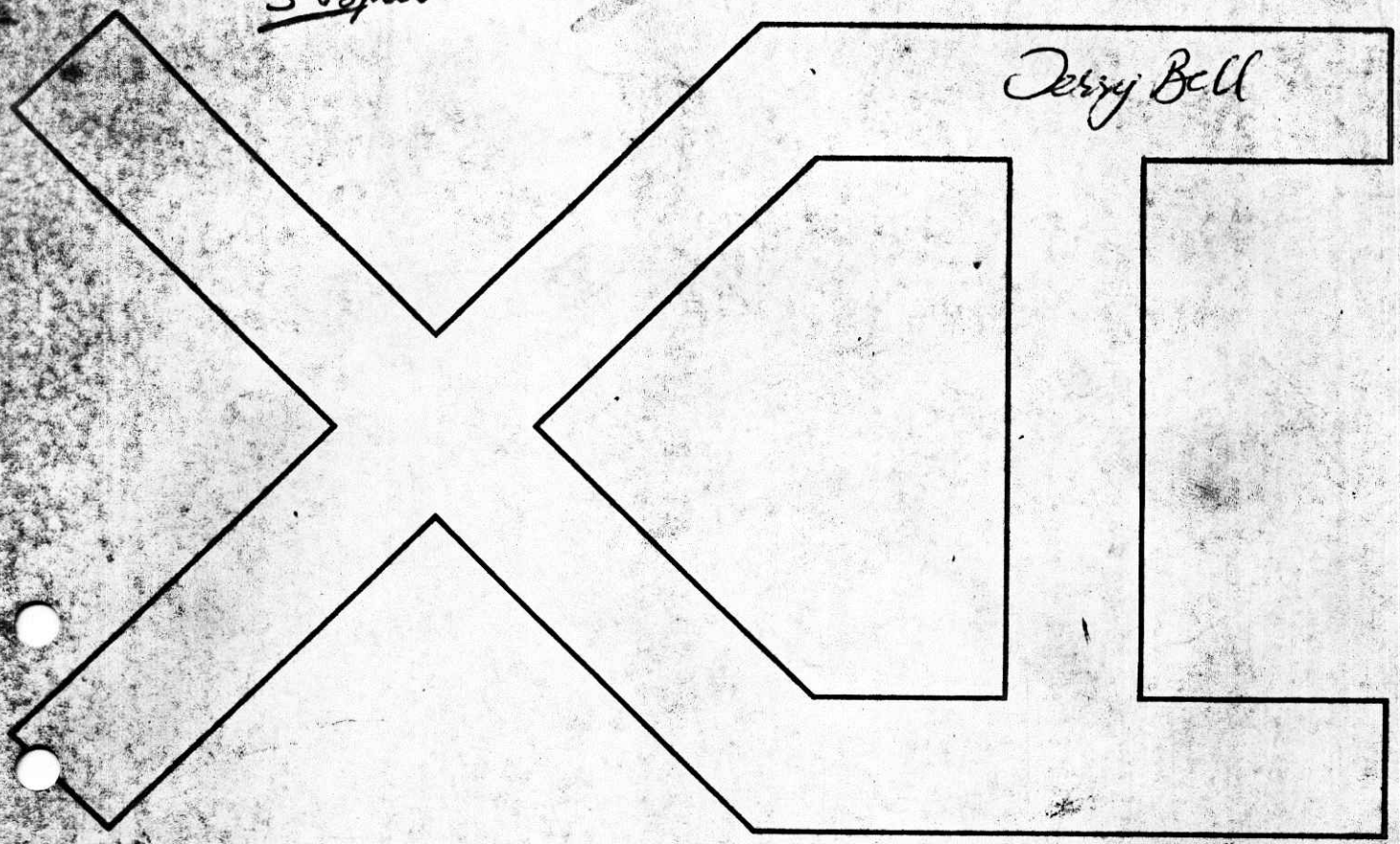
5 Copies

Jerry Bell



MONROBOT XI

THEORY OF ALPHANUMERIC INPUT/OUTPUT

COPYRIGHT 1964 BY MONROE INTERNATIONAL, INC.
ALL RIGHTS RESERVED

MQ-253 ED 2-64
PRINTED IN U.S.A.

ALPHANUMERIC FUNCTIONS

In almost any data processing system the programmer is primarily concerned with the handling of numeric values and "getting the right answers." However, a payroll check cannot be printed without an employee's name, an invoice or statement cannot be mailed without a name, street address, city and state. Even the mathematician likes to have titles above his columns of figures. The world outside the computer speaks a language recorded in alphabetic and numeric symbols. While numerics record values, the alphabetics record identities. Very few computer programs can be written without the handling of some alphanumeric information.

