected.

Timing. \( T = 0.0115 (L_I + 1) \text{ ms} + T_M \). (See section on Tape Timing.)

Note. When writing high-density tape on an IBM 729 II, IV, or V, only start time of the tape unit can be overlapped (minimum overlapped times are 7.3 ms in 729 II; 4.9 ms in 729 IV; or 7.3 ms in 729 V).

Special Features 101
Writing low-density tape on a 729 II, IV, or V, or a 7330 in either high- or low-density permits overlapping of start time and processing time. That is, processing cycles are taken in the time between the writing of characters on magnetic tape.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>Group mark + I</td>
</tr>
</tbody>
</table>

Example. Transfer the contents of the core-storage area labeled TPOUT (0525) to the tape unit labeled 3. The transfer of data is stopped by a group-mark with a word-mark in core storage (Figure 184).

For autocoder coding see the IBM 1401 Autocoder Bulletin (J24-1434).

Assembled Instruction: M @U3 525 W

Figure 184. Write Tape in Overlap Mode

Read a Card in Overlap Mode

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>1</td>
</tr>
</tbody>
</table>

Function. This code, if overlap is on, causes a card to feed, and causes all 80 columns of information to be read into core-storage locations 001 through 080 while processing continues.

Word Marks. Word marks are undisturbed.

Timing. $T = 0.0115(L_1 + 1) ms + I/O$.

Note: Processing is interrupted each time a row in the card is scanned (9-row, 8-row, etc.). The bits in a column are developed in the A-register, and the O-address register keeps track of which column is being scanned. While the scanning operation is being performed, the data required for processing is temporarily stored in the O-register. When the row scan is completed, the data in the O-register is sent back to the A-register and processing continues.

The read-in area of core storage must not be addressed while an overlap operation is being performed.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>081</td>
</tr>
</tbody>
</table>

Example. Read a card in overlap mode (Figure 185).

Read in Overlap Mode and Branch

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>R</td>
<td>1</td>
<td>xxx</td>
</tr>
</tbody>
</table>

Function. This is the same as the READ A CARD instruction, except that the next instruction is taken from the I-address instead of from the next, sequential instruction address. The program branch occurs when the instruction is processed, that is, before card reading is completed.

Word Marks. Word marks are not affected.

Timing. $T = 0.0115(L_1 + 1) ms + I/O$.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>081</td>
</tr>
</tbody>
</table>

Example. Read a card in the overlap mode, and branch to the location labeled OVERCD (1500), Figure 186.

Punch a Card in Overlap Mode

Instruction Format.

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>4</td>
</tr>
</tbody>
</table>

Function. This code, if overlap mode is on, causes the data in storage locations 101 through 180 to be punched into an INX card while processing continues.

Word Marks. Word marks are not affected.
Timing. \( T = 0.0115 \left( L_t + 1 \right) \text{ ms} + I/O. \)

**Note.** Processing is interrupted each time a row in the card (12-row, 11-row, etc.) is scanned for punching. The data to be punched is developed in the B-register, and the O-address register keeps track of which column is being punched. While the punch-scan operation is being performed, the data required for processing (from the A-register) is temporarily stored in the O-register. When the row punch-scan is completed, the data in the O-register is sent back to the A-register and processing continues.

The punch-out area of core storage must not be addressed while an overlap operation is being performed.

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>181</td>
</tr>
</tbody>
</table>

**Example.** Feed a card, and punch in overlap mode (Figure 187).

**Figures 187.** Punch in Overlap Mode

---

**Punch and Branch in Overlap Mode**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>4</td>
<td>xxx</td>
</tr>
</tbody>
</table>

**Function.** This is the same as the PUNCH A CARD instruction, except that the next instruction is taken from the I-address instead of from the next, sequential instruction address. The branch occurs when the instruction is processed; that is, before punching has been completed.

**Word Marks.** Word marks are not affected.

**Timing.** \( T = 0.0115 \left( L_t + 1 \right) \text{ ms} + I/O. \)

**Address Registers After Operation.**

<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>181</td>
</tr>
</tbody>
</table>

**Example.** Punch a card in overlap mode, and branch to the location labeled OVSTAT (1758), Figure 188.

**Figures 188.** Punch and Branch in Overlap Mode

---

**Branch if Indicator On**

**Instruction Format.**

<table>
<thead>
<tr>
<th>Mnemonic</th>
<th>Op Code</th>
<th>I-address</th>
<th>d-character</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPS</td>
<td>B</td>
<td>xxx</td>
<td>A BIN</td>
</tr>
</tbody>
</table>

**Function.** The d-character specifies the indicator tested. If the indicator is on, the next instruction is taken from the I-address. If the indicator is off, the next sequential instruction is taken. Figure 189 shows the symbols that are valid d-characters and the indicators they test.

**Indicators. Reader Busy.** This indicator turns on when the IBM 1401 is performing an overlapped read operation. The indicator automatically turns off when the overlapped read operation is completed.

**Punch Busy.** This indicator turns on when the 1401 is performing an overlapped punch operation. The indicator automatically turns off when the overlapped punch operation is completed.

**Tape or Input-Output Busy.** This indicator turns on when the 1401 is performing an overlapped magnetic tape or input-output (1419, or 1011) operation. The indicator automatically turns off when the overlapped operation is completed. **Note:** Without the advanced programming special feature, the B-address register is reset when the next sequential instruction is taken. If the system has the advanced programming special feature, the B-address register

**Figures 189.** Processing Overlap Indicators

---

<table>
<thead>
<tr>
<th>d-CHARACTER</th>
<th>INDICATORS</th>
</tr>
</thead>
<tbody>
<tr>
<td>H</td>
<td>Reader Busy</td>
</tr>
<tr>
<td>I</td>
<td>Punch Busy</td>
</tr>
<tr>
<td>J</td>
<td>Tape or Input-Output Busy</td>
</tr>
</tbody>
</table>
is reset and the contents of the O-address register are transferred to the B-address register before the next sequential instruction is taken.

Word Marks. Word marks are not affected.

Timing. \( T = 0.115 \left( L_i + 1 \right) \text{ms} \),
\[ T = 0.115 \left( L_i + 2 \right) \text{ms} \] for tape or I/O busy indicator if it is not ON.

Address Registers After Operation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>NSI</td>
<td>BI</td>
<td>dbb</td>
</tr>
</tbody>
</table>

Example. Test the reader-busy indicator, and branch to a routine beginning at the location labeled RDBUSY (0891), if the indicator is ON (Figure 190).

<table>
<thead>
<tr>
<th>SPS</th>
<th></th>
<th>CANAL</th>
<th>OPERATE</th>
<th>OPERATE</th>
<th>SYST</th>
<th>REA</th>
<th>PUN</th>
<th>ALTO</th>
<th>BRA</th>
<th>LIT</th>
<th>RDBUSY</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Autocoder

<table>
<thead>
<tr>
<th>Label</th>
<th>Operation</th>
<th>Address</th>
<th>Address</th>
<th>Address</th>
<th>Address</th>
<th>Address</th>
<th>Address</th>
<th>Address</th>
<th>Address</th>
<th>Address</th>
<th>Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>BS10</td>
<td>RDBUSY</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
</tr>
</tbody>
</table>

Assembled Instruction: B BS10 RDBUSY

Figure 190. Branch if Reader-Busy Indicator On

Programming Considerations

A one-character B-field arithmetic operation, where recomplementing can occur, must not be executed while an input-output unit is operating in the overlap mode.

In general, no other input-output operation can be initiated until the preceding overlapped I/O function is completed. The exceptions to this rule are:

1. If the 1401 is equipped with the print storage special feature, the following print instructions can be executed simultaneously with an overlapped input-output function:

   PRINT
   PRINT AND BRANCH
   PRINT WORD MARKS
   PRINT WORD MARKS AND BRANCH

2. When performing a normal CARD READ instruction in the overlap mode, the following output instructions can be performed:

   PUNCH
   PUNCH AND BRANCH
   PRINT AND PUNCH
   PRINT, PUNCH AND BRANCH
   PRINT WORD MARKS AND PUNCH
   PRINT WORD MARKS, PUNCH AND BRANCH

   Note: The last four instructions involving the printer would require the print storage feature to be installed on the system in order for it to be overlapped.

It is possible to maintain an approximate ratio of three card read operations to one card punch operation with the proper placement of BRANCH ON BUSY instructions. That is, after a punch operation, test for punch busy. If it is not busy, branch back to punch. If it is busy, test for reader busy. If the reader is not busy, perform a read operation and branch back to the punch operation. If the reader was busy, go to the main program (which eventually does the punch operation).

3. When performing a normal CARD PUNCH instruction, the following additional input-output instructions can be performed:

   READ
   READ AND BRANCH
   PRINT AND READ
   PRINT, READ AND BRANCH
   PRINT WORD MARKS AND READ
   PRINT WORD MARKS, READ, AND BRANCH

   Note: The last four instructions would require print storage in order to be overlapped.

4. If performing a tape operation in the overlap mode, a READ or PUNCH instruction in the overlap mode can be given successfully so long as the total tape operation is completed before the first read or punch scan (inability to store address of last position of tape record can result).

5. The last card indicator is turned ON during the time the last card is being read. Therefore, to ensure proper reading of the card, test for a reader busy condition before the branch on last card test is given.

The last card indicator is turned OFF if the system is in an overlap mode, and the program completes a successful branch if the last card indicator is ON.

The following conditions cause the 1401 system to interlock:

1. When the FMI 1401 is operating in an overlap mode, the system interlocks under certain conditions.

   \( B \ (xx) \) ?
   BRANCH ON READER ERROR. If the reader is operating when this instruction is given, the 1401 interlocks until the overlapped operation is completed, before making the error test. If punching is being overlapped at the same time, the 1401 interlocks until the end of the read and the punch operations.

   \( B \ (xx) \) !
   BRANCH ON PUNCH ERROR. If the punch is operating when this instruction is given, the 1401 interlocks until the overlapped operation is completed before making the error test. If the reader is being overlapped at the same time, the 1401 interlocks until the end of the read and the punch operations.
branch back to the test punch busy operation. If the reader was busy, go to the main program (which eventually goes to the test punch busy operation).
L (xxx) L  BRANCH ON TAPE TRANSMISSION ERROR. If the tape adapter unit is busy when this instruction is given, the 1401 interlocks until the TAU is not busy, before making the test for a tape transmission error.

H (xxx) H  BRANCH ON READER BUSY. If the reader is busy when this instruction is given, and the interlock-stop condition is on in the 1402, the 1401 stops. The stop occurs for these conditions:
   a. a card jam in the 1402 read or punch feed
   b. a full stacker
   c. an empty hopper
   d. the I/O check-stop switch on the 1401 console is on, and one of the following conditions occur:
      Hole-count check
      Validity check
      Punch check

I (xxx) I  BRANCH ON PUNCH BUSY. If the punch is busy when this instruction is given, and the interlock-stop condition is on in the 1402, the 1401 stops. The stop occurs for these conditions:
   a. a card jam in the 1402 read or punch feed
   b. a full stacker
   c. an empty hopper
   d. the I/O check-stop switch on the 1401 console is on, and one of the following conditions occur:
      Hole-count check
      Validity check
      Punch check

2. When operating in the overlap mode, the following conditions cause a processing unit stop:
   a. When a storage-address error occurs, processing stops.
   b. If a process error occurs during an overlapped I/O cycle, processing stops immediately; however, the input-output operation continues.
   c. If a process error occurs between overlapped I/O cycles, processing stops immediately, and the input-output operation continues.

3. Read and punch-release operation codes (8 and 9) are not operative in the overlap mode. However, the read or punch (1 and 4) operation codes, when used in the overlap mode, perform the same function. If a tape operation overextends the time allotted it (data to or from tape is still being processed when data from a card is available), a tape transmission error occurs and the data from tape does not enter/leave core storage. Normal tape-error procedure can correct this type of error. If a tape operation is given after a read or punch (1 or 4) operation, the tape operation and processing are interlocked until the read or punch operation is completed.

Space Suppression Feature
This feature enables the system to suppress the automatic space taken by the printer after a line has been printed. The d-character S is used with any printer instruction (operation codes 2, 3, 6, or 7) to prevent the IBM 1403 from spacing after a print operation. Therefore, the next line printed appears on the same line as a line printed by using a SPACE SUPPRESSION instruction (2S).
Address Modification

It becomes necessary in some 1401 programs to perform the same operations repetitively, with a change only in the A- or B-address. The changing of an address while retaining the rest of the instruction is called address modification. Address modification can result in savings in the number of program steps and in the number of storage requirements. In some cases, the program itself determines if and how addresses are to be changed in order to perform the correct program steps for conditions that arise during the processing of data.

There are two basic methods of address modification. The first method does not require the indexing feature. The second method makes use of the indexing feature, which is a special feature for the 1401 system.

Address Modification Without Indexing Feature

Address modification uses the A- and B-bit accumulation that can occur in the hundreds and units positions of a field. This accumulation has already been noted in connection with overflow indication in the Arithmetic Operations section of the manual.

Using Modulus 4 Arithmetic

For systems of 4,000 storage positions or less, A- and B-bit accumulation should occur only in the hundreds position, and is based on modulus 4 arithmetic. To understand how a modulus 4 arithmetic operation is accomplished, let us assign digital values to the A- and B-bit configurations:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>AB</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>

In a modulus 4 system, the highest digit is 3. Values in excess of three are equal to that value minus four.

\[
\begin{align*}
A + A &= B & \text{or } 1 + 1 &= 2 \\
A + B &= AB & \text{or } 1 + 2 &= 3 \\
B + B &= AB & \text{or } 2 + 2 &= 0 \\
A + AB &= AB & \text{or } 1 + 3 &= 0 \\
A + AB &= AB & \text{or } 1 + 0 &= 1 \\
B + AB &= A & \text{or } 2 + 1 &= 1 \\
B + AB &= A & \text{or } 2 + 0 &= 2 \\
AB + AB &= B & \text{or } 3 + 3 &= 2
\end{align*}
\]

Figure 191. A-Bit and B-Bit Values

For example, 5 is a digit 1. In this system, only two factors can be accumulated at a time (Figure 191).

Digit values in the high-order position of a field accumulate in the normal manner. In systems of 4,000 core-storage positions or less, it is assumed that there is a word mark in the high-order position of the address being modified.

Modification to a higher address in 000-999 address range is:

Increase address 472 by 345.

\[ 472 + 345 = 817 \]

Modification to an address greater than 1000 is:

Increase address 912 by 314.

\[ 912 + 314 = 1226 \text{ or } S26 \]

\( S = A2 \) (overflow in high-order position sets an A-bit using modulus 4 arithmetic and turns on the arithmetic overflow indicator).

Increase address 1754 (X54) by 1204 (S04).

\[ X54 + S04 = R58 \]

\( X = (A7) \)

\( S = (A2) \)

Using the rules of modulus 4 arithmetic, \( A + A = B \)-bit, the new address is:

958 with a B-bit over the high-order position \((B9 = R) \text{ or } R58 \text{ (2958)}\).

To decrease an address, a different means must be used. Modulus 4 arithmetic operates for addition only. Decreasing an address requires the addition of a complement, rather than doing a conventional subtract operation.

In systems of 4,000 core storage positions or less, the
16,000's complement of the decrement is added to the address to be modified.

Decrease address 879 by 148.

16000 - 148 = 15,852 (H3B) = complement
879 + H3B = 731 (with arithmetic overflow)
H = BA8
B = BA2

Using the modulus 16 rules, the arithmetic overflow adds an A-bit in the hundreds position (the hundreds position already contains A- and B-bits, and the units position contains A- and B-bits the combination of which indicates a 15,000 to 15,999 block address). The addition of the A-bit increases the value of the zone bits to 16, according to modulus 16 rules has a new address value of 0 (600-999 block address). Therefore, the new address is 731, and the overflow indicator is on.

Using the Modify Address Instruction

For systems with more than 4,000 positions of core-storage (with the IBM 1406 Storage Unit), A- and B-bit accumulation occurs in both the hundreds and units position. In these systems, the MODIFY ADDRESS instruction must be used. This instruction makes use of modulus 16 arithmetic to perform its operations.

In a modulus 16 system, the highest digit is 15. Values in excess of fifteen are equal to that value minus sixteen. For example, 16 is a digit 0, 17 is a digit 1, etc. In this system, only two factors can be accumulated at a time (Figure 192).

Digit values in high- and low-order positions of a field accumulate in the normal manner.

The MODIFY ADDRESS instruction is standard in systems with more than 4,000 core-storage positions, and does not require word marks in the high-order positions of the addresses.

<table>
<thead>
<tr>
<th>Hundreds</th>
<th>Units</th>
</tr>
</thead>
<tbody>
<tr>
<td>A + A = B</td>
<td>2 8</td>
</tr>
<tr>
<td>A + B = AB</td>
<td>3 12</td>
</tr>
<tr>
<td>B + B = No A No B</td>
<td>0 0</td>
</tr>
<tr>
<td>A + AB = No A No B</td>
<td>0 0</td>
</tr>
<tr>
<td>A + No A No B = A</td>
<td>1 4</td>
</tr>
<tr>
<td>B + AB = A</td>
<td>1 4</td>
</tr>
<tr>
<td>B + No A No B = B</td>
<td>2 8</td>
</tr>
<tr>
<td>AB + AB = B</td>
<td>2 8</td>
</tr>
</tbody>
</table>

Figure 192. Modulus 16 Arithmetic

Increase address 7355 (C5V) by 1800 (Y00).

C5V + Y00 = 85N (with arithmetic overflow)
C = \{AB3\}
V = \{A5\}
Y = \{A8\}
N = \{B5\}

Decrease address 1829 (Y29) by 161.

16,000 - 161 = 15,839 (H3I) complement
Y29 + H3I = W68 (with arithmetic overflow)
Y = A8
H = AB8
I = AB9

Using the modulus 16 rules and the MODIFY ADDRESS instruction, the addition of the existing zone bits results in zero (hundreds position, A + AB = 4; units position, AB = 12, 4 + 12 = 16, which has an address value of 0). Then the arithmetic overflow adds a new A-bit in the hundreds position. The result is address W68 (1068).

Because there are more than 4,000 core storage positions, the MODIFY ADDRESS instruction must be used. Using the rules of modulus 16 arithmetic, the addition of the existing zone bits results in a No A, No B (0 zone value) in the hundreds position, and an A (zone value 4) in the units position. The addition of the AB and A-bits (= 0) in the hundreds position results in a zone carry-back (A-bit) to the units position. This A-bit added to the A-bit already there results in a B-bit (zone value 8) in the units position. The arithmetic digit carry adds a new A-bit in the hundreds position. The result is address/5N (9155).

To decrease an address, only modulus 16 arithmetic is used. Because modulus 16 arithmetic operates for addition only, decreasing an address requires adding a complement, rather than doing a conventional subtract operation. The 16,000's complement of the decrement is added to the address to be modified. If the result is an address outside the storage limit of the system, an invalid address condition is indicated.

Decrease 1829 (Y29) by 161.

16,000 - 161 = 15,839 (H3I) complement
Y29 + H3I = W68 (with arithmetic overflow)
Y = A8
H = AB8
I = AB9

Using the modulus 16 rules and the MODIFY ADDRESS instruction, the addition of the existing zone bits results in zero (hundreds position, A + AB = 4; units position, AB = 12, 4 + 12 = 16, which has an address value of 0). Then the arithmetic overflow adds a new A-bit in the hundreds position. The result is address W68 (1068).
Address Modification with Indexing Feature

Three index locations are provided in the IBM 1401 Processing Unit to modify addresses automatically. This means that fewer instructions are needed to accomplish the modification of addresses. For a description of the indexing feature refer to the Special Features section of this manual.

The 1401, equipped with the indexing feature has three index locations.

<table>
<thead>
<tr>
<th>Index Location</th>
<th>Storage Locations</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>087-089</td>
</tr>
<tr>
<td>2</td>
<td>092-094</td>
</tr>
<tr>
<td>3</td>
<td>097-099</td>
</tr>
</tbody>
</table>

The programmer selects those instructions to be indexed, by designating the index location using the tens positions of the instruction address. This designation is referred to as tagging.

**Positions Tagged**

| OP | AAAAA | BBBBB |

Zone bits designate the index location involved.

<table>
<thead>
<tr>
<th>Index Location</th>
<th>Tens Position</th>
<th>Zone Punch</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>A-bit</td>
<td>zero</td>
</tr>
<tr>
<td>2</td>
<td>B-bit</td>
<td>eleven</td>
</tr>
<tr>
<td>3</td>
<td>A- and B-bits</td>
<td>twelve</td>
</tr>
</tbody>
</table>

When a program is written using the symbolic programming system, indexing is indicated by writing a 1, 2, and 3 in the index column of the symbolic coding sheet (cc 27 for (A) operand—cc 38 for (B) operand).

**Digit Result**

| 1 | Index operand by contents of 087-089 |
| 2 | Index operand by contents of 092-094 |
| 3 | Index operand by contents of 097-099 |

The (A) and (B) operands can be symbolic or actual addresses. In the processing of instructions:

1. The A-address and B-address are analyzed for indexing as they are moved into the address registers.
2. The contents of the proper index location (indexing factor) is added to the contents of the address register and develops the effective address there, when indexing is indicated.
3. Three or four additional cycles are required for each address indexed.

### Increasing an Address

To increase a core storage address using the indexing feature, the contents of the index location is added to the selected address register. Figure 193 illustrates various methods of address modification using the index locations.

### Decreasing an Address

To decrease an address, the 16000's complement of the amount to be subtracted from the address must be stored in the index location.

**Example:**

**Decreasing**

**Required**

Decrease a B-address by 10

**Indexing factor** = 16000 - 10 (15990)

(Complement)

The 15990 converts to the three digit factor 19? (Figure 194).

