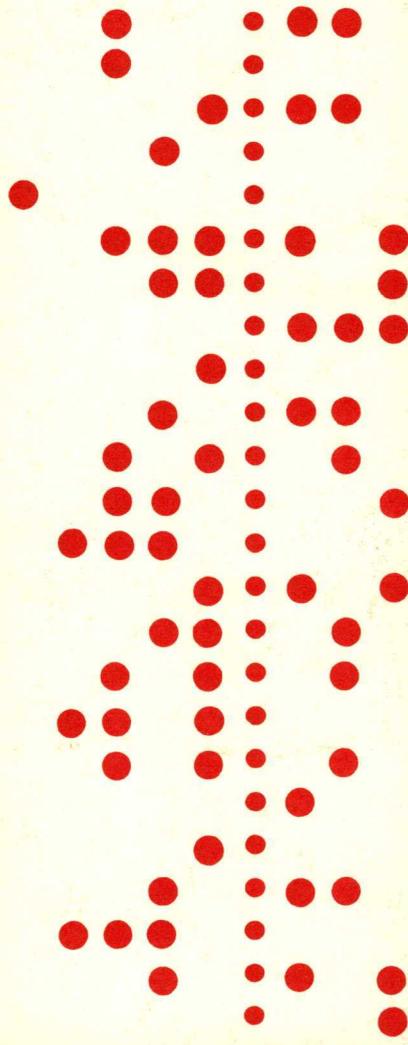


# Friden



*Programming Manual for*

## **Friden Flexowriter®**

AUTOMATIC WRITING MACHINE



MODELS  
SFD, SPS, SPD

MASTER

DATA

SELECT

## **PURPOSE OF THE MANUAL**

The purpose of this manual is to instruct the beginner in the basic steps of programming Friden Systems Models Flexowriters. The programming, illustrated throughout, takes the reader through a step-by-step Purchase Order Writing application, written on the Friden Flexowriter, Model SPS (Systems Programatic Single Case). The program includes a by-product tape for tape-to-card conversion. All steps, from the beginning through the end-product, are explained in detail.

Copyright, 1959, by Friden, Inc.  
Revised, 1963, by Friden, Inc.

"Flexowriter" - Trademark of Friden, Inc.

Composition for this manual was set on the Friden  
Justowriter® automatic tape-operated copy setting machine,  
a product of

FRIDEN, INC.

## INTRODUCTION

PROGRAMMING A FLEXOWRITER IS SIMPLE. It does not require a great deal of instruction or special skill.

Programming is merely the setting up of instructions for a piece of automatic equipment. In the case of an extremely complex unit, such as an electronic computer, the programming can be very involved and requires a highly trained specialist. A computer programmer may have to choose between fifty or sixty different commands or instructions to accomplish a step in his program. On a Flexowriter, however, the programmer seldom has to choose from more than six or eight instructional codes at any time.

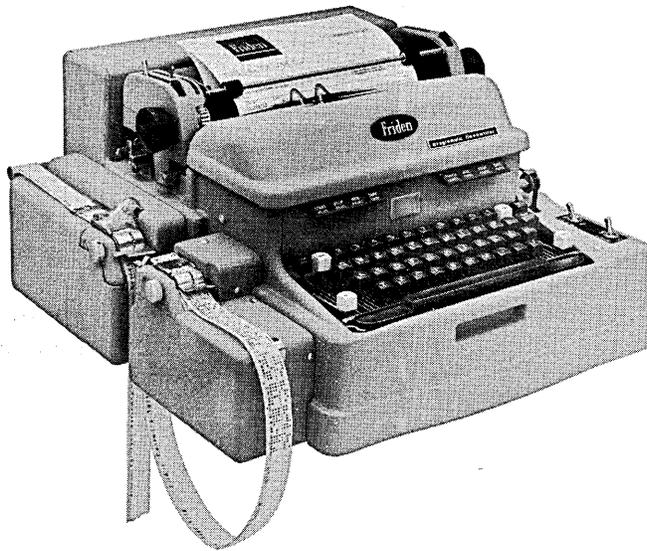
The purpose of this manual is to enable anyone, without previous experience in programming, to program a basic Flexowriter. There are several different models of the Flexowriter currently available. The differences between them is actually very slight. If a person can learn to program one, it is not at all difficult to advance to the other models in the complete Friden Flexowriter line.

We have chosen one model as the basis for this manual. It is the Systems Programmatic Single Case (Model SPS). This machine is the most popular and most versatile of all Flexowriter Programmatic models. For this reason, it is the model the student should master first.

It is suggested that a person familiarize himself with the basic functions of a Flexowriter before attempting to learn actual programming. In other words, know how to punch and read tapes on the Flexowriter and be familiar with the operation of the Start Read, Stop Read, Stop Code and Tape Feed functions of the machine. Such basic information can be found in a manual of instructions for the Systems Programmatic and mastered in a very short time.

## THE PROGRAM TAPE

A Flexowriter is programmed through the use of a tape known as the Program Tape. The Program Tape is a punched paper tape prepared on a Flexowriter containing instructional codes which tell the Flexowriter how to function in order to do a specific job. When inserted into the Flexowriter Reader, it operates the Flexowriter automatically as the programmer intends. It might be possible that the same Flexowriter would be used for two different jobs or applications, such as Check Writing and Purchase Order Writing. In each case, the Flexowriter would be expected to operate somewhat differently. Therefore, two separate Program Tapes would be used. One tape would contain the instructional codes necessary to prepare a Check and the other would contain the instructional codes needed to prepare the Purchase Order. The purpose of this manual is to show you how simple it is to determine the requirements of the Program Tape for any type of form.



### The Functions of the Program Tape

The Program Tape usually contains the following kinds of codes, or instructions

1. Codes to Space the Carriage of the Flexowriter.
2. Codes to turn the Tape Punch ON and OFF.
3. Codes for the Control of Other Equipment, such as the Tape-to-Card Punch.

Now let us look at the above codes in detail and see their exact functions.

**CODES TO SPACE THE CARRIAGE OF THE FLEXOWRITER.** There are only three codes on a Flexowriter Programatic which Space the Carriage and, therefore, the form. These are the Space Code, Tab Code and Carriage Return Code. Through the use of these codes, either individually or in combinations, we can advance to any place on a form we desire. In programming a Flexowriter, it is a general rule to put these codes for Spacing the Carriage into the Program Tape instead of having the operator manually touch the keys. Here is an example which illustrates how this works.

**THE AJAX COMPANY . . . .**

**Purchase Order**

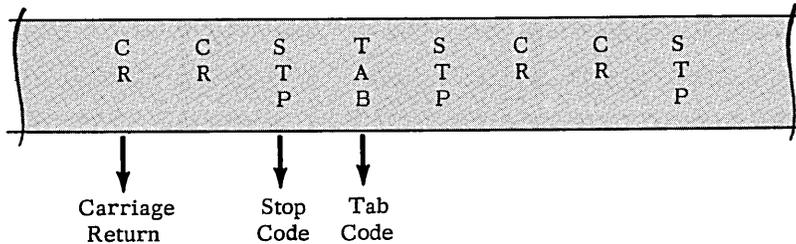
Order No.  
63205

Order Date  
05/14/58

To:  SMITH SPECIALTY CO.

This is a portion of a typical business form. We want to prepare a Program Tape which will guide us in writing this form. The tape will contain only codes to Space the Carriage of the Flexowriter and Stop Codes to stop the Tape Reader in the proper fields to allow a fill-in of needed data. In this case, we will be preparing no punched tape as a by-product of the typing.

The Program Tape would look like this.



Notice, it starts with two Carriage Return Codes. This is a good way to start any Program Tape. It gives us positive assurance that the Carriage is at the exact left margin of the form when the first entry (Order No.) is made. By starting with more than one Carriage Return Code, the operator will be able to line up the Writing Line Guide with a pre-determined location on the form. In this case, the line which borders the top of the Order No. field would be used to line up the form.

After inserting the Program Tape in the Reader, the operator touches the Start Read Switch, and the first two Carriage Return Codes are read. Following the Carriage Return Codes is a Stop Code. This code will stop the Reader so the operator can key-in the Order No. (63205).

After so doing, the operator touches the Start Read Switch, causing the next code in the Reader to be sensed. This is a Tab Code which causes tabulation to the next field on the form (Order Date). (Of course, a tab stop must be set at the proper position on the tab rack to allow tabulation to terminate exactly at the beginning of the Order Date field).

Following the Tab Code is a Stop Code. This Stops the Reader, allowing the operator to key-in the Order Date (05/14/58). The operator then touches the Start Read Switch and the next two Carriage Return Codes are read. These Carriage Return Codes will space the form down to the Name field.

Here another Stop Code is read, the Tape Reader Stops allowing the operator to key-in the Name (SMITH SPECIALTY CO.).

You can see how simple programming really is. You probably had no difficulty understanding it. And, if you can understand this, you should have no trouble mastering the remaining principles in the manual.

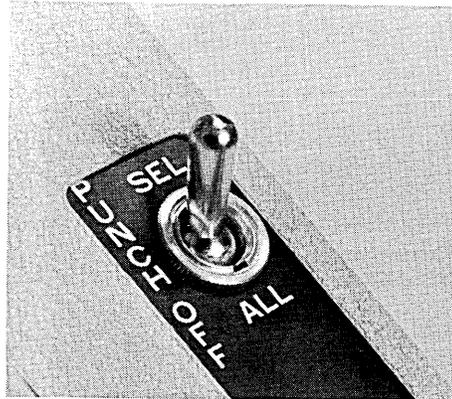
Notice the programming was such that it reduced the operator's job to simply touching the Start Read Switch and making an entry when the Reader stopped. This is extremely important. All programming, no matter how complex the application, should require no more than this of the operator -- THE TOUCH OF THE START READ SWITCH AND THE TYPING OF AN ENTRY WHEN THE

READER STOPS. This method of programming reduces to a minimum what are called OPERATOR DECISIONS. In other words, instead of having to decide which functional key to touch, the operator always touches the Start Read Switch and the Program Tape makes the decisions.

CODES TO TURN THE TAPE PUNCH ON AND OFF. Sometimes, when a document is written on a Flexowriter Programatic, no By-Product Tape is punched. At other times we wish to punch all of the information typed on the document into the By-Product Tape. Most of the time, however, it is necessary to turn the Punch ON and OFF at certain places on the form so only selected parts of the typing punch into the By-Product Tape.

We can regulate the Tape Punch to operate in any of these three ways through use of a Toggle Switch on the side of the Flexowriter, known as the Punch Switch (see illustration). This is a three position Switch.

In the Center position, labeled PUNCH OFF, the Flexowriter Tape Punch will be completely inoperative and no By-Product Tape will be punched.



The Rear position of the Switch is labeled PUNCH ALL. In this position, the Tape Punch will always be ON and can never be turned OFF. The By-Product Tape produced will contain all of the information typed. We call this tape a COMPOSITE TAPE.

The Forward position of the Punch Switch is called the PUNCH SELECT position. In this position, the Punch can be turned ON and OFF whenever needed. To turn the Punch ON we use the ON1 key on the keyboard to turn it OFF use the OFF key. If the Punch Switch is in the Select position and the ON1 key is touched, or its code read in the Tape Reader, the Punch will remain ON until an OFF code is read or the OFF key is touched. The OFF code will turn the Punch OFF and any further typing will not be punched into the By-Product Tape. This By-Product Tape will, therefore, contain only selected parts of the information that is typed on the document. We call such a By-Product Tape a SELECT TAPE.

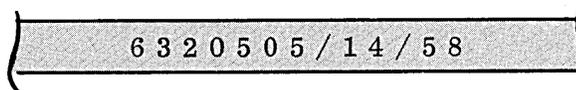
The Select position of the Punch Switch is often used when we are preparing a document on the Flexowriter and wish to produce a By-Product Tape to control another piece of tape operated equipment such as a Tape-to-Card Punch.

If the Punch Switch is in Select position and the ON1 or the OFF code is read, or if either key is manually touched, neither code itself will be punched in the By-

Product Tape. In other words, the Flexowriter is so designed that the ON 1 and OFF keys punch their respective codes only in the PUNCH ALL position and not in the PUNCH SELECT position. This is done to assure that the Select Tapes are free and clean of useless ON 1 and OFF codes when read in another piece of tape-operated equipment.

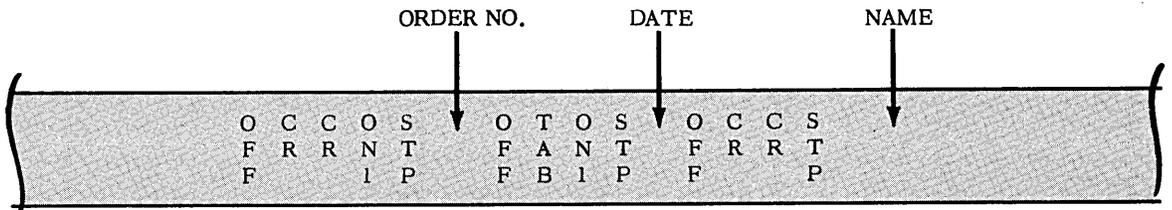
|                                 |   |   |
|---------------------------------|---|---|
| <b>THE AJAX COMPANY . . . .</b> |   | <b>Purchase Order</b>   |
|                                 | <div style="border: 1px solid black; padding: 2px; display: inline-block;">Order No.<br/><b>63205</b></div>                                   | <div style="border: 1px solid black; padding: 2px; display: inline-block;">Order Date<br/><b>05/14/58</b></div> |
| <b>To:</b>                      | <div style="border: 1px solid black; padding: 2px;"> <b>SMITH SPECIALTY CO.</b><br/> <b>100 ELM STREET</b><br/> <b>ANYTOWN, U.S.A.</b> </div> |   |

Here again we see the portion of the business form used as an illustration on Page 2. In that example, no tape was punched as a by-product of the writing operation. Now, let us assume we want to write the same form and also punch a By-Product Tape like the following.



