Digital Equipment Corporation
Maynard, Massachusetts

PDP-8/I

# DIBOL

## Programming Example

DEC-08-WDIA-D

# PDP-8/I
# DIBOL
# PROGRAMMING EXAMPLE

DIGITAL EQUIPMENT CORPORATION • MAYNARD, MASSACHUSETTS

TM
# DIBOL EXAMPLE

## TABLE OF CONTENTS

## I. INTRODUCTION

This document gives a detailed description of an inventory control system using the DIBOL language and the DIBOL data management facilities.

The data management facilities described are:

1) Generalized Input
2) SORT
3) Generalized Update

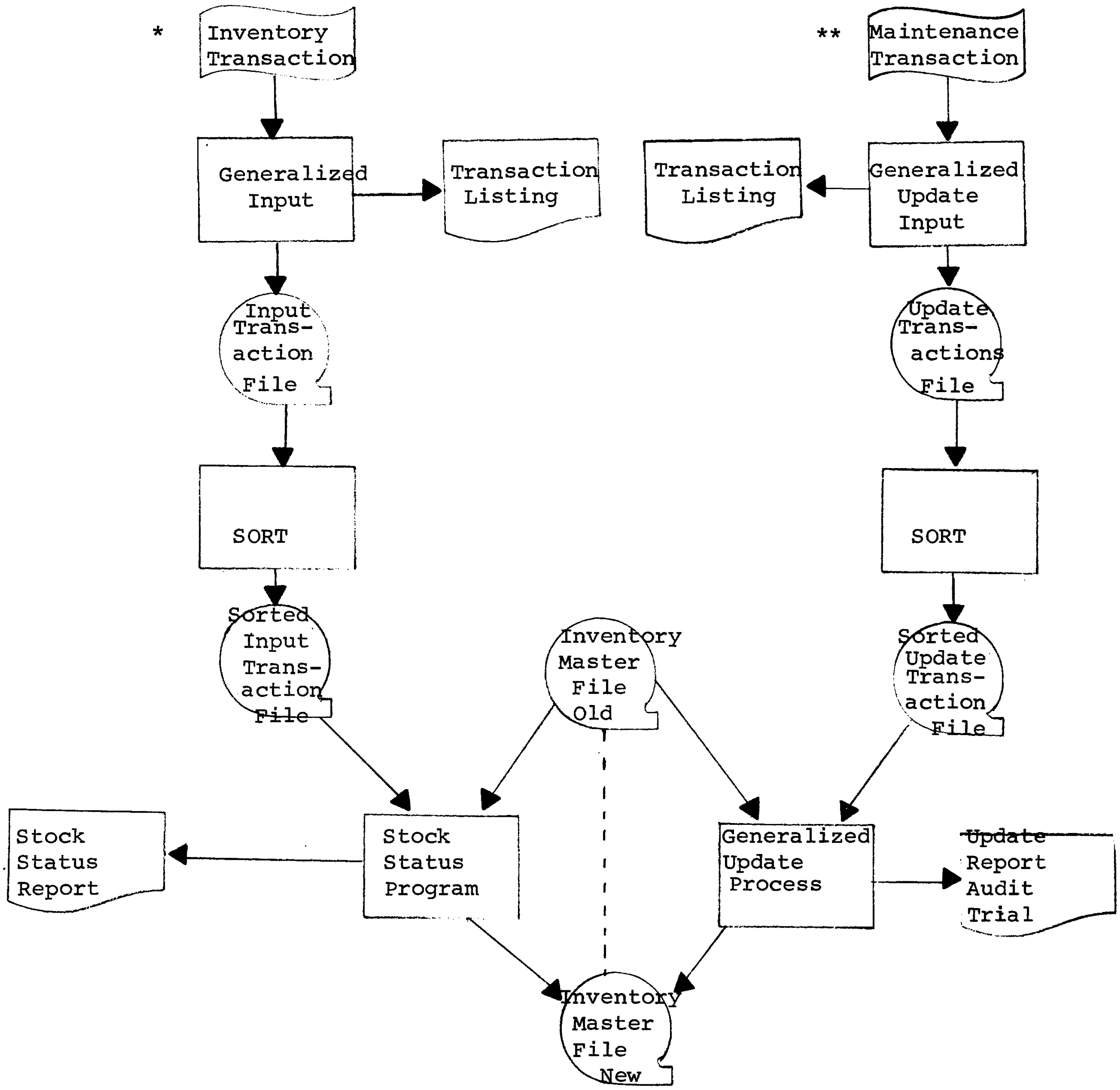
The appendix to this document contains additional information on the components of the DIBOL Software System. It is suggested that you review the Appendix before continuing with this example.

All the necessary steps required for a commercial application are contained in this description.

## II. SYSTEM FLOW CHART

The first step is to lay out the system flowchart. This establishes the sequence of operation, the various uses of input and output files and the reports generated.

The system flow chart for this inventory system is:



The Stock Status Program is the only user created program. All other programs are part of the Data Management Facility and the user need only to define the parameters.

* The first step in the flowchart is the data capture phase. It uses the generalized input to convert inventory transactions (receipts, orders, and amount used) to individual records on DECtape, and also produces a proof listing of all input transactions. The second step sorts these transactions into inventory master order. Finally, the third steps runs the sorted inventory transactions against the old master and creates a new inventory master file. In addition, a stock status print-out is generated showing current activity and year-to-date totals.

** In the case of maintenance transactions, the generalized update input converts the maintenance transactions (changes, insertions, deletions) to individual records on DECtape, while also generating a proof listing. The second step sorts these transactions into inventory master file order. The third step, runs the sorted updated transactions against the old master, and creates a new inventory master file. In addition, an update report (audit trail) is generated showing the old inventory item value and its new value after updating.

## III. FILE DEFINITIONS

After the flowchart has been designed, the content and format of all of the DECtape files must be precisely defined. In the case of this example, there are two files - the inventory master and the transaction file.

The formats are:

### Inventory Master

| Field Number | Description | Type | Size |
|---|---|---|---|
| 1 | Commodity Identification | Decimal | 4 |
| 2 | Description | Alpha | 16 |
| 3 | Number on Hand | Decimal | 4 |
| 4 | Number on Order | Decimal | 4 |
| 5 | Number Used Year-to-Date | Decimal | 4 |
| 6 | Minimum Balance | Decimal | 4 |
| 7 | Unit Cost | Decimal | 6 |
| | | | 42/Each inventory record is 42 characters long |

There is one inventory master record for each item stocked. The record contains a commodity identification number, a description of the item, the number on hand, the number on order, the number that has been used, a minimum balance and a unit cost. These are referred to as fields within the record.

### Transaction File

| Field Number | Description | Type | Size |
|---|---|---|---|
| 1 | Commodity Identification | Decimal | 4 |
| 2 | *Key | Decimal | 1 |
| 3 | Amount | Decimal | 3 |
| | | | 8/Each input transaction is 8 characters long |

*Key: 1 = order
2 = receipt
3 = use

There are three types of transactions. An "order" means an increase to the number on order. A receipt means an increase to the number on hand and a corresponding decrease to the number on order. Finally a "use" transaction means a decrease to the number on hand. Each transaction record contains a commodity identification, the key indicating transaction type and the number of items affected for that commodity.

## IV. DESIGN OF INPUT TRANSACTION

The next step is to design the input transactions. In this case, order, receipt and use are referred to as commands. For this application, the input format is: the command, followed by the commodity identification, followed by the amount. The command will indicate the transaction type. The commodity identification refers to the appropriate record in the inventory master. Finally, the amount indicates the number of units ordered, received, or used. An example of the input might be:

| (Key) | (ID) | (Amount) |
|---|---|---|
| *ORDER | 13 | 20 |
| ORDER | 30 | 50 |
| ORDER | 49 | 50 |
| *RECPT | 22 | 26 |
| RECPT | 23 | 20 |
| RECPT | 28 | 50 |
| RECPT | 50 | 90 |
| *USE | 20 | 5 |
| USE | 15 | 1 |
| USE | 14 | 1 |
| USE | 22 | 7 |
| USE | 22 | 10 |
| USE | 11 | 4 |
| USE | 34 | 3 |
| USE | 35 | 11 |
| USE | 15 | 19 |
| USE | 47 | 17 |

Punched on paper tape or cards.

*These transactions specify that 20 units of item 13 were ordered, 26 units of item 22 were received, and item 20 was used five times.

## V. GENERALIZED INPUT FACILITY

The Generalized Input Facility provides an easy means for capturing, editing and translating data which will later be used to update the master file.

Once the transaction inputs have been created, the Generalized Input Control will read the transaction items as defined, convert them to the proper DECtape format, and perform careful error checking. The procedure for the Generalized Input is: First, the Generalized Input Program from the subsystem tape is loaded into memory by the Monitor. This program enables the computer to read in the Generalized Input Control and the transaction data; Second, the Generalized Input Control is read from paper tape, card reader, or magtape (DECtape), which describes the input and output record formats; Third, the transaction paper tape is read in and converted to DECtape. There may be any number of paper tapes, each ending with a $ carriage return. After the paper tape has been read the teletype will write an asterisk. The user responds with a carriage return after loading the next paper tape or with an N followed by a carriage return at the end of the input.