**Figure 194. Converting Address**

Using the modulus 16 rules, the arithmetic overflow adds an A-bit in the hundreds position (both the hundreds and tens positions already contain A- and B-bits, the combination of which indicates a 15,000-15,999 block address). The addition of the A-bit increases the value of the zone bits to 16, which, according to modulus 16, has an address value of 0 (000-999 block address). Therefore, the new address is 927. With the indexing feature, even though there was an overflow, the arithmetic overflow indicator is not turned on.

**Note:** Refer to Address Validity in the Increase Core Storage section of Special Features.

---

**Figure 193. Indexing**

<table>
<thead>
<tr>
<th>1. INDEX THE B-ADDRESS</th>
<th>INDEX LOCATION</th>
<th>EFFECTIVE INSTRUCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEFORE</td>
<td>M 080 1A7</td>
<td>010 025 050</td>
</tr>
<tr>
<td>AFTER</td>
<td>M 080 1A7</td>
<td>010 025 050</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>2. INDEX A- AND B-ADDRESS</th>
<th>INDEX LOCATION</th>
<th>EFFECTIVE INSTRUCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEFORE</td>
<td>M 050 117</td>
<td>010 025 050</td>
</tr>
<tr>
<td>AFTER</td>
<td>M 050 117</td>
<td>010 025 050</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3. INDEX A- AND B-ADDRESS</th>
<th>INDEX LOCATION</th>
<th>EFFECTIVE INSTRUCTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>BEFORE</td>
<td>M 080 0C0</td>
<td>010 025 050</td>
</tr>
<tr>
<td>AFTER</td>
<td>M 080 0C0</td>
<td>010 025 050</td>
</tr>
</tbody>
</table>

**Figure 193. Indexing**

108
IBM 1401 Console Keys, Lights, and Switches

The IBM 1401 console has the operating keys, lights, and switches for the processing unit and connecting equipment. The console display has a data-flow schematic to assist the operator in tracing the path of data through the IBM 1401 Processing Unit (Figure 195).

Display lights are provided to indicate the bit-configuration of the data in the system registers.

The console keys are illuminated; when lit, they indicate that the particular function is on or is operative. During a manual operation, when a key is pressed, it lights, indicating that the function is controlled by the key.

Displaying Information

The 1401 console employs a back-lighting concept to display data. The bit configuration of each character in each of the machine’s logical elements is shown in binary-coded decimal form (including check-bit and word-mark status). The data is displayed, one character at a time. Also, an error at any display location is indicated by a red light that glows in the legend directly above the character display.

PROCESS LIGHT

This light comes on when an error occurs in the processing unit. The light and the error indication are reset by pressing the check-reset key. Machine operation is resumed by pressing the start key.

RAMAC LIGHT

This light comes on when an error condition exists in the IBM 1405 Disk Storage Unit. One of the indicator lights on the 1405 specifies the particular condition that caused the RAMAC light to come on.

TAPE LIGHT

This light comes on if a tape error occurs during a read or write operation. It also comes on if the tape-select switch is in other than the N or D position. It is turned off automatically when the error indicator is reset by a subsequent tape operation.

EXT I/O LIGHT (EXTERNAL INPUT-OUTPUT LIGHT)

This light comes on when one of the external input and/or output devices of the 1401 system requires operator attention. These devices are:
- The IBM 1009 Data Transmission Unit
- The IBM 1011 Paper Tape Reader
- The IBM 1412 Magnetic Tape Reader
- The IBM 1418 Optical Character Reader
- The IBM 1419 Magnetic Character Reader

READER LIGHT

This light comes on when the card reader requires operator attention (empty hopper, full stacker, card jam, clutch failure, hole-count check-error, or validity error). See Restart Procedure for corrective action.

PUNCH LIGHT

This light comes on when the card punch requires operator attention (empty hopper, full stacker, full chip box, clutch failure, or hole-count check-error). See Restart Procedure for corrective action.

OVERLAP LIGHT

This light comes on when any process error occurs during an overlap cycle. Ordinarily, the process light will also be lit along with an indication of the type of failure. However, if an A-register check occurs during a tape-overlap cycle, only the overlap light turns on. This light is reset by pressing the check-reset key. Press the start key to restart the machine.

PRINTER LIGHT

This light comes on when an error occurs in the printer. See Restart Procedure for corrective action.

STORAGE LIGHT

This light comes on when a parity check occurs on a character as it is read into storage. This light is reset by pressing the check-reset key. Press the start key to restart the machine.

B LIGHT

This light comes on when a B-register parity check occurs. The lights below the B-register light indicate the coded character, check-bit status and word-mark status of the B-register. This light is reset by pressing the check-reset key. Press the start key to restart the machine.

A LIGHT

This light comes on when an A-register parity check occurs. The lights below the A-register light indicate
the coded character, check-bit status and the word-mark status of the A-register. This light is reset by pressing the check-reset key. Pressing the start key restarts the machine.

**LOGIC LIGHT**

This light comes on when an error occurs in an arithmetic operation. It is reset by pressing the check-reset key, and the machine is restarted by pressing the start key.

---

**Logic Block Lights**

**O-FLOW LIGHT**

This light comes on when an overflow condition exists in the machine. The light remains on until the overflow indicator turns off by a programmed test, or by pressing START RESET.

**B ≠ A LIGHT**

This light comes on when an unequal condition exists during a COMPARE instruction.
This light comes on when an equal condition exists during a `COMPARE` instruction with the high, low, equal compare special feature.

This light comes on if the value of the B-field is greater than the value of the A-field, or if the A-field is shorter than the B-field. This works with the high, low, equal compare special feature.

This light also works in conjunction with the high, low, equal compare special feature and comes on when the value of the B-field is less than the value of the A-field in a `COMPARE` instruction.

These lights show the bit configuration (at the end of the B-cycle) of the sum of the characters being processed in an arithmetic operation.

The `OP` light comes on when an incorrect operation code exists in the Op-register. The lights below indicate the coded character and check-bit status of the character in the Op-register.

These lights indicate which character of the instruction is being read out during instruction cycles.

This light comes on when an addressing check occurs. The lights below it, displaying the address, can be checked for a parity-error condition.

This is a lighted key that turns the light on, and displays the contents of the I-address register in the storage-address display lights. If the machine is stopped during an overlap cycle, the I-address register light is on and the storage-address display lights indicate the last overlap address that addressed storage.

This is the same as the I-address register key-light except that it refers to the A-address register and the A-cycle of the processing operation.

Pressing either the A- or the B-auxiliary register key-light displays the contents of the corresponding register. The auxiliary registers are part of the multiply-divide special feature.

This is a lighted key that turns the light on, and displays the contents of the O-address register in the storage-address display lights. If the machine is stopped during an overlap cycle, the O-address register light is on and the storage-address display lights indicate the last overlap address that addressed storage.

Pressing the O-address register key does not turn off any other key lights, but pressing any other address-register key turns the O-address register light off. The O-address register key-light and any other address register key-light can be on at the same time.

The four dial switches labeled `Manual Address` select the address to be entered in the storage-address register. These switches are effective only with these selected positions of the mode switch:

1. Character Display
2. Alter
3. Address Stop
4. Storage Print Out
5. Storage Scan

To set the contents of the A-address register to 1200:

1. Set the mode switch to `Alter`
2. Set the manual address switches to 1200.
3. Press the A-address register key-light.
4. Press the start key.

The storage-address display lights then show the bit configurations for this address (1200).

The manual-address switches also select a storage location for a display or alteration, without disturbing the contents of the address registers.
SENSE SWITCHES

Seven sense switches can be included in the IBM 1401 Processing Unit. The manual toggle switches that control them are located on the console. Switch A controls last-card operations by making the BRANCH IF INDICATOR ON (d-character A) instruction a branch, only when the last card in the reader has passed the read station. Switch A is standard in all systems except Model D. Six additional sense switches (B, C, D, E, F, and G) are special features, except on Model C, which has all the sense switches as a standard feature.

The B xxx d instruction can interrogate the setting of the switch specified by the d-character, at any time during processing, and cause a branch to the I-address if the switch is on.

Mode Switch

The mode switch selects the nine modes of machine operation:

RUN

When the mode switch is set to RUN, the system is under control of the stored program.

I/EX (INSTRUCTION EXECUTION)

When the mode switch is set to I/EX, the first time the start key is pressed, the machine reads one complete instruction from storage and then stops. This is called the instruction phase.

The next time the start key is pressed, the machine executes the instruction. This is called the execution phase.

Subsequent pressing of the start key results in alternate instruction and execution phases.

When performing a 1407 operation, the 1401 does not stop between the instruction and execution phases.

SINGLE CYCLE PROCESS

In this mode, each pressing of the start key causes the machine to take one .0115-millisecond storage cycle and advance through the instruction and execution cycles of the program one character at a time. When the machine is single-cycled through an instruction, the following actions take place: At 1 Op time (first character time of an instruction (I) cycle), the I-address register key-light comes on to indicate that the I-address register is addressing storage. The storage-address display indicates the instruction address. The character at that address is read from storage to the B-register and displayed. Because the first I-cycle reads out the operation code of the instruction, the contents of the B-register is also read into the Op-register and displayed. The first light of the instruction length display comes on, displaying the legend OP.

The next pressing of the start key causes the storage address to increase by one and the next character of the instruction to enter the B-register. If this character does not have a word mark appearing with it in the B-register, it also enters the A-register. The instruction length is increased by 1, lighting the 1 on the display light. During this cycle, the character from the B-register enters the thousands and hundreds position of the A-address register and/or the B-address register except during MOVE, LOAD, or STORE B-ADDRESS REGISTER instruction when the character enters only the A-address register. During a STORE A-ADDRESS REGISTER instruction, this is a dummy I-Op, and the Op-light (instruction-length block) remains on. Pressing the key a third time during a store A-address register operation advances the I-instruction length to I-I, and the character in the B-register functions as previously stated, except it only enters the A-address register.

Each successive operation of the start key causes the storage address to increase by 1 and the instruction length to advance by 1.

The operations that occur for each of the eight instruction-length display cycles are:

<table>
<thead>
<tr>
<th>Cycle</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>The operation code enters the Op-register and the B-register. Because this is the first I-cycle, the A-register is undisturbed.</td>
</tr>
<tr>
<td>2</td>
<td>During the second cycle, the second instruction character (first character of the A-address) enters the hundreds position of the A- and B-address registers and the A-register by way of the B-register.</td>
</tr>
<tr>
<td>3</td>
<td>The third character of the instruction enters the tens position of the A- and B-address registers and the A-register through the B-register.</td>
</tr>
<tr>
<td>4</td>
<td>The fourth instruction character enters the units position of the A- and B-address registers, and the A-register through B-register.</td>
</tr>
<tr>
<td>5</td>
<td>The fifth instruction character (first character of the B-address) enters the hundreds position of the B-address register, and the A-register through the B-register.</td>
</tr>
<tr>
<td>6</td>
<td>The sixth instruction character goes to the tens position of the B-address register, and the A-register through the B-register.</td>
</tr>
</tbody>
</table>
The seventh character of the instruction (last character of the B-address) enters the B-address register, and the A-register through the B-register.

The first character of the next instruction enters the B-register only. Because this is the last I-cycle for this instruction, the A-register and the Op-register, the A- and B-address registers are undisturbed. The detection of a word mark associated with this character signals the machine that this is the Op code for the next instruction. The instruction phase of this operation stops, and the instruction that was just addressed is ready to be executed. Note that the I-address register contains the address of the high-order position of the next sequential instruction. If the character is a d-modifier, it enters the A-register by way of the B-register.

If the previous character was a d-modifier followed by the first character of the next instruction, the characters enter the B-register only. The same conditions occur as explained in cycle 8.

On all I-cycles in which no word mark is detected in the B-register, the data from this register transfers to both the A-register and the appropriate positions of the A- and B-address registers. Detection of a word mark in the B-register, after the first I-cycle, indicates the operation code of the next successive instruction. When this condition is sensed, the instruction phase is completed and the execute cycle is set up. One of the instruction length lights is on at this time to indicate the number of characters read.

The next operation of the start key causes the machine to take the first execution cycle. Each successive pressing of the start key causes an additional single execution cycle to occur until the operation has been completed. On I/O operations, except read release and punch release, all of the execute cycles are executed immediately.

Repeated operation of the start key advances the program through successive instruction and execution cycles.

SINGLE-CYCLE NON-PROCESS

This is similar to the single-cycle process mode, except that no data enters storage from the A-register or from the logic unit. Data always enters storage from the B-register only. This mode allows the operator to observe the results of arithmetic operations, one character at a time, in the logic display, without destroying the original B-field data.

CHARACTER DISPLAY

When the machine is operating in this mode, the start key is pressed, displaying in the B-register the character at the address selected by the manual-address switches.

STORAGE PRINT OUT

This mode of operation permits any 100-character block of storage to be printed. The hundreds and thousands manual address switches select the desired block of storage.

Example: 12xx is set in the manual-address switches and the start key is pressed. The 100 characters in the selected block (1201-1300) print automatically in print positions 1-100. Another automatic print cycle indicates the word marks for that block by printing 1's in their corresponding print positions on the second line. This is used to great advantage in program testing, because the contents of a block in core storage are printed and can be easily examined by the programmer. Thus, this feature increases both processing and programming efficiency. A specific storage position can be located by using the IBM Diagramming Template (Form X24-5884) which has a ten-to-the-inch printing scale.

ALTER

The operator can manually change the contents of any address register or storage location, if the mode switch is set to ALTER. To change the contents of address registers:

1. set the manual address switches at the desired location
2. press the appropriate address register key-light
3. press the start key
4. the selected address register contains the new address, and is displayed in the storage-address register.

To change the contents of a storage location:

1. set the manual address switches to the desired location
2. select the bit configuration of the character to be
entered by setting the eight bit-switches located on
the auxiliary console

3. turn on the enter switch (also on the auxiliary
console).

STORAGE SCAN

When the mode switch is set to STORAGE SCAN, holding
down the start key causes the 1401 to start reading out
of storage beginning at the address set in the manual-
address switches and continues to the end of core stor-
age. If an invalid character is detected in storage, the
machine stops, and the check light associated with the
error is turned on. The storage position in error is
shown in the storage address display unit. The B-
register displays the contents of the storage position
in which the error was detected. This character can
be corrected by using the bit-switches, manual-address
switches, and the enter key (see Alter mode).

After the error condition is corrected, the mode
switch is again set to STORAGE SCAN, and the start key
is held down to cause another read-out from storage.
This procedure insures that all positions of storage are
correct.

If there is a print, read, or punch check, scanning
stops at the storage position in error. Pressing the start
key a second time allows the scan to continue.

ADDRESS STOP

When the mode switch is set to ADDRESS STOP, pressing
the start key starts the program. The machine auto-
matically stops when the machine encounters the ad-
dress selected by the manual address switches. This
feature is commonly used in program testing opera-
tions, because the machine can be stopped at the
address of an instruction anywhere in the program,
except for input or output operations.

Control Keys and Switches

START KEY

This key is used to initiate or resume machine opera-
tion after a manual, programmed, or automatic stop.
Similar keys are found on each of the other units in
the system. The setting of the mode switch conditions
the operation of the start key:

1. During a normal run mode, or address stop mode,
the system can be started by pressing the start key
on any of the units.

2. To restart, following a process error indication, the
check-reset key on the console or on the I/O unit
must be pressed to reset the unit before the start key
is pressed.

3. Following a card jam or misfeed in either the reader
or punch, or a hole count or validity error, the cards
in the associated feed must be run out by means of
the non-process run-out key for that feed, and its
hopper must be reloaded before the I/O check-reset
and start keys are pressed.

START RESET KEY

This key resets the system (except for the data in
the address registers and core storage) so that the
operator can restart the operation. It turns off all
indicators that were turned on by the previous pro-
gram.

STOP KEY

This key stops processing in the system. It is not
effective until the instruction being executed is com-
pleted. Similar stop keys (without lights) are provided
on the 1402 and 1403. It is lighted only if a programmed
stop is encountered.

CHECK RESET KEY

An error detected by the checking circuits, lights this
key. It must be pressed following a processing error.
The system can then be restarted by pressing the
start key.

TAPE SELECT SWITCH

This rotary switch is set to the normal position (N)
during automatic operation. The switch can be set to
the numbers (1-6) that correspond to any of the at-
tached tape units, when manual operation is desired.
The D (diagnostic) setting of the switch allows char-
acters from tape to be read into 1401 core storage just
as they appear on the tape (or disk) record. If an
invalid character is transferred into storage, storage
can be scanned, allowing the operator to display the
character and make the necessary corrections. Im-
proper system operation results if the 1401 is operating
in the overlap mode with the tape-select switch in the
D-position.

BACKSPACE KEY

This key works in conjunction with the tape-select
switch. When the switch is set to a specific tape unit,
pressing this key causes the tape in the selected unit to
backspace over one record each time the key is
pressed.
**Tape Load Key**

When this key is pressed, tape unit 1 is automatically selected and tape data starts loading at address 001 and continues until an inter-record gap is sensed.

**I/O Check Stop Switch**

When this switch is in the **ON** position (up), the machine stops at the completion of an I/O operation, if an error occurs during that operation. In the **OFF** position (down), the machine does not stop, if it detects a hole-count check in the card reader or card punch, a validity check error in the card reader, or a print check error. With the switch in the **OFF** position, error detection must be accomplished by programming.

**Power On Switch**

This switch controls the main power supply for the entire system.

**Power Off Switch**

This switch turns off the main power supply.

**Emergency Off Switch**

This is a pull switch, located on the console. In an emergency, pulling this switch disconnects all the power from the entire system. This switch must be manually reset by a customer engineer before power is restored to the system.
**Auxiliary Console**

The auxiliary console panel is located below the main console of the IBM 1401 Processing Unit (Figure 196). Its purpose is to provide additional operator control of the system.

**Auxiliary Console Switches**

**BIT SWITCHES**

Eight bit-switches alter characters in storage. These switches are used in conjunction with the alter mode as explained in the mode switch description.

The dash over the digit indicates the off position.

**ENTER SWITCH**

This switch enters the characters selected by the bit switches into storage, when the mode switch is set to ALTE.

**CHECK STOP SWITCH**

This is normally on, to stop the machine automatically when a process check occurs. If the switch is in the off position, the machine does not stop on error conditions except for Op-register and address register checks, and input-output checks.

**I/O CHECK RESET SWITCH**

This switch resets error conditions sensed in an I/O unit. It is used primarily by customer engineering.

**AUXILIARY MODE SWITCH**

This switch is a three-position dial switch used with the IBM 1406 Storage Unit and the print storage special feature.

**Off.** This is the normal setting of the auxiliary mode switch. In this position, the switch is not functional.

**Full Storage Print.** When a 1401 system is equipped with an IBM 1406 Storage Unit, this position is used with the main mode switch to print the contents of all the storage positions. To accomplish this:

1. Set the auxiliary mode switch to FULL STORAGE PRINT.
2. Set the mode switch (1401 console) to STORAGE PRINTOUT.
3. Press the start key.

Printing starts at position 001 and continues until all positions in storage have been printed. Word marks print as 1's under the corresponding data positions.

**Print Storage Scan.** When a 1401 system has the print storage special feature, this position is used with the main mode switch to scan the 132 print storage positions, and the core storage print area (positions 201 through 332). To accomplish this:

1. Set the auxiliary mode switch to PRINT STORAGE SCAN.
2. Set the mode switch (1401 console) to STORAGE SCAN.

---

**Figure 196. Auxiliary Console**

116
3. Press the start key.
Scanning begins at position 201 of core storage, and the first position of print storage, and continues through 132 positions in each area. During this procedure, the contents of the main storage print area are displayed in the B-register, and the contents of the print storage area are displayed in the A-register.

If an error is found in either register, scanning stops; otherwise, it stops at position 332. After an error stop, resetting error condition and pressing the start key allows scanning to continue.

**Tape I/O.** This switch setting prevents magnetic tape or serial input-output operations from being performed in the overlap mode. It is used primarily by customer engineering (see *Note*).

**R/P (Read-Punch).** This switch setting prevents card operations from being performed in the overlap mode. It is used primarily by customer engineering.

**Tape I/O R/P.** This switch setting prevents card, tape, or serial input-output operations from being performed in the overlap mode. It is used primarily by customer engineering (see *Note*).

**Note:** With the switch in either the TAPE I/O or TAPE I/O R/P positions, tape operations are not performed in the overlap mode. Therefore, to store the address of the last position of a tape operation, a store B-address register operation must be performed. Storing the contents of the O-address register is incorrect.

**729 V density switch**
This three-position switch controls the low- and high-density rates of the IBM 729 V Magnetic Tape Units attached to the 1401 system. The three settings are 200-556, 200-800, and 556-800.

The IBM 729 II, 729 IV, and 7330 Magnetic Tape Units operate at either 200 or 556 characters-per-inch regardless of the tape-density switch setting. For example, if a 729 II tape unit is addressed for operation and the tape-density switch is set at the 556/800 position, the tape unit operates as if the switch were set at the 200/556 position. An IBM 729 V Magnetic Tape Unit, attached to the system, assumes the recording density actually designated by the tape-density switch.

**Disk write switch**
The disk write switch facilitates testing programs on an IBM RAMAC 1401 system. It prevents writing test data on permanent records in disk storage. When the switch is on, normal disk storage operations can be performed.

When this switch is off, all disk storage instructions, with exception of WRITE DISK and WRITE DISK WITH WORD MARKS, are performed normally. When these two instructions are encountered, the switch prevents the transfer of data from core storage to the disk surface. Automatic comparison of the record address in core storage and the address on the disk record is performed, however, and the unequal-address compare indicator turns on if an unequal condition occurs.

**Note:** A WRITE DISK CHECK instruction must be performed following a write operation, regardless of the setting of the disk write switch. Because the record data in core storage is not written on the disk, a read-back check occurs each time a write check operation is performed.

**Sync. points**
Ten test points and two rectifier circuit points are made available to simplify the servicing of the system. These points are only for the use of customer engineering.
IBM 1402 Card Read-Punch
Operating Keys, Lights, and Switches

START KEY
This key is used to initiate machine operation after a manual, programmed, or automatic stop (Figure 197).