Notice that only the Order No. and the Order Date are to be punched into the By-Product Tape and there are no form spacing codes (Tab, Space, Carriage Return) included.

Our Program Tape should look like this.



It is the same as the Program Tape on Page 3, except for the addition of ON 1 and OFF codes at certain places to regulate what is and what is not to punch into the By-Product Tape.

An OFF code should start the tape. This is to assure that the Punch is turned OFF when the following two Carriage Return codes are read. These, we have already determined, must not go into the By-Product Tape.

Following the Carriage Return codes, is an ON 1 code. This will turn the Flexowriter Tape Punch ON so that when the Reader stops for the manual entry of the Order No. (63205), it will be punched into the By-Product Tape when the operator

keys-in the digits. The Stop Code, itself, will not punch into the By-Product Tape. Later we will see the reason why this is so.

Immediately following the Stop Code is an OFF code. This will prevent the Tab code, from punching into the tape. After the Tab code is another ON 1 code which turns the Punch back ON. When the Reader stops for the manual entry of the Order Date (05/14/58), it will be punched into the By-Product Tape when the operator keys-in the digits. Again, the Stop Code is not punched into the tape.

Following the Stop Code is an OFF code. This will turn the Punch OFF and no more information will be punched into the By-Product Tape.

The Program Tape is prepared with the Punch Switch in ALL position and all codes are punched in the exact sequence indicated. The PUNCH ALL position is used because we want all codes to punch. Now, the Program Tape can be inserted into the Flexowriter Reader and used to guide us in writing the document. As we read the tape, the Punch Switch can either be in OFF, ALL, or SELECT position, depending on the result you desire.

In OFF position, no By-Product Tape will be punched.

In ALL position, all codes, except Stop Codes, will punch into the By-Product Tape. The result would be a Composite Tape which can be run back through the Reader to produce the document automatically.

In most cases, however, we use the SELECT position to prepare the document. In this position, the ON 1 and OFF codes in the Program Tape would be effective and the Select information desired would be punched. This By-Product Tape is a SELECT TAPE.

Notice that our programming has followed the same basic rule. The operator's task is reduced to touching the Start Read Switch and making the correct entry when the Reader stops. There is no need to turn the Punch ON or OFF. These functions are under the control of the Program Tape and are completely automatic.

### Capturing Repetitive Data

In addition to the foregoing, where the Flexowriter is used to prepare documents such as Purchase Orders, Sales Orders, etc., the Program Tape will often include repetitive data.

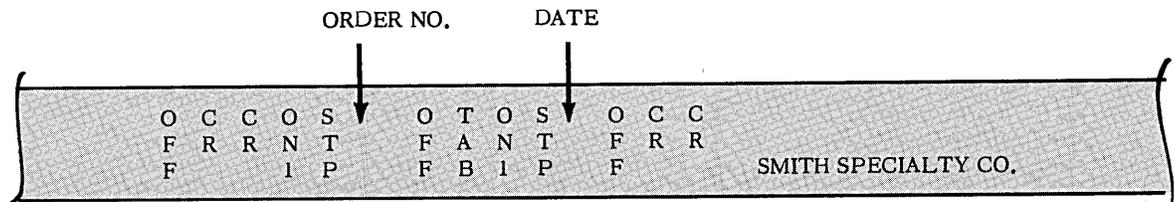
In preparing Purchase Orders, for example, most companies restrict their purchases to a limited number of Vendors. Often the great majority of their purchases will also be restricted to a limited number of Items. In this case, the use of the Program Tape can be carried one step further. A separate Program Tape can be prepared for each of the Vendors and one for each regularly purchased Item.

In addition to containing codes to space the form, turn the Punch ON and OFF, Stop Codes for manual entries, this new Program Tape will contain any data that

is repetitive for a given Vendor or Item. A Vendor Tape contains the Vendor's name and address, since these would always be constant when preparing a Purchase Order for that Vendor. By the same token, the Item Tape would contain the item description, the item code number and probably the unit price since this is constant information always written when a Purchase Order is prepared for a given Item.

Let us go back to the programming sample we have been using to illustrate the functions of a basic Program Tape. You will recall, this Program Tape contained Stop Codes to allow entry of the Order No., Order Date and Vendor's Name. We could never include the Order No. or the Order Date in our Vendor Program Tape. These are called variable entries because they vary from one Purchase Order to another. The Vendor's Name is, on the other hand, a constant entry and can be included in the Program Tape made up for a particular Vendor.

The new Vendor Tape will be as follows.



Notice it is the same as the Program Tape illustrated on Page 5, except the Vendor's Name is actually punched in the Program Tape in place of the Stop Code. When this Program Tape is used, the Reader of the Flexowriter will not stop to allow entry of the Vendor's Name, but, without stopping, will write the name automatically from the Program Tape.

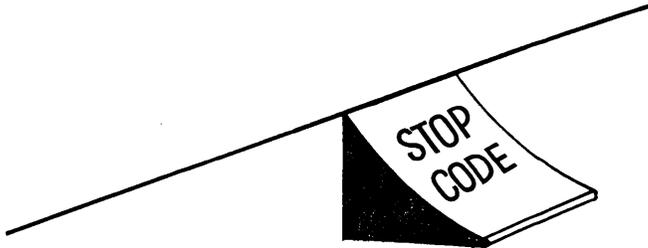
In a typical Purchase Order application, and in many other areas of business, it will often be found that eighty percent or more of the data to be entered on a form can be included in individual Program Tapes. This means the greatest part of the documents will be typed automatically at a speed of 100 words per minute. Since these individual Program Tapes for the Vendors and the Items are verified before use to assure accuracy, their use will considerably reduce the number of typing errors normally occurring in the manual preparation of documents.

Speed and accuracy are, therefore, two of the major benefits derived from the Capturing of Repetitive Data in Program Tape. These individual Program Tapes containing such repetitive data are often called Data Program Tapes. A Program Tape containing only functional codes, as illustrated on Page 5, is called a Master Program Tape.

It will often be found that the Data Program Tape is not a tape at all, but an edge-punched card. Edge-punched cards lend themselves well to such usage, since they are easy to handle, file, and can be labeled. In either case, whether tapes or edge-punched cards are used, the principles of programming will remain the same.

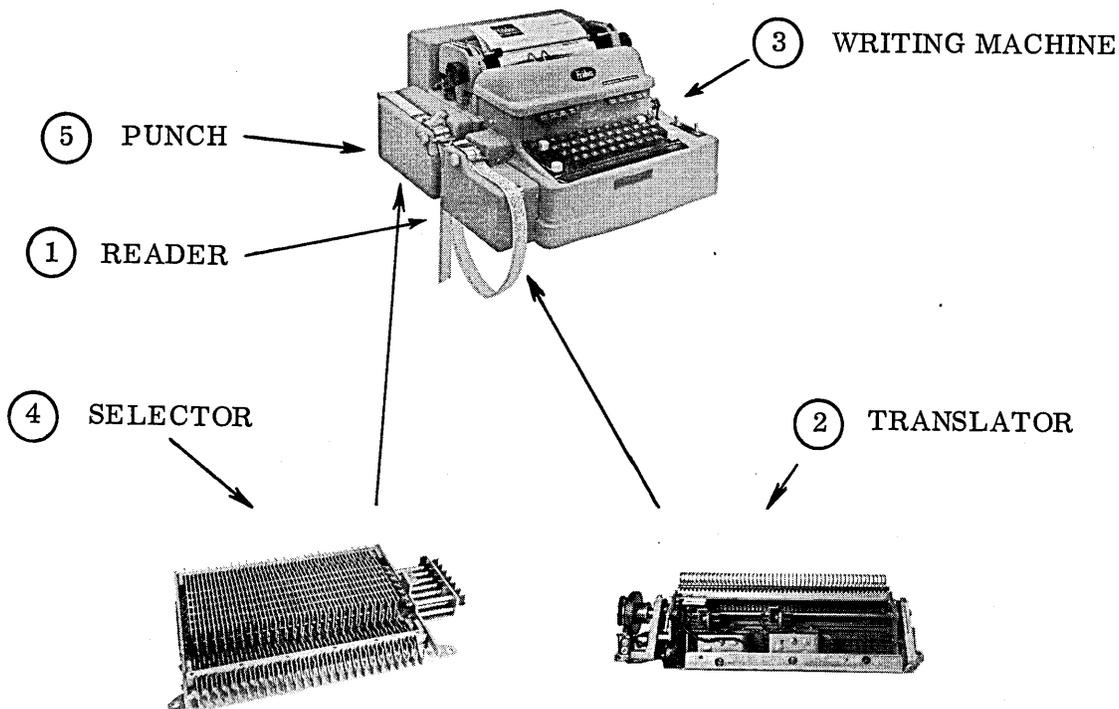
## Reproducing and Non-Reproducing Codes

If you recall, the programming sample on Page 5, you remember the Stop Codes read with the Flexowriter Tape Punch turned ON did not punch into the By-Product Tape. At that time it was said this would be explained later in the manual. This is the explanation: A Stop Code is called a NON-REPRODUCING CODE. This means when it is read in the Flexowriter it will not punch into a By-Product



Tape even if the Tape Punch is turned ON when it is sensed. This is because the Stop Code is not found on the Flexowriter keyboard. A CODE MUST BE ON THE KEYBOARD OF THE FLEXOWRITER IF IT IS TO REPRODUCE, OR PUNCH INTO A BY-PRODUCT TAPE, WHEN READ. The Stop Code is located on the front panel above the keyboard.

To have a better understanding of the above, it is best to have an idea of the mechanics of a Flexowriter. A basic Flexowriter is composed of five components.



1. THE TAPE READER senses codes in a tape and sends electrical impulses to the Flexowriter corresponding to the code read.
2. THE CODE TRANSLATOR receives these electrical impulses from the Reader and converts them into a mechanical action, namely pulling down a keylever corresponding to the code read.
3. THE WRITING MACHINE actually writes and spaces the document being prepared.
4. THE CODE SELECTOR is operated by the keylevers of the Writing Machine. It converts the mechanical keylever action into an electrical code impulse which is sent to the Tape Punch.
5. THE TAPE PUNCH, when turned ON, will accept the impulses from the Code Selector and punch the code into a By-Product Tape.

When a code is reproduced, all five components of the Flexowriter come into play. For example, let it be assumed the numeral 5 is being read. When this code is sensed in the Reader, an impulse corresponding to the code will be sent to the Code Translator. The Translator will operate upon receiving the code and cause the keylever for the numeral 5 to be pulled down. The numeral 5 is written on the document and, at the same time, this action causes the Selector to operate and send an impulse to the Tape Punch. The Punch receives this impulse if it has previously been turned ON by an ON 1 code or if the Punch Switch was in ALL position. The Punch then perforates the By-Product Tape with the proper code for the numeral 5.

Now, if the code read were a Stop Code, or any other code not represented on the keyboard of the Flexowriter, the result would be quite different. The code would be sensed by the Reader and, in the case of a Stop Code, the Reader would stop. An impulse would still be sent to the Code Translator but then it could go no further. There is no keylever for the Translator to pull down. As a result, the code would not be able to reproduce into a By-Product Tape.

We can establish the following as a basic rule. A CODE WILL NOT REPRODUCE WHEN READ UNDER NORMAL CONDITIONS UNLESS IT IS REPRESENTED ON THE FLEXOWRITER KEYBOARD.

#### Automatic Non-Print

Our rule on Non-Reproducing Codes states that codes not represented by keylevers will not reproduce when read under NORMAL CONDITIONS. It is not difficult to see, however, there are times when it is necessary to be able to reproduce such codes.

Let us assume a Sales Order is to be prepared on the Flexowriter, and at the same time we desire to punch a By-Product Tape to be used on another Flexowriter to prepare an Invoice automatically. Such applications require the reproduction of Stop Codes. It will be necessary to punch Stop Codes into the tape

that will be used to write the Invoice to allow the typist to make manual fill-ins. (For example, Invoice Number and Date would be manual entries). This means we must have a method of reproducing the Stop Codes, from the Program Tape that will create the Sales Order, into the By-Product tape that will write the Invoice.

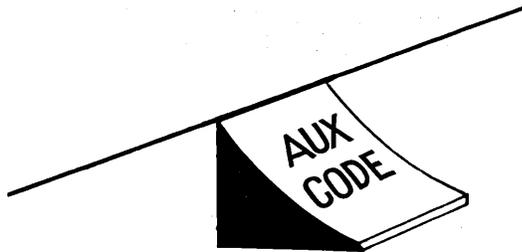
The Automatic Non-Print feature of the Flexowriter Programatic allows the reproduction of any code. Normally, it should be recalled, the Flexowriter Tape Reader impulses the Code Translator, which in turn operates the Writing Machine. When a code, known as a Non-Print Code, is sensed in the Reader all of this changes.