The Generalized Input Control facility requires the following information:

1. Definition of all data field, both input and output.

2. Definition of each output record stating which fields the record contains, their order and the output device (normally DECtape).

3. Definition of each input record stating which fields the record contains, their order of occurrence (if they are required) and a replacement or alternate field.

4. A summary of the output devices.

This information is recorded on a series of statements or lines, with each unique statement assigned a letter from A through F. The Generalize input control must follow this indicated sequence; i.e., A,B-C,D-E,F.

In our Inventory Control problem the Generalized Input Control is:

| | | | | | |
|---|---|---|---|---|---|
| A | 1 | 1 | 4 | | Definition of Data Fields |
| A | 2 | 1 | 3 | | |
| A | 3 | 1 | 1 | 1 | |
| A | 4 | 1 | 1 | 2 | |
| A | 5 | 1 | 1 | 3 | |
| B | 1 | 1 | | | Definition of Output Record (Order) |
| C | | 1 | | | |
| C | | 3 | | | |
| C | | 2 | | | |
| B | 2 | 1 | | | Definition of Output Record (Recpt) |
| C | | 1 | | | |
| C | | 4 | | | |
| C | | 2 | | | |
| B | 3 | 1 | | | Definition of Output Record (Use) |
| C | | 1 | | | |
| C | | 5 | | | |
| C | | 2 | | | |
| D | ORDER | 1 | | | Definition of Input Record |
| E | | 1 | 1 | | |
| E | | 2 | 1 | | |
| D | RECPT | 2 | | | Definition of Input Record |
| E | | 1 | 1 | | |
| E | | 2 | 1 | | |
| D | USE | 3 | | | Definition of Input Record |
| E | | 1 | 1 | | |
| E | | 2 | 1 | | |
| F | 1 | 1 | | | Summary of Output Devices |

Let's examine each line in detail to understand the parameters.

1. Data Field--Statement( A lines)
   Each Field is assigned a number, and defined as to size (number of characters) and type (alphanumeric or decimal). If a field is used as both input and output, it need be defined only once. A field may be input only, output only or both input and output. For the output only field, an option is available to initialize the field with a constant.

The A statement format is:

```
A   <Field Number>     <type>  <size>   <constant>
```

Field Number-- a one or two digit number
Type-- Ø=Alpha, 1=Decimal
Size-- Any one or two digit number
Constant-- Any value up to 30 alphanumeric characters or 15 decimal digits long. The constant must be consistent with the field type and with an alpha constant enclosed in quotes. This value may be changed by some command received later.

In our inventory example, the input and output formats are defined as:

| Input<br>Field # | Description | Type<br>Size | Output<br>Field # | Description | Type<br>Size |
|---|---|---|---|---|---|
| (3,or 4,or 5) | Transaction Key | D1 | (1) | Commodity ID | D4 |
| (1) | Commodity ID | D4 | (3,4,or 5) | Transaction Key | D1 |
| (2) | Amount | D3 | (2) | Amount | D3 |

The Data Fields would then be:

```
   <Field>  <Type>  <Count>  <Constant>
A     1       1       4                   /field 1 is a decimal number 4 characters lon
A     2       1       3
A     3       1       1      1(order)     /field 3 is a decimal number 1 character long
                                          with an initial constant value of one at the
                                          beginning of the run (remember values 1,2,3,
                                          refer to transaction key order, receipt, and
                                          use, respectively).
A     4       1       1      2 (receipt)
A     5       1       1      3 (use)
```

2. Output Statements (B & C lines)

Each output record is assigned a record number, a device or file number, and the fields (in order) which make up the record. Each record may be written to a different device, or multiple records may be written on the same device.

The B statement format is:

```
     B   <record number>       <file number>
        Record number-- any two digit number which is referenced by
                        an input statement. (D line)
        File Number--   any one digit number which is assigned to an
                        output file.  Output files are normally
                        DECtapes 1-4.
```

The C statement format is:

```
     C   <field number>
        Field Number-- any number previously assigned in an A
                       statement.
```

In our example:

```
B    1  1     /output record for the input command ORDER (1), on
              DECtape #1.  The following C lines describe the output
              format.
C       1     /data field 1, commodity ID
C       3     /data field 3, transaction key 1, ORDER
C       2     /data field 2, amount
B    2  2     /output record for the input command RECPT (2) on
               DECtape #1.
C       1
C       4
C       2
B    3  1     /output record for the input command USE (3) on DECtape
               #1.
C       1
C       5
C       2
```

3. Input Statements (D & E lines)

Each input record contains a transaction code or command, the fields (in order) which comprise the record, if they are required, and the options for field substitutions. Additionally, the output record number for the transaction is identified. Options allow sequence counts to be kept, fields to be cleared before output, and indicators to be set which discloses the presence of a field. This last option is useful when a legal value for the field is zero.

The D & E statement formats are:

```
     D  <Transaction Code> <Output Record> <Zero Field> <Increment>
            or Command          Number                    Field

          Transaction Code-- --up to a 6 character code or command
          Output Record Number-any one or two digit number that has
                               previously been defined by a B statement.
                               The B statement describes the output
```

Output Record Number-(continued) format for this input command. This number may be zero which indicates this transaction is not to be written out. This can be used to initial fields via input data.

Zero Field-- If this parameter contains a number other than zero, it refers to a field number as defined in an A statement. This field will be set to zero prior to writing out the record.

Increment Field-- If this parameter contains a number other than zero, it refers to a field number as defined in an A statement. This field must be decimal and it is incremented by one each time the transaction code or command is encountered. This is useful as a sequence number or counter, e.g. invoice numbering.

```
E   <Field>      <Necessary>      <Present>      <Default>
    Number        Switch           Field          Field
```

Field Number-- Any two digit number which refers to a field previously defined by an A statement.

Necessary Switch- A one digit number which has a value of zero or one. Zero indicates the field is optional; (i.e., it may or may not be present). One indicates the field is required. When required, if the field is missing from the input record, it is flagged as an error; e.g. in a payroll program, the input must contain the employee's name.

Present Field-- A one or two digit number which refers to a field, previous defined by an A statement. This is used in conjunction with the necessary switch. If the necessary switch is zero, then the field referenced by the present switch is set to one when the field is found on the input record and set to zero when the field is missing. The field referenced by the present field must be defined as decimal 1 (D1). In our inventory example the present field was set to zero. See end of this section for a detailed explanation of the present field.

Default Field-- A one or two digit number which refers tc a field previously defined by an A statement. This is used in conjunction with the necessary switch. If the necessary switch is set to zero, (indicating an optional field) and the default field is non-zero, then the field referenced (by the default field) is substituted when the input field is missing. The fields must be the same size. For example, E 3 0 0 5 says if field 3 data is missing, substitute field 5 in its place.

Note, only the present field or the default field may be used. They cannot be used together, and they may be used only when the necessary switch is zero.

In our example, the D & E lines are:

```
D    ORDER     1              /Every time there is a transaction
                               ORDER input it will be outputted
                               according to B1
E              1    1         /Item number-must be present
E              2    1         /Amount--must be present
D    RECPT     2
E              1    1
E              2    1
D    USE       3
E              1    1
E              2    1
```

## DETAILED EXPLANATION OF PRESENT FIELD

In order to give you a better understanding of the usage of the present field, another example might be helpful. Suppose we had a billing operation in which there could be price overrides.

The inputs to this example would be:
Commodity ID,
# shipped, and
price override (if present).