Short Titles and Column Headings

The programmer's simplest problem is the output of short, unvarying fields such as DATE, TOTAL, TAX, etc. The Control (or Instruction) Register output command of the Monrobot XI will prove to be the simplest and most efficient means for such output. The word TAX may be printed with three instructions:

S313	Output a T to device #1
S331	Output an A
S357	Output an X

This technique is so simple it need not be discussed further, it is left to the programmer when it is best to employ it.

ALPHANUMERIC INPUT/STORAGE/OUTPUT

It is assumed that the programmer is thoroughly familiar with the various techniques of numeric input and output. If so, the programming of alphanumeric input, storage, and output will be relatively simple. The problems which arise are:

1. Input followed by immediate output of an alphanumeric field until a designated end code is detected.

2. Input and storage of an alphanumeric field.

3. Output from storage of this alphanumeric field at a later time.

These are the basic problems and on the surface they seem exceedingly simple. Indeed, problem #1 is that, as illustrated on a following flow chart and coded example. However, some minor complications which must be considered are:

4. In exactly what manner is alphanumeric information to be stored in the computer?

5. Is it not necessary to provide for selection of input devices?

6. How is an area in memory reserved for storage of alphanumeric fields? What if more than one field must be input before output of the first field takes place?

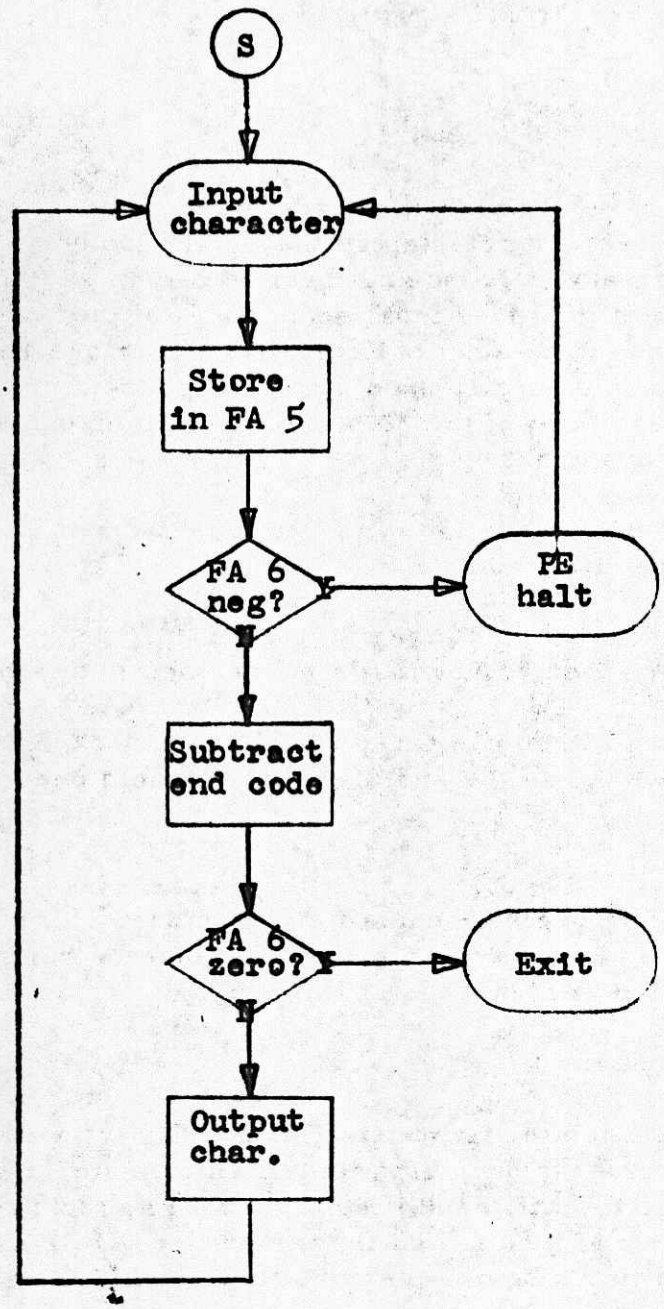
7. What if it is necessary to input, store, and also print the alphanumeric field at the time of input?