STOP KEY
This key is used to stop the system. If a program step is in process, it is completed before the stop occurs.

NON-PROCESS RUN-OUT READ KEY
This key is pressed to clear the read feed. The last two cards in the normal stacker have not been processed.

NON-PROCESS RUN-OUT PUNCH KEY
This key is pressed to clear the punch feed. The last two cards in the normal stacker have not been punched, and the third-from-last card has not been checked.

LOAD KEY
This key is used to start loading instruction cards. Pressing the load key operates the read feed until a card has passed the read station. The I-address register is set to 001, and a word mark is set in address 091. All other word marks in addresses 002 through 080 are removed.

When the card is read at the read and check station, the program starts and executes the instruction that is punched in the first columns of the card.

Continued operation is completely under control of any program in that card or succeeding cards, as conditioned by the first instruction in the first card. When the punch switch is on, pressing the load key also starts the punch.

CHECK RESET KEY
This key must be pressed to reset any error indication by a punch, read, or validity check, before the start key can become effective. This key is not effective until the feed unit in error is cleared of cards.

PUNCH SWITCH
This switch controls the punch unit of the machine. When this switch is off, the punch is inoperative. When it is on, the machine operates if all the interlock circuits in the punch are satisfied.

READ SWITCH
This switch controls the read section of the machine. When this switch is off, the read feed is inoperative.

POWER ON LIGHT
When power is supplied to the read-punch unit, the power-on light is on.

READER STOP LIGHT
A feed failure, a card jam, a clutch failure, or pressing the reader non-process run-out key stops the machine and turns the reader stop light on.

PUNCH STOP LIGHT
A feed failure, a card jam, a clutch failure, or pressing the punch non-process run-out key stops the machine, and turns the punch stop light on.

VALIDITY LIGHT
This light is on, if an invalid character is detected during a read operation.

READ CHECK LIGHT
This light is on if a hole-count error is detected during card reading. If the count from the read check and read stations for a given card do not agree, the read check light indicates an error. The light also comes on if read release time is over-extended.

PUNCH CHECK LIGHT
This light is on if an unequal hole count is detected in the punch unit or if a B-register error occurs during a punch scan. It also comes on if the A-register detects an improper character or if punch-release time is over-extended.

TRANSPORT LIGHT
This light turns on to indicate a card jam in either feed. The light turns off when the card jam is cleared.

STACKER LIGHT
If any of the five stackers become full, the machine stops, and this light signals the operator.

FUSE LIGHT
When a fuse in the card read-punch burns out, this light signals the condition.

CHIP LIGHT
When the chip box is full or not properly positioned this light comes on and the machine interlocks. It turns off by resetting the chip switch, when the cover is opened to empty the chip box.

Figure 197. IBM 1402 Keys, Lights, and Switches
IBM 1403 Printer Operating Keys, Lights, and Switches

Printer Controls
These controls are shown in Figure 198.

START KEY
This key starts the machine. There is a duplicate start key located at the rear of the printer for operator convenience (Figure 199).

CHECK RESET KEY
This key is used to reset a printer error indication. The start key is then pressed to resume operation.

STOP KEY
This key stops the machine at completion of the instruction in process. There is also a duplicate stop key at the rear of the machine for operator convenience.

END OF FORMS LIGHT
This light shows an end-of-form condition, and the machine stops.

If an end-of-forms condition occurs during a skip from one form to another form, the skip in the process of being performed is completed, but printing does not occur.

The operator must then determine the tape channel to which the skip was directed. The new form should be inserted and aligned at the skipped-to channel. After properly inserting the new forms (see Forms Insertion), press the start key to resume normal operation. Printing occurs on the proper (skipped-to) line.

If an end-of-form occurs during a skip, or while spacing within the last form in the printer, the operator should single-cycle print until a skip to the next form occurs. When the last form is skipped out, follow the same procedure described for inserting and determining the first printing line on a new form.

FORMS CHECK LIGHT
This light indicates paper feed trouble in the forms tractor. This light turns off when corrective action is taken, and the check reset key is pressed.

READY LIGHT
This light turns on when the printer is in condition to print.

PRINT CHECK LIGHT
This light indicates a print error.

SYNC CHECK LIGHT
This light turns on to show that the chain was not in synchronism, at all times, with the compare counter for the printer. The timing is automatically corrected. The light is extinguished by pressing the check-reset key.
**Carriage Controls**

The carriage controls are shown in Figure 198.

**CARRIAGE RESTORE KEY**

Pressing this key positions the carriage at channel 1 (home position). If the carriage feed clutch is disengaged, the form does not move. If it is engaged, the form moves in synchronization with the control tape.

**CARRIAGE STOP KEY**

Pressing this key stops carriage operation.

**CARRIAGE SPACE KEY**

Each time it is pressed this key causes the form to advance one space.

---

**Single Cycle Key**

This key initiates the operation of the printer for one print cycle on each pressing of the key. If the end-of-form light is on, pressing this key causes printing until channel 1 in the carriage tape is sensed.

**Manual Controls**

The manual controls are shown in Figure 200.

**Feed Clutch**

The feed clutch controls the carriage-tape drive and form-feeding mechanism. If it is set to neutral, automatic form-feeding cannot take place. It is also used to select six- or eight-lines-to-the-inch spacing.
Each pressing of this key causes an additional line to print, provided either of the following conditions is met:

1. A print operation is set up, and end-of-form and tape channel 1 are not sensed.

2. A print operation is set up, and end-of-form has occurred. This permits printing on the remainder of the form that was being printed when the form check was detected.
PAPER ADVANCE KNOB

This knob positions the form vertically. It can be used only when the feed clutch is disengaged.

VERTICAL PRINT ADJUSTMENT

This knob makes possible fine spacing adjustments of forms at the print line. Carriage tape is not affected by this knob.

LATERAL PRINT VERNIER

This knob permits fine horizontal positioning.

PRINT-DENSITY CONTROL-LEVER

As many as six forms can be printed at one time, and the print hammer unit is designed to adjust automatically for different thicknesses of forms. However, to provide a vernier control for print impression, a print-density control-lever is used. When this lever is set at position E, print impression is lightest. When this lever is set at position A, print impression is darkest. Between these two settings are intermediate settings. Position C is considered the normal setting. This lever moves the type chain closer to or farther from the hammer unit (Figure 200).

The setting of this lever must be considered together with the form's thickness, to determine the normal setting of the print timing dial. A chart is provided to determine this normal setting.

PRINT-TIMING DIAL

A movable-dial is set to a fixed indicator and is used for fine adjustment of printing quality (Figure 201).

The proper setting of the timing dial is obtained from the print timing dial chart (Figure 202) located on the ribbon cover. The dial is set by using the data on the chart. The setting of the print density lever, in conjunction with the thickness of the form, refers to a nominal setting of the print timing dial which can be read from the print timing dial chart.

The setting from the chart can be adjusted to a finer degree by the operator. For a finer setting of the timing dial, turn the timing dial clockwise till printing of characters on the left side of the form becomes light. Then rotate the dial counterclockwise till the printing of the characters on the right side of the form becomes light. The optimum setting of the print timing dial is halfway between the two readings.

PRINT UNIT RELEASE LEVER

This lever permits access to the form transport area (Figure 201).

PRINT LINE INDICATOR AND RIBBON SHIELD

The lower ribbon shield is also used as a print line indicator. It pivots along with the ribbon mechanism. The front side of this shield is marked to show print position location (Figure 203).

When used as a print line indicator, the ends of the shield indicate where the lower edge of characters will print.

When the printer frame is open, the indicator pivots against the forms so that the print line can be set.

HORIZONTAL ADJUSTMENT

This device positions the printing mechanism, horizontally. When the lever is raised, the print mechanism unlocks, and can be positioned horizontally within its 2.4-inch travel limit.

R. H. TRACTOR VERNIER

This knob allows for fine adjustments in paper tension. It can be used for adjustments of up to one-half inch.

Tractor Slide Bar: There are two tractor slide bars, upper and lower. The forms tractors are mounted on
these bars. The forms tractors are movable, and to facilitate this movement there are notches in the tractor slide bar. A description of the use of these notches for proper adjustment for the form to be used is given for the upper tractor slide bar. The description would be the same for the lower slide bar.

The left hand tractor is locked in place by a spring-loaded latch in one of the nine notches located one inch apart on the tractor slide bar. The third notch from the left end is the normal location for most applications.

The first notch is used for forms from $5\%$ to $18\%$ inches wide. When this notch is used, the print unit's lateral movement is limited to .4 inch.

The second notch is used for forms from $4\%$ to $17\%$ inches in width. When this notch is used, the print unit's lateral movement is limited to 1.4 inch.

The third notch is used for forms from $3\%$ to $16\%$ inches wide. When this notch or notches 4 through 9 are used, full lateral print unit movement (2.4 inches) is possible.

The ninth (last) notch can be used for forms from $3\frac{1}{2}$ to $10\frac{3}{4}$ inches wide. When this notch is used, the first usable print position is 38.

The right-hand tractor is locked in place by spring-loaded pins snapped into any one of 27 holes, located one-half inch apart on the tractor slide bar.

The movement of the tractor slide bar, in which the holes are located, is controlled by the right hand tractor vernier. Movement of up to $\frac{1}{2}$ inch can be made by using the vernier knob.

**Indicator Panel Lights**

*Gate Interlock:* This light is turned on when the print unit is not locked in position (Figure 204).

*Brush Interlock:* This light is on if the carriage tape brushes are not latched in position for operation.
Shift Interlock: This light turns on to indicate that the manual feed clutch is not properly positioned.

Thermal Interlock: This light turns on to indicate that a thermal unit has caused a fuse to burn out. If it is on, the IBM customer engineering department should be notified.

High Speed Start: This light turns on when a high-speed skip has been initiated.

Low Speed Start: This light turns on when a low-speed skip or line spacing has been initiated.

High Speed Stop: This light turns on to indicate that high-speed skipping is to be stopped.

Low Speed Stop: This light turns on to indicate that a low-speed skip stop has been initiated. It is on when the carriage is not in motion.

Tape-Controlled Carriage

The tape-controlled carriage (Figure 205) controls high-speed feeding and spacing of continuous forms. The carriage is controlled by punched holes in a paper tape that corresponds in length to the length of one or more forms. Holes punched in the tape stop the form when it reaches any predetermined position.

Forms skip at a rate of 33 inches per second. With the dual-speed carriage, distances of less than eight lines are skipped at 33 inches per second, and those of more than eight lines at 75 inches per second. The last eight spaces skipped in a high-speed skip are skipped at 33 inches per second.

The carriage accommodates continuous forms, up to a maximum of 22 inches in length (at 6 lines per inch) or 16⅔ inches (at 8 lines per inch). The minimum length is 1 inch. For efficient stacking of forms, the recommended maximum forms length is 17 inches. The width of the form can vary from a recommended minimum of 3¾ inches to a maximum of 15¾ inches, including punched margins.

Forms can be designed to permit printing in practically any desired arrangement. Skipping to different sections of the form can be controlled by the program and by holes punched in the carriage tape.

Control Tape

The control tape (Figure 205) has 12 columnar positions indicated by vertical lines. These positions are called channels. Holes can be punched in each channel throughout the length of the tape. A maximum of 132 lines can be used to control a form, although for convenience, the tape blanks are slightly longer. Horizontal lines are spaced six to the inch for the entire length of the tape. Round holes in the center of the tape are pre-punched for the pin-feed drive that advances the tape in synchronism with the movement of a printed form through the carriage. The effect is exactly the same as though the control holes were punched along the edge of each form.
Figure 206. Tape Punch

PUNCHING THE TAPE

A small, compact punch (Figure 206) is provided for punching the tape. The tape is first marked in the channels in which the holes are to be punched. This can be done easily by laying the tape beside the left edge of the form it is to control, with the top line (immediately under the glue portion) even with the top edge of the form. A mark is then made in the first channel, on the line that corresponds to the first printing line of the form. Additional marks are made in the appropriate channels for each of the other skip stops, and for the overflow signal required for the form.

The marking for one form should be repeated as many times as the usable length of the tape (22 inches) allows. With the tape thus controlling several forms in one revolution through the sensing mechanism, the life of the tape is increased. Finally, the line corresponding to the bottom edge of the last form should be marked for cutting after the tape is punched.

The tape is inserted in the punch by placing the line to be punched over a guide line on the base of the punch and placing the center feed holes of the tape over the pins projecting from the base. The dial is then turned until the arrow points at the number of the channel to be punched. Pressing on the top of the punch, toward the back, cuts a rectangular hole at the intersection of a vertical and horizontal line in the required channel of the tape. The tape should never be punched in more than one channel on the same line. Holes in the same channel should not be spaced closer than 8 lines apart. After the tape is punched, it is cut and looped into a belt. The bottom end is glued to the top section, marked glue, with the bottom line coinciding with the first line. Before the tape is glued, the glaze on the tape should be removed by an ink eraser; if this is not done, the tape ends may come apart. The center feed holes should coincide, when the two ends of the tape are glued together.

The last hole punched in the tape should be at least four lines from the cut edge, because approximately the last half inch of the tape overlaps the glue section when the two ends are spliced. If it is necessary to punch a hole lower than four lines from the bottom of the form, the tape should be placed with the top line (immediately under the glue portion) four lines lower than the top edge of the form, before making the channels. To compensate for the loss, the tape should then be cut four lines lower than the bottom edge of the form.

8-Lines-per-inch Spacing

The control tape for 8-lines-per-inch spacing is punched as it would be for normal 6-lines-per-inch spacing. Each line on the tape always equals one line on the form, regardless of whether the latter be 6 or 8 lines-per-inch. In measuring a control tape for a document printed 8 lines to the inch, every ⅜ inch on the form represents one line on the tape.

Carriage Tape Brushes

Two sets of reading brushes (Figure 207) mounted on the same frame, are used to sense holes in the carriage control tape. A small contact roll is used for each set of brushes. One set is called the slow brushes. The other set is called the stop brushes. Seven spaces, as measured by the control tape, separate the brush sets. The slow brushes are positioned ahead of the stop brushes.

The slow brushes are used to control high-speed skipping. They regulate the speed of the last eight spaces of a high-speed skip.

All carriage tape brushes can function to stop a carriage skip under control of the stored program.

Inserting Control Tape in Carriage

1. Raise the counter-balanced cover of the printer to gain access to the tape-reading mechanism.
2. Turn the feed clutch knob to a disengaged position (see Figure 200).
3. Raise the brushes by moving to the left the latch located on the side of the brush holder.
4. Place one end of the tape loop, held so that the printed captions can be read, over the pin-feed drive wheel so that the pins engage the center drive holes.

5. Place the opposite end of the loop around the adjustable carriage control tape idler.

6. Remove the excess slack from the tape by loosening the locking knob on the idler and moving the idler in its track. Tighten the knob when the desired tension is reached. The tape should be just tight enough so that it gives slightly when the top and bottom portions of the loop are pressed together (see Figure 205). If it fits too tightly, damage occurs to the pin-feed holes.

7. Press the brushes down until they latch, and close the printer cover, when the tape is in position.

8. Press the carriage restore key to bring the tape to its home position, and turn the feed clutch knob back to the engaged position. The carriage is ready to operate.

**Ribbon Changing**

To change the ribbon (Figure 208) on the IBM 1403 Printer:

1. Turn off the power in the printer.
2. Lift up the printer cover.
3. Pull back and unlock the print unit release lever. Swing the print unit out.
4. Open the top ribbon cover.
5. Unlatch the print-line indicator ribbon shield and swing it against the form.
6. Push the top ribbon roll to the right (hinged side of print unit), lift out the left end of the ribbon roll, and remove roll from the drive end of mechanism.
7. Slip the ribbon out from under the ribbon correction roll.
8. To remove the bottom roll, press the ribbon roll to the right, and lower the left end of the ribbon roll and remove it from the drive end of the mechanism.

When replacing the ribbon in the machine, hand-tighten the ribbon to remove slack from in front of the printing mechanism. Ribbons are available in widths of 5, 8, and 11 inches in addition to the standard 14 inches. The ribbon width lever (Figure 209) can adjust the ribbon-feed mechanism to accommodate the various ribbon widths.

*Note:* When installing a new ribbon in the printer, always load the full ribbon spool on the bottom spindle to assure proper ribbon skew on the first winding of the ribbon (Figure 210).

**Forms Insertion**

1. Raise the counterbalanced cover of the printer to gain access to the print and forms area.
2. Turn the feed clutch knob to a neutral position.
3. Unlock and swing back the print unit by using the print unit release lever.

4. Open the upper and lower forms tractors (Figure 211).

5. Set the left forms tractors slightly to the left of the first unit position by pulling up or down in the tractor lock (upper and lower tractor). See Figure 203.

6. Insert form on pins and close tractor cover.

7. Pull out on right tractor pin and move tractor to desired location to line up the right side of form. The pin should latch in one of the recessions in the tractor slide bars. See Figure 203.

8. Insert form on pins and close tractor covers.

9. Use the tractor vernier knob to tighten the tension on the form. This knob is used for adjustments of up to one-half inch.

10. Check the position and line where printing will occur, by swinging the ribbon shield against the form (it is marked with each print position). If the horizontal alignment is not correct, it can be adjusted by using the horizontal adjustment knob and/or the lateral print vernier knob for slight adjustments. The vertical adjustment can be made by using the paper advance knob and/or vertical print adjustment knob.

11. Return the print unit to its normal position and lock it in place.

12. Restore the carriage tape to the first printing position by pressing the carriage restore button.

13. Return the feed-clutch knob to a drive position at either six or eight lines-per-inch, depending on the form to be printed.

14. Close the outside cover of the printer.

**Paper Stacker**

The paper stacker provides a manual control for optimum stacking of paper at the rear of the printer. Two controls (Figure 212) permit the operator to set up the paper stacker for each individual run.

The upper lever controls the position of the paper guide at the stacker. This lever is indexed (0-6) so that the setup position can be recorded for reference in the operator’s procedures.

The lower lever is a speed control that is set to keep light tension on the paper form feeding into the
stacker. The speed control has five settings. The setting of this control is selected according to the carriage operation being used. For example, if the job is a listing operation with no long skips, the slow position is selected. However, this must also be conditioned by the kind of forms being used because of varying weight of the paper.

**Cleaning the Chain**

The printing chain on the IBM 1403 Printer should only be cleaned by qualified personnel. When cleaning the chain becomes necessary, a customer engineer should be called to perform the task.
IBM 1405 Disk Storage Unit Indicator Lights

POWER ON LIGHT
When power is supplied to the disk storage unit, the power-on light is on (Figure 213).

READY LIGHT
This light is on when the 1405 is available for use in the 1401 system. It goes off to indicate either low air pressure, power supply failure, or improper functioning of the reading or writing circuits. When this light goes off the RAMAC light on the 1401 console comes on. Processing can continue until the stored program addresses the disk storage unit. This prevents inaccurate disk storage operation.

PARITY LIGHT
This light comes on when a parity check occurs as a character is read from, or written on a disk record. This light is reset by pressing the start reset key on the 1401 console or by testing the READ-WRITE PARITY CHECK indicator. The 1401 system does not stop for a parity error.

ACCESS 0-1-2 LIGHTS
These access lights correspond to the addresses of the access arms on the 1405. Each light indicates that the corresponding access arm has been placed in an inoperable condition by either a logic safety circuit or by a customer engineer. The RAMAC light on the 1401 console turns on when the access light is on. An indicator that can be tested by a branch instruction turns on when the access light comes on.

RECORD LENGTH LIGHT
This light comes on when the wrong-length record indicator turns on. It turns off when the indicator turns off (see Branch if Indicator On instruction for 1405).

ADDRESS COMPARE LIGHT
This light comes on when the unequal-address compare indicator turns on. The light goes off when the indicator turns off (See Branch if Indicator On instruction for 1405).

ADDRESS INVALID LIGHT
This light comes on when an instruction for an access arm or disk storage unit that is not in the system is processed by the 1401. When this light comes on the RAMAC light on the 1401 console turns on. The system does not stop.

Figure 213. IBM 1405 Indicator Lights
IBM 1407 Console Inquiry Station Keys and Lights

REQUEST/ENTER KEY-LIGHT

If the 1401 is in the RUN mode, pressing this key-light (Figure 214) and the processing of a console instruction causes:
1. the inquiry request indicator to turn ON, and the request light comes ON
2. the enter light to come ON when the 1401 is ready to accept data from the 1407
3. the typewriter keyboard to unlock when a 1407 instruction is processed
4. the inquiry request indicator to turn OFF.

RESPOND/TYPETOUT KEY-LIGHT

If the 1401 is in the RUN mode, and a read-into-storage operation was performed, pressing this key causes:
1. a group-mark with a word-mark to enter 1401 storage
2. the enter light to turn OFF
3. the carriage on the typewriter to return
4. the typewriter keyboard to lock
5. the 1401 program to go to the next program step.

If the 1401 is in the RUN mode, and a write-out-of-storage operation is to be performed by the stored program, the typeout light comes ON when the 1407 begins to type data from 1401 storage.

If the 1401 is in the ALTER mode, a write-out-of-storage operation is to be performed, pressing this key causes:
1. data from 1401 storage to be typed on the 1407
2. the light to come ON when typing begins.

CLEAR KEY-LIGHT

If the 1401 is in the RUN mode, and a read-into-storage operation has been performed, pressing this key causes:
1. the inquiry clear indicator to turn ON
2. the carriage on the typewriter to return
3. the typewriter keyboard to lock
4. the enter light to turn OFF
5. the 1401 program to go on to the next program step.

If the 1401 is in a RUN mode, and a read-into-storage operation is being performed, the clear light comes ON and locks the typewriter if a group-mark with a word-mark is sensed in 1401 storage before the end of the read-in operation.