The output would be:
Commodity ID,
# shipped,
price override present (yes/no)
price override

The Generalized Input Control for this example would be:

```
         A   1   0   5          /Commodity ID
Data     A   2   1   4          /# Shipped
Fields   A   3   1   1          /price present
         A   4   1   6          /price override

         B   1   1              /output the input command 1 on DECtape 1

Output   C       1
Records  C       2
         C       3
         C       4

         D   I   1              /output this input transaction I
                                 according to B1

Input
Records  E       1   1              /Commodity ID-required
         E       2   1              /# shipped-required
         E       4   0     3        /override price not required but,if
                                     present, Set A3 to 1 and use
                                     override price (special price) of
                                     A4.  If override price not present
                                     the A3 would be zero and the price
                                     on the commodity master would be
                                     used.  In this manner, a special
                                     price to customers may be recorded.
```

## 4. Summary of Output Devices (F line)

F statement defines file specifications. This simply indicates the number of different DECtapes specified as the file number on all the B lines. The format is:

```
F   <number of files>  <file 1>   <file 2>   <file 3>   <file 4>
```

In our inventory example:

```
     F   1   1               /says total number of one file, on
                              DECtape #1
```

Now, let's put all of these statements back together in our Generalized Input Control to see the entire operation.

```
A    1    1    4             /Data Field for commodity number,4 digits
A    2    1    3             /Data field for amount, 3 digits
A    3    1    1    1        /Data field for order transaction key,
                                constant of 1
A    4    1    1    2        /Data Field for receipt transaction key,
                                constant of 2
A    5    1    1    3        /Data Field for use transaction key,
                                constant of 3
B    1    1                  /Output record for the input command 1
                                (ORDER) on DECtape #1
C         1                  /(Output format) field 1-item number
                                              field 3-order key
                                              field 2-amount
B    2    1                  /Output record for the input command 2
                                (RECPT) on DECtape #1
C         1
C         4
C         2
B    3    1                  /Output record for the input command 3
                                on DECtape #1



C         1
C         5
C         2
D  ORDER  1                  /Input transaction record ORDER to be
                                outputted according to B1
E         1    1             /Input item number--required
E         2    1             /Input Amount--required
D  RECPT  2
E         1    1
E         2    1
D  USE    3
E         1    1
E         2    1
F    1    1                  /Total number of files 1, on DECtape #1
```

As each transaction is read by the Generalized Input program, it is listed and any transactions in error will have an error message printed to the left. Only valid transactions are written on the output tape.

From the transaction input, i.e.,, USE 20 5, this would be outputted on DECtape 1 as:

```
20
3  (use key)
5
```

This would be continued for all valid input transactions.

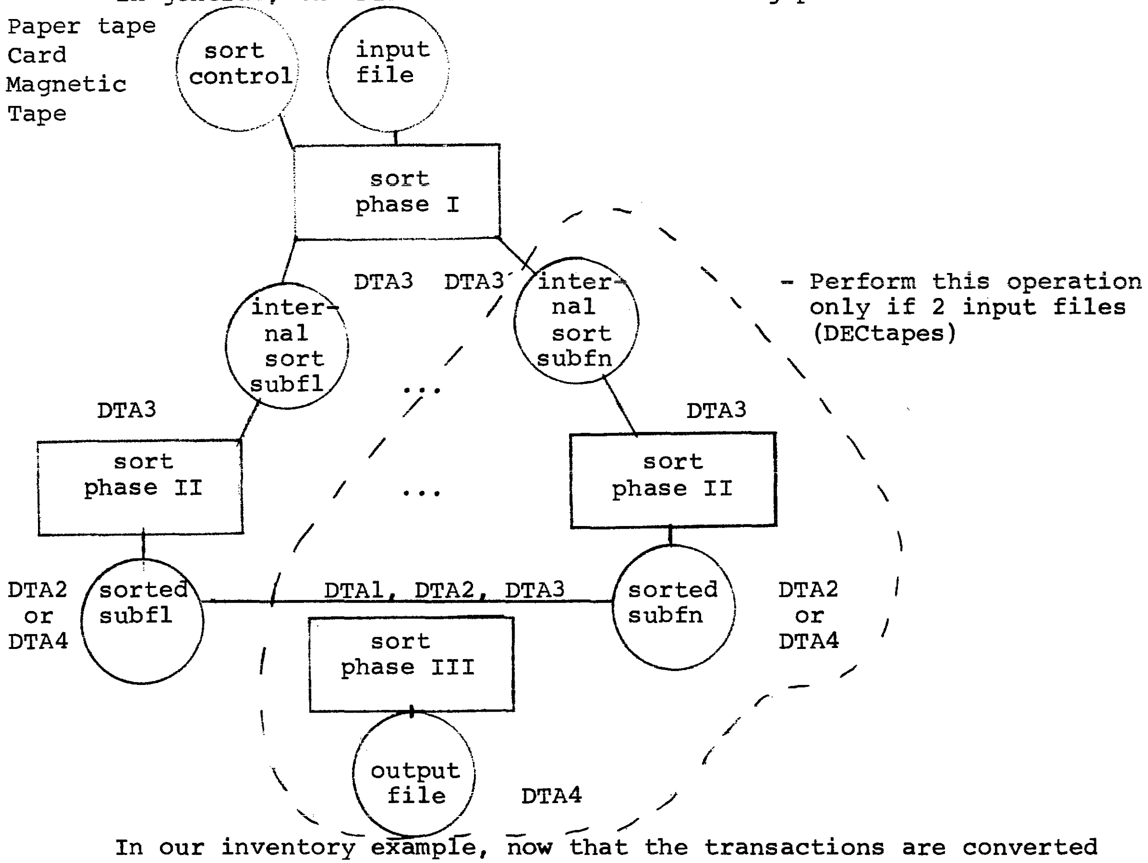
## VI. SORT

The sort package allows the user to arrange a data file in any specified order. The user defines the records to be sorted and gives "sort keys" within the record. The sort run is a two or three phase process, depending on the number of input DECtape files. The system's tape contains the programs for sort phase 1, sort phase 2, and sort phase 3. Each program enables the system to perform specific sort operating instructions. These programs are called into core by the DIBOL monitor. The user must create the sort control program defining the sort parameters.

The three phases of the sort operation follow:

> The first phase is an internal sort. It reads in a predetermined number of records from the input file, sorts them as specified by the sort control, and writes them out on DECtape #3. This operation is continued until the end of the input DECtape. The output from Phase I will contain a string of sorted subfiles on DECtape 3. The number of input DECtapes will normally equal the number of output DECtapes after Phase I. In the case of a single reel input, the Phase I output will be a single subfile. In the case of multi-reel inputs, when an output subfile is filled, the operator is requested to take it down and put up a new tape for the next Phase I output subfile.
>
> Phase II then uses a multi-phase merge to sort each output from Phase I; i.e., strings of sorted subfiles so that only one sorted string remains. Phase II requires the use of all four DECtapes. Phase II must be run as many times as there are output files from Phase I.
>
> The Phase III sort is used only in the case of multi-reel files. Phase III merges the Phase II outputs into a single multi-reel output file. (Phase III is not required if there is only one output subfile from Phase I).
>
> If there are 2 input files (DECtapes), then there would be 2 output files (DECtapes) after Phase I; each containing strings of sorted subfiles. Each output file would then be individually sorted (sort strings of subfiles into one) so that after Phase II there are two files, each on DECtape and each separately sorted. However, this still means there are two sorted subfiles. Phase III will merge the two sorted subfiles into one sorted file.

In general, the DIBOL sort has the following process chart:



In our inventory example, now that the transactions are converted from input format to DECtape format, vou need to sort them into the order of the inventory master,that is by commodity identification. To do so, order the records in the ascending order of the first four characters (commodity identification). The sort control for this is simply:

```
first line   ØØ518ØØ1Ø1Ø5Ø
sec. line    ØØ2Ø2ØØ
             $
```

(Punched on paper tape, cards or magnetic tape)

The first line (13 characters) defines the sort keys. The second line (7 characters) defines the record (sort field) to be sorted, and its position on the DECtape file. In our example the sort field is by commodity identification. Let's examine what each character means:

FIRST LINE-13 Character, i.e.

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| 0 | 0 | 5 | 1 | 8 | 0 | 0 | 1 | 0 | 1 | 0 | 5 | 0 |

1-3 (005) /Record input size in words
to calculate the size of the input record in words, add 1 to the record character length divided by 2 (drop any remainder) plus 1. In our example the transaction input record is 8 characters long; i.e., commodity ID, key, and amount (refer to transaction file)
8 + 1 = 9   2= 4 + 1= 5 (coded as 005)

4-7 (1800) /Internal Sort Buffer Size
Must be a multiple of record size in words and less than 2000. One normally chooses this number as large as possible in order to sort as many records internally in Phase I. (There is no common practice for determining this buffer size, it is dependent on the record size)
1800= (5 x 360 a constant)

8 (1) Number of sort fields
(If two sort fields are contiguous, they are considered as 1). In our example there is only one sort field--commodity identification.

9-12 (0105) /Number of Blocks per DECtape
This field times the size of the internal sort buffer must be less than 190,000 (words per DECtape). This is calculated by dividing internal sort buffer into 190,000
190,000 ÷ 1800= 105 (coded as 0105)

13 (0) /Phase 2 Input Save Switch
If the input from Phase 2 should be saved, Code 1; If not, Code 0. Sort coded 0 will write over the input to Phase 2. Normally this field is 0 and the input tapes to Phase 2 are written over as part of the multiphase merge.