8. Is it not necessary to provide for selection of output devices?

9. Is it necessary to provide for ending an input field on more than one end code, or to be able to select end codes?

ALPHANUMERIC INPUT/OUTPUT UNTIL END CODE

PROBLEM #1



THE "PACKED WORD"

Problem #4

To preserve identities each alphabetic (and numeric) character must be allotted a specified area in a register so that it may be contained without altering any other character. Since no legal alphanumeric character requires more than 6 bits it proves most economical of storage to "pack" at least five characters into each register. (The carriage return code which requires 8 bits is normally used as an end of input code and is excepted). This will leave two of the 32 bits of a register unused and they can be employed to advantage.

If five alphanumeric characters (30 bits) are to be packed into a register it is necessary to know two things:

1. During input, when has a register been fully packed with five characters? (Remember that a space code is 6 zero bits.)

2. During output, when has a register been unpacked with more to follow and when has the end of the output field been reached?

The Flag Bit

The two unused bits in a register containing alphanumeric information can be used during input and output to help identify when the register has been packed or unpacked and to signal the end of a field.

It is done in this manner:

A virgin packed word register is prepared with a 2's bit acting as a flag placed in the low order. As each character is input, stored and tested this packed word register will be shifted left six bits, the character added to it, the word stored and then tested for the presence of a high order bit. This will happen five times before the condition is met.

If the end character has not been input and detected the packed word register will be shifted left two bits (dropping the input flag) and a 1's bit placed in the low order to indicate end of packed word. It is then stored and another virgin packed word set up.

When the end character has been detected it is not packed, rather the packed word is shifted left two bits and a 2's bit added to the low order to indicate end of field. The word will then be shifted left four bits and tested for the input flag. It will then be shifted left by six bits until the condition is met

and then two more bits to drop the input flag. It is then stored. This manner of storing information in a register is called left justification, ie, it is packed against the high order of the register.

During output FA 5 is cleared and the first packed word is loaded into FA 6. After that successive 6 bit shifts will place each character in FA 5, FA 6 is tested for zero and if the condition is not met the character is output. When the condition is met the low order flag bit will have been shifted into FA 5. This flag is brought into FA 6 and by shifting off the correct number of bits to remove the "end of word" flag it can be determined whether or not additional words are to be unpacked and output or whether the output field has been terminated. In the latter case FA 6 will not be zero.

The following diagram shows graphically how alphanumeric characters are packed into a register.

Alpha Pack Progression

	T7	T6	T5	T4	T3	T2	T1	T0
Flag virgin							·	0010
1st character							10	A
2nd character					0010		A	L
3rd character				10	A		L	P
4th character		0010		A	L		P	H
5th character	10	A		L	P		H	A
Stored as EOW	A		L		P		H	A 01
Stored as EOF	A		L		P		H	A 10

Alpha Unpack Progression

	T7	T6	T5	T4	T3	T2	T1	T0
Stored word	A		L		P		H	A 01
1st character	L		P		H		A	0100
2nd character	P		H		A	01		
3rd character	H		A		0100			
4th character	A	01						
5th character	0100							
Flag to FA 5								
Load FA 5								0010 EOW flag 0100 EOF flag
BSR 2 for EOW								FA 6 = zero if EOW
								0001 FA 6 ≠ zero if EOF

EOW = end of word
EOF = end of field

Problem #5 Selection of Input Device

Alphanumeric fields usually appear in only one input source in a program and a great deal of flexibility is not required. When necessary a virgin input instruction with the correct device code can be set into the program.

Problem #6 Storage of Alphanumeric Fields

An area must be reserved in memory which can contain the maximum number of characters which may appear in each field. A field is usually defined as a line of alpha information, ie, customer or company name, street address, city and state, would each be considered a field.

Problem #7 Input, print, and store alpha

Each input character must be stored while checking to determine if is the end code character. If FA 5 is used for this purpose an output instruction can be substituted for a no operation instruction and each character except the end code may be output.

Problem #8 Selection of Output Devices

It is often necessary to "turn on or off" devices for alpha output. This is done in the same manner as input devices by insertion of the correct virgin output instruction in the output subroutine.

Problem #9 Selection of End Codes

A single end code, usually the carriage return code, is usually adequate to signal the end of an alpha field. In some instances two codes may be required, one to indicate end of alpha field, the other to indicate end of all alpha input. This would be a possibility to permit variation from two to four lines of name and address fields.