If the 1401 is in the ALTER mode of operation, a read-into-storage operation is being performed, and this key is pressed:
1. the inquiry clear indicator is not affected
2. the read-in operation is cancelled
3. the carriage on the typewriter returns
4. the typewriter keyboard locks.

If the 1401 is in the ALTER mode of operation, a write-out-of-storage operation is being performed and this key is pressed:
1. the inquiry clear indicator is not affected
2. the write-out operation is cancelled
3. the carriage on the typewriter returns.

ALTER KEY-LIGHT

This key-light is operative only if the 1401 is in the ALTER mode of operation. The light comes ON when the mode switch on the 1401 console is in the ALTER position. Pressing the key:
1. signals the 1401 that a 1407 alter operation is to take place
2. causes the 1407 keyboard to unlock
3. causes the enter light to come ON.

WORD MARK KEY

Pressing this key causes a word mark to enter 1401 core storage with the next character typed except during a READ FROM CONSOLE PRINTER instruction. The character with the word mark is typed in red.

FORM KEY LIGHT

This light comes ON when an end-of-forms condition is sensed. Typing stops immediately. Pressing the key, and holding it down, allows typing to resume.

After forms are replaced, pressing the key resets the end-of-forms condition and turns OFF the light.

Figure 214. IBM 1407 Keys and Lights
Operating Pointers

This section describes the functional keys and levers of the console printer. Instructions describing console printer setup are also included.

Keys and Levers

IMPRESSION INDICATOR LEVER

This lever (Figure 215) can be moved forward and backward permitting changes to be made in the amount of force with which the type bars strike the paper. The higher the indicator is moved, the harder the type bars strike. To determine the correct setting for each type of work, use the comma and period as a test, adjusting the impression indicator so that they print distinctly but not heavily. As a general rule, use a higher setting for multiple carbon copies.

MARGIN RELEASE KEY

To print beyond the right or left hand margin, press the margin release key and space beyond that point.

TAB CLEAR KEY

To clear a tab stop, space to the point to be cleared and press the tab clear key. To clear all stops at once, move the carriage to the end of the printing line, press the tab clear key, and, while holding it down, press the carriage return key.

TAB SET KEY

To set a tabulator stop, position the carriage and press the tab set key.

MARGIN SET KEY

The margin set key is used to position the left and right margin stops. To reset the left or right margin:
1. Move the carriage to the present margin stop position.
2. Press the margin set key, and, while holding the key down, move the carriage to the new location.
3. Release the margin set key, and the margin stop is repositioned.

Impression and Carriage Control

LINE POSITION RESET

This lever (Figure 216) locks out standard line spacing and provides a free-rolling platen. To return to the regular printing line accurately and automatically, restore the lever to its forward position. The LINE POSITION RESET makes it possible to print above or below the line — double underscore, subscript, superscript, exponent — and then continue with regular printing. Follow these steps: push back the reset lever, roll the platen to the desired place, print. When the reset lever is returned to the forward position, regular spacing is resumed.

CARRIAGE RELEASE BUTTONS

To move the carriage by hand, press either the left or the right carriage release button located on either side of the carriage. (As the left margin is approached, the air-cushioning action that quiets the electric carriage return will be felt.)

PLATEN VARIABLE BUTTON

When the PLATEN VARIABLE is pressed, the platen moves freely backward or forward. It should be used
when re-inserting a printed page, and it is necessary to locate the same printed line for corrections or additions.

LINE SPACE LEVER
To select single, double, or triple line spacing, move the line space lever to the positions labeled 1, 2, or 3.

PAPER SCALE
This scale indicates the position of the carriage and assists in setting margins and tabs accurately.

PAPER RELEASE LEVER
This lever frees the paper for positioning and removal.

MULTIPLE COPY CONTROL LEVER
This lever permits movement of the platen backward or forward to compensate for the thickness of copy paper, allowing the type bar to strike the paper evenly. The weight of paper and carbon paper used are important factors in determining the correct setting. Normally, the multiple copy control lever should be left at A when printing one to four copies. As a general rule, the lever should be advanced one position for every three extra copies (after the first four).

To Reverse a Ribbon
Open the top cover of the console printer by grasping the cover at the center opening and pulling up and away from the console printer. To close, press lightly and the cover will snap shut.

A ribbon reverse lever (Figure 217) is located beside each ribbon spool. These levers reverse the ribbon auto-
matically on either side at the end of the ribbon. The
direction of the ribbon can be changed by pressing
down on the left-hand lever to wind the ribbon onto
the right spool and vice versa.

Press ELECTRIC RIBBON REWIND (Figure 217), and the
ribbon is automatically wound onto either spool. (To
lock, push down and back.)

To Install a New Ribbon

Press ELECTRIC RIBBON REWIND and wind all of the old
ribbon onto the spool most nearly filled. Hold back
the ribbon guide located over the spool; press the
ribbon reverse lever next to the spool; pull out the
small knob in the center of the spool. Lift out and
discard the used ribbon and spool, and insert a new
ribbon making certain the ribbon reverse lever beside
this spool is down. Spool teeth on top must point to-
ward the platen. Thread the ribbon through the guides
indicated (Figure 218). Hook the end of the ribbon
on the hub of the empty spool.

Removing the Platen

1. Center the carriage; raise the paper bail, carriage
and covers, and copy guide.
2. Push back and lift up latches (Figure 219). Lift out
platen.

To replace platen, center the groove in the right
platen shaft on the carriage end plate, and bring
latches forward and down.

Check these points for proper console printer op-
eration:

1. If the motor and console printer are operating
but typebars will not print, make certain that
MULTIPLE COPY CONTROL is at A.
2. If the carriage will not move, turn OFF for several
seconds, then ON. With switch ON, press the margin
release key.

Pin-Feed Platen

A pin-feed platen is standard on the 1407 Con-
sole Inquiry Station. The pin-feed platen maintains
proper registration and alignment of continuous forms.

The platen is available in three degrees of hardness:
hard, medium, and soft. A hard platen should be used
whenever forms with five or more parts are used, but
it should never be used with less than a five-part form.

Form Feeding

For the best form feeding operation, place the
stack of forms directly behind the carriage when it
is in the right-hand margin. This allows the forms to
align directly behind the carriage at the time that it
line spaces.
Two types of forms feeding can be used on the 1407. One type is roll paper which, attached to the console printer, is shown in Figure 219. The other is continuous forms.

When using continuous forms, the forms are placed on the forms platform. The form is then inserted in a slot between the console printer and the table (Figure 220) and then inserted in the console printer.

The printed forms, in either type of forms-feeding operation, are stacked on the portion of the console inquiry station table behind the console printer.
IBM 729 and 7330 Magnetic Tape Units
Operating Keys and Lights

The operating keys and lights of the IBM 729 and 7330 magnetic tape units are located at the top of the unit, above the tape reels (Figure 221). The lights are all on the upper row, and the keys are on the lower row. The address selection dial is at the left.

DENSITY SWITCH
This selects high or low density operation, depending on the tape operating mode desired.

ADDRESS SELECTION DIAL
This dial assigns a number, from 1 to 6, to the tape unit, to identify it to the stored program. If some other number (7, 8, 9, 0, or blank) is set, the tape unit cannot be used by the stored program.

The setting should not be changed when a tape operation is in progress.

SELECT LIGHT
The select light turns on automatically, when the address selection dial is properly positioned and the unit is addressed by the computer, provided that the unit is ready.

READY LIGHT
This light, when on, indicates that the tape unit is ready for operation. See Start Key for method of turning this light on. The reel door should never be opened when the ready light is on.

TAPE INDICATE ON LIGHT
This indicator is turned on by:
1. sensing the end-of-reel marker while writing on tape
2. sensing the tape mark while reading tape.

The indicator may be turned off by:
1. pressing the unload key on the tape unit
2. executing a REWIND TAPE AND UNLOAD OR BRANCH IF END OF REEL instruction in the stored program.

FILE PROTECTION LIGHT
This light automatically turns on if the unit is loaded with a reel that does not have the file-protection ring inserted in the back of the reel. The tape cannot be written as long as the file-protection light is on. This light is on whenever the tape unit is not in ready status.

FUSE LIGHT
This light turns on automatically whenever a fuse in the unit has blown.

TAPE DENSITY LIGHTS
These two lights (high and low) indicate the density in which the tape unit is operating. They are controlled by the setting of the density switch.

LOAD REWIND KEY
This key is operative only when the reel door is closed and the ready light is off. Use of this key, after tape has been properly mounted in the magnetic tape unit, lowers tape into the columns, lowers the head assembly, and moves tape in the rewind direction until the load point reflective spot is sensed. If the reflective spot is not to the right of the read-write head when this key is pressed, the tape will unwind from the machine reel.

Caution: Do not open the reel door during rewind or load point searching.

START KEY
Use of this key places the tape unit in ready status and turns on the ready light, provided that:
1. the reel door is closed
2. tape has been loaded into the columns
3. the tape unit is not in the process of finding the load point (rewind or load-point operation).

UNLOAD KEY
This key is operative only when the ready light is off, the tape is in the vacuum columns, and the reel door is closed. Use of this key raises the head assembly, and removes the tape from the columns, regardless of the distribution of tape on the two reels. If the tape is not at load point when the operator wishes to change tape reels, a load-point search should be initiated first by pressing the load-rewind key. Pressing the unload key also turns off the tape-indicate-on light.

RESET KEY
On a 729 II or IV, this key turns off the ready light. It also stops any tape operation except load and unload. If this key is pressed during a high-speed rewind, the operation stops, and then continues as a slow-speed rewind. If the reset key is pressed during a slow-speed rewind, the operation stops.

On a 7330, this key turns off the ready light. It also stops any operation being performed. Do not press this key while the 7330 is performing a high-speed rewind. If the key is pressed while the tape unit is performing a high-speed rewind, damage to the magnetic tape results. Press the reset key only after the high-speed rewind has been completed.

REEL DOOR INTERLOCK
When the door is open, the interlock contact prevents any normal operation of the tape unit. The reel door should never be opened when the ready light is on, or during any load-rewind operation.

REEL RELEASE KEY
When this key is pressed, both reels may be turned manually for threading tape or removing the file reel. To operate the reel release key, open the reel door.

Operating Pointers
Consider the following points whenever a tape unit is in operation.
1. Do not change the address of a tape unit by turning the address selection dial, during the execution of a program that uses other tape units. This applies whether the unit is in ready status or not.
2. Never set two tape units to the same address.
3. Do not open the door of a tape unit unless the tape inside is out of the vacuum columns and the read-write head is raised.
4. If the power in the 1401 goes off with tape units in ready status, have an IBM customer engineer remove the tape from the read-write head and the vacuum columns, of every unit in ready status, before power is restored.
Console Operation

Starting the Machine
The power-on key controls the main power supply for the entire system. Pressing this key has the following results:
1. ac power is supplied to the entire system
2. power light on the reader-punch and ready light on the printer are lighted
3. blowers in the cubes and printer are started
4. dc power is automatically supplied to the system.

Turning Off the Machine
Pressing the power-off key removes the power from the systems:
1. dc power goes off
2. ready and power-on lights go out on all units
3. blower motors are turned off
4. ac power supply goes off.

Entering Information
Information can be entered into a storage location manually as follows:
1. Set the mode switch to ALTER.
2. Set the address of the storage location for information to be entered in the manual-address switches.
3. Set the bit switches on the auxiliary console for the correct BCD code for the information to be entered into the selected storage location.
4. Press the enter key on the auxiliary console.
5. Check B-register bit lights to verify information entered.

Notes. The B-register check light should be off. If it is on, the information entered is not valid.

In some cases the B-register check light gives an error indication on an enter operation if the previous character in the selected area is invalid. Restart with step 4 if the character appears valid but a B-register check is still found.

The storage address bit lights should indicate the storage position selected.

To enter the same character into more than one position of core storage:
1. Set mode switch to storage scan.
2. Set bit switches on the auxiliary console to the desired bit configuration.

Note: Invalid characters can be entered if the check stop switch is off.
3. Hold enter key on auxiliary console.
4. Press the start key.

The process unit continues in the storage scan mode until both keys are released.

Manual Display
1. Set mode switch to CHARACTER DISPLAY.
2. Set manual address switches to desired storage location.
3. Press the start key.

Notes. The B-register contains bit configuration for selected location.

The storage address lights display the bit configuration for the address set in the manual address switches.

Check Manual Entry of Address Registers
1. Set mode switch to ALTER.
2. Set manual address switches to the desired address location.
3. Press the I-, A-, or B-address register key.
4. Press the start key.

Note. The storage-display lights show the bit configuration for the particular address register selected in step 3. There should be no error indication.

Console Inquiry Station Operation

Entering Information in Alter Mode
A method of entering data typed on the IBM 1407 into 1401 core storage is:
1. Set the mode switch (1401 console) to ALTER. Alter light on 1407 turns on.
2. Set the manual address switches (1401 console) to the high-order position in storage, where the typed data is to be placed.
3. Press the B-address register key-light.
4. Press the start key (1401 console) to put the address, set up by the manual address switches, in the B-address register.
5. Press the ALTER key-light (1407 console).
6. The operator can now begin typing the data into 1401 storage beginning at the position specified by setting of the manual address switches. Word marks are cleared from 1401 storage as each character enters from the inquiry station. Characters typed can have an associated word mark entered with them by using the word-mark key.
7. When the transmission is completed, press the clear key (1407 console).
8. To continue programming in the 1401 turn the mode switch to RUN and press the start key (1401 console). If necessary set the address where programming is to begin in the manual address switches (1401 console) and enter it as previously explained.
Read Out of Storage in Alter Mode
A method of transferring an area of 1401 core storage to the 1407 and typing that information is:
1. Set the mode switch (1401 console) to ALTER. Alter light (1407) comes on.
2. Set the manual address switches (1401 console) to the high-order position in storage of the data to be typed out.
3. Press the B-address register key-light.
4. Press the start key (1401 console) to set the address set up by the manual address switches in the B-address register.
5. Press the respond/typeout key-light (1407 console). The typeout light comes on.
6. The typewriter prints the data from 1401 storage beginning at the position specified by the setting of the manual address switches. Characters that have a word mark in the same position are typed in red. Blank positions in core storage are printed as lowercase b's.
7. When the transmission is completed, press the clear key.
8. To continue programming in the 1401, turn the mode switch (1401 console) to RUN and press the start key (1401 console). If necessary, set the address of the next program step in the manual address switches (1401 console), and enter it as previously explained.

Console Inquiry with IBM 1401 in Run Mode with Inquiry Routine
A method of sending an inquiry to the 1401, when an inquiry routine is in 1401 core storage and the 1401 is under operator control, is:
1. Set the manual address switches (1401 console) to the high-order position of the inquiry routine in 1401 storage.
2. Set the mode switch (1401 console) to ALTER. Alter light on 1407 comes on.
3. Press the I-address register key-light and the start key (1401 console) to set the address set up by the manual address switches in the I-address register.
4. Set the mode switch (1401 console) to RUN. The alter light on the 1407 turns off. Press the start reset key (1401 console).
5. Press the request enter key on 1407, and the request light comes on. Press the start key (1401 console).
6. The 1401 then starts executing the instruction of the inquiry routine. When the move or load instruction of the routine is reached, the 1401 halts, the enter key-light comes on, request light goes off, and the system is ready to receive the inquiry.
7. The operator types the inquiry on the 1407. When the message is completed, press the respond key, and a group-mark with a word-mark is inserted in 1401 storage to the right of the inquiry.
8. The 1401 then processes the inquiry. A branch operation can be performed if it is not desired to proceed with the main program after processing the inquiry by the stored program. This operation allows the operator to make the decision to continue processing or take any action necessary.
Restart Procedures

Reader
When a validity or read check error occurs in the card reader, it is necessary to go through the following restart procedure to continue 1401 operation:

1. Open the joggler.
2. Remove the cards in the hopper.
3. Close joggler partially (leave open approximately 45 degrees).
4. Press the non-process run-out key to clear two cards out of the read feed. Cards will not drop from the file-feed magazine.
5. Press the check reset key on the 1402 to reset error indicators.
6. Remove the last three cards from the normal read stacker, correct the card causing the error, and place the cards in the card feed hopper.
7. Replace the other cards in the hopper on top of the three cards.
8. Press the start key to restart the program.

In the case of a read check error, a storage scan operation can be used to determine the position in error. If the error occurred on a punch feed read operation, the scan should be started in core-storage position 101 because the scan stops in each position of the read-in area (core-storage locations 001-080). This is the area into which data has been read from the punch feed read station.

In case of a punch stop due to a card feed failure or card jam, and if there is no punch check indication, the punch can be restarted after the punch feed is cleared of cards. The card that was punched on the cycle when the stop occurred, is repunched. This corresponds to card number 2 of step 4.

Punch
When a punch check error occurs in the punch unit, this procedure can be used to continue 1401 operation:

1. Lift the cards out of the card punch hopper.
2. Press the non-process run-out key to clear three cards out of the punch.
3. Press the check reset key on the 1402 to reset the error indicators.
4. Remove the last four cards from the stacker. The first of these cards is the error card. This card should be reconstructed and placed with card four in front of the card file.
5. Place the card file in the card punch hopper.
6. Press START RESET (1401) and the start key, and restart the program at the instruction that caused card three to be read by the punch feed read brushes.

The validity check procedure is:

1. Lift the cards out of the card punch hopper.
2. Press the non-process run-out key to clear three cards out of the punch.
3. Press the check reset key on the 1402 to reset the error indicators.
4. Remove the last four cards from the stacker. The first of these cards is correct. The second card must be checked, because it was read and punched but not checked. Card three is the error card. This card should be reconstructed and placed with card four in front of the card file.
5. Place the card file in the card punch hopper.
6. Press START RESET (1401) and the start key, and restart the program at the instruction that caused card three to be read by the punch feed read brushes.

Note: If validity and punch-check errors occur at the same time, use the punch-check restart procedure and check card three for the validity error.
cleared of cards. On systems equipped with the processing overlap feature, it is necessary to press the start key twice to resume processing.
**Printer**

When a print check error occurs in the printer, the following restart procedure can be used to continue 1401 operation:

**WITHOUT PRINT STORAGE**

1. Press the print check reset key to reset the error indicators.
2. Turn the mode switch on the processing unit console to STORAGE SCAN.
3. Press the start key. The 1401 stops at the location in storage where the print error occurred and the print check light comes on.
4. Press the print check reset key to turn off the error indication.
5. Perform a storage scan to determine the position or positions in error if the system stopped due to a B-register error during printing. A position in error can be corrected by entering the correct character.
6. Press the start key a second time to re-scan storage to ensure that not more than one print error or B-register error occurred during the last printed line.
7. Set the core-storage address of the print instruction in the I-address register (see Storage Address Dial Switches, 1401 Console).
8. Turn mode switch to RUN.
9. Press the start reset key and then the start key to continue the program.

**WITH PRINT STORAGE**

1. Press the print check reset key to reset the error indicators.
2. Turn the load switch on the 1401 console to STORAGE SCAN and the mode switch on the auxiliary console to PRINT STORAGE SCAN.
3. Press the start key. The 1401 stops at the location in storage where the print error occurred, and the print check light comes on.
4. Press the print check reset key to turn off the error indication.
5. Press the start key a second time to re-scan storage to ensure that not more than one print error or B-register error occurred during the last printed line.
6. Because the system stops due to the print error at the print instruction following the operation that was in error, the information to be reprinted must be reconstructed in the print area.
7. If the system stops due to a B-register error during a print transfer, the core-storage position in error can be corrected by entering the correct character after pressing the start reset key (see Alter Mode, 1401 Console). The B-register and print error indicators are reset by pressing the process check reset key and the print check reset key.
8. Press the start key a second time to re-scan storage to ensure that not more than one print error or B-register error occurred during the last printed line.
9. Set the core-storage address of the print instruction in the I-address register (see Storage Address Dial Switches, 1401 Console).
10. Turn the mode switch to RUN.
11. Press the start reset key and then the start key to continue the program.
Timing

The input, output, and processing abilities of the IBM 1401 Data Processing System are used to greatest advantage, if careful timing consideration is given in the development of each 1401 program. In a card system, computing can usually overlap the input and output operations.

This section includes methods of estimating timing requirements for input-output, calculation, and total job time. Each of these areas is discussed in detail, and timing charts are included where necessary. An understanding of this material allows the programmer to place efficiently the input-output operation codes. This efficient placement can often save valuable processing time.

The formulas used for calculating the time required to execute an instruction are included. This enables the programmer to figure the time required for the processing of data, and, in combination with the input-output operations, the efficient placement of the various operation codes can be achieved.

Timing Input and Output Operations

Card Reader

The feeding mechanism of the card reader is controlled by a single tooth clutch that completes one revolution every 75 ms. If a signal to read is received by the clutch, the feed reads a card as soon as the tooth is at the proper position (clutch point) Clutch points are always 75 ms apart.

The read release special feature permits processing during the 21 ms read start time (RST), and makes a total of 31 ms, of the 75-ms cycle, available for processing. The start read feed instruction is given during the 10 ms normally available for processing. It sends a signal to engage the clutch for the next read cycle as soon as the current cycle is completed. The read a card instruction must then be given during the 21 ms read start time of the next cycle (see Start Read Feed operation code).

If processing time exceeds the 10 ms allowed, but is less than 85 ms and read release is not used, card feeding stops for a period of 75 ms (Figure 222). The timing chart shows that two card-read cycles (150 ms) are required to complete such an operation. Of this time, 85 ms are available for processing. If read release is used, an additional 21 ms (read start time from the third card-read cycle) is also available for processing.

![Figure 222. Read Timing Chart (Read Release)](image-url)
Figure 223. Card Reading Speeds

Note: In this case the START READ FEED instruction can be given any time after clutch point 2 and before clutch point 3.

Figure 223 is a table showing card reading speeds, and the processing time allotted for each.

The early card read feature provides two additional points (clutch points) where the feeding mechanism can engage. When processing between cards exceeds 10 ms, the feed mechanism can engage 50 ms sooner than before. The time between card feeding, is reduced to 100 ms rather than 150 ms. Instead of a 50 per cent reduction in the rated speed (to 400 cpm), there is only a reduction of 25 per cent (to 600 cpm).