The Non-Print Code causes the Reader to become automatically disconnected from the Code Translator and instead directly connects it to the Tape Punch. Since the Reader and the Translator have become disconnected from each other, no codes that follow it will have any effect on the Writing Machine. Codes for the alphabet, numerals, and punctuation will not print on the form, thus the term Non-Print. The Tab, Carriage Return and Space Bar will not be operated during this Non-Print condition and the operation of the Punch ON and OFF keys will be suppressed. Neither will the Stop Code stop the Reader when read in a Non-Print condition.

Since the Reader becomes connected to the Punch directly after a Non-Print Code is read, all codes that follow it WILL REPRODUCE into the By-Product Tape. Therefore, reading a Non-Print Code causes two changes to take place in the Flexowriter operation.

1. All Flexowriter functions become inoperative.
2. All codes read WILL REPRODUCE into the By-Product Tape provided the Punch was turned ON before the Non-Print Code was read.

The Non-Print Code is put into the Program Tape by holding down the Aux Code Switch and, at the same time, touching the Space Bar. The Aux Code, when depressed in conjunction with other Flexowriter keys, will provide the programmer with many different control or functional codes for the Program Tape.



By depressing the Aux Code Switch and touching the key for the numeral Zero together, a code known as Print Restore will be punched into the Program Tape. When the Print Restore Code is read, the Non-Print operation is discontinued and the Flexowriter is restored back to normal typing and punching.

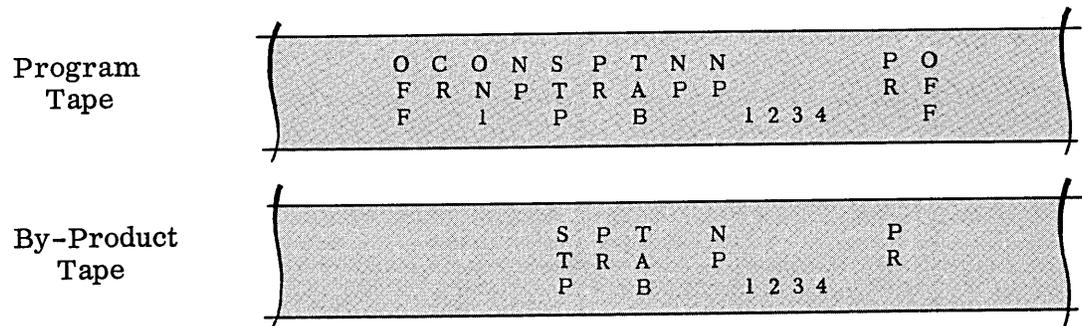
Since the Non-Print Code and the Print Restore Code are not represented by keys on the Flexowriter keyboard, they are Non-Reproducing codes. This means

when they are read under normal conditions they will not punch into a By-Product Tape even if the Punch is turned ON at the time. If, however, either of these codes are read in an Automatic Non-Print condition, they will reproduce into the By-Product Tape, as will any code.

Due to the above, the following rules can be formulated.

1. The Non-Print Code which starts a Non-Print operation will not reproduce. This is because the Flexowriter isn't in a Non-Print condition when it is read.
2. Any Non-Print Codes following the first one will reproduce. By the time these codes are read, the Flexowriter will be in a Non-Print condition allowing any code to reproduce,
3. Thus, a By-Product Tape always contains one less Non-Print Code than a Program Tape. If there is one Non-Print Code in the Program Tape, the By-Product Tape will contain none. If there are two Non-Print Codes in a row in the Program Tape, the By-Product Tape will contain one. By putting the proper number of these Non-Print Codes into the Program Tape we can regulate whether information is to be Non-Printed in the Program Tape alone (one Non-Print Code) or in both the Program Tape and the By-Product Tape (two Non-Print Codes).
4. The Print Restore Code which terminates a Non-Print operation will reproduce since the Flexowriter will still be in a Non-Print condition when this code is read.

The following example illustrates the above possibilities.



In this example, we see a portion of a Program Tape to be run through the Flexowriter Reader in Select position for preparing the By-Product tape illustrated. Notice, first of all, only those codes following the ON 1 code can possibly punch into the By-Product Tape. This is because there can never be any punching in the Select position unless an ON 1 code is read or the ON 1 key is touched.

The first Non-Print Code in the illustrated Program Tape is used to allow the Stop Code, immediately following it, to reproduce into the By-Product Tape. When this Non-Print Code is read, the Flexowriter is put into a Non-Print condition. Since all codes reproduce during Automatic Non-Print, the Stop Code will punch or reproduce into the By-Product Tape as we desire. When the Print Restore Code, following the Stop Code, is read it punches into the By-Product

Tape and also causes the Non-Print operation to cease. The Print Restore Code will reproduce because it will be read in a Non-Print condition.

Note that the By-Product Tape at this point contains only the Stop Code and the Print Restore Code. This is because these were the only codes that reproduced.

Therefore, if this section of the tape were read back through the Reader, the Flexowriter would stop on the Stop Code. When the Print Restore Code was read, the Flexowriter would skip over it without any function taking place.

Further along in this sample Program Tape, we find two Non-Print Codes in a row. The purpose of this Non-Print operation is to prevent the numerals 1234, which follow, from printing on the document being written and yet allow them to punch into the By-Product Tape. The reason two Non-Print Codes, instead of one, are used here is that we desire the numerals 1234 to be Non-Printed when the By-Product Tape itself is run through the Flexowriter.

The first of the two Non-Print Codes will put the Flexowriter into a Non-Print condition. It will not punch into the By-Product Tape because it is a Non-Reproducing Code. By the time the second Non-Print is read, the Flexowriter will be in the desired Non-Print condition. Since any code read, whether represented on the keyboard or not, will reproduce in a Non-Print operation, the second Non-Print Code does punch into the By-Product Tape. The numerals 1234 will not print on the document when read because no printing takes place during a Non-Print operation. They will, however, punch into the By-Product Tape. When the Print Restore Code, following these numerals, is read it punches into the By-Product Tape and simultaneously takes the Flexowriter out of Non-Print and restores it back to normal operation.

Notice the By-Product Tape at this point contains one Non-Print Code, the numerals 1234, and the Print Restore Code. When this section of the Tape is read back through the Flexowriter, these numerals will once more be Non-Printed.

Compare the two Non-Print operations in this sample. Notice how different the result is when two Non-Print Codes are employed rather than one.

### How to Prepare Data Tapes

The concept of punching individual Data Tapes for Vendors, Items and Customers has been explained. If a Flexowriter user employs this concept, it means upon delivery of the machine, hundreds or even thousands of these individual tapes or edge-punched cards will have to be prepared. If the operator was required to manually punch out each code in every Data Tape, the task would be gigantic. Fortunately, this is not the case.

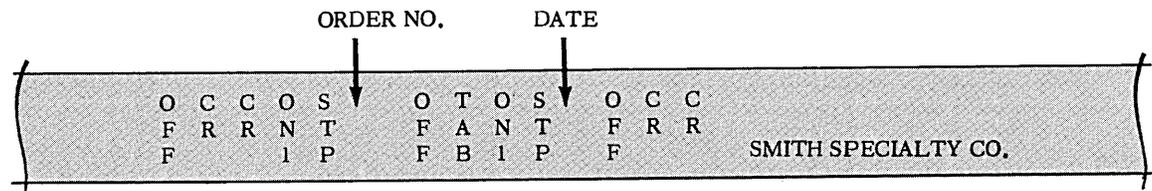
A MASTER PROGRAM TAPE IS ALWAYS EMPLOYED AS A BASIS FOR PREPARING INDIVIDUAL DATA TAPES. This Master Tape contains only functional or control codes. When a Flexowriter operator desires to prepare a Data Tape, she inserts the Master Tape into the Reader of the Flexowriter and puts the Punch Switch in ALL position. The operator merely touches the Start Read Switch and waits for the Reader to stop. When it does, the Data needed is entered. The

process continues in this manner until the tape is completed.

The task is reduced to simply touching Start Read and entering the Data when the Reader stops. All of the programming will be punched into the Data Tapes automatically by the Master Program Tape.

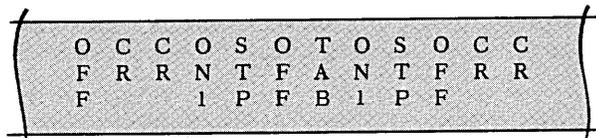
Once the codes needed in the Data Tape have been determined, it is quite simple to determine the codes required to make the Master Program Tape which will produce the Data Tape.

On Page 7, a portion of a sample Data Tape for a Vendor in a Purchase Order application was illustrated. This tape is shown again below.

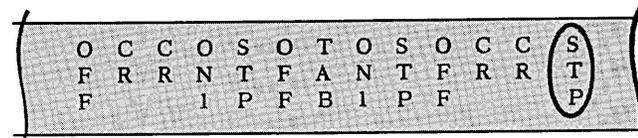


To determine the Master Program Tape that will guide us in making this Data Tape, or others like it, we follow three simple rules.

1. COPY ALL OF THE FUNCTIONAL CODES IN THE DATA TAPE. Do not copy any repetitive data.



2. IN PLACE OF THE REPETITIVE DATA, INSERT STOP CODES. This is done so the Master Tape will stop at these places to allow the manual entry of the repetitive data. Therefore, in our sample we will add a Stop Code where the Vendor's Name appears.



3. WHERE THERE ARE ALREADY STOP CODES IN THE DATA TAPE, USE THE NON-PRINT AND PRINT RESTORE CODES TO ALLOW THE STOP CODES TO REPRODUCE FROM THE MASTER PROGRAM TAPE TO THE DATA TAPE. Here we see that one set of Non-Print and Print Restore Codes will suffice to allow both the Stop Code for the Order No. and the Stop Code for the Order Date to reproduce from the Master Program Tape to the Data Tape.



Now the requirements of the Master Program Tape have been completely determined. To punch the Master Program Tape the Punch Switch is put in ALL position and the codes are punched in the exact sequence indicated. Once these

codes have been manually punched into a Master Program Tape, it will never be necessary to manually punch them again, no matter how many individual Data Tapes are to be made.

To prepare Data Tapes from the Master Program Tape, the Punch Switch remains in ALL position and the Master Program Tape is inserted into the Reader of the Flexowriter. The operator then touches the Start Read Switch and the Program Tape starts to read and cause punching of the Data Tape. The Reader will not stop until the Stop Code for the Vendor's Name is read. At this point, the operator will type the Vendor's Name (SMITH SPECIALTY CO.). This Stop Code will not punch into the Data Tape.

The Stop Codes for the Order No. and the Order Date did reproduce into the Data Tape, but did not cause the tape reading to stop. This is because they are read after a Non-Print Code. We have seen earlier how Stop Codes, when read in a Non-Print condition, will not stop tape reading but will reproduce into the By-Product Tape, as will any code.

The Non-Print Code, which caused Non-Printing to occur, does not punch into the Data Tape. Since there is no second Non-Print Code following it, the Data Tape will contain no Non-Print Codes. Therefore, when the Data Tape is read to prepare a Purchase Order, the Flexowriter WILL STOP to allow the entry of the Order No. and the Order Date, which are variable entries that cannot be included in the Data Tapes.

#### Controlling Allied Equipment with the Flexowriter

In many Flexowriter applications, the tapes produced as a By-Product of writing a document are used in a piece of tape reading equipment other than the Flexowriter itself. Teletype machines, Automatic Graphotypes, Electronic Computers and Tape-to-Card Punches are some of this Allied Equipment which can be used. The requirements of each are different. Each type requires certain control codes in the tape they read. These tell the particular machine how and when to perform its various functions. For example, the Automatic Graphotype, which produces address plates automatically from tape, requires a code known as a Plate Discharge Code to tell it to eject the completed plate.