SECOND LINE (One for each sort field) 7 characters

| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|
| 0 | 0 | 1 | 0 | 2 | 0 | 0 |

1-3 (001) Beginning Location of field on the DECtape file
First position of field + 1 ÷ 2 (drop remainder)
In our example commodity ID begins in Position 1
(See Transaction file)

1 + 1 ÷ 2 = 1

4-5 (02) Number of words spanned
The formula for calculating words spanned is dependent on the beginning position of the field on the DECtape file, i.e., even or odd, and the number of characters in the field, again even or odd. The formula for words spanned is:

| Beginning Position | # of Characters in Field | | Calculation |
|---|---|---|---|
| Even | Even | = | # of Characters ÷2+1 |
| Even | Odd | = | # of Characters ÷2+1 |
| Odd | Odd | = | # of Characters ÷2+1 |
| Odd | Even | = | # of Characters÷ 2 |
| | | | (all remainders are dropped) |

In our example Commodity ID begins in Position 1, and is four characters long

4 ÷ 2=2 (Odd,Even)

6 (0) /First word switch
If final character of sort key is even, code as 1; if odd, code 0 (it's odd in our example, it starts in Position 1)

7 (0) /Last word switch
If ending position for sort key is odd, code as 1; if even, code 0. In our example commodity ID ends in Position 4 (4 characters)

## VII. STOCK STATUS PROGRAM

The major portion of this application is the user created program which updates the inventory master and prints the stock status report. This program consists of only 164 DIBOL statements, of which 80 are executable. The other 84 statements are used to define the various data elements. The program is:

```
START           ;X003 - DEMO UPD + RPT
BLOCK M
          M1,   D4
          M2,   A16
          M3,   D4
          M4,   D4
          M5,   D4
          M6,   D4
          M7,   D6
BLOCK T
          T1,   D4
          T2,   D1
          T3,   D3
BLOCK HOF,C
          ,     D2,70
          ,     A1
          ,     D1
BLOCK CHD, C
          ,     D2
          ,     A11,'  ID DESC'
          ,     A14
          ,     A9,'POH  XCP '
          ,     A11,'#CUR #YTD '
          ,     A13,'NOH  ORD MBAL'
          ,     A4
          ,     A5,'UCOST'
          ,     A8
          ,     A5,'XCOST'
          ,     D1
BLOCK PHD,C
          ,     D2
          ,     A10,'STOCK STAT'
          ,     A10,'US REPORT '
          ,     A20
          PHD1, A8
          ,     A24
          ,     A4,'PAGE'
          PHD2, A4
          ,     D1
BLOCK BLK,C
          ,     D2
          ,     A1
          ,     D1
BLOCK P,C
          ,     D2
          PP,   A80
          ,     D1
BLOCK,X
          ,     D2
          ,     A2
```

```
          P1,     A4
          ,       A1
          P2      A16
          ,       A1
          P3,     A4
          ,       A1
          P4,     A4
          ,       A1
          P5,     A4
          ,       A1
          P6,     A4
          ,       A1
          P7,     A4
          ,       A1
          P8      A4
          ,       A1
          P9,     A4
          ,       A1
          P10     A8
          ,       A1
          P11     A12
          ,       D1
BLOCK
          M8,     D4
          M9,     D4
          M10,    D4
          PG,     D3
          LN,     D2
          TOT,    D9
          DATE,   D6,P
          TRAN,   D1,P
          XCOST,  D9
          EOFM,   D1
          EOFT,   D1
PROC
          TOT=
          PG=
          LN=50
          INIT(2,V,IN)
          IF(TRAN.EQ.0)GO TO Q6
          INIT(1,V,IN)
Q6        INIT(3,V,OUT)
          CALL READM
          CALL READT
Q5        IF(EOFM.EQ.1)GO TO Q1
Q3        IF(EOFT.EQ.1)GO TO Q2
          IF(T1.GT.M1)GO TO Q2
          GO TO (ORD,RED,USE),T2
ORD,      M4=M4+T3
          CALL READT
```

```
          GO TO Q3
REC,      M4=M4+T3
          M3=M3+T3
          M9=M9+T3
          CALL READT
          GO TO Q3
USE,      M3=M3+T3
          M5=M5+T3
          M10=M10+T3
          CALL READT
          GO TO Q3
Q2,       PP=
          IF(M3,GE,M6)GO TO Q4
          PP(1,1)='*'
Q4,       P1=M1
          P2=M2
          P3=M8
          P4=M9
          P5=M10
          P6=M5
          P7=M3
          P8=M4
          P9=M6
          P10=M7,'X,XXX,XX'
          XCOST=M3*M7
          P11=XCOST,'X,XXX,XXX.XX'
          TOT=TOT+XCOST
          CALL PRINT
          XMIT (3,M)
          CALL READM
          GO TO Q5
Q1,       XMIT(6,BLK)
          PP=
          PP(67,67)='*'
          P11=TOT,'X,XXX,XXX.XX'
          XMIT (6,P)
          FINI(3)
          XMIT (6,HOF)
          STOP
READM,    EOFM=1
          XMIT(2,M,Z1)
          EOFM=
          M8=M3
          M9=
          M10=
Z1,       RETURN
READT,    EOFT=1
          IF(TRAN,EQ,0)GOTOZ2
          SMIT(1,T,Z2)
          EOFT=
```

```
Z2,       RETURN
PRINT,    LN=LN+1
          IF(LN.LE,50)GO TO Z3
          LN=
          PG=PG+1
          PHD1=DATE,'XX/XX/XX'
          PHD2=PG
          XMIT(6,HOF)
          XMIT(6,PHD)
          XMIT(6,BLK)
          XMIT(6,CHD)
          XMIT(6,BLK)
Z3,       XMIT(6,P)
          RETURN
END (0430,0481)
```