Figure 224 illustrates these timings and rates in detail.

When early card read is used with read release, additional processing time between the reading of two cards is provided. This is accomplished by using the read start time (21 ms at the beginning of each cycle) as processing time.

The value of both features is realized in those programs with processing routines that exceed the 10 ms allotted for processing during each basic card reading cycle. Although they accomplish separate objectives, they are complementary. Together, they permit the IBM 1401 Data Processing System to operate at speeds closer to capacity, thus effecting shorter over-all job time.

Figures 225 and 226 illustrate cycle timings with the early card read feature. A read release instruction causes the read clutch to engage at the next clutch point, which occurs at 25-millisecond intervals. Therefore, this instruction should not be given more than 25 milliseconds before the time the clutch is to be engaged.

Card Punch

The card punch is operated by a 4-tooth clutch that completes one revolution every 240 ms. Because the clutch has 4 teeth, there are 4 clutch points occurring at 60-ms intervals in the punch cycle, during which the punch feed can be engaged to operate. As soon as the PUNCH A CARD instruction is interpreted by the program, a signal is sent to the clutch. When the clutch reaches a clutch point, punch start time (PST) begins, followed by punch time and processing time.

The punch release special feature permits processing during the 37 ms punch start time, and increases to 59 ms the total available processing time between successive punch cycles. It initiates punch start time for the next punch cycle as soon as the current punch cycle is completed. See Start Punch Feed operation code.

If processing time exceeds 22 ms and START PUNCH FEED is not used, the punch cycle is delayed for 60 ms (Figure 227).

Figure 228 shows card punching speeds and the processing time available with each.

Printer

The IBM 1403 Printer operates at a maximum rated speed of 800 lines per minute. The 100 ms print cycle allows 16 ms of processing time.

The 1401 system is interlocked for the 84 ms print time. The print storage special feature releases all but approximately 2 ms of the entire print cycle, thus making 98 ms available for processing (Figure 229). The print storage area, however, is interlocked for the 84 ms during printing. Form movement for single spacing is overlapped during the last 20 ms of the cycle. If additional forms movement time or additional processing time is required by the application, this time must

Timing Input and Output Operations
Figure 225. 600 CPM with Early Card Read

Figure 226. 480 CPM with Early Card Read

Figure 227. Punch Timing Chart (Punch Release)
be added to the 100-ms cycle to determine printing speed.

Figure 230 shows the effective printing speeds under various processing and forms movement considerations with and without the print storage special feature.

Additional forms-skipping time beyond the first 8 lines is calculated by multiplying the number of lines skipped by 2.3 ms.

Some program instructions cause form movement to start immediately. If the printer is printing without using print storage when an immediate forms control instruction is given, or if the carriage is already in motion, the 1401 waits until the previous carriage operation is completed before the immediate skip is executed. If the system is equipped with the print storage special feature, and a CONTROL CARRIAGE instruction is given during the 84-ms print time, the processing unit is interlocked. In this case, skipping takes place when printing and carriage operations are completed, if the program calls for an immediate skip, or follows the next print operation if a delayed skip is indicated.

Immediate skips require 20 ms for the first space, 5 ms for each additional space up to 8, and then 2.3 ms

Each space over 8 requires an additional 2.3 ms for models B, C, D. Model A requires 5 ms for each space after the first.
for each space thereafter, except Model A, which requires an additional 5 ms for each space after the first.

Figure 231 shows form movement timing requirements for immediate skip instructions.

With the numerical print feature installed, and the 1403 operating in the numerical mode, the print cycle is 46.7 ms as opposed to the 100 ms cycle in the alphabetical mode. Because changing from one mode to the other affects neither spacing nor skipping, forms movement time remains 20 ms. Interlock time, however, has been reduced from 84 ms to 30.7 ms (Figure 232). Therefore, the maximum rated speed in the numerical mode is increased to 1285 lines per minute.

In either mode the print cycle permits 16 ms of overlapped processing time.

**Simultaneous Input-Output Operations**

Total job-time improvements are often made through use of combination operation codes, and the read release, punch release, and print storage special features.

The first time a simultaneous operation is performed, the I/O cycle time may not exactly correspond to the cycle shown in the charts; but as the operation is continuously performed, the cycle time will be the same as the time specified in the timing charts.

**Combination Print and Read**

If reading and printing are to be done as a combined operation (operation code 3) and print storage is not installed, the 1401 can print 400 lines per minute and read 400 cards per minute. The combined cycle takes 150 ms; 18 ms are available for processing.

Figure 233 shows the division of the print cycle, and the functions that are performed.

*Note:* Read start time extends for four milliseconds beyond print time.

The printer takes priority and operates first. The system is interlocked for the first 84 ms of the cycle. Forms movement can start 80 ms after printing begins and can continue until the end of the cycle. The read operation interlocks the processing unit for 44 ms. The remaining 18 ms are used for processing. A START-READ-FEED instruction should not be given preceding a WRITE AND READ instruction.

*Note:* If the system is equipped with the print storage and read release special features, the use of a single PRINT A LINE and READ A CARD instruction permits the same speeds as the combination WRITE AND READ instruction — 533 cards and 533 lines per minute. In this case, it becomes more practical to use the single instruction because more processing time is available with the print storage special feature.
Figure 234 shows the reading and printing operations performed under individual instructions. The 225 ms functional cycle shows the pattern for 2 cards and 2 lines. Repeating this cycle produces the desired speed — 533 cards and 533 lines per minute.

A read instruction starts the operation. The first print instruction is given at point A to initiate print line 1. The line to be printed is transferred to print storage in 2 ms. Eight ms from the first read cycle remain and are available for processing. A read release command must be given during this 8 ms to release the normal 21 ms RST interlock for card-cycle 2. Now there is a total of 29-ms processing time available between the transfer of line 1 to print storage and the READ A CARD instruction. The 1401 is interlocked for 44 ms while card 2 reads into the input area. Because the print storage feature assumes all but 2 ms of the normal printer interlock, the actual read operation is completely overlapped with print time.

An instruction to print given after the read operation would cause the 1401 to wait until the print storage interlock is released. Another READ A CARD instruction would cause the machine to be interlocked at the time print storage is free to receive another WRITE A LINE instruction. Therefore, processing can take place for 25 ms before the next print instruction initiates the printing of the second line. After the 2-ms interlock for the transfer to print storage, a START READ FEED can be given to release the RST interlock for the third read cycle and to make a total of 79 ms available for processing before the next read operation.

At point B the functional cycle (starting at point A) can begin.

A maximum of three spaces between print lines can be effected during this operation.

**Combination Read and Punch**

The 1401 can read 250 cards per minute and punch 250 cards per minute if the READ AND PUNCH instruction (operation code 5) is used. The entire read operation is overlapped with card punching during the 240-ms punch cycle. The 22 ms normally available for processing during a punch cycle are also available during this combination operation.

The punch release special feature makes it possible to extend the available processing time to a total of 59 ms. If a START PUNCH FEED instruction is given during the 22 ms following punching, the RST interlock is released.

Figure 235 is a chart showing the relationship between the read and punch operations.

**Note:** The position of the punch clutch can cause a maximum delay of 60 ms before the first read and punch operation is initiated. However, if processing time is kept within the prescribed time limits, continued use of the READ AND PUNCH instruction permits operation at 250 cards per minute.
Figure 235. Read and Punch Timing Chart

Figure 236. Print and Punch Timing Chart

Figure 237. Print and Punch Timing Chart (with Print Storage)
Combination Print and Punch

The 1401 can execute a combination WRITE AND PUNCH instruction (operation code 6) during a 300-ms cycle without using the print storage special feature. Maximum output under these conditions is 200 lines and 200 cards per minute.

Figure 236 shows the relationship between printing and punching with 28 ms processing time available between successive combined operations. The printer always takes priority and operates first. The signal to engage the clutch and initiate the punch operation is automatically given by the machine during the 84-ms print time when operation code 6 is used.

Note: START PUNCH FEED instruction cannot be given preceding a WRITE AND PUNCH combination instruction.

If the print storage special feature is used, print time can be overlapped with punching. This timing is illustrated in Figure 237. Single instructions and the punch release special feature are used to obtain maximum processing time. Output is 400 lines and 200 cards per minute.

Combination Print, Read, and Punch

Cards can be read and punched at the rate of 200 cards per minute and printing can occur at 200 lines per minute if the WRITE, READ, AND PUNCH instruction (operation code 7) is used. The printer takes priority and operates first. Reading time is completely overlapped with punching, and processing time available during the 300-ms cycle is 28 ms. Because the signals to start the reader and the punch are automatic in the combination instruction, START READ FEED and START PUNCH FEED cannot be used. Figure 238 shows the timing for the operation.

Note: The first WRITE, READ AND PUNCH instruction could be extended to a maximum of 360 ms because of clutch wait time. Subsequent successive WRITE, READ, AND PUNCH instructions require 300 ms each.

If the print storage special feature is installed in the system, the input-output rates can be increased to 250 cards and 250 lines per minute and the cycle time can be reduced to 240 ms. The punch release special feature makes it possible to increase processing time when print storage is available. Figure 239 is a card and printer summary timing chart.
Print, Read, Punch Cycle - 300 ms

![Print, Read, Punch Cycle Diagram]

Note: Punch start time extends for 7 ms beyond Print Time

Figure 238. Print, Read, and Punch Timing Chart

<table>
<thead>
<tr>
<th>OP CODE</th>
<th>OPERATION</th>
<th>CYCLE</th>
<th>RATE</th>
<th>PROCESSING TIME AVAILABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>WITHOUT RELEASE INSTRUCTION</td>
</tr>
<tr>
<td>1</td>
<td>Read</td>
<td>75 ms</td>
<td>800 cpm</td>
<td>10 ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>150 ms</td>
<td>400 cpm</td>
<td>85 ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>225 ms</td>
<td>266 cpm</td>
<td>160 ms</td>
</tr>
<tr>
<td>2</td>
<td>Print - Without print storage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single space</td>
<td>100 ms</td>
<td>600 lpm</td>
<td>16 ms</td>
</tr>
<tr>
<td></td>
<td>Double space</td>
<td>105 ms</td>
<td>572 lpm</td>
<td>21 ms</td>
</tr>
<tr>
<td></td>
<td>Triple space</td>
<td>110 ms</td>
<td>545 lpm</td>
<td>26 ms</td>
</tr>
<tr>
<td></td>
<td>Print - With print storage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Single space</td>
<td>100 ms</td>
<td>600 lpm</td>
<td>96 ms</td>
</tr>
<tr>
<td></td>
<td>Double space</td>
<td>105 ms</td>
<td>572 lpm</td>
<td>103 ms</td>
</tr>
<tr>
<td></td>
<td>Triple space</td>
<td>110 ms</td>
<td>545 lpm</td>
<td>108 ms</td>
</tr>
<tr>
<td>1 and 2</td>
<td>Print &amp; Read</td>
<td>112.5 ms</td>
<td>533 cpm</td>
<td>8 ms</td>
</tr>
<tr>
<td></td>
<td>With print storage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Print &amp; Read</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Without print storage</td>
<td>150 ms</td>
<td>400 cpm</td>
<td>18 ms</td>
</tr>
<tr>
<td>4</td>
<td>Punch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>240 ms</td>
<td>250 cpm</td>
<td>22 ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>300 ms</td>
<td>200 cpm</td>
<td>82 ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>360 ms</td>
<td>165 cpm</td>
<td>142 ms</td>
</tr>
<tr>
<td>5</td>
<td>Read &amp; Punch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>240 ms</td>
<td>250 cpm/rd</td>
<td>28 ms</td>
</tr>
<tr>
<td></td>
<td></td>
<td>300 ms</td>
<td>250 cpm/pu</td>
<td>22 ms</td>
</tr>
<tr>
<td>6</td>
<td>Print &amp; Punch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Without print storage</td>
<td>300 ms</td>
<td>200 lpm/pr</td>
<td>28 ms</td>
</tr>
<tr>
<td></td>
<td>With print storage</td>
<td>240 ms</td>
<td>250 lpm/pr</td>
<td>20 ms</td>
</tr>
<tr>
<td>7</td>
<td>Print, Read, &amp; Punch</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Without print storage</td>
<td>300 ms</td>
<td>200 lpm/pr</td>
<td>28 ms</td>
</tr>
<tr>
<td></td>
<td>With print storage</td>
<td>240 ms</td>
<td>250 lpm/pr</td>
<td>20 ms</td>
</tr>
</tbody>
</table>

Figure 239. Card and Printer Summary Timing Chart

148
Processing Time
To make realistic timing estimates for processing, it is necessary to consider the individual instructions used and the number of data characters involved in each operation. One approach that can be used is:
1. Develop a general block diagram for the problem to be solved.
2. Define the operation performed in each block.
3. Determine the number of and type of instructions required to accomplish the operations in the block.
4. Using the formulas listed in System Timings (Figure 240), calculate the time required to perform the operations. The timings shown are expressed in milliseconds.
5. Four charts have been included that give approximate timings for multiplication and division operations. Two of the charts (Figure 241 and 243) give the time required when the multiply and divide special features are included in the system. The other charts (Figure 242 and 244) give the timings for multiply and divide, based upon the subroutines written in actual language in the section, Multiplication and Division Subroutines.

### System Timings

<table>
<thead>
<tr>
<th>Key to abbreviations used in formulas</th>
<th>Key to abbreviations used in formulas</th>
</tr>
</thead>
<tbody>
<tr>
<td>L = length of the A-field</td>
<td>L = length of the A-field</td>
</tr>
<tr>
<td>l = length of the B-field</td>
<td>l = length of the B-field</td>
</tr>
<tr>
<td>L = length of Multiword field</td>
<td>L = length of Multiword field</td>
</tr>
<tr>
<td>i = length of instruction</td>
<td>i = length of instruction</td>
</tr>
<tr>
<td>l = length of Multiplexer field</td>
<td>l = length of Multiplexer field</td>
</tr>
<tr>
<td>L = length of Quasiword field</td>
<td>L = length of Quasiword field</td>
</tr>
<tr>
<td>l = length of Divisor field</td>
<td>l = length of Divisor field</td>
</tr>
<tr>
<td>S = Number of significant digits in Divisor (Excludes high-order 0's and blanks)</td>
<td>S = Number of significant digits in Divisor (Excludes high-order 0's and blanks)</td>
</tr>
<tr>
<td>Lw = Length of A- or B-field, whichever is shorter</td>
<td>Lw = Length of A- or B-field, whichever is shorter</td>
</tr>
<tr>
<td>La = Number of characters to be cleared</td>
<td>La = Number of characters to be cleared</td>
</tr>
<tr>
<td>Lr = Number of characters back to right-most '0' in control field</td>
<td>Lr = Number of characters back to right-most '0' in control field</td>
</tr>
<tr>
<td>l = Number of '1's inserted in a field</td>
<td>l = Number of '1's inserted in a field</td>
</tr>
<tr>
<td>s = Timing for Input or Output cycle</td>
<td>s = Timing for Input or Output cycle</td>
</tr>
<tr>
<td>Tm = Form movement times. Allow 20 ms for first space, plus 5 ms for each additional space</td>
<td>Tm = Form movement times. Allow 20 ms for first space, plus 5 ms for each additional space</td>
</tr>
<tr>
<td>Tn = Tape movement times.</td>
<td>Tn = Tape movement times.</td>
</tr>
<tr>
<td>N = Number of fields included in an operation</td>
<td>N = Number of fields included in an operation</td>
</tr>
</tbody>
</table>

### System Timings

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>OP CODE</th>
<th>FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Punch a Card</td>
<td>4</td>
<td>.0115 (l + 1) + l/O</td>
</tr>
<tr>
<td>Read a Card*</td>
<td>1</td>
<td>.0115 (l + 1) + l/O</td>
</tr>
<tr>
<td>Read and Punch</td>
<td>5</td>
<td>.0115 (l + 1) + l/O</td>
</tr>
<tr>
<td>Select Stack</td>
<td>K</td>
<td>.0115 (l + 1)</td>
</tr>
<tr>
<td>Set Word Mark</td>
<td>9</td>
<td>.0115 (l + 2)</td>
</tr>
<tr>
<td>Start Punch Feed*</td>
<td>9</td>
<td>.0115 (l + 2)</td>
</tr>
<tr>
<td>Short Read Feed*</td>
<td>8</td>
<td>.0115 (l + 1)</td>
</tr>
<tr>
<td>Store B-address Register*</td>
<td>Q</td>
<td>.0115 (l + 3)</td>
</tr>
<tr>
<td>Store B-address Register*</td>
<td>6</td>
<td>.0115 (l + 4)</td>
</tr>
<tr>
<td>Subtract (no recomplementation)</td>
<td>5</td>
<td>.0115 (l + 1 + l + l + l)</td>
</tr>
<tr>
<td>Subtract (recomplementation)</td>
<td>5</td>
<td>.0115 (l + 1 + l + l + l + l)</td>
</tr>
<tr>
<td>Write a Line</td>
<td>2</td>
<td>.0115 (l + 1) + l/O</td>
</tr>
<tr>
<td>Write and Punch</td>
<td>6</td>
<td>.0115 (l + 1) + l/O</td>
</tr>
<tr>
<td>Write and Read</td>
<td>3</td>
<td>.0115 (l + 1) + l/O</td>
</tr>
<tr>
<td>Write, Read and Punch</td>
<td>7</td>
<td>.0115 (l + 1) + l/O</td>
</tr>
<tr>
<td>Zone and Add</td>
<td>7</td>
<td>.0115 (l + 1 + l + l + l)</td>
</tr>
<tr>
<td>Zero and Subtract</td>
<td>7</td>
<td>.0115 (l + 1 + l + l + l)</td>
</tr>
</tbody>
</table>

### Tape Operations

- **Tm**: Tape movement can be determined from the following:
  - Tm = Number of Characters
  - C = Character Rate

#### 729 Model I

- 729 Model I at 200 cpi = .057 ms
- 729 Model I at 556 cpi = .024 ms
- 729 Model I at 200 cpi = .044 ms
- 729 Model I at 556 cpi = .016 ms
- 729 Model I at 100 cpi = .139 ms
- 729 Model I at 556 cpi = .053 ms

#### 729 Model II

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write 1.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
<tr>
<td>Write 1.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
<tr>
<td>Write 1.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
</tbody>
</table>

#### 729 Model III

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write 10.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
<tr>
<td>Write 10.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
</tbody>
</table>

#### 729 Model IV

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write 10.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
<tr>
<td>Write 10.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
</tbody>
</table>

#### 729 Model V

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write 10.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
<tr>
<td>Write 10.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
</tbody>
</table>

#### 729 Model VI

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write 10.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
<tr>
<td>Write 10.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
</tbody>
</table>

#### 729 Model VII

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>FORMULA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Write 10.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
<tr>
<td>Write 10.5 + CN ms = TAU Interlocked</td>
<td>10.5 + CN ms = Processing Interlocked</td>
</tr>
</tbody>
</table>

### Processing Time

149
### Figure 241. IBM 1401 Divide Times (with Special Feature)

<table>
<thead>
<tr>
<th>No. of Pos. in Quotient</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.276</td>
<td>.449</td>
<td>.621</td>
<td>.794</td>
<td>.966</td>
<td>1.139</td>
<td>1.311</td>
<td>1.484</td>
<td>1.656</td>
<td>1.829</td>
</tr>
<tr>
<td>2</td>
<td>.337</td>
<td>.610</td>
<td>.783</td>
<td>1.116</td>
<td>1.359</td>
<td>1.622</td>
<td>1.875</td>
<td>2.128</td>
<td>2.381</td>
<td>2.634</td>
</tr>
<tr>
<td>3</td>
<td>.438</td>
<td>.771</td>
<td>1.104</td>
<td>1.437</td>
<td>1.770</td>
<td>2.103</td>
<td>2.436</td>
<td>2.769</td>
<td>3.102</td>
<td>3.425</td>
</tr>
<tr>
<td>5</td>
<td>.598</td>
<td>1.093</td>
<td>1.588</td>
<td>2.083</td>
<td>2.578</td>
<td>3.073</td>
<td>3.568</td>
<td>4.063</td>
<td>4.559</td>
<td>5.053</td>
</tr>
<tr>
<td>7</td>
<td>.759</td>
<td>1.413</td>
<td>2.071</td>
<td>2.727</td>
<td>3.383</td>
<td>4.039</td>
<td>4.692</td>
<td>5.351</td>
<td>6.007</td>
<td>6.663</td>
</tr>
<tr>
<td>8</td>
<td>.840</td>
<td>1.576</td>
<td>2.312</td>
<td>3.048</td>
<td>3.784</td>
<td>4.520</td>
<td>5.256</td>
<td>5.992</td>
<td>6.728</td>
<td>7.464</td>
</tr>
</tbody>
</table>

(Add 1 ms to times shown above if signing of quotient is required.)