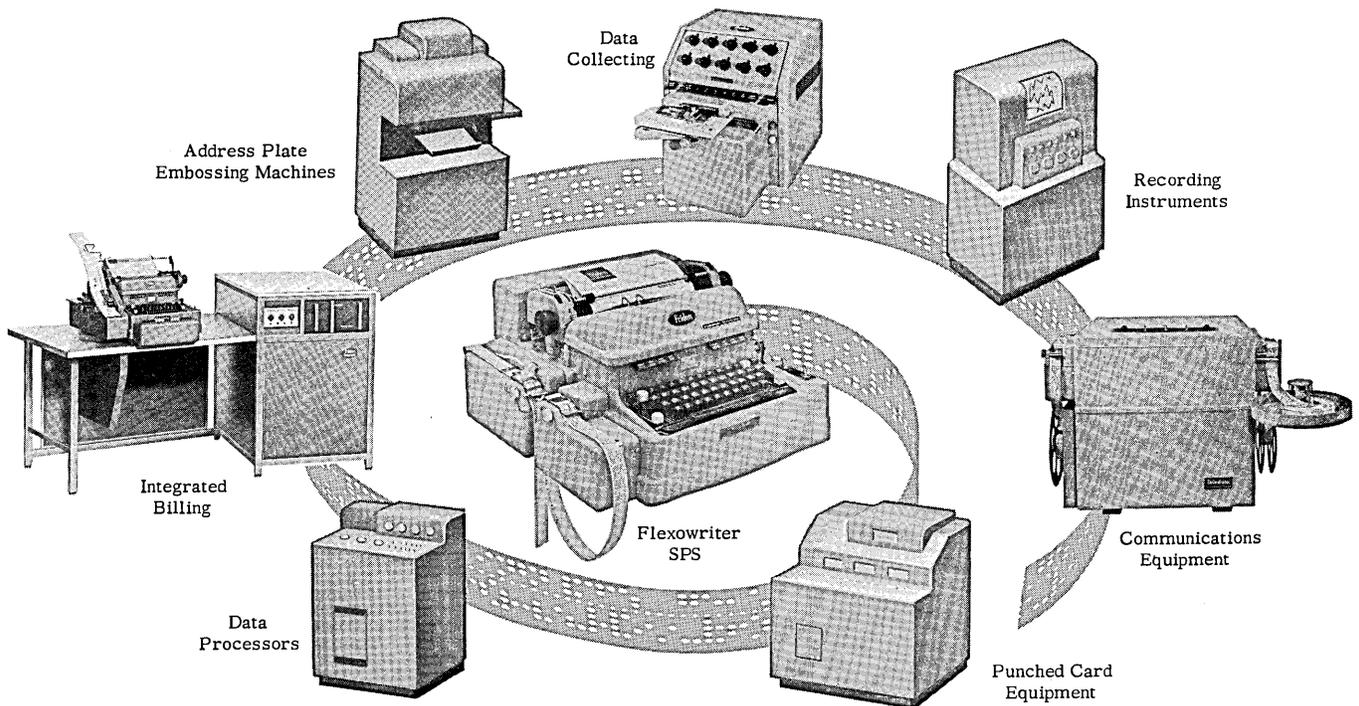
The control code idea is not a new one. The Flexowriter itself requires certain codes, such as the Stop Code, the Non-Print and Print Restore, in the tapes it reads. The Flexowriter programmer, when programming it to work with Allied Tape-Operated Equipment, must know the exact requirements of the Equipment. If he does not have first-hand knowledge, it would be well to work closely with those who do.

#### The IBM Type 046 and 047 Tape-to-Card Punch

Many Flexowriter Programatics are used in applications where tapes produced on the Flexowriter are read by the IBM Tape-to-Card Punch to produce IBM tabulating cards automatically. To show you this, in the next few pages we will

discuss the IBM Tape-to-Card Punch and the requirements of a Flexowriter By-Product Tape to be read by it. This information will not only be helpful when programming the Flexowriter in an application involving the Tape-to-Card Punch, but should also give you an idea of how to proceed in programming the Flexowriter for various other pieces of Allied Equipment.

This is by no means to be considered a full and complete treatment of the Tape-to-Card Punch. Such coverage would require a complete manual itself. In fact, anyone desiring this knowledge can obtain a complete Manual of Operation published by International Business Machines Corporation. Our sole purpose here is to give the reader the requirements of a Flexowriter By-Product Tape which will produce Tabulating Cards.



### The IBM Tabulating Card

The IBM Tabulating Card has a capacity of eighty codes. These codes are represented by combinations of punched holes in the Card in much the same way holes perforated in punched paper tape represent various codes. Of course, the coding system employed in Punched Cards is quite different. From the illustration on Page 16, you can see that the IBM Card is divided into eighty vertical columns. These columns are numbered from one to eighty and each can contain one code.

The IBM Card is also divided into vertical divisions known as Fields. A Field is a column or group of columns set aside to contain some specific data. Illustrated is a portion of the Purchase Order form we have discussed earlier and a Tabulating Card which was produced as a By-Product of typing this portion of the Purchase Order.

THE AJAX COMPANY . . . .

Purchase Order

To: Order No.  
63205 Order Date  
05/14/58  
SMITH SPECIALTY CO.  
100 ELM STREET  
ANYTOWN, U.S.A.  
Deliver By:  
06/14/58

|              |   |   |   |   |   |   |   |   |    |    |    |    |                    |    |    |    |    |    |    |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
|--------------|---|---|---|---|---|---|---|---|----|----|----|----|--------------------|----|----|----|----|----|----|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 163205051458 |   |   |   |   |   |   |   |   |    |    |    |    | SMITH SPECIALTY CO |    |    |    |    |    |    |    |    |    |    |    |    | 061458 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 1            | 2 | 3 |   |   |   |   |   |   |    |    |    |    | 4                  |    |    |    |    |    |    |    |    |    |    | 5  |    |        |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 0            | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0                  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0      | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 1            | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1  | 1  | 1  | 1  | 1                  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1      | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  | 1  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 2            | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2  | 2  | 2  | 2  | 2                  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2      | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  | 2  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 3            | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  | 3  | 3  | 3  | 3                  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3      | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 4            | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4  | 4  | 4  | 4  | 4                  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4      | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  | 4  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 5            | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5  | 5  | 5  | 5  | 5                  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5      | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  | 5  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 6            | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6  | 6  | 6  | 6  | 6                  | 6  | 6  | 6  | 6  | 6  | 6  | 6  | 6  | 6  | 6  | 6  | 6  | 6      | 6  | 6  | 6  | 6  | 6  | 6  | 6  | 6  | 6  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 7            | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7 | 7  | 7  | 7  | 7  | 7                  | 7  | 7  | 7  | 7  | 7  | 7  | 7  | 7  | 7  | 7  | 7  | 7  | 7      | 7  | 7  | 7  | 7  | 7  | 7  | 7  | 7  | 7  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 8            | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8 | 8  | 8  | 8  | 8  | 8                  | 8  | 8  | 8  | 8  | 8  | 8  | 8  | 8  | 8  | 8  | 8  | 8  | 8      | 8  | 8  | 8  | 8  | 8  | 8  | 8  | 8  | 8  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 9            | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9 | 9  | 9  | 9  | 9  | 9                  | 9  | 9  | 9  | 9  | 9  | 9  | 9  | 9  | 9  | 9  | 9  | 9  | 9      | 9  | 9  | 9  | 9  | 9  | 9  | 9  | 9  | 9  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |
| 1            | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14                 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27     | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 |

As you can see, this Card contains five different Fields. The first of these is confined to Column 1 alone. It is the Card Number field and is used in the Punched Card Accounting System to distinguish that type of Tabulating Card from others which might be used with it.

Columns 2 - 6 make up the Order No. field. The Purchase Order No. on the form will always be punched in this field.

The Date is always punched in the Date field made up of Columns 7 - 12.

Columns 13 - 66 make up the Vendor's field. If the Vendor's Name does not take up the complete field, as it rarely will, it will be necessary to Skip the remainder of the Name field before punching the Deliver By date.

Columns 67 - 72 make up the Deliver By field.

The value of dividing the Card into Fields in this manner is because various Punched Card Accounting Machines sense predetermined Columns in the Card in order to read a particular bit of information. We can be sure, for example, when sensing Column 12, that only the year of the Date is being sensed.

Notice the Date is punched as 051458. One might expect the first "zero" to be omitted and the Date to be punched as 51458. The "zero" is employed to fill the field completely. If the month were October, instead of May, the Date would be punched as 101458. Therefore, it is essential to extend the Date field to six Columns to allow for the maximum Date length.

Compare this example with the Name field. There, when the Name did not fill the complete field, the remainder of the field was Skipped. This practice is usually restricted to Name fields alone.

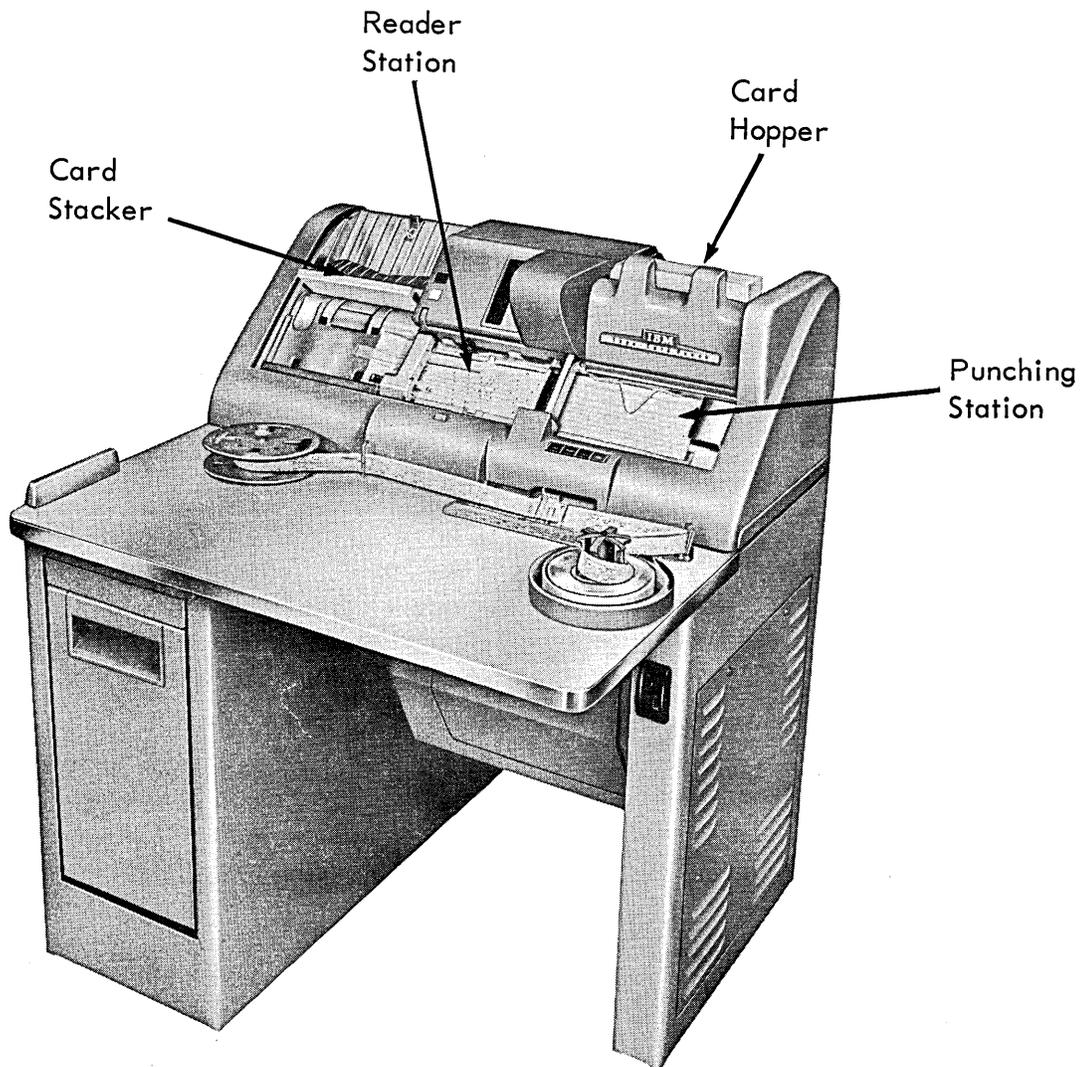
In most numeric fields, like the above Date field, it is essential to punch "spaces" or "zeros" to the left of the first significant digit to fill the field. So, a "zero" must be added in Column 7 before the first digit of the Date.

### The Components of the Tape-to-Card Punch

There are two basic Tape-to-Card Punches manufactured by IBM. These are the Type 046 and Type 047. The only difference between the two machines is this. Type 047 (illustrated on Page 18) punches the Card and, at the same time, prints along the top of it. Type 046 only punches the Card without printing on it.

On the Tape-to-Card Punch there are four different Stations through which the Tabulating Cards must pass as they are punched.

1. THE CARD HOPPER. This unit stores the Cards before they are punched. It has a capacity of approximately 500 Cards.
2. THE PUNCHING STATION. Here the Cards are actually punched. Alphabetic, numerals and special symbols can be punched into the Card at this point. Here, too, the Card may be spaced in the same manner in which the Space Bar of a typewriter can space a document.



IBM TAPE - TO - CARD PUNCH

Type 047





## The Functions of the Tape-to-Card Punch

1. TAPE READING. This is the most common function of the machine. It consists of Reading Codes in the Tape and Punching them into Tabulating Card. These codes must be read in the same sequence as they are to be punched into the Tabulating Card. For this reason, it is essential that the form, being written by the Flexowriter, follow the same sequence of information as the Tabulating Card.

If an Order No. is to be punched into the Tabulating Card before a Date is punched, it is necessary to design the form so that the Order No. is written before the Date. Not the other way around.

The Control Panel has a great influence over the Tape Reading and subsequent Card Punching of the Tape-to-Card Punch. Actually the Control Panel wiring determines what code will be punched into the Card when a given code is read in the tape.

This means, for example, that the punched tape code for a "space" may be wired to cause the punching of a "zero" in the Tabulating Card; or, depending on how the Control Panel has been wired, it may cause "spacing" to occur.

The only place where the Control Panel wiring does not regulate Card punching is in the case of alphabetic characters. This means that if the code for the letter "A" is read in tape, it will punch an "A" into the Card, irrespective of the Control Panel wiring.