## EXPLANATION OF PROGRAM

```
START
BLOCK M                                         ;(INVENTORY MASTER FILE)
        M1,    D4                               ;Commodity ID, 4 digits
        M2,    A16                              ;Description, 16 character alpha
                                                 word
        M3,    D4                               ;Number on hand, 4 digits
        M4,    D4                               ;Number on order, 4 digits
        M5,    D4                               ;Number used year to date, 4 digits
        M6,    D4                               ;Minimum balance, 4 digits
        M7,    D6                               ;Unit cost, 6 digits
BLOCK T                                         ;(TRANSACTION FILE)
        T1,    D4                               ;Commodity ID
        T2,    D1                               ;Transaction key (order, receipt,
                                                 used)
        T3,    D3                               ;Amount
BLOCK HOF,C                                     ;(HEAD OF FORM) Required for
                                                 line printer
        ,      D2,70
        ,      A1
        ,      D1
BLOCK CHD,C                                     ;(RECORD HEADING ON PAGE)-See
                                                  sample report
        ,      D2                               ;When using line printer as
                                                 output, start block with D2.
                                                 This is a device control and does not
                                                 mean 2 zeros
        ,      A11,'        ID DESC'
        ,      A14                              ;Spaces
        ,      A9, 'POH     XCP'
        ,      A11,'#CUR    #YTD'                  Column Headings
        ,      A13,'NOH ORD MBAL'
        ,      A4
        ,      A5, 'UCOST'
        ,      A8
        ,      A5, 'XCOST'
        ,      D1                               ;When using line printer as output
                                                 End Block with D1. This is a
                                                 control and does not mean 1 zero
BLOCK PHD,C                                     (PAGE DESCRIPTION)-See  sample
                                                report
        ,      D2                               ;Start block with printer control
        ,      A10,'STOCK STAT'                 ;Stock Status Report
        ,      A10,'US REPORT'
        ,      A20
     PHD1,     A8                               ;Date location to be referenced
                                                 in program
        ,      A24
        ,      A4,'PAGE'
     PHD2,     A4                               ;Page number location
               D1                               ;End Block with printer control
```

```
BLOCK BLK,C                                 ;(BLANK LINE)
        ,      D2
        ,      A1                           ;Prints blank digits
        ,      D1
BLOCK P,C                                   ;(PRINTER BUFFER)
        ,      D2
        PP,    A80                          ;Printer buffer 80 columns
        ,      D1
BLOCK ,X                                    ;(DATA)-Data Overlay into printer
                                                   buffer
        ,      D2
        ,      A2                           ;2 spaces
        P1,    A4                           ;ID Number (in columns 3-7)
        ,      A1                           ;Space
        P2,    A16                          ;Description
        ,      A1
        P3,    A4                           ;Previous on hand
        ,      A1
        P4,    A4                           ;Receipt
        ,      A1
        P5,    A4                           ;Current #
        ,      A1
        P6,    A4                           ;Year to date #
        ,      A1
        P7,    A4                           ;New on hand
        ,      A1
        P8,    A4                           ;Ordered
        ,      A1
        P9,    A4                           ;Minimum Balance
        ,      A1
        P10,   A8                           ;Unit Cost
        ,      A1
        P11,   A12                          ;Extended Cost
        ,      D1
BLOCK                                       ;(TEMPORARY STORAGE)
        M8,    D4                           ;Temporary Storage
        M9,    D4
        M10,   D4
        PG,    D3                           ;Temporary Storage for page number
        LN,    D2                           ;Temporary Storage for line count
        TOT,   D9                           ;Temporary Storage for Total
        DATE,  D6,P                         ;Allows entry of DATE from console
                                                before program execution
        TRAN   D1,P                         ;Allows entry of  parameter  from
                                                console. If no input transactions
                                                =0, if there are transaction
        XCOST  D9                           ;Temporary storage for extended cost
        EOFM   D1                           ;End of Master Indicator
        EOFT   D1                           ;End of Transaction
PROC                                        ;BEGIN PROCESSING SECTION
        TOT=                               ;Set TOTAL to zero
        PG=                                 ;Set page to zero
        LN=    50                           ;Set line count  to 50
        INIT (2,V,IN)                       ;Initialize DECtape #2 (old master) as
                                             an input file with variable length
                                             records
```

```
          IF (TRAN.EQ.0) Go to Q6        ;If no transaction(Trans=0) go
                                          to Q6--would rewrite inventory master
                                          and produce a listing of the master file
          INIT (1,V,IN)                  ;Initialize DECtape #1 (sorted
                                          transaction) as an input file
Q6,       INIT (3,V,OUT)                 ;Initialize DECtape #3 (new
                                          master) as an input file
          CALL READM                     ;Execute subroutine READM (read
                                          old master)
          CALL READT                     ;Execute subroutine READT (reads
                                          transaction record)
Q5,       IF (EOFM.EQ.1) Go To Q1        ;If end of master=1 (all records
                                          read) go to Q1 (print Total)
Q3,       IF (EOFT.EQ.1) Go To Q2        ;If end of transaction=1 go to
                                          Q2
          IF (T1.GT.M1) Go To Q2         ;If transaction commodity ID# is
                                          greater than master ID# go to
                                          Q2 (see footnote)
          Go To (ORD,REC, USE),T2        ;If T2 (transaction file key)
                                          equals 1, go to ORD, If=2 go to
                                          REC, If=3 go to USE
ORD,      M4 = M4 + T3                   ;Number on order in old master +
                                          number ordered in transaction
                                          will be new number on order
                                          figure held in M4
          CALL READT                     ;Execute subroutine (reads
                                          another transaction record)
          Go To Q3                       ;Jump statement
REC,      M4 = M4 - T3                   ;Number on order in new master
                                          will be number on order (old
                                          master)-number received (trans-
                                          action record) held in M4
          M3 = M3 + T3                   ;Number on hand in new master
                                          will be number on hand (old
                                          master) + amount received (trans-
                                          action record) held in M3
          M9 = M9 + T3                   ;M9 will store amount received
          CALL READT                     ;Execute subroutine READT (reads
                                          another transaction record)
          Go To Q3                       ;Jump statement
USE,      M3 = M3 - T3                   ;Number on hand in new master will
                                          equal number on hand (old master)
                                          minus amount used
          M5 = M5 + T3                   ;Number used year to date in new
                                          master will equal number used
                                          year to date plus amount (trans-
                                          action record) held in M5
          M10 = M10 + T3                 ;M10 will store current amount used
          CALL READT                     ;Execute subroutine (read another
                                          transaction record)
          Go To Q3                       ;Jump statement
```

```
Q2,      PP=                            ;Set printer buffer to zero
         If(M3.GE.M6)Go To Q4           ;If number on hand is greater
                                         than or equal to minimum balance
                                         go to Q4
         PP (1,1) = '*'                 ;Print an * when number on hand
                                         is less than MBAL

Q4,      P1=M1                          ;Commodity ID will be in P1
         P2=M2                          ;Description
         P3=M8                          ;Previous on hand(see M8)
         P4=M9                          ;Amount received
         P5=M10                         ;Current number used
         P6=M5                          ;Total used year to date
         P7=M3                          ;Total number on hand
         P8=M4                          ;Total number on order
         P9=M6                          ;Minimum balance
        P10=M7,'X,XXX.XX'               ;Unit cost (formatted)
         XCOST = M3 * M7                ;Extended cost=number on hand x
                                         unit cost
         P11=XCost,'X,XXX,XXX.XX'       ;Stores extended cost (formatted)
                                         in P11
         TOT=TOT + XCost                ;Incremental Total Cost by adding
                                         XCost for each record
         CALL PRINT                     ;Execute subroutine   print
         XMIT (3,M)                     ;Write onto DECtape #3 (new master)
                                         block M (master file)
         CALL READM                     ;Execute subroutine(read old master)
         Go To Q5                       ;Jump statement

Q1,      XMIT (6,BLK)                   ;Write onto device 6 (line printer)
                                         contents of block BLK which is
                                         blank line
         PP=                            ;Set print buffer to zero
         PP (67,67)='*'                 ;Print * in column 67
         P11=TOT,'X,XXX,XXX.XX'         ;Store total cost in P11
         XMIT (6,P)                     ;Write onto line printer block
                                         P (Block P will now contain only the
                                        total figure)
         FINI (3)                       ;Finalize DECtape #3 (new master)
                                         is rewound
         XMIT (6,HOF)                   ;Write Head of Forms Block
         STOP                           ;Return Control to DIBOL monitor

READM,   EOFM=1                         ;Set EOF Master file to 1
         XMIT (2,M,Z1)                  ;Read a record from DECtape #2
                                         old inventory master; at end of
                                         file go to Z1
         EOFM=                          ;Set end of master to zero
         M8=M3                          ;Put number on hand into location
                                         M8
         M9=                            ;Set location M9 to zero
         M10=                           ;Set location M10 to zero

Z1,      RETURN                         ;Return to next statement after
                                         CALL READM
```

```
READT     EOFT=1                         ; Set end of transaction to 1
          IF(TRAN.EQ.O)Go To Z2          ; If no transaction(0) go to Z2
                                            When no more transactions, the
                                            program could continue to read
                                            the rest of the older inventory
                                            items
          XMIT (1.T.Z2)                  ;Read from DECtape #1 a trans-
                                          action record; at end of file
                                          go to Z2
          EOFT=                          ;Set end of transaction to zero
Z2,       RETURN                         ;Return to statement after
                                          CALL READT
PRINT,    LN=LN+1                        ;Increment line count
          IF (LN.LE.50)Go To Z3          ;If line number is equal to or
                                          less than 50, go to Z3
          LN=                            ;Set line  count to zero
          PG=PG+1                        ;Increment page number
          PHD1=DATE,'XX/XX/XX'           ;Format date for page header
          PHD2=PG                        ;Inserts page number for page
                                          description
          XMIT (6,HOF)                   ;Write on device 6 (line printer)
                                          block HOF (head of forms)
          XMIT (6,PHD)                   ;Write on device 6 (line printer)
                                          block PHD (page description)
          XMIT (6,BLK)                   ;Write on device 6 (line printer)
                                          block BLK (blank line)
          XMIT (6,CHD)                   ;Write on device 6 (line printer)
                                          block CHD (record heading)
          XMIT (6,BLK)                   ;Write on device 6 (line printer)
                                          block BLK (blank line)
Z3,       XMIT (6,P)                     ;Write on device 6 (line printer)
                                          block P (data)
          RETURN                         ;Return to statement after CALL
                                          PRINT
          End  (0430,0481)               ;1st set of numbers is the number of
                                         words used in the DATA
                                          section. (Maximum is 1170) 2nd
                                          set of numbers is the number of
                                          words used in the Procedure Section-
                                          after PROC. (maximum 1050)
```

Footnote

| * Master Commodity ID | Transaction after sorted commodity ID | |
|---|---|---|
| 1 | 1 | |
| 2 | 1 | |
| 3 | 1 | |
| 4 | 2 | At this point trans commodity ID is greater than master ID go to Q2 (print subroutine) |
| etc. | | |
| | 3 | |
| | 4 | |
| | 4 | |