### Figure 242. IBM 1401 Divide Times (Based on Divide Subroutine)

<table>
<thead>
<tr>
<th>No. of Pos. in Quotient</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11.815</td>
<td>15.468</td>
<td>19.113</td>
<td>22.762</td>
<td>26.411</td>
<td>30.060</td>
<td>33.709</td>
<td>37.358</td>
<td>41.007</td>
<td>44.655</td>
</tr>
<tr>
<td>2</td>
<td>11.446</td>
<td>15.247</td>
<td>19.008</td>
<td>22.909</td>
<td>26.760</td>
<td>30.551</td>
<td>34.372</td>
<td>38.193</td>
<td>42.014</td>
<td>45.835</td>
</tr>
<tr>
<td>3</td>
<td>11.078</td>
<td>15.072</td>
<td>19.000</td>
<td>23.000</td>
<td>26.960</td>
<td>31.048</td>
<td>35.000</td>
<td>39.000</td>
<td>43.000</td>
<td>47.024</td>
</tr>
<tr>
<td>5</td>
<td>10.342</td>
<td>14.481</td>
<td>19.200</td>
<td>23.359</td>
<td>27.698</td>
<td>32.037</td>
<td>36.376</td>
<td>40.715</td>
<td>45.054</td>
<td>49.393</td>
</tr>
<tr>
<td>7</td>
<td>9.666</td>
<td>14.290</td>
<td>18.974</td>
<td>23.638</td>
<td>28.342</td>
<td>33.026</td>
<td>37.710</td>
<td>42.394</td>
<td>47.076</td>
<td>51.762</td>
</tr>
<tr>
<td>9</td>
<td>8.870</td>
<td>13.899</td>
<td>18.928</td>
<td>23.957</td>
<td>28.986</td>
<td>34.015</td>
<td>39.044</td>
<td>44.073</td>
<td>49.102</td>
<td>54.131</td>
</tr>
<tr>
<td>10</td>
<td>8.502</td>
<td>13.703</td>
<td>18.904</td>
<td>24.105</td>
<td>29.306</td>
<td>34.507</td>
<td>39.708</td>
<td>44.909</td>
<td>50.110</td>
<td>55.311</td>
</tr>
</tbody>
</table>

### Figure 243. IBM 1401 Multiply Times (with Special Feature)

<table>
<thead>
<tr>
<th>No. Pos. Multiplied</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>.276</td>
<td>.347</td>
<td>.418</td>
<td>.489</td>
<td>.560</td>
<td>.631</td>
<td>.702</td>
<td>.773</td>
<td>.844</td>
<td>.915</td>
</tr>
<tr>
<td>2</td>
<td>.414</td>
<td>.552</td>
<td>.690</td>
<td>.826</td>
<td>.966</td>
<td>1.104</td>
<td>1.242</td>
<td>1.380</td>
<td>1.518</td>
<td>1.656</td>
</tr>
<tr>
<td>3</td>
<td>.552</td>
<td>.748</td>
<td>.944</td>
<td>1.140</td>
<td>1.336</td>
<td>1.532</td>
<td>1.728</td>
<td>1.924</td>
<td>2.120</td>
<td>2.316</td>
</tr>
<tr>
<td>4</td>
<td>.690</td>
<td>.942</td>
<td>1.196</td>
<td>1.449</td>
<td>1.703</td>
<td>1.956</td>
<td>2.209</td>
<td>2.452</td>
<td>2.715</td>
<td>2.968</td>
</tr>
<tr>
<td>5</td>
<td>1.004</td>
<td>1.350</td>
<td>1.956</td>
<td>2.382</td>
<td>2.808</td>
<td>3.234</td>
<td>3.660</td>
<td>4.096</td>
<td>4.512</td>
<td>4.928</td>
</tr>
<tr>
<td>7</td>
<td>1.360</td>
<td>1.921</td>
<td>2.442</td>
<td>3.000</td>
<td>3.544</td>
<td>4.085</td>
<td>4.626</td>
<td>5.167</td>
<td>5.708</td>
<td>6.249</td>
</tr>
</tbody>
</table>

---

150
### Table: 1401 Multiply Times Based on Multiply Subroutine

| No. | Pos. | Multiplier Digit | Value of each
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>3.51 3.63 3.74 3.86 3.97 4.08 4.20 4.32 4.43 4.55 4.67</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>7.10 7.33 7.56 7.79 8.02 8.25 8.48 8.71 8.94 9.17 9.40</td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>10.74 11.08 11.43 11.77 12.12 12.46 12.81 13.15 13.50 13.84 14.19</td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td>14.42 14.88 15.34 15.80 16.26 16.72 17.18 17.64 18.10 18.56 19.02</td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>18.15 18.73 19.30 19.88 20.45 21.03 21.60 22.18 22.75 23.33 23.90</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td>25.76 26.57 27.37 28.18 28.98 29.79 30.59 31.40 32.20 33.01 33.81</td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td>29.63 30.55 31.47 32.39 33.31 34.23 35.15 36.07 36.99 37.91 38.83</td>
</tr>
<tr>
<td>9</td>
<td></td>
<td></td>
<td>33.54 34.58 35.61 36.65 37.68 38.72 39.75 40.79 41.82 42.85 43.89</td>
</tr>
</tbody>
</table>

(Add 1 ms to times shown above if it is required to 'sign' the product.)

---

**Figure 244.** IBM 1401 Multiply Times (Based on Multiply Subroutine)
Tape Processing Time

All tape units in a 1401 system are under the control of a tape adapter unit (TAU). TAU can control the operations of only one tape unit at a time. If one tape unit is busy, no other tape unit can be used until all operations on the busy one have been completed. The execute time of 1401 tape instruction varies according to the type and model tape units used in the system (Figure 240).

$C$ is the character rate in milliseconds based on the setting of the tape density switch.

$N$ is the number of characters in the record.

$CN$ is record time (number of characters in the record, times the character rate).

$Start$ time is the time necessary for the tape unit to accelerate to operating speed.

$Stop$ time is the time necessary for the tape unit to decelerate and stop.

$Record$ check time is the time it takes to read or write the check character. This time is based on the read-write head gap (the distance that separates the read and write heads) and the time it takes a single character written on tape to travel from the write head to the read head.

IBM 729 II Tape Timings

During a 729 II read operation, the tape adapter unit is interlocked for $10.7 + CN$ ms (Figure 245). This includes:

- $10.5$ ms — start time
- $.2$ ms — record check time for high-density tape (.6 for low-density tape)
- $CN$ ms — record time

During the same read operation, the processing unit is interlocked for $10.5 + CN$ ms. This includes:

- $10.5$ ms — start time
- $CN$ ms — record time

Therefore, in a tape read operation, processing can take place during the $.2$ ms stop time. A tape transmission error condition can be recognized $.2$ ms after the processing interlock is released. If the tape transmission error test instruction is given during this $.2$ ms period, the processing unit is interlocked until the error indicator can be interrogated.

### READ OPERATION

<table>
<thead>
<tr>
<th>Start Time</th>
<th>Record Time</th>
<th>Stop Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$10.5$ ms</td>
<td>$CF$</td>
<td>$2.1$ ms</td>
</tr>
<tr>
<td>Tape Adapter Unit Interlocked</td>
<td>$10.7 + CN$ ms</td>
<td></td>
</tr>
<tr>
<td>Processing Unit Interlocked</td>
<td>$10.5 + CN$ ms</td>
<td></td>
</tr>
<tr>
<td>Can Test Tape Error Indicator</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### WRITE OPERATION

<table>
<thead>
<tr>
<th>Start Time</th>
<th>Record Time</th>
<th>Stop Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>$7.5$ ms</td>
<td>$CF$</td>
<td>$5.1$ ms</td>
</tr>
<tr>
<td>Tape Adapter Unit Interlocked</td>
<td>$11.7 + CN$ ms</td>
<td></td>
</tr>
<tr>
<td>Processing Unit Interlocked</td>
<td>$7.5 + CN$ ms</td>
<td></td>
</tr>
<tr>
<td>Can Test Tape Error Indicator</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 245. IBM 729, Model II, Read-Write Tape Timing
During a 729 II tape write operation, the tape adapter unit is interlocked for $11.7 + eN$ ms (Figure 245). This includes:

- $7.5 \text{ ms} - \text{start time}$
- $4.2 \text{ ms} - \text{record check time}$ for high-density tape ($4.6$ for low-density tape)
- $CN \text{ ms} - \text{record time}$

During the same write operation, the processing unit is interlocked for $7.5 + CN$ ms. This includes:

- $7.5 \text{ ms} - \text{start time}$
- $CN \text{ ms} - \text{record time}$

Therefore, in a tape write operation, processing can take place during the $5.1 \text{ ms}$ stop time. A tape transmission error condition can be recognized $4.2 \text{ ms}$ after the processing interlock is released. If the tape transmission error test instruction is given during this $4.2 \text{ ms}$ period, the processing unit is interlocked until the error indicator can be interrogated. The difference between the $2.2 \text{ ms}$ record check time of reading and the $4.2 \text{ ms}$ record check time of writing is due to the read-write head gap time ($4.0 \text{ ms}$).

For job timing estimates of tape read-write operations, the nominal formula $10.8 + CN \text{ ms}$ can be used.

**IBM 729 IV Tape Timings**

During a 729 IV read operation, the tape adapter unit is interlocked for $6.8 + CN \text{ ms}$ (Figure 246). This includes:

- $6.7 \text{ ms} - \text{start time}$
- $.1 \text{ ms} - \text{record check time}$ for high-density tape ($0.4$ for low-density tape)
- $CN \text{ ms} - \text{record time}$

During the same read operation, the processing unit is interlocked for $6.7 + CN \text{ ms}$. This includes:

- $6.7 \text{ ms} - \text{start time}$
- $CN \text{ ms} - \text{record time}$

Therefore, in a tape read operation, processing can take place during the $2.1 \text{ ms}$ stop time. A tape transmission error condition can be recognized $.1 \text{ ms}$ after the processing interlock is released. If the tape transmission error test instruction is given during this $.1 \text{ ms}$ period, the processing unit is interlocked until the error indicator can be tested.

During a 729 IV tape write operation, the tape adapter unit is interlocked for $7.8 + CN \text{ ms}$ (Figure 246).

---

**Figure 246.** IBM 729, Model IV, Read-Write Tape Timing
This includes:

- 5 ms - start time
- 2.8 ms - record check time for high-density tape (3.0 ms for low-density tape)
- CN ms - record time

During the same write operation, the processing unit is interlocked for 5 + CN ms. This includes:

- 5 ms - start time
- CN ms - record time

Therefore, in a tape write operation, processing can take place during the 3.8 ms stop time. A tape transmission error condition can be recognized 2.8 ms after the processing interlock is released.

During a 7330 tape write operation, the tape adapter unit is interlocked for 20.3 + CN ms (Figure 248). This includes:

- 5.0 ms - start time
- 6.6 ms - stop time
- 8.7 ms - record check time for high-density tape (9.3 ms for low-density tape)
- CN ms - record time

During the same write operation, the processing unit is interlocked for 5 + CN ms. This includes:

- 5.0 ms - start time
- CN ms - record time

Therefore, in a tape read operation, processing can take place during 12.8 ms of stop time and record-check time. A tape transmission error condition can be recognized .3 ms after the processing interlock is released.

During a 7330 tape write operation, the tape adapter unit is interlocked for 20.3 + CN ms (Figure 248). This includes:

- 5.0 ms - start time
- 6.6 ms - stop time
- 8.7 ms - record check time for high-density tape (9.3 ms for low-density tape)
- CN ms - record time

During the same write operation, the processing unit is interlocked for 5 + CN ms. This includes:

- 5.0 ms - start time
- CN ms - record time

Therefore, in a tape write operation, processing can take place during the 15.3 ms stop time. A tape transmission error condition can be recognized 8.7 ms after the processing interlock is released. If the tape transmission test instruction is given during this 8.7 ms period, the processing unit is interlocked until the error indicator can be interrogated. The difference between the .4 ms record check time of reading and the 2.8 record check time of writing is due to the read-write head gap time (2.7 ms).

For job timing estimates of tape read-write operations, the nominal formula 7.3 + CN ms can be used.

**IBM 7330 Tape Timings**

During a 7330 tape read operation, the tape adapter unit is interlocked for 20.5 + CN ms (Figure 247). This includes:

- 7.6 ms - start time
- 12.5 ms - stop time
- .4 ms - record check time for high-density tape (1.0 ms for low-density tape)
- CN ms - record time

During the same read operation, the processing unit is interlocked for 7.7 + CN ms. This includes:

- 7.6 ms - start time
- .4 ms - record check time for high-density tape (1.0 ms for low-density tape)
- CN ms - record time

For job timing estimates of read operations in either high- or low-density, use the formula 20.1 + C(N + 7) ms, where the factor C(7) is the record check time.

For job timing estimates of write operations in either high- or low-density, use the formula 19.9 + C(N + 7) ms, where the factor C(7) is the record check time, and 8.3 ms of the 19.9 ms is the read-write head gap time.
Figure 248. IBM 7330 Write Tape Timing
Processing Overlap Timing

The addition of the processing overlap special feature can provide great reductions in over-all job times. To use the overlap feature to the greatest advantage, careful timing consideration should be given in the development of each 1401 program and to the operations that are to be overlapped.

Charts on the increase in processing time through the use of the overlap feature are included to aid in efficient placement of overlapped instructions (Figure 249).

A careful analysis of these charts allows the programmer to place efficiently the input-output operations codes. This efficient placing of I/O operations can reduce over-all job time by the overlapping of I/O operations and processing (Figure 250).

Magnetic Tape Operations

The factors that are added to the overlapped tape start and stop times have been derived by subtracting one storage cycle from the character rate of the tape unit and compensating for tape unit timing. The character rate and tape timings for each tape unit can be found in the section on Processing Time.

<table>
<thead>
<tr>
<th>OPERATION</th>
<th>1402 CYCLE</th>
<th>PROCESSING TIME AVAILABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ms</td>
<td>STANDARD ms</td>
</tr>
<tr>
<td>Read</td>
<td>75</td>
<td>10</td>
</tr>
<tr>
<td>Read Column Binary</td>
<td>75</td>
<td>10</td>
</tr>
<tr>
<td>Punch</td>
<td>240</td>
<td>22</td>
</tr>
<tr>
<td>Punch Column Binary</td>
<td>240</td>
<td>22</td>
</tr>
<tr>
<td>Read and Punch</td>
<td>240</td>
<td>22</td>
</tr>
<tr>
<td>Read-Punch Feed</td>
<td>240</td>
<td>22</td>
</tr>
</tbody>
</table>

Figure 249. Summary of Overlapped Card Operation Timing

Figure 250. Comparison of Standard and Overlapped Processing Times
Figure 251 can be used to help in determining the amount of processing time available for overlapped tape read and write operations.

Reading a thousand-character tape record written in low-density on an IBM 729 II tape unit, without the processing overlap feature, interlocks the 1401 processing unit for:

\[
10.5 + CN \\
10.5 + (0.67 \times 1000) \\
10.5 + 67 \\
77.5 \text{ ms}
\]

Using the processing overlap feature, 66.1 ms are available for processing during tape-movement time.

\[
11.1 + (0.055 \times 1000) \\
11.1 + 55 \\
66.1 \text{ ms}
\]

This means that the IBM 1401 Processing Unit is interlocked for only 12 ms. Therefore, the resulting effect is that the processing unit is interlocked only for the time it takes to transfer characters from magnetic tape to core-storage.

The time used for tape movement (T_m) remains the same for an overlap or non-overlap operation. Tape movement time for the same read operation regardless of processing overlap is:

\[
T_u = 11.1 + CN \\
T_u = 11.1 + (0.67 \times 1000) \\
T_m = 11.1 + 67 \\
T_u = 78.1 \text{ ms}
\]

With the processing overlap feature 66.1 ms are available for processing during the reading of a thousand-character tape record. If the 1401 system does not have the processing overlap feature, only 0.6 ms of the total tape-movement time are available for processing. Thus, the processing overlap feature results in a saving of 65.5 ms of what would normally be processing-unit interlock time.

<table>
<thead>
<tr>
<th>TAPE UNIT</th>
<th>TAPE DENSITY</th>
<th>OPERATION</th>
<th>TAPE ADAPTER UNIT INTERLOCKED MS</th>
<th>APPROXIMATE PROCESSING TIME AVAILABLE</th>
<th>APPROXIMATE GAIN IN PROCESSING TIME MS</th>
</tr>
</thead>
<tbody>
<tr>
<td>729 II</td>
<td>High</td>
<td>Read</td>
<td>10.7 + CN</td>
<td>0.2</td>
<td>6.8</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write</td>
<td>11.7 + CN</td>
<td>4.2</td>
<td>11.5</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>Read</td>
<td>11.1 + CN</td>
<td>0.6</td>
<td>11.1 + 0.056N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write</td>
<td>12.1 + CN</td>
<td>4.6</td>
<td>12.1 + 0.056N</td>
</tr>
<tr>
<td>729 IV</td>
<td>High</td>
<td>Read</td>
<td>6.8 + CN</td>
<td>0.4</td>
<td>7.1 + 0.031N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write</td>
<td>7.8 + CN</td>
<td>2.8</td>
<td>7.8 + 0.031N</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>Read</td>
<td>8.1 + CN</td>
<td>0.4</td>
<td>8.1 + 0.031N</td>
</tr>
<tr>
<td>729 V</td>
<td>High</td>
<td>Read</td>
<td>10.7 + CN</td>
<td>0.2</td>
<td>10.8</td>
</tr>
<tr>
<td>(556 or 800)</td>
<td></td>
<td>Write</td>
<td>11.7 + CN</td>
<td>4.2</td>
<td>11.5</td>
</tr>
<tr>
<td></td>
<td>Low (200)</td>
<td>Read</td>
<td>11.1 + CN</td>
<td>0.6</td>
<td>11.1 + 0.056N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write</td>
<td>12.1 + CN</td>
<td>4.6</td>
<td>12.1 + 0.056N</td>
</tr>
<tr>
<td>7330</td>
<td>High</td>
<td>Read</td>
<td>20.5 + CN</td>
<td>12.8</td>
<td>20.5 + 0.036N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write</td>
<td>20.3 + CN</td>
<td>15.9</td>
<td>20.3 + 0.036N</td>
</tr>
<tr>
<td></td>
<td>Low</td>
<td>Read</td>
<td>21.1 + CN</td>
<td>13.2</td>
<td>21.1 + 0.127N</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Write</td>
<td>20.9 + CN</td>
<td>15.9</td>
<td>20.9 + 0.127N</td>
</tr>
</tbody>
</table>

Figure 251. Summary of Available Processing Time
Disk Storage Access Time

To calculate timing for magnetic disk operations, it is necessary to estimate the average time it takes to seek the records needed for a particular application. If input to the operation is in sequence, the average access time is less than if the input data is unsorted. This can be explained by the fact that the duration of the seek depends on how far the access arm must travel.

To seek a track on another disk, the access arm moves horizontally, vertically, and horizontally again. The minimum time to move from the outside track of one disk to the outside track of an adjacent disk is 415 milliseconds. The maximum length of a seek operation is from the inside track of the top disk to the inside track of the bottom disk and takes 800 milliseconds. Figure 252 shows track-to-track access times.

To seek a different-track on the same disk (top or bottom face), the arm moves horizontally only. In this case, minimum seek time is 90 milliseconds and maximum seek time is 250 milliseconds (Figure 253).

Disk-to-disk access time ranges from 100 to 315 milliseconds. Figure 254 shows timing for these operations.

---

**Figure 252. Track-to-Track Motion Time—Different Disk**

**Figure 253. Track-to-Track Motion Time—Same Disk**
Total Job Time

It is important to estimate the total time required to complete the processing of each job considered for a data processing system application. One suggested approach that can be used to calculate total job time is outlined here:

1. Design a general flow-chart of the machine operation.
2. Analyze the operation and determine where input-output operations can be combined effectively. For example, a combination read and punch instruction can sometimes be used to save time if a card is to be punched for each card read.
3. Determine the amount of processing time required. This can be done by drawing a block diagram and estimating the type and number of program steps in each block. In most cases, an average instruction time of .25 ms can produce a rough timing estimate. In some special cases it may be necessary to program portions of the program and determine actual timing from the formulas provided.
4. After processing time is calculated, the proper placing of input-output instructions in the program must be determined to insure that enough processing time is available between input-output interlocks.
   Note: If magnetic tape operations are performed, tape timing must also be calculated to determine overlapping with print operations if print storage is installed.
5. After all cycle lengths and overlapping have been established, calculate the total job time for each type of record including input, processing, and output, for each transaction and multiply by the job volume.
Form Design

Some of the customary rules for designing forms should be reconsidered in the light of the many new features introduced by the IBM 1403 Printer.

1. The print unit contains 100 print positions in a 10.0-inch width or a maximum of 132 print positions (special feature) in a 13.2-inch width. Each print position can print any character.

2. Editing, high-speed skipping and other features are included in the system.

One of the basic tools used in designing forms is the spacing chart shown in Figure 255. The numbers across the top from 0 to 13 represent the tens and hundreds positions of the print-position number, and the numbers directly beneath represent the units position of the print-position number. Print-position 42 can be located by referring first to the 4 column and then to the digit 2 within the 4 column. Print-position 9 can be located by referring to the 0 column and then to the digit 9 within that column.

A facsimile of the carriage-control tape is shown at the left (in Figure 255) for marking the control punching for a specific form. Notations have been included relative to standard form-widths and form-depths, lateral movement of the carriage, and instructions to forms manufacturers.

The IBM 1403 Printer carriage is designed to feed marginally-punched continuous forms satisfactorily under the conditions and specifications outlined in Figure 256. These specifications, if followed, give maximum operating efficiency when the 1403 carriage is used. They are not intended to be restrictive, but rather they are intended to permit customers to purchase their continuous forms from the manufacturer of their choice.

![IBM 1403 and 1404 Printer Spacing Chart](image-url)

Figure 255. Forms Spacing Chart
Form Design as Affected by the Print Unit

In view of the 100 or 132 print positions and the 13.2-inch print unit, these factors should be considered when designing forms to be used on the IBM 1403 Printer:

1. The maximum form width is 18 3/4 inches, and the minimum is 3 1/2 inches (see Figure 256).
2. The maximum form length is 22 inches at six-lines-per-inch spacing, or 16 1/2 inches at 8 lines per inch. For efficient stacking of forms, the recommended maximum forms length is 17 inches.
3. Because all print positions can print all characters, form depth can be reduced, and carbon paper eliminated, by the use of side-by-side printing. For example, sold to and ship to names can be printed on the same line, one on the left side of the form and the other on the right.
4. Forms can be designed for printing six or eight lines to the inch. Single-space, eight-lines-per-inch printing is not recommended when the registration between lines is critical.
5. Forms can be designed for variable line spacing within a form by use of single-, double-, or selective-space control.
6. It is possible to dispense with many vertical lines, because the system can be programmed to print commas, decimals, oblique lines, dashes, and other symbols.
7. A vertical line should not be printed between two
adjacent printing positions because there is an over-all maximum tolerance of only .013 inch between adjacent characters.