The Control Panel wiring may also be SELECTIVE. This means that a tape code can be wired to perform one thing in one portion of the Card, and something different elsewhere in the Card. A common example of this is to wire the Control Panel so that the tape code for a "space" causes "spacing" of the punched Card in all alphabetic field, and the punching of a "zero" in numeric fields.

It is important to note that the Control Panel can be wired to ignore certain codes in the tape being read. It is common practice to type the Date on a form with slashes separating the day from the month and year (05/14/58). In the Tabulating Card, these slashes are not to be punched. The Date is to be punched as 051458. It is possible to leave unwired in the Control Panel, the code for the slash, or any other code.

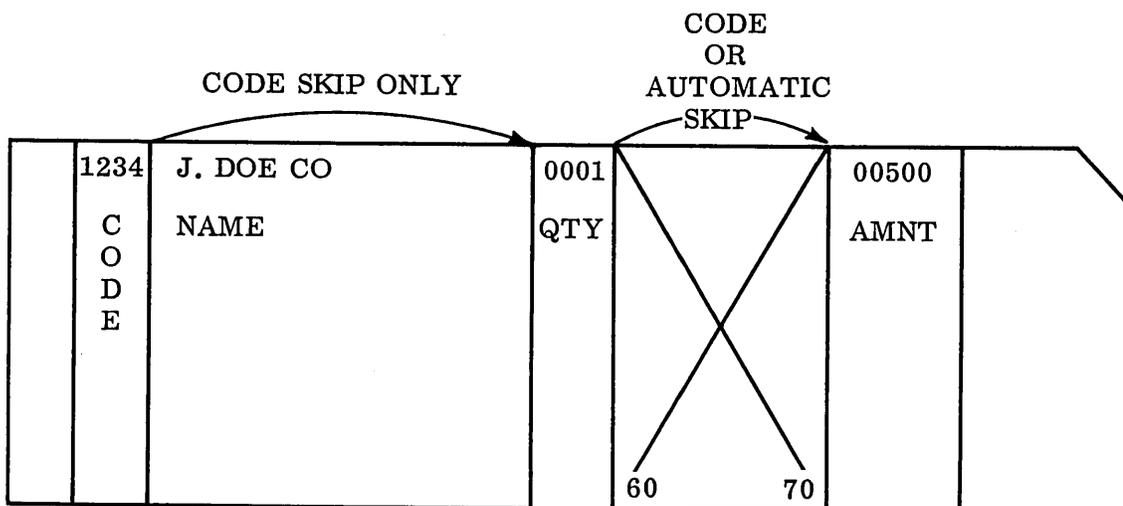
2. SKIPPING. The Skip function of the Tape-to-Card Punch is quite similar to the Tab function of a Flexowriter. A skip means to pass over rapidly, or Skip, one or more Columns of a Tabulating Card. In a Name field, for example, if the Name does not occupy the entire field, it is necessary to Skip the remainder of the field so that the next Data in the tape will be punched into its designated field.

There are two ways to cause a Skip to occur on the Tape-to-Card Punch. One method is to put a code in the tape that is being read. This causes a Skip to occur when it is sensed. It is common practice to use the Tab Code on the Flexowriter to cause Skip operation. It is through Control Panel wiring that we determine which code in the tape will cause Card Skipping.

An alternate method of causing a Skip to occur is through Control Panel wiring.

The Control Panel of the Tape-to-Card Punch can be so wired to allow a Skip to occur automatically, whenever the Card reaches a specific Column in the Punching Station. This automatic type of Skip can only be employed when the starting point of the Skip is predetermined. It could never be employed in a Name field because we do not know beforehand, in which Column the Skip is to start. It will be based, in this case, on the length of the Name punched in the Card. A code, usually a Tab Code, must be used to start the Skip.

If, on the other hand, we always want to Skip Columns 60 - 70, and we have a predetermined starting point (Column 60), we can initiate the Skip from the Control Panel. In either event, it will be the Program Drum that will terminate the Skip operation.



3. CARD RELEASE. This ejects the Card after the last code has been punched. We can cause a Release to occur in one of two ways.

A code can be put into the tape which, when read, will cause the Card to be Released. Or, the Control Panel can be wired to cause the Card to be Released whenever a certain Column is reached.

The use of the Control Panel method is discouraged. The preferred method is use of the code. Although any control code can be wired to cause Release, the Carriage Return Code is most often employed.

The reason a code is preferred to Control Panel wiring is because a code offers a very valuable safety check at the end of the Card.

Let us assume a particular Card is to be punched up to and including Column 50. This means the Carriage Return Code, which is to cause Release, is read when Column 51 is at the Punching Station.

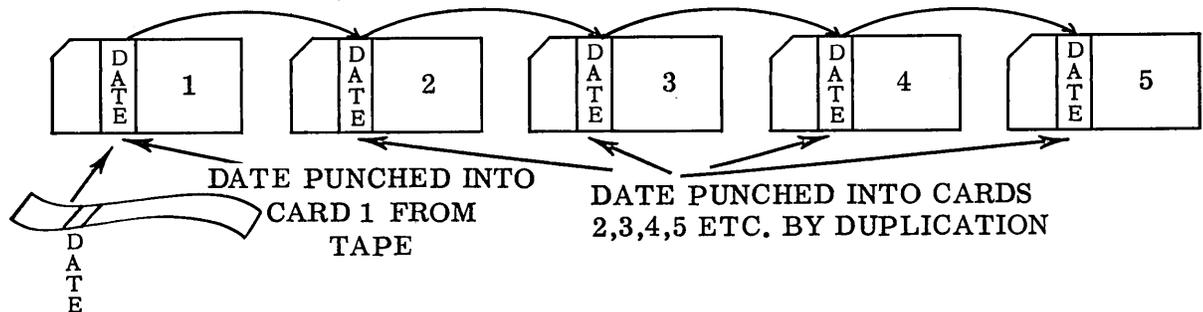
If the Carriage Return Code is read at any other Column, we can be sure there are either too many, or too few, codes in the Card. It is possible, through Control Panel wiring, to make the Tape-to-Card Punch stop if the code causing Card Release is read in the wrong Column. This safety check can only be employed when a code is used to cause Card Release.

4. **DUPLICATION.** This is one of the most important functions of a Tape-to-Card Punch. It consists of reading repetitive information from the Card in the Reading Station and punching it into the Card at the Punch Station.

For example, assume the Date is to be punched in the first six Columns of each Tabulating Card. This Date need be in the tape, only once. When the very first Card is at the Punch Station, the Date will be read from the tape and punched into the Card. The same Date will punch into all subsequent Cards through Duplication.

When the first Card is completely punched, it advances from the Punch Station to the Reader Station. At the same time, a new Card advances to the Punch Station.

When this new Card advances, the Read Station reads the Date from the Card passing through it, causing it to be punched into the Card in the Punch Station. When this new Card has been completely punched, it, too, passes to the Reader Station and the Date is Duplicated in the next Card, and so on.



There is one important rule to remember in Duplication. Information to be Duplicated must be punched in the same Columns in which it is found at the Reader Station. Therefore, if the Date from the tape is punched in Columns 1 - 6 in the first Card, it can be Duplicated in subsequent Cards in Columns 1 - 6, only.

Like Skip and Release, Duplication can be caused in either of two ways. A code in the tape can cause Duplication to start when the Card at the mpunch Station reaches a specific Column.

Unlike Skip and Release, there is no special code to cause Duplication to occur. Any Flexwriter functional code can be used. Later, we will show how certain codes work well for initiating many Duplicate operations.

The length of the Duplicate operation is determined by the punching of the Program Drum Card. Like the Skip function, the Duplicate function can be initiated from a code in the tape or from wiring the Control Panel. Its length will always be governed by the punching of the Card wrapped around the Program Drum. If we desire to Duplicate the first ten Columns of a Tabulating Card, the Program Drum Card must be punched to allow Duplication to occur only through Column ten.

In addition to the functions mentioned above, there are several other functions employed when the Tape-to-Card Punch is reading tapes produced on the Flexo-

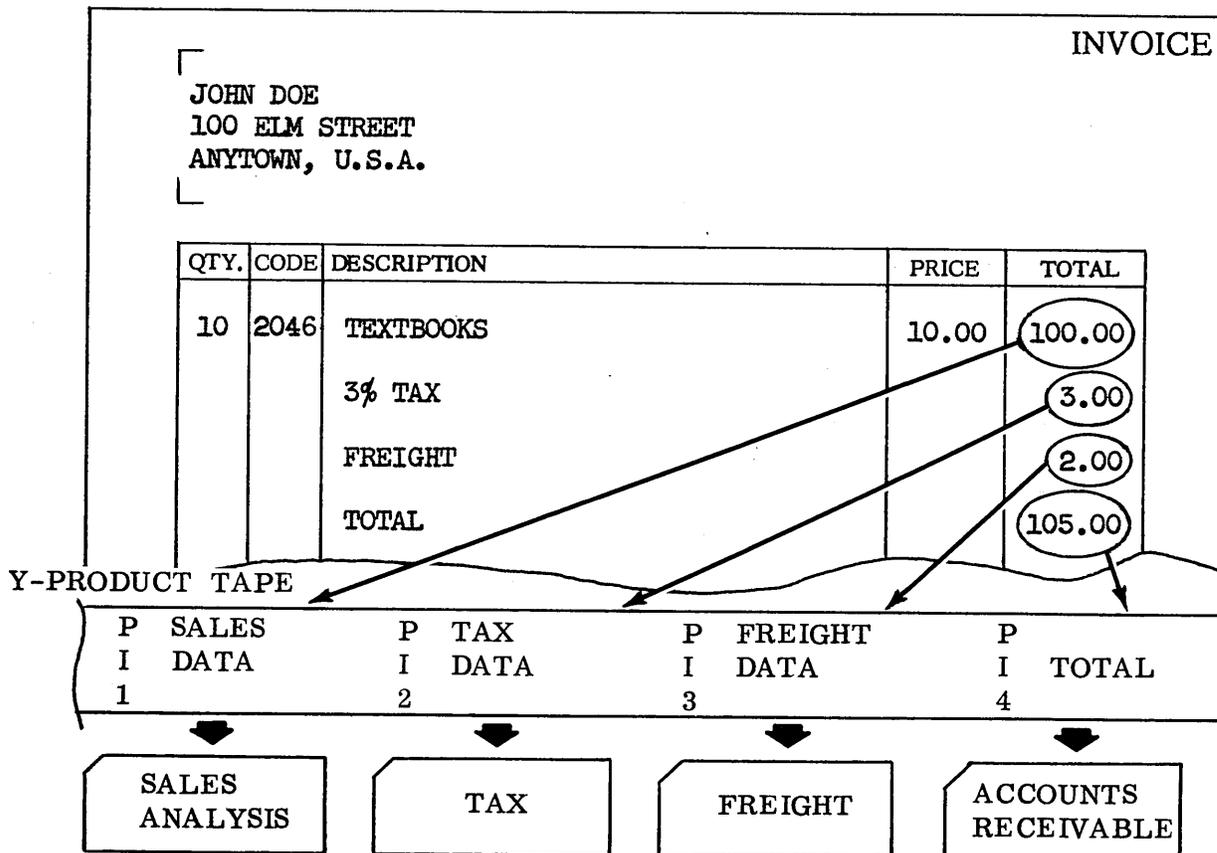
writer. When other tape punching equipment, such as the Friden Add-Punch, produces tapes for the Tape-to-Card Punch, these other features are employed more often. It is not our intention here to discuss in detail the other features of the Tape-to-Card Punch. A detailed description of all the operating features of the machine may be found in the IBM Manual of Operation previously mentioned.

### The Requirments of the Flexowriter By-Product Tape

The tape produced on the Flexowriter, as a By-Product of typing a document, contains all the Data that is to be punched into the Tabulating Card. Through the use of ON 1 and OFF codes and the Punch SELECT position of the Flexowriter, this tape should contain only the Data to be punched into the Tabulating Card, and no more. If certain information on the form, such as Description, is not to be punched into the Tabulating Card, an OFF code will prevent this Data from punching into the By-Product Tape for the Tape-to-Card Punch. This By-Product Tape is, therefore, properly called a SELECT TAPE.

### The Program Indication Code

Very often we find it necessary to produce more than one kind of Tabulating Card from the typing of a single document on the Flexowriter. When typing an Invoice, for example, we might be required to produce a Tax Card from the Taxes shown on the Invoice, a Freight Card from any Freight charges, and an Accounts Receivable Card from the Total Amount of the Invoice.



## THE USE OF THE PROGRAM INDICATION CODES

The requirements as far as Skip, Duplicate, Release, etc., for each of these Cards may be quite different. For this reason, we must have some way of telling the Tape-to-Card Punch what kind of Card is next to be read from the tape. Through Control Panel wiring, the correct Card will be punched at the correct time.

We use a code, known as a Program Indication Code, to tell the Tape-to-Card Punch what kind of Card is to be punched at a given time. There are seven different Program Indication Codes (more than enough to handle any one Flexowriter application). These Program Indication Codes must be included in the Flexowriter Program Tape and made to reproduce into the By-Product Tape.

The Program Indication Codes are put into the Flexowriter Master Program Tapes through a process known as Overpunching. Program Indication Code 1, known henceforth as PI 1, is produced in the following manner.