## VIII. STOCK STATUS REPORT

Suppose that the transactions shown in Section IV were run against the master file; you would receive the following report print out:

```
STOCK STATUS REPORT           3/1Ø/7Ø              PAGE 1               (Page Description)

   ID DESC              POH XCP #CUR #YTD NOH  ORD MBAL   UCOST     XCOST    (Record Heading)

    1 FLANGE, PURPLE    118   Ø    Ø  699 118   74   57  873.79 1Ø3,1Ø7.22
    2 SCREW, BLACK      1Ø6   Ø    Ø  793 1Ø6   57   52  835.57  88,57Ø.42
    3 NUT, BROWN        119   Ø    Ø   17 119   35   79  675.72  8Ø,41Ø.68
    4 BRACKET, BLACK    154   Ø    Ø  334 154   86   9Ø  931.Ø5 143,381.7Ø
    5 BRACKET, WHITE    1Ø9   Ø    Ø  797 1Ø9   91   31   59.87   6,525.83
    6 RIVIT, WHITE      146   Ø    Ø  323 146   31   95  755.73 11Ø,336.58
    7 PIN, YELLOW       146   Ø    Ø  44Ø 146   54   66  912.Ø4 133,157.84
    8 RIVIT, RED        113   Ø    Ø  8Ø8 113   95   38  295.69  33,412.97
    9 SCREW, PURPLE     1Ø7   Ø    Ø  3Ø1 1Ø7   62   6Ø  273.15  29,227.Ø5
   1Ø DOWEL, GREEN       94   Ø    Ø  542  94   5Ø   54   5Ø.29   4,727.26
   11 HINGE, GREEN      174   Ø    4    6 17Ø   35   93  537.16  91,317.2Ø
   12 CLIP, RED         143   Ø    Ø   8Ø 143   73   8Ø  717.93 1Ø2,663.99
   13 FLANGE, GREEN      83   Ø    Ø  441  83   78   11  161.63  13,415.29
   14 FLANGE, PURPLE    169   Ø    1  115 168   95   2Ø  815.25 136.962.ØØ
   15 BRACKET, BLACK     99   Ø   2Ø  451  79   2Ø   62  343.Ø8  27,1Ø3.32
   16 PIN, BLUE         154   Ø    Ø  4Ø7 154   62   61  981.77 151,192.58
   17 DOWEL, BLACK       97   Ø    Ø  33Ø  97   86   53  693.31  67,251.Ø7
   18 NUT, YELLOW       171   Ø    Ø  465 171   19   92  786.61 134,51Ø.31
   19 PIN, BLUE          84   Ø    Ø  976  84   94   9Ø  4Ø1.8Ø  33,751.2Ø
   2Ø RIVIT, YELLOW      92   Ø    5   89  87   94   59  431.21  37,515.27
   21 BRACKET, PINK      81   Ø    Ø  16Ø  81   27   2Ø  956.19  77,451.39
   22 HINGE, YELLOW     147  26   17  193 156    Ø   97  94Ø.37 146,697.72
   23 DOWEL, RED         79  2Ø    Ø  2Ø3  99   27   54  265.32  26,266.68
   24 FLANGE, RED        8Ø   Ø    Ø   47  8Ø    8   62  37Ø.25  29,62Ø.ØØ
   25 PIN, BROWN        122   Ø    Ø  4Ø3 122   81   69  958.27 116,9Ø8.94
   26 FLANGE, WHITE     146   Ø    Ø  394 146   53   1Ø   92.53  13,5Ø9.38
   27 BOLT, BROWN       144   Ø    Ø  8Ø2 144   33   22  165.64  23,852.16
   28 NUT, BLACK        156  5Ø    Ø  473 2Ø6   22   51  462.89  95,355.34
   29 HINGE, BROWN      134   Ø    Ø   1Ø 134    1   42  427.55  57,291.7Ø
   3Ø FLANGE, PURPLE     81   Ø    Ø  221  81  1ØØ   76  499.Ø9  4Ø,426.29
   31 HINGE, BLUE       167   Ø    Ø  38Ø 167   29    2  214.76  35,864.92
   32 DOWEL, PINK       114   Ø    Ø  377 114    Ø   79  133.13  15,176.82
   33 CLIP, GREEN       147   Ø    Ø  615 147   38   24  612.Ø3  89,968.41
   34 PIN, RED           91   Ø    3  125  88   57   24  456.Ø5  4Ø,132.4Ø
   35 PIN, BLACK         82   Ø   11  276  71   97   75  6Ø4.68  42,932.28
   36 PIN, ORANGE       146   Ø    Ø  215 146   76   5Ø  964.97 14Ø,885.62
   37 CLIP, RED          83   Ø    Ø  728  83   79    6   79.71   6,615.93
   38 FLANGE, GREEN      97   Ø    Ø  32Ø  97   86   94  899.41  87,242.77
   39 DOWEL, BLUE        93   Ø    Ø  3Ø6  93   43   41  8Ø7.4Ø  75,Ø88.2Ø
   4Ø NUT, PURPLE        77   Ø    Ø  484  77   7Ø    2  1Ø2.41   7,885.57
   41 FLANGE, BLUE      133   Ø    Ø  942 133   85   39  518.59  68,972.47
   42 PIN, YELLOW       164   Ø    Ø  2ØØ 164    3   53    5.17     847.88
   43 NUT, YELLOW        91   Ø    Ø  67Ø  91   75   53  774.92  7Ø,517.72
   44 HINGE, ORANGE     1Ø9   Ø    Ø  32Ø 1Ø9   63   83  649.45  7Ø,79Ø.Ø5
   45 PIN, BLACK        143   Ø    Ø  128 143   35    4  536.67  76,743.81
   46 BOLT, RED         162   Ø    Ø  7Ø5 162    4   17  789.33 127,871.46
   47 SCREW, YELLOW     174   Ø   17  661 157   69   62  139.24  21,86Ø.68
   48 BRACKET, GREEN    162   Ø    Ø   97 162   6Ø   33   7Ø.Ø9  11,354.58
   49 BRACKET, YELLOW    89   Ø    Ø  64Ø  89  114   6Ø  461.95  41,113.55
   5Ø RIVIT, RED        125  9Ø    Ø  74Ø 215    8   69  7Ø7.89 152,196.35
   51 DOWEL, BROWN      172   Ø    Ø  439 172   2Ø   54  412.36  7Ø,925.92
```

Notice that item 14 was used once. Item 22 was received 26 times and used 17 times - for a total change of nine in the "on hand" column. The new on-hand balance for item 19 is less than the minimum balance - thus an asterisk is printed out to the left.

## IX. FILE MAINTENANCE - GENERALIZED UPDATE FACILITY

In addition to the regular master file activity update and stock status report, it is necessary to have some means of creating and changing a master file. For example, it may be necessary to change the minimum balance or description of some existing item, or completely delete an item from stock. The Generalized Update Facility provides the means for the user to perform insertions, changes, and deletions.

The update run is a three stage process:

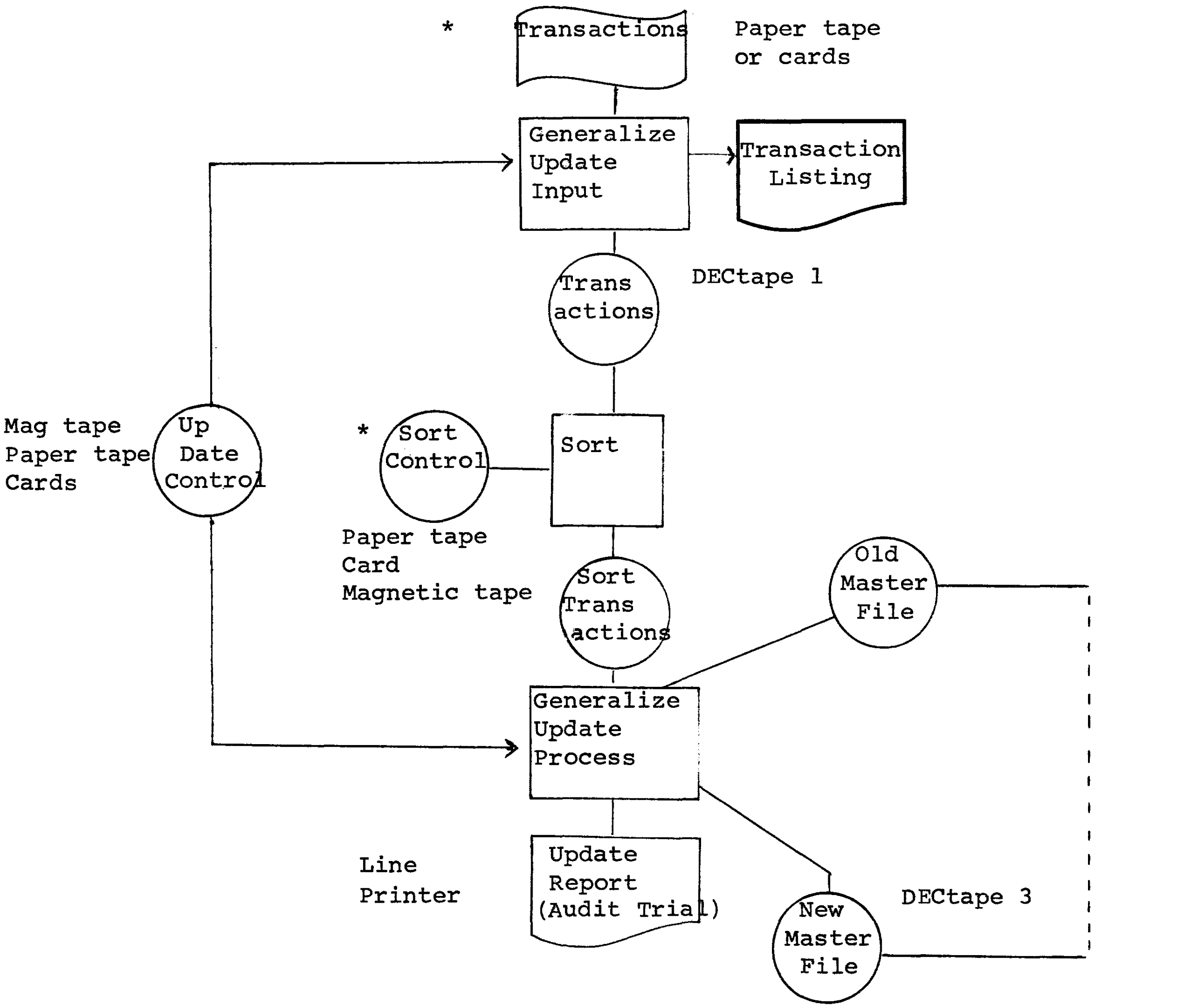
First, the Generalized Update Input Program is read into memory from the system tape by the monitor (this enables the computer to read data from paper tape). The update control, defining file parameters, is then read into memory before the transaction data. The transactions are then converted from paper tape to DECtape according to the update control.