8. The number of legible copies that can be produced depends on the weight of the paper used for each form, and on the carbon coating.

Because the striking force of the print hammers is not adjustable, paper and carbon should be tested in conjunction with the print-density control-lever and the print timing dial.

9. The CR (credit symbol) prints from two print positions and the minus sign prints from one. For this reason the minus sign is recommended as a credit symbol instead of the CR symbol.

10. The dollar symbol does not have to be preprinted on a check form, because this symbol can be programmed to print immediately to the left of significant digits.

Form Specifications and Dimensions

**Paper Characteristics**

The paper used for continuous forms must be of sufficient weight and strength to prevent the holes from tearing out during feeding or ejecting of the form. This is particularly important when single-part forms are being used.

The paper must not be so stiff as to cause improper feeding or excessive bulging, particularly at the outfold.

Paper must be as free from paper dust or lint as possible.

**Weight**

The number of legible copies required is a factor in determining the weight of the paper to be used in a multiple-part set.

Best results on multiple-copy forms require a lightweight paper of 13 pounds or less, except for the last copy. Again, the number of copies, as well as the distance of the form away from the hammers (this distance can be varied by use of the print-density control-lever), affects the determination of paper weight.

Feeding and legibility performance can best be determined by making test runs of sample sets of forms.

**Friction**

During the feeding operation, friction on marginally-punched continuous forms should be eliminated by the following means:

1. Place the pack of forms directly beneath the front of the printer on the forms stand, in a position that eliminates any abnormal drag on the forms.

2. Allow sufficient clearance between the hammers and the print chain, to permit the forms to be fed by the pins freely, and without interference. This can be accomplished by properly setting the print-density control-lever.

**Perforated Lines**

The perforations between forms should be sufficiently deep to permit easy separation, but not so deep as to tear in ordinary handling or feeding through the machine.

The perforated lines at the end of the form should always be located at 90 degrees to a vertical center line through the marginal holes.

Cut and uncut portions should be uniformly accurate in length and spacing to insure proper and efficient tearing.

Vertical perforations, at the margin for removal of the marginally punched strip, can vary, depending upon requirements. The distance from the edge of the form to the marginal perforations is usually 1/2 inch.

**Marginal Holes**

Continuous forms should have holes in both right and left margins, 1/8 inch in diameter, spaced vertically 3/8 inch apart from center to center, the full length of the form. The holes should be located this way on all copies of all sets throughout each pack of forms.

It is possible, however, to use holes of any size, shape, and spacing that accomplish the equivalent feeding conditions.

Vertical lines passing through the two vertical rows of pin holes must be parallel. It is recommended that the edges of the form be ¼ inch from the vertical center lines through the holes.

A horizontal line passing through the center of any two marginal holes on the same line should be at a 90-degree angle to either vertical center lines through the marginal holes.

Spacing between holes, center-to-center, must be such that the pins in the forms tractor, 3/8 inch in diameter and spaced ½ inch apart, enter and leave the holes in the paper, freely without tearing the paper.

**Width of Forms**

Although forms of any width within the extremes of those shown in Figure 256 can be used, it is recommended that form widths be confined to the standard sizes shown in Figure 257.

**Length of Forms Between Perforated Lines**

The 1403 accommodates marginally-punched continuous forms up to a maximum length of 22 inches, at
6-lines-per-inch. It is recommended, however, that form lengths be confined to regular lengths, such as 3, 3½, 3⅛, 4, 4½, 5, 5½, 6, 7, 8, 8½, 10, 11, 12, 14, 16, and 17 inches.

**LINE SPACING**

The forms tractor of the IBM 1403 can be set by the operator for single-space printing, 6- or 8-lines-per-inch. For 6-lines-to-the-inch spacing, the length of the form must be evenly divisible by ⅛ inch for single spacing, by ⅜ inch for double spacing, and by ½ inch for triple spacing. Similarly, 8-lines-to-the-inch spacing requires that the length of the form be evenly divisible by ⅜ inch for single spacing, by ¾ inch for double spacing, and by ⅝ inch for triple spacing.

Single-space, 8-lines-per-inch printing on the 1403 is not recommended when the registration between lines is critical.

**MULTIPLE COPIES**

Multiple-copy forms consisting of more than four parts, and forms with the first part made of paper of more than 13-pound weight, should be tested under operating conditions to determine the suitability of feeding and legibility.

If multiple-copy forms are not fastened together, the carbon paper must be kept in line with the form by an acceptable method. One such method is center carbon without pin holes, glued to the set, or full-width carbon paper punched with substantially larger marginal holes that are approximately centered with the corresponding holes in the form. Marginal holes in the carbon that are substantially larger than the corresponding holes in the forms make allowance for carbon shrinkage and provide the processing tolerance necessary for some of the commonly used form structures.

One-time carbon paper or carbon-backed paper can be used. The carbon paper or coating should produce the required number of legible copies without excessive smudging. This can be determined best by making test runs with sample sets of forms containing different qualities of carbon paper.

**FASTENING OF MULTIPLE-COPY FORMS**

The width, length, and number of copies of the form determine the fastening requirements for satisfactory feeding through the forms tractor. For most efficient stacking, however, it is recommended that a suitable fastening method always be used with multiple copy forms.

If the construction of the form is such that the parts are of different widths, the necessity for, and the method of, fastening the form should be determined by the width of the parts, the depth of the form, and weight of paper (shown in Figure 258).

![Table: Fastening Requirements for Multiple-Copy Forms](image)

**Figure 258.** Fastening Requirements for Multiple-Copy Forms

Forms of fanfold construction can be used on the IBM 1403 Printer.

When card-tag or rag-content paper stock is used, a test of sample sets of forms should be made to determine the exact fastening requirements. The fastening may consist of any satisfactory method, such as stitching or gluing, that prevents the copies from shifting. It is essential, however, that whatever fastening medium is used should not impair the feeding or printing alignment of the form.

**REGISTRATION OF FORMS**

The assembly of multiple-copy forms should insure that the punching and printing of all copies of the form
are in absolute registration with the material printed by the 1403. The following tolerances should be maintained.

Vertical Lines: Vertical columns of print positions are spaced 1/10 inch apart. There are 50 printing spaces in 5 inches. Vertical rules printed on a form should be spaced in multiples of 1/10 inch.

The center line of any one character, with reference to any other character on the same line, may have a plus or minus tolerance of .0065 inch, or a maximum over-all tolerance of .013 inch. From a forms viewpoint, it is practically impossible to guarantee that the cumulative tolerance of printing-plate shrinkage, paper shrinkage, and marginal-hole perforations does not exceed .0065 inch. This precludes the possibility of retaining satisfactory registration if vertical rules are spaced to split between print positions.

Where vertical lines are required, such rules should split the respective print position, thereby assigning that particular position for the columnar field (dollars and cents, for example) separation. However, in view of the fact that the 1403 can print special characters such as period and comma in every print position, the use of these symbols as decimal points, etc., avoids the need for vertical lines for such separations.

Vertical printed lines should parallel a vertical center line passing through the marginal holes.

Horizontal Lines: Horizontal printed lines on the form should be at a 90-degree angle to the vertical center line passing through the paper-feed pin holes.

The spacing should conform to the setting of the 1403 forms tractor—6- or 8-lines-to-the-inch.

Margins: It is recommended that no staples or other metal fasteners be used with multiple-copy forms. If unavoidable, it is important that, by careful use of printer area of storage (positions 201-332), either the left or right margin (whichever has the staples) be set outside the print hammer area, so that staples or other metal fasteners do not pass between the chain and hammer unit.

Program Loading Routine

This is a procedure for loading information into the IBM 1403 Data Processing System. It is not the only method that can be used, but it is typical of methods used by programmers.

This loading procedure pre-supposes use of an instruction card format as shown on the chart in Figure 259.

The rules to be followed in preparing each of the 6 types of instruction cards used for loading are:

Rule 1

Card formats must follow those shown in the storage-layout chart. The first three cards are used to set word marks necessary for succeeding operations.

Card 1. Does not need a work mark for location 001. The load key automatically sets a program to this location. No word mark is necessary for location 008 for the first card. The 1401 recognizes the end of a SET WORD MARK instruction when it has placed seven characters into the program registers. The card sets word marks in locations 008 and 012, initiates the reading of card 2, and branches to location 001.

Card 2. Sets two word marks for locations 050 and 067, initiates the reading of card 3 and branches to location 001.

Card 3. Sets two word marks, for locations 074 and 078. It causes card 4 to be read, and branches to location 060 for the next instruction.

Instruction Card. A standard load card. The information contained in card columns 1-4, 8-11, and 60-80 is constant, and should be pre-punched. The data punched in card columns 5-7 identifies the location of the instruction (high-order position). The instruction to be loaded is punched starting in card-column 12, and may continue through card-column 19.

Constant Load Card. The length of the constant can vary from 1 character to 48 characters. The constant load card is a standard card and may be pre-punched. The location (XXX) of variable data

Figure 259. Instruction Cards
(units position) is in columns 5-7. The constant to be loaded is in card columns 12-59, and the number of characters (YYY) to be loaded is in columns 2-4.

**Note:** The information to be pre-punched differs from that pre-punched in the instruction load cards.

**Trailer Card.** Used to clear input area and to branch to the first program step.

**Rule 2**

Pressing the load key on the reader causes an instruction card to feed, places the contents of the card into locations 001 through 080, and automatically starts execution of the load program at location 001. This eliminates the need for manual setting of console dials in preparation for loading.

**Clear Storage Routine**

This is a procedure that can be used to prepare core storage to accept program and data information. This is not the only clearing routine that can be used; others are left to individual creativity of the programmer.

The following two-card program (Figure 260) can be used to clear storage of all characters and word marks. The character in column 27 of card 2 is variable according to the number of storage positions available.

- **T** for 1400 positions
- **Z** for 2000 positions
- **I** for 4000 positions

**Figure 260. Clear Routine**

---

### IBM 1401 PROGRAM CHART

#### Card 1 of 2

<table>
<thead>
<tr>
<th>Step</th>
<th>Int. No.</th>
<th>Address</th>
<th>Instruction</th>
<th>Remarks</th>
<th>Rows</th>
<th>Effective No. of Characters</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>001</td>
<td>J 076</td>
<td></td>
<td>Load Instructions into Punch Area</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>002</td>
<td>J 106</td>
<td></td>
<td>Define Instructions in</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>003</td>
<td>J 112</td>
<td></td>
<td>Punch Area</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>004</td>
<td>J 101</td>
<td></td>
<td>Branch to Punch Area</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>005</td>
<td>J 197</td>
<td></td>
<td>Clear Block of Storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>006</td>
<td>J 197</td>
<td></td>
<td>Move to Next Location</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>007</td>
<td>J 101</td>
<td></td>
<td>Branch to Clear Block of Storage</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>008</td>
<td>J 099</td>
<td></td>
<td>Clear 099 to 000</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>009</td>
<td></td>
<td></td>
<td>Field Card</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>010</td>
<td>J 091</td>
<td></td>
<td>Set Word Mark to 001</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>011</td>
<td>J 117</td>
<td></td>
<td>Clear Punch Area</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>012</td>
<td></td>
<td></td>
<td>Constant</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Figure 260. Clear Routine**

---

| Column 27 contains an I for a Model 3, P for a Model 2, 1411, a Z for a Model 2, 1412, and a T for a Model 1, 1411 |
**Multiplication and Division Subroutines**

These are subroutines for multiplication and division operations, discussed here to illustrate programming methods and to aid programming for machines not equipped with the multiply-divide special feature. These are not the only methods of performing these operations—they are typical methods.

**Multiplication**

The block diagram in Figure 261 illustrates the logic used in developing the two multiply subroutines discussed here. Both subroutines provide for a maximum of a 9-digit multiplier, 11-digit multiplicand, and a 20-digit product. Both routines use positive factors.

The subroutine written in actual language (Figure 262) occupies the 900 block of storage. A multiplier area is provided in storage positions 901-909, and the product area is assigned in storage positions 910-929. The multiplicand can be located anywhere.

Any program that uses this subroutine must include a step that moves the multiplier address (XXX) to location 937 and the multiplicand address (YYY) to location 952.

At the completion of the multiply subroutine, the program instruction step 12 can use a branch to the main program or stop instruction.

The routine starts in storage position 930. The product is found in 929 for a 9-digit multiplier, 928 for 8-digit, 927 for 7-digit, 926 for 6-digit, etc.

The subroutine written using the symbolic programming system (Figure 263) parallels the one written in actual. In this instance, the 1401 (through the processor program) controls the location of the instructions, and the data and work areas.

---

**Figure 261. Multiply Flow Chart**

**Figure 262. Multiply Subroutine (Actual)**
Figure 263. Multiply Subroutine (Symbolic)

Note: The multiplication subroutine results in blanks instead of zeros in the low-order position of a product when the multiplier contains low-order zeros. To correct this situation, set the product area to zeros.

Division
The restrictions placed on this subroutine (Figures 264 and 265) are:
1. The dividend and quotient fields must be of equal length.
2. The dividend and divisor must both be positive.
3. The divisor must have no zone for its positive indication. This is necessary only if the divisor could be zero.
4. The divisor cannot contain more than nine leading zeros.
5. All fields must be located completely below address 999.
6. At the completion of the subroutine, the address of the units position of the quotient can be found in the B-address of the instruction located in 651.
7. The remainder is left in the dividend field.
8. A word mark must be located immediately to the right of the units position of the dividend.
9. The quotient area must be preset to zeros or blanks to develop the correct quotient. If the area is not zeroed or blanked, then the quotient will be added to whatever is there. The positions added will depend on the number of leading zeros in the divisor.
10. The information shown in Data for Division Subroutine, except the constant 1 in location 513, must be set initially for each desired execution of the divide subroutine. The two addresses in locations 507 and 510, associated with the divisor, are not altered. Thus, they do not have to be reinitialized if the divisor is contained in the same area.

Appendix 167
**Figure 264. Divide Flow Chart**

1. Adjust Addresses
2. Is Divisor Digit Zero?
   - Yes: Subtract Divisor from Dividend
   - No: Divide Digit Neg?
3. Add One to Dividend
4. Adjust Addresses One Position to the Right
5. Are There Any Words Mark?
   - Yes: Division Completed, Branch to Main Routine
   - No: Find Number of Significant Digits in Divisor

**Figure 265. Divide Subroutine**

- Subtract Divisor from Dividend
- Is Dividend Neg?
- Add Divider Back to Dividend
- Adjust Addresses One Position to the Right
- There a Word Mark?

---

**IBM 1401 Program Chart**

Program: **DIVISION ROUTINE EXAMPLE**

<table>
<thead>
<tr>
<th>Instruction</th>
<th>Data</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td>516 M 507 520</td>
<td>Store Address of Word Mark Position (High Order) of Divisor</td>
<td></td>
</tr>
<tr>
<td>523 B 662 TYY</td>
<td>Branch if Divisor Digits Equal Zero</td>
<td></td>
</tr>
<tr>
<td>521 S 510 515</td>
<td>(Length of Divisor) - (Order Exact)</td>
<td></td>
</tr>
<tr>
<td>528 A 518 521</td>
<td>Adjust Dividend Address</td>
<td></td>
</tr>
<tr>
<td>531 A 513 524</td>
<td>Modify Address</td>
<td></td>
</tr>
<tr>
<td>552 S 513 521</td>
<td></td>
<td></td>
</tr>
<tr>
<td>559 S 513 534</td>
<td>Clear zone from low-order position to prepare for address assignment</td>
<td></td>
</tr>
<tr>
<td>561 Y 755 561</td>
<td>Clear zone from low-order position to prepare for address assignment</td>
<td></td>
</tr>
<tr>
<td>566 Y 765 561</td>
<td>Modify Addresses into Routine</td>
<td></td>
</tr>
<tr>
<td>567 M 501 542</td>
<td>Set modified Addresses into Routine</td>
<td></td>
</tr>
</tbody>
</table>

**DATA FOR DIVISION SUBROUTINE**

- Operation of Data Word Data Word
- DESCRIPTION OF DATA
  - 501 WWW Address of word mark position (high order) of dividend
  - 504 XXX Address of word mark position (high order) of quotient
  - 507 YYY Address of word mark position of divisor
  - 510 ZZZ Divisor Address
  - 517 9C Counter for number of zeros in divisor
  - 513 EH Constant
  - 515 NN Length of the divisor

---

**Notes**

- **Address**
  - 501 Address of dividend
  - 504 Address of quotient
  - 507 Address of divisor
  - 510 ZZZ Divisor Address
  - 517 9C Counter for number of zeros in divisor
  - 513 EH Constant
  - 515 NN Length of the divisor

---

**Instructions**

- 623 B 323 XXX Add One to Quotient
- 624 4 256 Store Divisor
- 625 A 513 529 Add One to YYYY Address
- 626 A 513 552 Increase Counter by One
- 627 C 112 555 If Equal, Divisor Equals Zero
- 653 B 565 Branch Unequal
- 655 L 765 Halts - cannot Divide by Zero
- 656 A 112 565 Add Divisor to Dividend
- 657 S 175 565 More Words to Zero word mark position of dividend
- 706 A 543 219
- 713 S 250 565 Test for End of Division
- 721 A 513 642 Move Addresses to Develop Next Quotient Digit
- 728 A 513 642 " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " " 

**Return to Divide Calculations**

**Divide Complete**
### Operation Codes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Read Card</td>
<td>R</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Write Card</td>
<td>W</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>23</td>
<td>Write Word Marks</td>
<td>WRM</td>
<td>23</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Punch Card</td>
<td>P</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4R</td>
<td>Read/Punch Feed*</td>
<td>R</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>4WR</td>
<td>Read/Write Punch Feed*</td>
<td>W</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Read-Punch</td>
<td>RP</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>6</td>
<td>Write-Punch</td>
<td>WP</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>6P</td>
<td>Write-Punch Punch Feed*</td>
<td>P</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>7</td>
<td>Read-Write Punch and Branch*</td>
<td>RWB</td>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>Read-Branch</td>
<td>RB</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>8P</td>
<td>Write-Branch Punch Feed*</td>
<td>WPB</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>Read-Branch Punch Feed*</td>
<td>RBP</td>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>9P</td>
<td>Write-Branch Punch Feed*</td>
<td>WPB</td>
<td>9</td>
<td>9</td>
</tr>
</tbody>
</table>

### Input-Output Codes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Add</td>
<td>A</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Subtract</td>
<td>S</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>Zero and Add</td>
<td>ZA</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>Multiply</td>
<td>M</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>Divide</td>
<td>D</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

### Logical Operations Codes

<table>
<thead>
<tr>
<th>Logical Operation Code</th>
<th>BRANCH ON</th>
<th>BRANCH OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>BRANCH</td>
<td>BRANCH</td>
</tr>
<tr>
<td>1</td>
<td>BRANCH</td>
<td>BRANCH</td>
</tr>
<tr>
<td>2</td>
<td>BRANCH</td>
<td>BRANCH</td>
</tr>
</tbody>
</table>

### Arithmetic Codes

<table>
<thead>
<tr>
<th>Arithmetic Code</th>
<th>BCD Code</th>
<th>Card Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Add</td>
<td>BA1</td>
<td>12-1</td>
</tr>
<tr>
<td>S Subtract</td>
<td>CA2</td>
<td>02-0</td>
</tr>
<tr>
<td>Z Zero and Add</td>
<td>CB2</td>
<td>12-0</td>
</tr>
<tr>
<td>M Multiply</td>
<td>CB4</td>
<td>04-0</td>
</tr>
<tr>
<td>D Divide</td>
<td>DAB4</td>
<td>04-0</td>
</tr>
</tbody>
</table>

### Move and Load Codes

<table>
<thead>
<tr>
<th>Move and Load Code</th>
<th>BCD Code</th>
<th>Card Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>D Move Numerical</td>
<td>BA4</td>
<td>12-4</td>
</tr>
<tr>
<td>L Load Character</td>
<td>LCA</td>
<td>11-3</td>
</tr>
<tr>
<td>M Move Characters</td>
<td>MC8</td>
<td>11-4</td>
</tr>
<tr>
<td>Z Move Zone</td>
<td>CA2</td>
<td>05-8</td>
</tr>
<tr>
<td>S Set Word Marks</td>
<td>SCB21</td>
<td>03-8</td>
</tr>
<tr>
<td>C Clear Word Mark</td>
<td>CWCB44</td>
<td>12-8</td>
</tr>
</tbody>
</table>

### Character at d for BID Branch

<table>
<thead>
<tr>
<th>Character at d</th>
<th>BRANCH ON</th>
<th>BRANCH OFF</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp; Unconditional</td>
<td>R</td>
<td>U</td>
</tr>
<tr>
<td>B Corr. Chem.</td>
<td>T</td>
<td>V</td>
</tr>
<tr>
<td>C &quot;Last Card&quot;</td>
<td>U</td>
<td>W</td>
</tr>
<tr>
<td>D Sense Switch</td>
<td>U</td>
<td>X</td>
</tr>
<tr>
<td>E Sense Switch</td>
<td>U</td>
<td>Y</td>
</tr>
<tr>
<td>F Sense Switch</td>
<td>U</td>
<td>Z</td>
</tr>
<tr>
<td>G Sense Switch</td>
<td>U</td>
<td>A</td>
</tr>
<tr>
<td>H Sense Switch</td>
<td>U</td>
<td>B</td>
</tr>
<tr>
<td>I Sense Switch</td>
<td>U</td>
<td>C</td>
</tr>
</tbody>
</table>