The operator touches the numeral 2 and its code is punched from the keyboard into the tape. The tape is rolled backward in the Punch one notch. Then the numeral 8 is touched combining the 2 and 8 in one code. The operator again rolls the tape back one notch and touches the Space Bar. This combines codes 2, 8, and Space which make up the PI 1 code.

Following is a list of the seven Program Indication Codes and the combinations of Overpunched Codes that will produce them

|      |   |   |               |
|------|---|---|---------------|
| PI 1 | 2 | 8 | SPACE         |
| PI 2 | 2 | 8 | HYPHEN (DASH) |
| PI 3 | 2 | 8 | ZERO          |
| PI 4 | 5 | Z |               |
| PI 5 | N | Z |               |
| PI 6 | 9 | N |               |
| PI 7 | 1 | 4 | 8             |

Although it may seem awkward to produce these Program Indication Codes at first, it must be remembered that these codes have only to be produced once when making the Master Program Tape. From then on, the codes will automatically be reproduced into all of the Data Tapes and the SELECT By-Product Tapes.

Since there are no keybuttons on the Flexowriter representing them, the PI Codes are Non-Reproducing Codes. This means WE MUST EMPLOY THE NON-PRINT AND PRINT RESTORE CODES TO MAKE THEM REPRODUCE FROM ONE TAPE TO THE NEXT.

The Program Indication Code should be punched into the By-Product Tape for the Tape-to-Card Punch at the beginning of the section of tape that is to produce an individual Tabulating Card. For each different type of Tabulating Card, there should be a different Program Indication Code.

The chief function of the Program Indication Code is to tell the Tape-to-Card Punch what kind of Card is coming next. In addition, the same Program Indication Code can be used simultaneously to cause other things to occur.

1. It can cause a number to be punched in Column 1. This number can be

used to distinguish the various types of Cards in a tabulating system.

In the illustration on Page 24, for example, a numeral "one" in Column 1 might identify a Sales Analysis Card "two" might identify a Tax Card "three" a Freight Card and "four" an Accounts Receivable Card. PI 1, 2, 3, and 4 Codes, respectively, would cause the punching of these identifying digits.

Through Control Panel wiring, the Program Indication Code can be made to punch any desired digit into the Card. A PI 1 Code could cause a "one" or a "six", for example, to be punched into Column 1 of a Tabulating Card, depending on the Control Panel wiring.

2. The Program Indication Code can also be used to cause a Skip or a Duplication to occur at Column 1, if the Control Panel is so wired. It is common to use them to cause Duplication, since most Duplication does occur in Column 1 of a Tabulating Card.

#### Control Codes in the By-Product Tape

We have shown that the Skip, Release and Duplicate functions of the Tape-to-Card Punch can be started either by inserting a code into the By-Product Tape, or by wiring the Control Panel to start the function when the proper Column of the Tabulating Card reaches the Punching Station. The following is a suggested rule for determining which method to employ at any time.

IT IS SUGGESTED THAT, WHENEVER POSSIBLE, A CODE BE USED TO START ALL FUNCTIONS OF THE TAPE-TO-CARD PUNCH, EXCEPT A SKIP FUNCTION WHICH ALWAYS BEGINS IN A PREDETERMINED COLUMN OF THE CARD. IN THE CASE OF SUCH A SKIP, THE CONTROL PANEL MIGHT BETTER BE EMPLOYED.

#### A Sample Problem

Illustrated ( on Page 28 ) is a sample Purchase Order form with the Flexowriter By-Product Tape punched as a result of writing the form. The two Tabulating Cards produced when this tape is read by the Tape-to-Card Punch are also shown.

Notice the Order No. and the Date are punched in Columns 2 - 12 of both Cards. This is a typical example of a situation where the Duplicate feature of the Tape-to-Card Punch should be employed. The Order No. and Date need be punched into the By-Product Tape only once, causing it to punch into the first Card. It will be punched into the second Card by Duplication from the first Card.

Following is a point-by-point breakdown of the programming.

- A. Start the tape with a PI 1 Code. We should always put a Program Indication Code at the start of a section of tape that is to produce a Tabulating Card. The PI 1 Code has two functions in this case.

1. It tells the Tape-to-Card Punch a new Card is coming and what type it is. In this case it is the first Card on a document.
2. It punches the numeral "one" in Column 1. This will serve

to identify this type of Card and distinguish it from the next Card.

- B. Immediately following the PI 1 Code we find the Order No. (63205). This is punched directly into Columns 2 - 6 of the Card.
- C. The Date (05/14/58) is punched into the tape next. Notice there are slashes between the day, month, and year. They do not affect the Tabulating Card because the slash is not wired in the Control Panel to punch.
- D. The Name (SMITH SPECIALTY CO.) is punched into the tape next. Since it does not occupy the entire field, it will be necessary to Skip the remainder of the field.

The starting point of the Skip cannot be pre-determined, so we must use a code to cause the Skip to occur. The Tab Code is used in this case. The Program Drum Card will regulate the length of this Skip and stop it in Column 67.

- E. The last entry in the tape (taken from the Heading of the document) is Deliver By (06/14/58). It is punched in the Card the same way the Order Date was punched in Columns 7 - 12.
- F. To Release the Card in Column 45, we read the Carriage Return Code. This will cause the Card to be ejected and the next Card to advance into the Punch Station.
- G. The next code in the tape is a PI 2 Code. In this case the PI 2 Code has three functions.

1. It indicates that a new Card is to be punched. The new Card will be different from the first.
2. It causes the numeral "two" to be punched in Column 1. This is to identify the Card.
3. It causes Duplication of the Order No. and the Order Date.

The functions of the various Program Indication Codes will be determined through Control Panel wiring. The Program Drum Card will regulate the length of the Duplication and cause it to cease in Column 13.

- H. The Quantity Ordered (15) is punched into the tape next. Notice the Card field allows a maximum of five digits. However, in this case there are only two digits.

Since this is a numeric field, we cannot employ the Skip feature to get through the remainder of the field (as we did in the Name field). We must use "spaces" to the left of the actual Quantity Ordered. The Space Codes, it should be noticed, are punching "zeros" in the Quantity Ordered field of the Card. (Space Codes in the Name field caused "spacing".)



E

F

C  
R

0 6 / 1 4 / 5 8

0 6 | 1 4 | 5 8

41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

L

M

S S S  
P P P

C  
R

9 0 0

S T I C K S  
T  
A  
B

S T I C K S

0 0 0 9 | 0 0

P R I C E

41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80



## Programming the Complete Application

We have now completed our discussion of the Flexowriter Programatic and the methods by which it is programmed. We have also seen the principles employed in programming a typical piece of Allied Equipment, the Tape-to-Card Punch. We are now ready to put the pieces together and make up the complete program of an application.

This will be our sample application. From a Vendor Data Tape and an Item Data Tape, we will automatically prepare a Purchase Order on the Model SPS Flexowriter. As we type the Purchase Order the Punch Switch will be in SELECT position. Thus a SELECT By-Product Tape, for the Tape-to-Card Punch, will be punched. The Purchase Order form will be the same as that used to illustrate the operation of the Tape-to-Card Punch shown on Page 18.

Before we can begin to program any Flexowriter application, we must have certain things at our disposal.

1. We must have a copy of the Form we are to prepare. This should not be a blank Form, but one fully typed.
2. We must have samples of any Tabulating Cards with which we will be working.

### The Form

Illustrated, on (Page 30), is the Form we are to use. Before we can start to program it, we must establish a few prerequisites.

First, we determine the Margin and Tab Stop Settings. You will note, these are clearly marked on the Form.

The Margin is set at the first writing Position, where the numeral "6" is written in the Order No. field.

Next we determine our Tab Stop Settings. We should try, if possible, to tabulate on a Flexowriter whenever we have to advance two spaces, or more, horizontally on the Form. It should be pointed out that the minimum distance a Flexowriter can tabulate is two spaces. If you attempt to tabulate only one space, the Carriage of the Flexowriter will pass over that Tab Stop and go on to the next Tab Stop.

On this sample Form we have two Tab Stop settings. The first will take care of the Tab from the Order No. field to the Order Date field. The same Tab Stop will allow tabulation (in the Body of the Form) from the Code field to the Description field. Proper Form Design should employ this principle of lining up the various fields on the Form so that one Tab Stop can handle several different tabulations.

There is a second Tab Stop setting (on the right side of the Form) to terminate the tabulation from the Description field to the Price field.

Next, we must determine what fields on the Form can be typed automatically from the Data Tapes, and where we will need Stop Codes to allow the manual entry of variable information.

On the Heading of the Form, the Order No., Order Date and Deliver By fields cannot be included in the Vendor Data Tape. These will change each time a Purchase Order is written to that particular Vendor. On the other hand, the Vendor's Name and Address is repetitive and can be included in the Vendor Data Tape.

In the Body of the Form the only field that cannot be included in the Item Data Tape is the Quantity Ordered. This will differ as each Purchase Order is written. Code, Description and Price are constant for each Item and can be included in the Item Data Tape.

We should next compare the Form with the Tabulating Cards to be produced, making certain they are compatible. The following fields on the Form should punch into the By-Product Tape for the Tape-to-Card Punch: Order No., Order Date, Quantity Ordered, Code, Description and Price. In this case, the fields on the Form maintain the same sequence as the fields on the Tabulating Cards. If they did not, we would have to re-design either the Form or the Tabulating Card to achieve the necessary compatibility.

#### The Various Tapes to be Employed

In this application, we will be working with three different kinds of tapes.

First we have the Master Program Tapes. These will be used by the operator as a guide in preparing the Data Tapes. We will need two Master Program Tapes, one will be used as a guide in preparing the Vendor Data Tape; the other in preparing the Item Data Tape.

The second kind of tapes with which we will be working are the Data Tapes. These will be used to actually type the Purchase Order. We will need a separate Vendor Data Tape for every Vendor to whom Purchase Orders are written. Also, a separate Item Data Tape for every regularly purchased Item.

Finally, we will be working with the SELECT By-Product Tape, which will be used in the Tape-to-Card Punch. This will be referred to as the Select Tape.

Our task in programming will be to determine exactly which codes will be in each of the above tapes, and to put these codes down on paper. One would normally expect to first, figure out the Master Program Tapes, then the Data Tapes, and finally the Select Tape. However, this is not the case.

There are great advantages to programming a Flexowriter application in the reverse order, or backwards. This means we must first determine the codes in the Select Tape, then the Data Tapes, and finally the Master Program Tapes. If we tried to program from Master Tapes to Data Tapes to Select Tape, we would often find we had failed to allow for certain functional codes needed in the Tape-to-Card Punch in the Select Tape programming. If we proceed from the Select Tape to Master Program Tapes, such a difficulty will not arise.

## The Use of the Layout Chart

Illustrated, on ( Page 36 ), is a Punched Tape Layout Chart. This Layout Chart is extremely helpful in programming a Flexowriter application. It gives a graphic picture of the complete program, and is a fine tool to give to the person who is responsible for the Control Panel wiring of the Tape-to-Card Punch. Notice we have labeled the various tapes with which we are to be working.

Our first step in setting up the program is to draw on the Layout Chart, in pencil, the Tabulating Cards to be employed.

1. The fields should be drawn in and then labeled.
2. The Data on the sample Form should be inserted into the correct fields of the Tabulating Card. The Layout Chart, with the two Tabulating Cards to be produced in this application, is illustrated with this sample Data inserted.

## The Select Tape

Our next step is to determine the requirements of the Select Tape, and print the needed codes on the Layout Chart. We have, in a previous section for the Tape-to-Card Punch, determined the Select Tape requirements for the Tabulating Cards used in this application.

If, at this point, the programming is not completely clear, it might be well to review the point-by-point breakdown of this Select Tape coding on Page 26.

We have illustrated, on ( Page 36 ), the Layout Chart with this Select Tape coding inserted in the Proper place. It is very helpful in preparing the Layout Chart to have the codes in the Select Tape fall exactly above their corresponding codes in the Tabulating Cards.

## The Programming of the Data Tapes

The heart of programming a Flexowriter application comes when we determine the codes needed in the Data Tapes. The steps involved are quite simple. It is necessary, however, that the programmer learn to proceed slowly and logically.

In programming the Vendor and Item Data Tapes, we proceed one field on the form at a time. By considering each field on the form as a separate block in our program, we arrive at a set of four simple questions which will enable us to program any application:

1. HOW DO I GET TO THE NEXT FIELD ON THE FORM?
2. DO THE CODES THAT GET ME THERE PUNCH INTO THE SELECT TAPE?
3. DOES THE FIELD PUNCH IN THE SELECT TAPE?
4. IS THE ENTRY A CONSTANT OR A VARIABLE?

Now let us analyze each question, in detail.

## HOW DO I GET TO THE NEXT FIELD ON THE FORM?

We saw earlier that there are only three codes to Space the Carriage (Space, Tab and Carriage Return), and they are always incorporated into Program Tapes.

In asking ourselves this first question, we are merely asking which of these three codes, or combinations of them, is needed in our Data Tape to Space the Carriage to the next Field on the Form. A forms ruler, which is calibrated in tenths of an inch for Spaces and sixths of an inch for Carriage Returns, is very helpful at this point.

|   |
|---|
| T |
| A |
| B |

If we find, for example, that a Tab is needed, we mark the Tab on our Layout Chart.