Second, the sort program is read into memory by the monitor. The sort control program, which will define the sort arrangement, (refer to Sort Transaction) is then read from paper tape. The transaction DECtape would then be sorted according to the sort control. In our example, the transaction data is sorted by Commodity ID.

Third, the Generalize Update Process Program (enables computer to read DECtape data formats from transactions and old master, compare, write a new master, and generate update report) is read into memory. The update process program and the update control program will run the sorted transactions against the old master, producing a new master and an update report (audit trail).

As each transaction is read by the Generalize Update Input Program, it is listed and any transactions in error will have an error message printed to the left. Only valid transactions are written on the output tape.

The process chart for any generalized update is:



* The user would have to create the input transactions,the update control, and the sort control. The input transaction contains transaction code for insert or delete. The insert code is used either to insert a new item or change an existing item. Through these commands, the transactions are passed against a master file, resulting in an update master tape and a listing of transactions on the line printer (audit trail). The update transaction statements are limited to 50 characters per transaction. The update control defines the content of the file records and the format of the desired "audit trail" report. This control is read by both sections of the update (input and process) before execution.

Let's examine each of the programs generated by the user, i.e., update control and input transaction programs, sort and the audit trail report.

## 1. Update Control

The update control has three sections. The first section consists of one line and contains information as to the number of fields and the record size. The second section consists of four lines and gives the column headings for the update report. Finally, the third section has one line for each field on the inventory master and describes where the field is, how big it is, the data type, and where and if it should be printed on the update report.

The generalized update control for our inventory example would be:

```
07042001041                                         Section I
                                                    (Control line)
--------------------------------------------------
                  IDNT DESCRIPTION           /
ONHND    ORD        #YTD      MBAL    UCOST  /      Section II
                                             /      (Column headings)
--------------------------------------------------
0010410091
0051600151
0210410351                                          Section III
0250410411                                          (Field Specification)
0290410471
0330410531
0370610591
```

SECTION I--Control Line, 11 characters; for our example it is:

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 7 | 0 | 4 | 2 | 0 | 0 | 1 | 0 | 4 | 1 |

Characters — Explanation

1-2 (07) /The number of fields in the master file. In our example there are 7; i.e. IDNT DESCRIPTION, ONHND, ORD, #YTD, MBAL, UCOST, from the inventory master file.

3-5 (042) /Record size: Total number of characters for the field on the inventory master. The total in our example is 42 characters.

Characters — Explanation

6-8 (001) — /Process key base, starting position (character position of the sort key as it is on the Inventory Master File). In this example the sort key is commodity ID and it occupies first four characters in each master file record. Beginning position is 1.

9-10 (04) — /Number of characters in the process key (Commodity ID 4 characters).

11 (1) — /Process key type (0 = alpha, 1 = decimal). In our example commodity ID is a decimal number.

SECTION II

The column heading section consists of four lines with 132 characters of column heading. The first line contains character positions 1-33, the second line contains character positions 34-66, the third line contains character positions 100-132.

In the inventory example the column heading section is:

```
                IDNT DESCRIPTION         /
  ONHND    ORD     #YTD    MBAL    UCOST  /
                                          /
                                          /
```

All four lines must be there with a / in column 34.Presently we utilize only the first two lines because we only offer an 80-column line printer. (A 132 line printer will be available in the near future).

The column headings are only for those headings you want printed on the audit trail. These headings may correspond to all of the heading fields that are on the master file, or only a few. In our example we have all.

SECTION III

The field specification section contains one line for each field in the master file. In our inventory example there are 7 fields, therefore, 7 lines. Each line is 10 characters. The format for each line is:

Characters — Explanation

1-3 — /Field position; the first character position of the field as it is on each inventory master record.

| Characters | Explanation |
|---|---|
| 4-5 | /Field size; the number of characters in the field. |
| 6 | /Data type; 0 = alpha, 1 = decimal. |
| 7-9 | /Print left index; the first print position. This will be zeroes if this column is not printed. |
| 10 | /Field print switch; 0 means, do not print on the update report; 1 means print on update report. |

(If the file to be updated is fairly small, it will be possible to print all fields on the update report. However, if more than 132 characters are needed, it will be necessary to "non-print" some fields by giving them an 0 in character position 7-10 of the field specification section).

In our inventory example the 7 field lines are:

| Field base | Field Size | Data type | Print left Index | Field Print Switch | | |
|---|---|---|---|---|---|---|
| 001 | 04 | 1 | 009 | 1 | - | Commodity ID* |
| 005 | 16 | 0 | 015 | 1 | - | Description |
| 021 | 04 | 1 | 035 | 1 | - | Number on hand |
| 025 | 04 | 1 | 041 | 1 | - | Number on order |
| 029 | 04 | 1 | 047 | 1 | - | Number used YTD |
| 033 | 04 | 1 | 053 | 1 | - | Minimum Balance |
| 037 | 06 | 1 | 059 | 1 | - | Unit Cost |

*Line one is for the commodity ID. It says: the field base start at first position in inventory master, it's a 4-digit number, the data is decimal. On the update report, commodity ID will start in column 9 and it is to be printed on the update report.

## 2. Input Data Records

Now suppose the following records are to be updated: delete item 30, insert a new item number 125, change the unit cost on item 23, input would be prepared for the above changes

```
D  30
I  125  1  125
I  125  2  'HOOK, COPPER'
I  125  4  100
I  125  7  10000
I   23  7  28000
I   27  2  'BOLT, RED'
$
```

The I line is used either for an insert or change. It specifies; the process key (Commodity ID), the field number, and the data to be inserted. To change one item of an existing record, the I line is used with the process key (Commodity ID) of the existing record, the field to be changed, and the new data. To add a record, simply use an I line for each field on the inventory master, where the process key is the new record identification. To delete an item, simply use the D line with the appropriate process key.

The input data records have the following general format:

```
I  <process key>  <field number>  <data>
D  <process key>
```

In our inventory example:

```
  D  30                        /delete item 30
* I  125  1  125               /insert new item number 125 in ID field
* I  125  2  'HOOK,COPPER'     /insert HOOK,COPPER in description field
                                of item 125
* I  125  4  100               /insert 100 in order field of item 125
* I  125  7  1000              /insert 1000 in unit cost field of
                                item 125
  I  23   7  2800              /insert 2800 in unit cost field of
                                item 23 (this would be a change to an
                                existing item).
  I  27   2  'BOLT,RED'        /insert BOLT,RED in description field
                                of item 27.
```

* A new item number - 125 is inserted with the description, number on order, and unit cost. The other field, i.e. number on hand and #YTD would be zeroed since no data was inserted into these fields (see Audit Trail).

As each input transaction is read, it is listed, and any transaction in error will have an error message printed to the left. Only valid transactions are written on the output tape.

## 3. Update Sort

The sort control has already been discussed - refer back to explanation of SORT TRANSACTION. The only difference in the update sort is, First; Update records must be regarded as 50 characters long for the sort record input size (this is a maximum figure, and even though your update records may not be that long, 50 is still the number used in the sort control) and Second; the field sorted must always be regarded as beginning in Position one(1).The user does not have to concern himself with the update record Format on DECtape - he need only remember the two above rules.

The update sort program puts the transactions (Insertions, Changes, Deletions) into the order of the appropriate master file to be updated. In our example the inventory master is sorted by commodity I.D.

The sort control for our update would be

```
026130010460
0010200
$
```

First Line

1-3 (026) - Record input size, normal practice is to use 50 for update record size (Rule 1)
50 + 1 = 51 ÷ 2 = 25 + 1 = 26
4-7 (1300)- Internal sort buffer size
1300 = 26 x 50 (result must be less than 2000)
8 (1) - Number of sort fields
9-12 (0146)- Number of blocks per DECtape
190,000 ÷ 1300 = 146
13 (0) - Phase 2 input save switch

Second Line

1-3 (001) - First position of sort field. This will always be Position 1 (Rule 2).
4-5 (02) - Number of words spanned
6 (0) - First word switch
7 (0) - Last word switch

4. Update Report (Audit Trail)

After the generalize update process program and the update control have been loaded into core, they will run the sorted transaction against the old master, producing a new master and an update report on the line printer. The update report in our example would be:

```
        IDNT    DESCRIPTION        ONHND   ORD     #YTD    MBAL    UCOST

OLD     0023    DOWEL, RED          0079   0047    0203    0054    026532
NEW     0023    DOWEL, RED          0079   0047    0203    0054    028000

OLD     0027    BOLT, BROWN         0144   0033    0802    0022    016564
NEW     0027    BOLT, RED           0144   0033    0802    0022    016564

DLT     0030    FLANGE, PURPLE      0081   0050    0221    0076    049909

NEW     0125    HOOK, COPPER        0000   0100    0000    0000    001000
```

Notice that the old item for the deleted record is displayed. For the inserted item, 125, the new master record is printed. For each of the changed items (23 and 27), both the old and the new master records are printed.