### Character at d for Tape Storage

<table>
<thead>
<tr>
<th>Character at d</th>
<th>Tape Operation</th>
<th>BRANCH ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp; Branch</td>
<td>BRANCH</td>
<td></td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>R Access</td>
<td></td>
</tr>
<tr>
<td>&amp; Branch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>V Read/Write</td>
<td></td>
</tr>
<tr>
<td>&amp; Branch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>W Write/Read</td>
<td></td>
</tr>
<tr>
<td>&amp; Branch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>X Rewrite</td>
<td></td>
</tr>
<tr>
<td>&amp; Branch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>Y Rewrite</td>
<td></td>
</tr>
<tr>
<td>&amp; Branch</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Character at d for Disk Storage

<table>
<thead>
<tr>
<th>Character at d</th>
<th>Disk Operation</th>
<th>BRANCH ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp; Branch</td>
<td>BRANCH</td>
<td></td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>R Access</td>
<td></td>
</tr>
<tr>
<td>&amp; Branch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>V Read/Write</td>
<td></td>
</tr>
<tr>
<td>&amp; Branch</td>
<td></td>
<td></td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>W Write/Read</td>
<td></td>
</tr>
<tr>
<td>&amp; Branch</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Character at d for Inquiry

<table>
<thead>
<tr>
<th>Character at d</th>
<th>Inquiry Operation</th>
<th>BRANCH ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp; Inquiry</td>
<td>BRANCH</td>
<td></td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>R Inquiry</td>
<td></td>
</tr>
<tr>
<td>&amp; Inquiry</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Magnetic Tape %UX Tape Unit Address

<table>
<thead>
<tr>
<th>Magnetic Tape %UX</th>
<th>Tape Unit Address</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Read/Write Tape</td>
</tr>
<tr>
<td>L2</td>
<td>With Word Marks</td>
</tr>
<tr>
<td>L3</td>
<td>Read/Write Tape</td>
</tr>
<tr>
<td>L4</td>
<td>With Word Marks</td>
</tr>
<tr>
<td>L5</td>
<td>Read Compressed Tape</td>
</tr>
<tr>
<td>L6</td>
<td>Address of Tape Unit</td>
</tr>
<tr>
<td>L7</td>
<td>Create Options</td>
</tr>
<tr>
<td>L8</td>
<td>Write Options</td>
</tr>
</tbody>
</table>

### Column Binary

<table>
<thead>
<tr>
<th>Column Binary</th>
<th>BRANCH ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp; Create</td>
<td>BRANCH</td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>R Create</td>
</tr>
<tr>
<td>&amp; Branch</td>
<td>BRANCH</td>
</tr>
</tbody>
</table>

### Disk Storage %FX Disk Operation

<table>
<thead>
<tr>
<th>Disk Storage %FX</th>
<th>Disk Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>L1</td>
<td>Read Disk</td>
</tr>
<tr>
<td>L2</td>
<td>Write Disk</td>
</tr>
<tr>
<td>L3</td>
<td>Read Disk with Word Marks</td>
</tr>
<tr>
<td>L4</td>
<td>Write Disk with Word Marks</td>
</tr>
</tbody>
</table>

### Inquiry %To Address

<table>
<thead>
<tr>
<th>Inquiry %To</th>
<th>BRANCH ON</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp; Inquiry</td>
<td>BRANCH</td>
</tr>
<tr>
<td>&amp; Unconditional</td>
<td>R Inquiry</td>
</tr>
<tr>
<td>&amp; Inquiry</td>
<td></td>
</tr>
</tbody>
</table>

---

*Special Feature
The 1401 has the ability to read MLP card codes in the read feed only. The 1401 ignores the 8-9 punches when they appear in the same column. The 1401 does not punch out MLP card codes.

Note 1. If specified, this code can be made compatible with 705 Group Mark Code (12-5-8).

Note 2. The A-bit coding must be program generated in the 1401 (it cannot be read from a card; it can be punched as a zero). It is used in conjunction with the C-bit to indicate a blank position on tape that was written in even-bit parity.

Note 3. Other special character printing arrangements can be obtained.
Index

A-Address ................................................................. 16
A-Address Register ..................................................... 111
A-Address Register Key-Light ...................................... 111
A- and B-Auxiliary Register Key-Lights ......................... 111
A-Light ....................................................................... 169
A-Register .................................................................. 106
Access Arm.................................................................... 63
Access المجال .................................................................. 68
Access Lights ................................................................ 128
Access Timing ................................................................ 138
Add .............................................................................. 92
Add A-Address ................................................................ 29
Add A- and B-Address .................................................... 29
Add to Special .................................................................. 58
Add to Storage Logic ...................................................... 12
Additional Print Control Feature ................................. 82
Additional Quotient Digits .......................................... 75
Address Arithmetic, Increased Core Storage .................. 78
Address Checking .......................................................... 14
Address Codes ................................................................ 18
Address Compare Light ............................................... 128
Address Invalid Light ................................................... 128
Address Modification .................................................... 106
Address Modification, Increased Core Storage ............... 78
Address Modification, Indexing .................................... 84
Address Modification with Indexing Feature .................... 78
Address Modification Without Indexing Feature ............... 106
Address Registers ........................................................ 19
Address Selection Dial ................................................ 134
Address Stop .................................................................. 114
Address Validity Checking ........................................... 14
Address Validity, Increased Core Storage ....................... 77
Addressing ................................................................. 10
Addressing, Increased Core Storage ............................. 77
Adjusting the Secondary Stackers ................................. 98
Advanced Systems Design ............................................. 9
Algebraic Sign Control, Addition .................................... 28
Algebraic Sign Control, Division .................................... 75
Algebraic Sign Control, Multiplication ......................... 74
Algebraic Sign Control, Subtraction .............................. 30
Alter Key-Light, 1407 .................................................. 125
A-Cartridge .................................................................. 125
Any Disk Unit Error Condition, 1405 ......................... 123
Appendix ...................................................................... 160
Area Definition ............................................................ 23
Arithmetic ..................................................................... 28
Arithmetic Operations .................................................... 28
Arithmetic Operations Sign .......................................... 28
Arithmetic Overflow .................................................... 28
Asterisk Control ............................................................. 82
Asterisk Protection ........................................................ 82
Autocoder ...................................................................... 23
Autocoder Coding Sheet ............................................... 23
Autocoder Mnemonics .................................................. 23
 Auxiliary Console ........................................................ 116
Auxiliary Console Switches ......................................... 118
Auxiliary Mode Switch ................................................ 118

B = A Light .................................................................. 110
B = A Light .................................................................. 111
B = A Light .................................................................. 111
B < A Light .................................................................. 111
B-Address .................................................................... 16
B-Address Register ....................................................... 20
B-Address Register After Branch ................................... 22
B-Address Register Key-Light ....................................... 111
B-Light ......................................................................... 109
B-Register .................................................................... 20
B Special Character, 1407 .......................................... 72
Backlight Key .............................................................. 114
Backspace Tape Record ................................................ 60
Batching ....................................................................... 12
BCD Code and Column Binary Punches ......................... 90
Binary-Code Decimal ................................................... 13
Binary Tape Instructions .............................................. 92
Bit Configuration, Increased Core Storage .................... 77
Bit Display Lights ........................................................ 111
Bit Switches .................................................................. 118
Bit Values ..................................................................... 106
Body Control Word ...................................................... 41
Branch ......................................................................... 33
Branch if Bit Equal ........................................................ 92
Branch if Character Equal .......................................... 53
Branch if End of Read .................................................. 62
Branch if Indicator On .................................................. 33
Branch if Indicator On, 1407 ........................................ 73
Branch if Indicator On, Overlap Mode ........................... 103
Branch if Indicator On, Sense Switches ......................... 87
Branch if Overflow Indicator On .................................... 29
Branch if Tape Error .................................................... 62
Branch if Word Mark and/or Zone ................................. 33
Branch Indicators, Overlap ........................................... 103
Branch Unconditional .................................................. 32
Brush Interlock, 1403 ................................................... 122
Calculating Processing Time ......................................... 149
Calculating Total Job Time ........................................... 159
Card-Deck Supports ..................................................... 98
Card-Deck Schematic, 1402 .......................................... 10
Card Feeding ............................................................... 10
Card Punch Timing ...................................................... 141
Card Punching Time ..................................................... 45
Card Reader Instructions ............................................... 44
Card Reader Timing ..................................................... 140
Card Reading Time ..................................................... 44
Carriage Busy Indicator ............................................... 82
Carriage Control Tape .................................................. 123
Carriage Controls ........................................................ 120
Carriage Release Button, 1407 .................................... 130
Carriage Restore Key ................................................... 120
Carriage Space Key ..................................................... 150
Carriage Stop Key ........................................................ 120
Carriage Tape Brushes ................................................ 124
Chain Cartridge Adapter ............................................... 94
Chaining ...................................................................... 20
Chaining Instructions ................................................... 20
Character Code Chart .................................................. 171
Character Display ........................................................ 113
Character Rate Tape ..................................................... 149
Character Registers ..................................................... 19
Check Cartridge, Magnetic Tape .................................... 53
Check Manual Entry of Address Registers ....................... 138
Check Reset Key, 1401 ................................................ 114
Check Reset Key, 1402 ................................................ 118
Check Reset Key, 1403 ................................................ 119
Check Stop Switch ....................................................... 116
Chip Light ................................................................. 118
Cleaning the Chain ..................................................... 127
Clear Key, 1407 .......................................................... 71
Clear Key Light ........................................................... 129
Clear, Move, Load and Word Mark Operations ............... 35
Clear Storage .............................................................. 38
Clear Storage and Branch ............................................ 36
Clear Storage Routine .................................................. 165
Clear Word Mark, A-Address ........................................ 37
Clear Word Mark, A- and B-Address ............................ 37
IBM 1404 Printer .................................................. 8
IBM 1405 Disk Storage Instructions ...................... 68
IBM 1406 Indicator Lights ..................................... 68
IBM 1405 Disk Storage Unit Indicator Lights .......... 125
IBM 1405 Indicators ............................................. 68
IBM 1405, Model 1 ............................................. 63
IBM 1406 Storage Unit ......................................... 76
IBM 1406 Storage Unit Models ......................... 76
IBM 1406, Model 1 ............................................. 77
IBM 1406, Model 2 ............................................. 77
IBM 1407 Console Inquiry Station ....................... 70
IBM 1407 Console Inquiry Station Instructions ...... 71
IBM 1407 Indicator Lights ................................... 73
IBM 1407 Indicators ........................................... 73
IBM 1418 Optical Character Reader ...................... 8
IBM 1419 Magnetic Character Reader .................. 8
IBM 7070 Compressed Tape .................................. 87
IBM 7070 Tape Processing .................................... 54
IBM 7090 Data Processing Systems ...................... 8, 89
IBM 730 Magnetic Tape Unit ................................ 57
IBM 730 Magnetic Tape Unit Models ............. 57
IBM 7330 Tape Tunnings .................................... 154
IBM 7701 Magnetic Tape Transmission Terminal .... 35
IBM BASIC System ........................................... 65
IBM Scientific Data Processing System ............... 8
IBM Imperative Operations ............................... 23
IBM Impression and Carriage Control, 1407 .......... 130
IBM Impression Indicator Lever ......................... 130
IBM Index Card Switching ................................. 63
IBM Increased Core Storage ............................... 78
IBM Increasing an Address ................................. 106
IBM Increasing an Address, Indexing ................. 108
IBM Index Checking ........................................... 84
IBM Index Locations .......................................... 84
IBM Indexing Feature ......................................... 84
IBM Indicator Lights, 1405 .............................. 103
IBM Indicators, Processing Overlap ................. 103
IBM Indicators, Sense Switches ......................... 87
IBM Input-Output Control System (IOCS) ............ 27
IBM Input-Output Operations ............................ 44
IBM Input-Output Storage Assignments .............. 18
IBM Inquiry, 1407 ............................................ 70
IBM Inquiry Clear, 1407 .................................... 73
IBM Inquiry Station, Printer ................................ 70
IBM Inserting Control Tape in Carriage .............. 124
IBM Installing a Ribbon, 1407 .......................... 132
IBM Instruction Describing ............................... 16
IBM Instruction Format ...................................... 16
IBM Instruction Format, 1405 ........................... 64
IBM Instruction-Length Lights ............................ 111
IBM Instruction Loading, Schematic .................... 113
IBM Instruction Phase ........................................ 113
IBM Instruction Sizes ........................................ 17
IBM Instructions, No-Address ............................ 22
IBM Instructions, Single-Address ....................... 22
IBM Instructions, 1407 ...................................... 71
IBM Instructions, Processing Overlap ................. 99
IBM Inter-Record Gap (IRG) ............................ 53, 60
IBM Interchangeable Cartridge Adapter ............. 94
IBM Interchangeable Cartridge Reader ............... 94
IBM Interlock Conditions, Processing Overlap ....... 104
IBM Internal Checking ........................................ 14
IBM Introduction, 1401 Reference Manual ........... 7
IBM IOCS (Input-Output Control System) ............ 27
IBM I/O Instruction Execution Mode .................. 112
IBM I/O Check Reset Switch ............................. 116
IBM I/O Check Stop Switch ............................... 115

Lateral Print Vernier ........................................ 131
Length of Forms Between Perforated Lines ........... 162
Library Unit ................................................... 63
Line Position Check, 1407 ................................ 130
Line Space ....................................................... 73
Line-Space Lever .............................................. 121
Line Spacing, Forms Design ............................. 163
Literals .......................................................... 23
Load Characters to A Word Mark, A-Address ....... 40
Load Characters to A Word Mark, A- and B-Address 39
Load Characters to A-Address ............................ 113
Load Rewind Key ............................................. 134
Logic ............................................................. 14
Logic Block Lights ........................................... 110
Logic Lights ..................................................... 110
Logic Operations ............................................. 32
Longitudinal Redundancy Check Register (LRCR) .... 50
Low-Speed Start, 1403 ...................................... 123
Low-Speed Stop, 1403 ...................................... 123

IBM 1408 Disk Storage Unit ......................... 63
IBM 1408 Disk Storage Unit Indicator Lights ........ 125
IBM 1408 Indicators ......................................... 68
IBM 1408 Model 1 ........................................... 63
IBM 1408 Model 2 ........................................... 77
IBM 1409 Indicators ......................................... 73
IBM 1417 Indicators ......................................... 73
IBM 1418 Optical Character Reader .................. 8
IBM 1419 Magnetic Character Reader ................. 8
IBM 7070 Compressed Tape ................................ 87
IBM 7070 Tape Processing .................................. 54
IBM 7090 Data Processing Systems ...................... 8, 89
IBM 730 Magnetic Tape Unit ............................. 57
IBM 7330 Tape Tunnings .................................... 154
IBM 7701 Magnetic Tape Transmission Terminal .... 35
IBM BASIC System ........................................... 65
IBM Scientific Data Processing System ............... 8
IBM Imperative Operations ............................... 23
IBM Impression and Carriage Control, 1407 .......... 130
IBM Impression Indicator Lever ......................... 130
IBM Index Card Switching ................................. 63
IBM Increased Core Storage ............................... 78
IBM Increasing an Address ................................. 106
IBM Increasing an Address, Indexing ................. 108
IBM Index Checking ........................................... 84
IBM Index Locations .......................................... 84
IBM Indexing Feature ......................................... 84
IBM Indicator Lights, 1405 .............................. 103
IBM Indicators, Processing Overlap ................. 103
IBM Indicators, Sense Switches ......................... 87
IBM Input-Output Control System (IOCS) ............ 27
IBM Input-Output Operations ............................ 44
IBM Input-Output Storage Assignments .............. 18
IBM Inquiry, 1407 ............................................ 70
IBM Inquiry Clear, 1407 .................................... 73
IBM Inquiry Station, Printer ................................ 70
IBM Inserting Control Tape in Carriage .............. 124
IBM Installing a Ribbon, 1407 .......................... 132
IBM Instruction Describing ............................... 16
IBM Instruction Format ...................................... 16
IBM Instruction Format, 1405 ........................... 64
IBM Instruction-Length Lights ............................ 111
IBM Instruction Loading, Schematic .................... 113
IBM Instruction Phase ........................................ 113
IBM Instruction Sizes ........................................ 17
IBM Instructions, No-Address ............................ 22
IBM Instructions, Single-Address ....................... 22
IBM Instructions, 1407 ...................................... 71
IBM Instructions, Processing Overlap ................. 99
IBM Inter-Record Gap (IRG) ............................ 53, 60
IBM Interchangeable Cartridge Adapter ............. 94
IBM Interchangeable Cartridge Reader ............... 94
IBM Interlock Conditions, Processing Overlap ....... 104
IBM Internal Checking ........................................ 14
IBM Introduction, 1401 Reference Manual ........... 7
IBM IOCS (Input-Output Control System) ............ 27
IBM I/O Instruction Execution Mode .................. 112
IBM I/O Check Reset Switch ............................. 116
IBM I/O Check Stop Switch ............................... 115

Keys and Levers, 1407 .................................... 130
Language ......................................................... 13
Last Card Indicator, Processing Overlap ............. 104
Last Card Switch .............................................. 87

Machine Features, 51-Column Feed Feature ........... 95
Magnetic Core Storage ........................................ 11
Magnetic Disk Storage ........................................ 13
Magnetic Tape .................................................. 55
Magnetic Tape Characteristics ........................... 55
Magnetic Tape Operations ................................... 57
Magnetic Tape Operations, Summary .................. 58
Magnetic Tape Storage ........................................ 13
Manual Address Switches .................................. 111
Manual Controls .............................................. 120
Manual Display ................................................ 136
Margin Release Key ......................................... 130
Margin-Set Key ............................................... 130
Margin Holes ................................................... 163
Method of Printing ........................................... 11
Methods, Programming ....................................... 7
Minus Sign ...................................................... 28
Minus Symbol, Signs, and Control Left ............... 83
Mnemonic Literatures ........................................ 28
Mnemonic, MPS ............................................... 24
Mode-Change Character ..................................... 68
Mode-Change, Compressed Tape ......................... 87, 88
Mode Switch .................................................... 112
Modify Address, A-Address ............................... 78
Modify Address, A- and B-Address ..................... 78
Modify Address Instructions .............................. 107
Modifying the File Feed ..................................... 95
Modulus 4 ....................................................... 108
Modulus 16 ...................................................... 107
Move and Binary Divide .................................... 92
Move and Insert Zeros ........................................ 88
Move Binary Code ............................................. 93
Move Characters and Edit ................................... 41
Move Characters and Suppress Zeros ................... 38
Move Characters to A or B Word Mark, A-Address 38
Move Characters to A or B Word Mark, A- and B-Address 37
Move Characters to Record or Group Mark .......... 38
Move Numerical .............................................. 39
Move Record Feature ......................................... 86
Move Zone ...................................................... 39
Multiple Copy Control Lever, 1407 .................... 131
Multiple Copy Forms ......................................... 183
Multiplication and Division Subroutines .............. 168
Multiplication Subroutine .................................. 168
Multiply ........................................................ 74
Multiply Divide Feature ...................................... 74
Multiply Times .................................................. 150, 151
Negative Sign ................................................... 28
No-Address Instructions ..................................... 22
No-Operation ................................................... 34
Non-Process Run-Out Punch Key ......................... 118
Non-Process Run-Out Read Key ......................... 118
Numerical Print Feature .................................... 94
Numerical Print Feature, Timing ......................... 144

Index 173
| Timing .......................................................... | 140 |
| Timing Chart, Summary .................................. | 148 |
| Timings, Divide .......................................... | 150 |
| Timings, Multiply ....................................... | 150, 151 |
| Timings, Systems ........................................ | 143 |
| Total Job Time .......................................... | 159 |
| Tractor Slide Bar ....................................... | 121 |
| Transport Light ......................................... | 118 |
| True Add .................................................. | 28 |
| Turning Off the System .................................. | 136 |
| Unequal Address Compare, 1405 ...................... | 69 |
| Unload Key ................................................ | 135 |
| Using Modulus Arithmetic ............................. | 106 |
| Using the Modify Address Instruction .............. | 107 |
| Validity Checking ....................................... | 14 |
| Validity Light ............................................ | 118 |
| Variable Word Length ................................... | 15 |
| Vertical Print Adjustment ............................. | 121 |
| Weight, Paper ............................................ | 162 |
| Width of Forms .......................................... | 161, 162 |
| Word Marks ............................................... | 15 |
| Word Separator Character ............................. | 50, 60 |
| Words ...................................................... | 15 |

| Write a Line ............................................. | 47 |
| Write and Branch ....................................... | 47 |
| Write and Punch ........................................ | 49 |
| Write and Read .......................................... | 48 |
| Write Binary Tape ...................................... | 93 |
| Write Check ................................................ | 67 |
| Write Disk Check ........................................ | 67 |
| Write Disk, Full Track ............................... | 66 |
| Write Disk, Full Track with Word Marks .......... | 66 |
| Write Disk, Single Record ........................... | 66 |
| Write Disk, Single Record with Word Marks ...... | 68 |
| Write on Console Printer ............................. | 71 |
| Write on Console Printer with Word Marks ........ | 72 |
| Write, Punch, and Branch ............................ | 50 |
| Write, Read, and Branch .............................. | 48 |
| Write, Read, and Punch .............................. | 50 |
| Write, Read, Punch, and Branch .................... | 50 |
| Write, Read Punch Feed .............................. | 81 |
| Write, Read Punch Feed and Branch ............... | 81 |
| Write Tape ................................................ | 58 |
| Write Tape in Overlap Mode ......................... | 101 |
| Write Tape Mark ......................................... | 69 |
| Write Tape with Word Marks ......................... | 59 |
| Write Word Mark ......................................... | 47 |
| Write Word Marks and Branch ....................... | 48 |
| Writing High-Density Tape in Overlap Mode ....... | 101 |
| Wrong-Length Record, 1405 .......................... | 69 |

| Zero and Add ............................................ | 30 |
| Zero and Subtract, A-Address ....................... | 31 |
| Zero and Subtract, A- and B-Address .............. | 31 |
| Zero Suppression, Editing ............................ | 43 |
| Zone Value, Increased Core Storage ................ | 77 |