## DO THE CODES THAT GET ME THERE PUNCH INTO THE SELECT TAPE?

This merely means, does the Tab, Space, or Carriage Return Code, that Space the Carriage to the next Field, actually punch into the Select Tape.

It is necessary to look at the Select Tape on the Layout Chart to determine this. If the Code IS to punch, we precede it with an ON 1 Code to turn the Punch ON. If the Code IS NOT to punch, we precede it with an OFF Code to turn the Punch OFF.

|   |   |
|---|---|
| O | T |
| F | A |
| F | B |

Let us assume that the Tab Code, which Spaced the Carriage to the next Field, IS NOT to punch into the Select Tape. We then will put an OFF Code before the TAB Code to prevent its punching.

## DOES THE FIELD PUNCH IN THE SELECT TAPE?

We have previously noted all of the Fields in our Form which are to punch into the Select Tape, and subsequently into the Tabulating Cards.

If the information in the Field IS to punch into the Select Tape, we turn the Punch ON with the ON 1 Code, after the code that Spaces the Carriage. If it IS NOT to punch into the Select Tape, we use an OFF Code to turn the Punch OFF.

|   |   |   |
|---|---|---|
| O | T | O |
| F | A | N |
| F | B | 1 |

In our example, let us say that the Field on the Form IS to be punched into the Select Tape. We would then add an ON 1 Code after the TAB Code to turn the Punch back ON.

## IS THE ENTRY A CONSTANT OR A VARIABLE?

This question asks whether the Entry can be included in the Data Tape; or, if a STOP Code is needed to stop the Tape Reader to allow the typist to fill-in the Variable Data required.

|   |   |   |   |
|---|---|---|---|
| O | T | O | S |
| F | A | N | T |
| F | B | 1 | P |

If we assume the Entry in our example is a Variable, we would add a STOP Code to our program to enable the Reader to stop for a fill-in of that Variable Data.

In addition to the above four points, we must also keep our eyes open for any Control Codes required for operation of the Tape-to-Card Punch. These include Program Indication Codes and codes to cause Skip, Duplicate and Release operations to occur. These codes can easily be determined by observing closely

the Select Tape which has been previously outlined on the Punch Tape Layout Chart.

More often than not, when these Control Codes are included in the program for the Data Tapes, they will have to be preceded by a Non-Print Code and followed by a Print Restore Code. This is necessary if the particular Control Code is not represented on the Flexowriter keyboard. CODES NOT REPRESENTED ON THE KEYBOARD WILL NOT REPRODUCE INTO THE SELECT TAPE UNLESS READ IN NON-PRINT CONDITION. (Example: Program Indication Codes.)

The Automatic Non-Print feature must also be employed if the Control Code for the Tape-to-Card Punch is a keyboard code but does not fit into the Flexowriter program at the time it is needed in the Tape-to-Card program. An example of this would be a Tab Code needed in the Select Tape to cause a Skip to occur in the Tabulating Card.

If the Field on the Tabulating Card coincides with a tabulation on the Form being prepared, a Non-Print Code is not needed before the Tab Code. If, on the other hand, a tabulation on the Form is not desirable at that time, it will be necessary to suppress tabulation by preceding the Tab Code with a Non-Print Code and following it with a Print Restore Code.

On the pages that follow, you will find a point-by-point breakdown of the programming of the Data Tapes for the Purchase Order application we are using for our example. Notice that each Field on the Form is considered separately and the four questions mentioned on (Page 33) will be applied in every case.

On (Page 40) you will find the complete Layout Chart with the Data Tapes included. Note that the programming on the Layout Chart and the corresponding Field on the Form (above the Layout Chart) have been marked with alphabetical characters A through J. We will discuss each requirement in sequence.

|    |                     |   |   |   |   |   |   |   |   |
|----|---------------------|---|---|---|---|---|---|---|---|
| A. | <u>ORDER NUMBER</u> | O | C | C | O | N | P | P | S |
|    |                     | F | R | R | N | P | I | R | T |
|    |                     | F |   | 1 |   | 1 |   |   | P |

Question 1: How do I get to the next field on the form?  
Answer: Carriage Return twice. (CR, CR)

This is the first Field on the Form. It is good practice to start all Flexowriter programs with one or more Carriage Return Codes. In this case, two Carriage Return Codes are suggested. This will enable the operator to position the Writing Line Finder of the Flexowriter on the horizontal line of the Form, just above the words "Order No.", each time a new document is written.

Question 2: Do the codes that get me there punch into the Select Tape?  
Answer: No. Precede the Carriage Return Codes with an OFF Code. (OFF, CR, CR)

Question 3: Does the field punch into the Select Tape?  
Answer: Yes. Add an ON 1 Code. (OFF, CR, CR, ON 1)

At this time we must provide programming for the Program Indication



C  
R

0 6 / 1 4 / 5 8

0 6 | 1 4 | 5 8

DELIVER BY

41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

T  
A  
B

S S S  
P P P

C  
R

S T I C K S

9 0 0

S T I C K S

0 0 0 9 | 0 0

P R I C E

41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

Code appearing immediately before the Order No. in the Select Tape.

Since Program Indication Codes are Non-Reproducing Codes, we must follow the method previously discussed in order to reproduce them in the Select Tape. Precede the Program Indication Code with a Non-Print Code and follow it with a Print Restore Code. (OFF, CR, CR, ON 1, NP, PI 1, PR)

This will cause a Print Restore Code to punch into the Select Tape after the PI 1 Code. The Print Restore Code will be ignored by the Tape-to-Card Punch.

Question 4: Is the entry a constant or a variable?

Answer: Variable. Add a Stop Code. (OFF, CR, CR, ON 1, NP, PI 1, PR, STP)

B. 

|                   |         |
|-------------------|---------|
| <u>ORDER DATE</u> | O T O S |
|                   | F A N T |
|                   | F B 1 P |

Question 1: How do I get to the next field on the form?

Answer: Tab. Add a Tab Code. (TAB)

Question 2: Do the codes that get me there punch into the Select Tape?

Answer: No. Precede the Tab Code with an OFF Code. (OFF, TAB)

Question 3: Does the field punch into the Select Tape?

Answer: Yes. Add an ON 1 Code. (OFF, TAB, ON 1)

Question 4: Is the entry a constant or a variable?

Answer: Variable. Add a Stop Code. (OFF, TAB, ON 1, STP)

C. 

|             |                         |
|-------------|-------------------------|
| <u>NAME</u> | O C C O                 |
|             | F R R N                 |
|             | F 1 SMITH SPECIALTY CO. |

Question 1: How do I get to the next field on the form?

Answer: Carriage Return twice. Add two Carriage Return Codes. (CR, CR)

Question 2: Do the codes that get me there punch into the Select Tape?

Answer: No. Precede the Carriage Return Codes with an OFF Code. (OFF, CR, CR)

Question 3: Does the field punch into the Select Tape?

Answer: Yes. Add an ON 1 Code. (OFF, CR, CR, ON 1)

Question 4: Is the entry a constant or a variable?

Answer: Constant. Add SMITH SPECIALTY CO. (OFF, CR, CR, ON 1, SMITH SPECIALTY CO.)

Notice, a Tab Code appears in the Select Tape at this time to cause a Skip to occur on the Tape-to-Card Punch. We do not, however, desire a tabulation to occur on the Form at this time.

THE AJAX COMPANY . . . .

Purchase Order

A

Order No.  
63205

B

Order Date  
05/14/58

To: C SMITH SPECIALTY CO.  
D 100 ELM STREET  
E ANYTOWN, U.S.A.

Deliver By:  
06/14/58

F

| Quantity Ordered | Code      | Description              | Price    |  |
|------------------|-----------|--------------------------|----------|--|
| G<br>15          | H<br>3208 | I<br>BRASS CANDLE STICKS | J<br>900 |  |





N T P  
P A R  
B

For this reason, we should now add a Tab Code to our Program Tape and precede it with a Non-Print Code and follow it with a Print Restore Code. This will cause a Print Restore Code to be punched into the Select Tape. It will, of course, be ignored by the Tape-to-Card Punch.

D. STREET ADDRESS            O C  
   F R  
   F    100 ELM STREET

Question 1: How do I get to the next field on the form?  
Answer: Carriage Return. Add a Carriage Return Code. (CR)

Question 2: Do the codes that get me there punch into the Select Tape?  
Answer: No. Precede the Carriage Return Code with an OFF Code. (OFF, CR)

Question 3: Does the field punch into the Select Tape?  
Answer: No. Since the Punch is already OFF (having preceded the Carriage Return Code with an OFF Code...see Question 2), there is no need to add another OFF Code following the Carriage Return Code.

Question 4: Is the entry a constant or a variable?  
Answer: Constant. Add 100 ELM STREET. (OFF, CR, 100 ELM STREET)

E. CITY                            C  
   R ANYTOWN, U. S. A.

Question 1: How do I get to the next field on the form?  
Answer: Carriage Return. Add a Carriage Return Code. (CR)

Question 2: Do the codes that get me there punch into the Select Tape?  
Answer: No. Since the Punch has previously been turned OFF, there is no need to precede this Carriage Return Code with an OFF Code.

Question 3: Does the field punch into the Select Tape?  
Answer: No. The Punch remains OFF.

Question 4: Is the entry a constant or a variable?  
Answer: Constant. Add ANYTOWN, U. S. A. (CR, ANYTOWN, U. S. A.)

F. DELIVER BY                    C C C O S  
   R R R N T  
   1 P

Question 1: How do I get to the next field on the form?  
Answer: Carriage Return three times. Add three Carriage Return Codes. (CR, CR, CR)

Question 2: Do the codes that get me there punch into the Select Tape?  
Answer: No. The Punch remains OFF.

Question 3: Does the field punch into the Select Tape?  
 Answer: Yes. Add an ON 1 Code. (CR, CR, CR, ON 1)

Question 4: Is the entry a constant or a variable?  
 Answer: Variable. Add a Stop Code. (CR, CR, CR, ON 1, STP)

The DELIVER BY date is the last entry on the Heading of the Purchase Order Form. We are now in a position to program the conclusion of the Vendor Data Tape.

|  |             |
|--|-------------|
| Programming to terminate Vendor Data Tape: | C O C C C S |
|  | R F R R R T |
|  | F P         |

Question 1: How do I get to the next field on the form?  
 Answer: Carriage Return four times. (CR, CR, CR, CR)

Before we proceed any further, let's examine the SELECT TAPE. Note, one Carriage Return Code is required to allow Card Release to occur on the Tape-to-Card Punch.

|          |  |
|----------|--|
| C        | Since the Punch has previously been turned ON, there is no need to precede the first of the four Carriage Return Codes with an ON 1 Code to reproduce the Carriage Return Code into the Select Tape. |
| <u>R</u> |  |

|           |   |
|-----------|---|
| C O C C C | The next three Carriage Return Codes, however, do not punch into the Select Tape. Therefore, we must precede them with an OFF Code. |
| R F R R R |   |
| <u>F</u>  |   |

|             |  |
|-------------|--|
| C O C C C S | Finally, we must have a Stop Code at the very end of the Vendor Data Tape. This causes the Tape Reader to Stop. (All Flexowriter tapes should end with a Stop Code.) |
| R F R R R T |  |
| F <u>P</u>  |  |

.....

Following is a point-by-point breakdown of the programming of the Item Data Tape.

|    |                         |               |
|----|-------------------------|---------------|
| G. | <u>QUANTITY ORDERED</u> | O C O N P P S |
|    |                         | F R N P I R T |
|    |                         | F 1 2 P       |

Question 1: How do I get to the next field on the form?  
 Answer: Carriage Return once. (CR)

(Note: The four Carriage Return Codes at the end of the Vendor Data Tape, brought us to a point just one line short of the first writing line of the Body of the Form. The Carriage Return Code at the beginning of the Item Data Tape takes us down to the desired writing line.)

Question 2: Do the codes that get me there punch into the Select Tape?  
 Answer: No. Precede the Carriage Return Code with an OFF Code. (OFF, CR)

Question 3: Does the field punch into the Select Tape?  
 Answer: Yes. Add an ON 1 Code. (OFF, CR, ON 1)

At this time we must provide programming for the Program Indication Code (PI 2) appearing immediately before the Quantity Ordered in the Select Tape.

Precede the Program Indication Code with a Non-Print Code, and follow it with a Print Restore Code. (OFF, CR, ON 1, NP, PI 2, PR)

Question 4: Is the entry a constant or a variable?  
 Answer: Variable. Add a Stop Code. (OFF, CR, ON 1, NP, PI 2, PR, STP)

H. 

|             |          |
|-------------|----------|
| <u>Code</u> | O S O    |
|             | F P N    |
|             | F 1 3208 |

Question 1: How do I get to the next field on the form?  
 Answer: Space once. Add a Space Code. (SP)

Question 2: Do the codes that get me there punch into the Select Tape?  
 Answer: No. Precede the Space Code with an OFF Code. (OFF, SP)

Question 3: Does the field punch into the Select Tape?  
 Answer: Yes. Add an ON 1 Code. (OFF, SP, ON 1)

Question 4: Is the entry a constant or a variable?  
 Answer: Constant. Add 3208. (OFF, SP, ON 1, 3208)

I. 

|                    |                           |
|--------------------|---------------------------|
| <u>DESCRIPTION</u> | O T O                     |
|                    | F A N                     |
|                    | F B 1 BRASS CANDLE STICKS |

Question 1: How do I get to the next field on the form?  
 Answer: Tab. Add a Tab Code. (TAB)

Question 2: Do the codes that get me there punch into the Select Tape?  
 Answer: No. Precede the Tab Code with an OFF Code. (OFF, TAB)

Question 3: Does the field punch into the Select Tape?  
 Answer: Yes. Add an ON 1 Code. (OFF, TAB, ON 1)

Question 4: Is the entry a constant or a variable?  
 Answer: Constant. Add BRASS CANDLE STICKS.  
 (OFF, TAB, ON 1, BRASS CANDLE STICKS)

J. 

|              |         |
|--------------|---------|
| <u>PRICE</u> | T S S S |
|              | A P P P |
|              | B 900   |

Question 1: How do I get to the next field on the form?  
 Answer: Tab. Add a Tab Code. (TAB)

Question 2: Do the codes that get me there punch into the Select Tape?  
 Answer: Yes. Since the Punch is already ON, there is no need to precede the Tab Code with an ON 1 Code.