*******************

This example has created a complete application package for inventory control. All of the program and control statements for the entire application appear in this memo. Not one additional line of code is needed. Even relatively inexperienced programmers can quickly learn to produce additional applications--billing, sales analysis, payroll, etc. DIBOL and the PDP-8 can bring you into the world of data processing, quickly and at an extremely low cost.

# APPENDIX

## Brief Description of the DIBOL Software System

The DIBOL Software System contains a DIBOL language, Generalize Input Facility, Sort Package, Generalized Update Facility and a DIBOL Monitor. The following is a summary of the features of the system:

### 1. DIBOL Language

Each DIBOL program is divided into two sections - the data section and the procedure section. The data section is used to describe the information used in the program and to allocate memory. The procedure section contains the executable statements.

The data section is broken up into a number of "blocks", each of which contains a number of "data elements". An example is:

```
BLOCK MASTER
        CUSTNO, D4
          NAME, A30
        AMOUNT, D8
BLOCK WORK, C
          TEMP, A10
         TABLE, 6D5
```

In the example, the data is contained in two blocks; "master" and "work". The first block has 42 bytes and is divided into data elements "customer number", "name", and "amount". The customer number is specified to be decimal (D) and to contain four bytes. The name is specified alphanumeric (A) with thirty bytes. The second block is specified with the clear option (C). This implies that a load time, TEMP will be filled with alphanumeric blanks and TABLE with decimal zeros. TABLE is specified to be an array of six elements, each of which contains five decimal bytes.

The procedure section contains statements to control input/output, compute arithmetic expressions, move data and control the program sequence of execution. Examples of each of the ten types of DIBOL statements are:

```
 1)   INIT (1, V, OUT)
 2)   XMIT (1, MASTER)
 3)   FINI (1)
 4)   PRINT (21, 45) = NAME
      PRICE = LIST * (100 - DISCT) / 100
      PRINT (60, 70) = AMOUNT, 'XXX,XXX.XX-'
 5)   BUFF =
 6)   GO TO LOOP
      GO TO (MASS, CONN, RI, NY), TAXCD
 7)   IF (CODE1 _ CODE2 .NE. 0) GO TO PRINT
      IF (ONHAND .LT. MINBAL) GO TO ORDER
 8)   CALL INPUT
 9)   RETURN
10)   STOP
```

The initialization statement (1) opens file-1 as an output file. The transmit statement (2) sends data from the block "master" to file-1. The finalization statement (3) performs the necessary operations to close file-1.

There are three examples of the data manipulation statement (4). In the first, the contents of "name: are moved into character positions 21 through 45 of the data element "print". In the second, a selling price is computed by marking down a "list price" by a "discount percentage" . The last example is similar to the first in that data is moved into specified positions or "print". The difference is that the move is formatted with a comma for the thousands digit, a decimal point for the units digit and a minus sign if "amount" is negative. Statement type (5) clears the contents of the specified data element. Decimal elements are set to zero and alphanumeric elements to blank.

Statement type (6) is used for conditional and unconditional program branches. In the first example, control goes to the statement with the tag LOOP. In the second example, control goes to MASS,CONN, RI or NY depending on whether "tax code" is one, two, three or four.

Statement type (7) may also be used for conditional branching. In the first example, control goes to "print" if either code-1 or code-2 is not equal to zero. Otherwise, control goes to the next statement. In the second example, control goes to "order" if the number "on hand" is less than the "minimum balance".

The CALL statement (8) transfers control to the first statement of a closed subroutine. The RETURN statement (9) transfers control back to the statement after the last executed CALL statement. Finally, the STOP statement (10) causes a return to the DIBOL monitor.

## 2. Generalized Input Facility

The Generalized Input Facility provides a simple means for getting data into the computer. The user specifies his data fields, the desired output record contents and the input line formats. The Generalized Input Facility reads the input data, converts it to the specified output formats and performs careful error checking.

The format of the Generalized Input control is:

| | | | | |
|---|---|---|---|---|
| A | field number | type count | optional constant | |
| B | record number | file number | | |
| C | field number | | | |
| D | command | record number | zero field | increment field |
| E | field number | necessary switch | present field | default field |

## 3. Sort Package

The sort package allows the user to arrange a data file in any specified order. The user defines the records to be sorted and gives "sort keys" within the record. The sort runs in two phases. In the first phase (the internal sort), as many records as possible are read into memory and sorted. The sorted groups of records are written as strings onto an intermediate file. In the second phase (the poly-phase merge), the original strings are successviely made longer by merging until there is only one string remaining. This last string is the sorted output file. A third phase of the sort ( a general single-pass merge) is used only in the case of multi-reel files.

The format for the sort control is:

| | | | | |
|---|---|---|---|---|
| record size | buffer size | number fields | buffers/tape | tape save |
| field base | field size | first word split | last word split | |

## 4. Generalized Update Facility

The Generalized Update Facility provides a simple means for the user to perform insertions, changes and deletions on existing data files of arbitrary format. The user defines the content of the file records and the format of the desired "audit trail" report. He prepares the transaction input referring to records by "process key" and to individual fields by field number. DIBOL then performs all necessary inputting, sorting and updating to apply the transactions to the old master and produce a new master. As a by-product, the audit trail report shows which records have been deleted or inserted and which fields of existing records have been changed.

The format of the Generalized Update Control is:

```
number fields     record size    key base    key size    key type
report heading 1-33
report heading 34-66
report heading 67-99
report heading 100-132
field base   field count   data type   print left index   print?
```

The general format for the user transactions are:

```
I   process key    field number    data
D   process key
```

## 5. DIBOL Monitor

The DIBOL monitor ties the various DIBOL facilities together into a unified data processing system. By means of a simple interactive dialog, the monitor executes the subsystems according to operator commands. The commands to call subsystems are:

| Command | Subsystem |
|---|---|
| COMP | DIBOL Compiler |
| RSYS | Run Time System |
| SRT1 | Sort Phase I |
| SRT2 | Sort Phase II |
| SRT3 | Sort Phase III |
| GENI | Generalized Input Facility |
| GUPI | Generalized Update Facility-input |
| GUPP | Generalized Update Facility-process |
| EDIT | DIBOL Editor |
| DUMP | Tape Dump |
| CDDT | Card to DECtape utility |
| DTPR | DECtape to printer utility |
| PTDT | Paper Tape to DECtape utility |
| DTFX | DECtape data edit utility |

The DIBOL compiler (COMP) takes it input from a source master file on DECtape. Its output goes onto an updated object master, also on DECtape. The run time system (RSYS) executes DIBOL programs which have been compiled. The three phases of the sort may be called by the commands SRT1,SRT2, and SRT3. The Generalized Input Facility (GENI), when loaded into memory, asks the user for the name of the control. After the user responds, the control is fetched from the system DECtape and the input process begins.

The Generalized Update Facility is operated in three phases--input, sort and process. The input phase (GUPI) reads in the transactions in an operation similar to that of the Generalized Input Facility. Then the sort (SRT1,SRT2 and possibly SRT3) is used to order the transactions on master file order. Finally, the process phase (GUPP) does the file update.

The DIBOL Editor (EDIT) provides a source program file maintenance capability. It is used for the creation and editing of DIBOL source programs, sort controls, generalized input controls and generalized update controls. It is also used to perform maintenance of the DIBOL system tape.

The tape dump (DUMP) is used to print any DIBOL generated data DECtape. It gives record count, record size and all data printed in both alphanumeric and decimal formats. The Card to DECtape utility (CDDT) creates DECtape files of alphanumeric card images. The DECtape to Printer utility (PTDT) is the same as CDDT, except the input comes from paper tape. The DECtape Fix utility (DTFX) provides a simple means of checking and/or changing any field within any record of a DIBOL generated data tape.

* * * * * * * * * * * * * * * * * * * * * * * * *

The DIBOL Software System provides a simple, but complete capability for commercial applications on the PDP-8 family of computers. It is the most easily learned total data processing package in the industry. DIBOL is ready to solve the data processing problems of small to medium sized businesses.

Digital Equipment Corporation
Maynard, Massachusetts

printed in U.S.A.