WHY must this Tab Code punch into the Select Tape?  
 If you will examine the Select Tape and the Tabulating Card now, you will note that a Tab Code is needed to cause Skipping to occur in the variable length Description Field.

Question 3: Does the field punch into the Select Tape?  
 Answer: Yes. Since the Punch is still ON, there is no need to follow the Tab Code with an ON 1 Code.

Question 4: Is the entry a constant or a variable?  
 Answer: Constant. Add three Space Codes and 900. (TAB, SP, SP, SP, 900)

|   |       |
|---|-------|
| <u>Programming to terminate Item Data Tape:</u> | C O S |
|   | R F T |
|   | F P   |

Question 1: How do I get to the next field on the form?  
 Answer: Carriage Return once. (CR)

Again a Carriage Return Code is required in the SELECT TAPE, to allow Card Release to occur on the Tape-to-Card Punch.

|          |  |
|----------|--|
| C        | Since the Punch is still ON, there is no need to precede the Carriage Return Code with an ON 1 Code, to reproduce the Carriage Return Code into the Select Tape. |
| <u>R</u> |  |

|          |   |
|----------|---|
| C O      | It is good practice to end all Data Tapes with the Punch in OFF condition. (Should the operator make any further entries on the Flexowriter, after concluding the Purchase Order, these entries will not punch into the Select Tape.) |
| R F      |   |
| <u>F</u> |   |

|            |   |
|------------|---|
| C O S      | Finally, we must have the required Stop Code to end the Item Data Tape. |
| R F T      |   |
| <u>F P</u> |   |

### Programming the Master Tapes

The final step in our programming will be to determine the codes needed in the Master Program Tapes.

Master Program Tapes are employed to facilitate the preparation of individual Data Tapes. Since we are dealing with a Vendor Data Tape and an Item Data Tape in this Purchase Order application, we will need to prepare a Master Tape for each of these Data Tapes.

Illustrated, on ( Page 46 ), are the Master Tapes, Data Tapes, Select Tapes and the Tabulating Cards for this application.

### Rules for Coding Master Tapes

The Rules which we use to determine the Codes in the Master Tapes are as follows:



C S  
R T  
P

C C C O N S P  
R R R R N P T R

C O C C C N S P S  
R F R R R P T R T  
P

C  
R

C C C O S P  
R R R R N T R  
I P

C O C C C S P  
R F R R R T R  
F P

100 ELM STREET

ANYTOWN, U. S. A.

C  
R

0 6 / 1 4 / 5 8

0 6 | 1 4 | 5 8

DELIVER BY

38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

T  
A  
B

S  
T  
P

C O N S P S  
R F P T R T  
F P

T  
A  
B

S S S  
P P P

C O S P  
R F T R  
F P

L E S T I C K S

9 0 0

T  
A  
B

S S S  
P P P

C  
R

L E S T I C K S

9 0 0

L E S T I C K S

0 0 0 9 | 0 0

P R I C E

38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80

1. COPY ALL OF THE CONTROL CODES IN THE DATA TAPES.  
These Control Codes include all ON 1, OFF, Space, Tab, Carriage Return, Stop, Non-Print, Print Restore and Program Indication Codes.

The only exception to this Rule is applied to the Price Field. Space Codes in the Price Field will vary depending on the Price of the Item.

EXAMPLE: If the Price was \$99.00 rather than \$9.00 we would require two Space Codes instead of three.

It must be remembered, the purpose of these Space Codes is to insure that the Price will always be six digits in length to completely fill the Price Field on the Tabulating Card.

2. WHERE THERE IS CONSTANT DATA IN THE DATA TAPES, INSERT STOP CODES IN THE MASTER TAPES.  
These will Stop the Tape Reader for entry of Constant Data as the operator prepares the individual Data Tapes.

There should be a Stop Code to replace the Name, Street Address, City, Code, Description and Price. (We have "circled" these Stop Codes on the Punched Tape Layout Chart on Page 46 ),

3. WHERE THERE ARE ALREADY STOP CODES IN THE DATA TAPES, EMPLOY NON-PRINT AND PRINT RESTORE CODES IN THE MASTER TAPES TO ALLOW THE STOP CODES TO REPRODUCE FROM THE MASTER TAPES TO THE DATA TAPES.  
We have put a "box" around all such Non-Print and Print Restore Codes on our Layout Chart.

Notice, there is only one set of Non-Print and Print Restore Codes needed to handle the Stop Codes for both the Order No. and the Order Date.

The Non-Print Codes will prevent these Stop Codes from stopping the Tape Reader as the Data Tapes are prepared. The same Non-Print Codes will not punch into the Data Tapes.

Notice, however, the Print Restore Codes do punch into the Data Tapes. We have found it necessary to adjust the Data Tapes by adding these Print Restore Codes. They will not affect the preparation of the Purchase Order in any way.

4. WHERE THERE ARE ALREADY NON-PRINT CODES IN THE DATA TAPES, WE SHOULD ADD A SECOND NON-PRINT CODE TO THE MASTER TAPE IN EACH CASE.  
In the section on Non-Print, we saw the combination of two Non-Print Codes in a row does allow the second of the two to reproduce. We have underscored all such Non-Print Codes on the Layout Chart.

(If we had only one Non-Print Code in the Master Tapes, it would not reproduce in the Data Tapes.)

5. AT THE VERY END OF EACH MASTER TAPE, PUT A STOP CODE TO STOP READING.

## Putting the Program Into Use

The programming of the Purchase Order application is now complete. All that remains is to actually prepare tapes from the Layout Chart, and use these tapes to prepare a Purchase Order.

### Preparing the Master Program Tapes

The first operation is to prepare the Vendor and Item Master Program Tapes. Follow each of these steps to prepare first the Vendor Master Tape and then the Item Master Tape.

1. Place the Punch Switch of the Flexowriter in ALL position.
2. Tape Feed about three inches of tape.
3. Punch the Codes outlined on the Layout Chart into the Vendor Master Tape. (It is good practice to have someone slowly read these Codes to you as you punch them.)
4. When you finish the Vendor Master Tape, Tape Feed about three inches of tape. Remove the Vendor Master Tape from the Punch.
5. Repeat Steps 1 through 4 to prepare the Item Master Tape.

Remember how these Codes are formed?

|               |  |
|---------------|--|
| NON-PRINT     | Depress the AUX CODE Panel Switch and touch the Space Bar, simultaneously. |
| PRINT RESTORE | Depress the AUX CODE Panel Switch and touch the Zero key, simultaneously.  |
| PI 1          | Overpunch 2 8 and a Space Code.  |
| PI 2          | Overpunch 2 8 and Hyphen (dash).   |

### Preparing the Data Tapes

The next step is to prepare the individual Data Tapes. In actual practice, we would prepare a separate Vendor Data Tape for each Vendor to whom we write Purchase Orders; also, a separate Item Data Tape for each regularly purchased Item.

The Master Program Tapes are used to guide us in preparing these individual Data Tapes. Normally, these Master Tapes would be spliced into a loop, with Friden Tape Cement, to make them cycle through the Tape Reader continuously.

#### VENDOR DATA TAPES

To prepare Vendor Data Tapes, follow each of these steps.

1. Insert the Vendor Master Tape into the Tape Reader.

2. Insert a blank sheet of paper into the Flexowriter. On this you will have visual proof of the entries made into the Vendor Data Tape.
3. Place the Punch Switch of the Flexowriter in ALL position.
4. Tape Feed about three inches of tape.
5. Touch the Start Read Switch.

The Program Codes in the Master Tape will be Read and Punched into the Data Tape. Stop Codes for the Order No. and Order Date will not stop the Reader, since they will be read in a Non-Print condition.

When the Stop Code for the Vendor's Name is Read, however, the Reader will stop. ENTER: SMITH SPECIALTY CO.

6. Touch the Start Read Switch.

When the Stop Code for the Street Address is Read, the Reader will stop. ENTER: 100 ELM STREET

7. Touch the Start Read Switch.

When the Stop Code for the City and State is Read, the Reader will stop. ENTER: ANYTOWN, U. S. A.

8. Touch the Start Read Switch.

The remainder of the Programming will be punched into the Vendor Data Tape.

9. Tape Feed about three inches of tape. Remove the completed Vendor Data Tape from the Punch.

You are now ready to prepare the next Vendor Data Tape.

#### ITEM DATA TAPES

To prepare Item Data Tapes, follow this procedure.

1. Insert the Item Master Tape into the Tape Reader.
- 2, 3 and 4. (Same procedure for Vendor Data Tapes, Page 40.)
5. Touch the Start Read Switch.

The Program Codes in the Master Tape will be Read and Punched into the Item Data Tape. The Stop Code for the Quantity Ordered will not stop the Reader, since it will be read in a Non-Print condition.

When the Stop Code for the Code number is Read, however, the Reader will stop. ENTER: 3208

6. Touch the Start Read Switch.

When the Stop Code for the Description is Read, the Reader will stop.  
ENTER: BRASS CANDLE STICKS

7. Touch the Start Read Switch.

When the Stop Code for the Price is Read, the Reader will stop. ENTER:  
SPACE SPACE SPACE 900 (3 Spaces, 900)

8. Touch the Start Read Switch.

The remainder of the programming will be punched into the Item Data Tape.

9. Tape Feed about three inches of tape. Remove the completed Item Data Tape from the Punch.

You are now ready to prepare the next Item Data Tape.

#### Preparing the Purchase Order

The final step is to use these Data Tapes to prepare a Purchase Order, and at the same time punch a Select By-Product Tape for the Tape-to-Card Punch.

This is the procedure to follow now.

1. Set the Margin and Tab Stops on the Flexowriter.

Margin is to be set at 10. Tab Stops to be set at 22 and 57.

2. Insert a blank Purchase Order form in the Flexowriter.

Position the Writing Line Finder on the horizontal line of the Purchase Order form, just above the words "Order No."

3. Tape Feed about three inches of tape.
4. Place the Punch Switch of the Flexowriter in SELECT position.
5. Insert the Vendor Data Tape in the Tape Reader.
6. Touch the Start Read Switch.

The Purchase Order form will be automatically spaced down to the Order No. field. The PI 1 Code will be punched into the Select Tape. The Reader will stop for the entry of the Order No. ENTER: 63205

7. Touch the Start Read Switch.

The Carriage of the Flexowriter will tabulate to the Order Date field. The Reader will stop for the entry of the Order Date. ENTER: 05/14/58

8. Touch the Start Read Switch.

The Purchase Order form will be automatically spaced down two lines, the Name and Address will be read.

The Purchase Order form will be automatically spaced down three lines to the Deliver By field. The Reader will stop for the entry of the Deliver By date. ENTER: 06/14/58

9. Touch the Start Read Switch.

The Purchase Order form will be automatically spaced down three lines. The Reader will stop.

10. Remove the Vendor Data Tape from the Reader.

11. Insert the Item Data Tape in the Reader.

12. Touch the Start Read Switch.

The Purchase Order form will be automatically spaced down one line. The PI 2 Code will be punched into the Select Tape. The Reader will stop for the entry of the Quantity Ordered. ENTER: SPACE SPACE SPACE 15 (3 Spaces, 15)

13. Touch the Start Read Switch.

The balance of the item (Code, Description, Price) will be typed automatically. The Carriage will return to the left Margin. The Reader will stop.

14. Remove the Item Data Tape from the Reader.

15. Touch the ON 1 key. Tape Feed about three inches of tape.

16. Remove the SELECT TAPE from the Punch.

17. Insert the SELECT TAPE in the Reader, to test it.

18. Touch the Start Read Switch.

The information Read should look like this:

|                                   |          |
|-----------------------------------|----------|
| 6320505/14/58 SMITH SPECIALTY CO. | 06/14/58 |
| 153208 BRASS CANDLE STICKS        | 900      |

It is important to note that all Flexowriter operations, both for the preparation of the Data Tapes and actual preparation of the Purchase Order, followed the same basic rule.

The operator's task was reduced to merely touching the Start Read Switch, and making an entry when the Tape Reader stopped.

This rule should be characteristic of all Flexowriter operations, no matter how complex the application.

# Friden, Inc.

HOME OFFICE AND MAIN PLANT  
SAN LEANDRO, CALIFORNIA  
SALES AND SERVICE THROUGHOUT THE WORLD