SUBROUTINE FOR ALPHANUMERIC INPUT AND STORE WITH PRINT OPTION

The flow charts and coding on the following pages illustrate a completely flexible alphanumeric input and store subroutine. Any one of three input devices may be selected; input will terminate on a tab or carriage return code; the tab end code may be changed to any other code desired; each input character except the end code may be output to selected devices if required. This subroutine should prove adequate for most purposes.

Operating Instructions

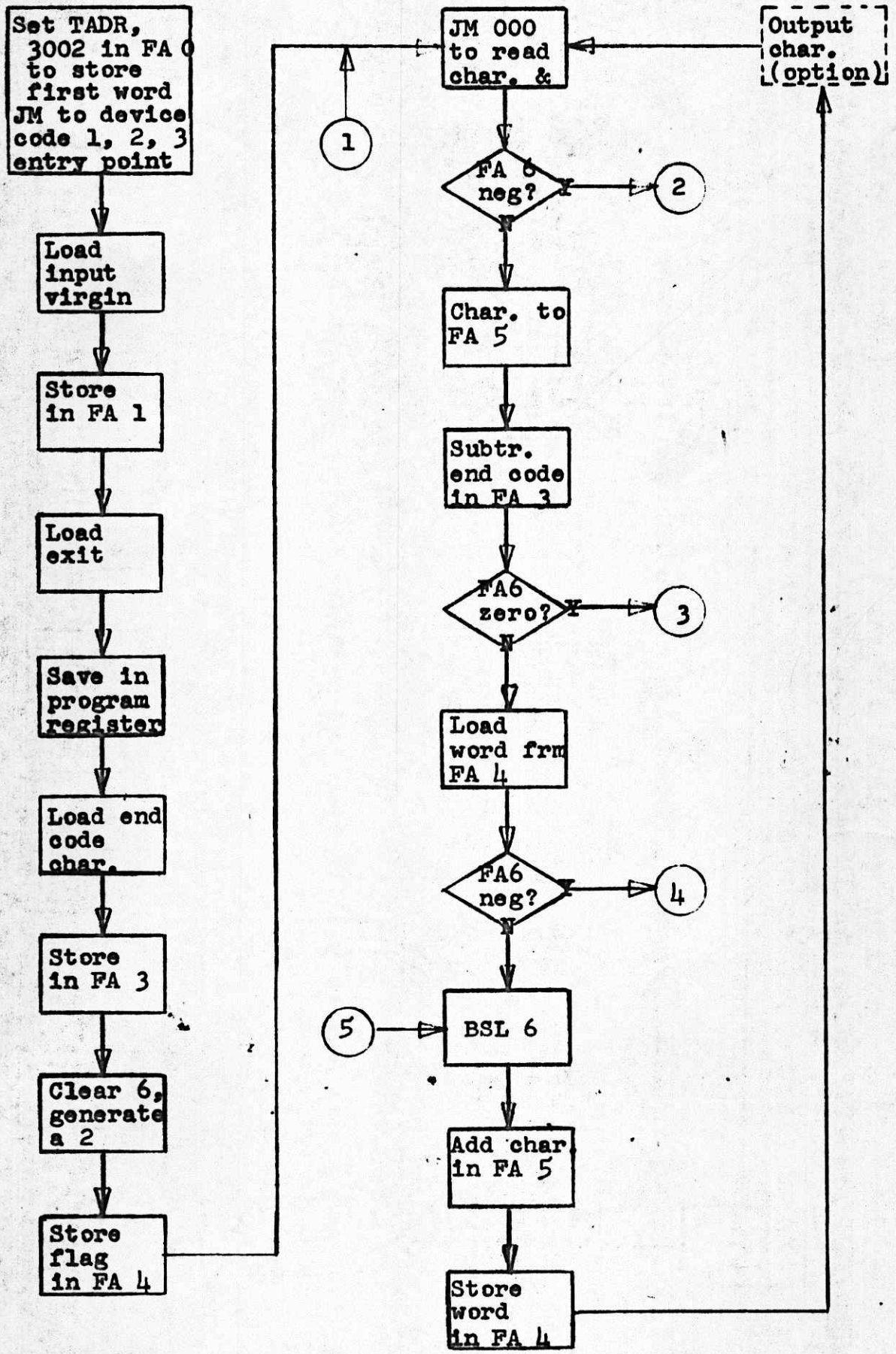
On entry to the subroutine FA 0 must contain TADR, 3002 where ADR is the first location of the alphanumeric storage table. Entry to the subroutine is by a jump mark instruction to one of three points selecting input from device 1, 2 or 3. The input field is terminated by a tab or a carriage return code.

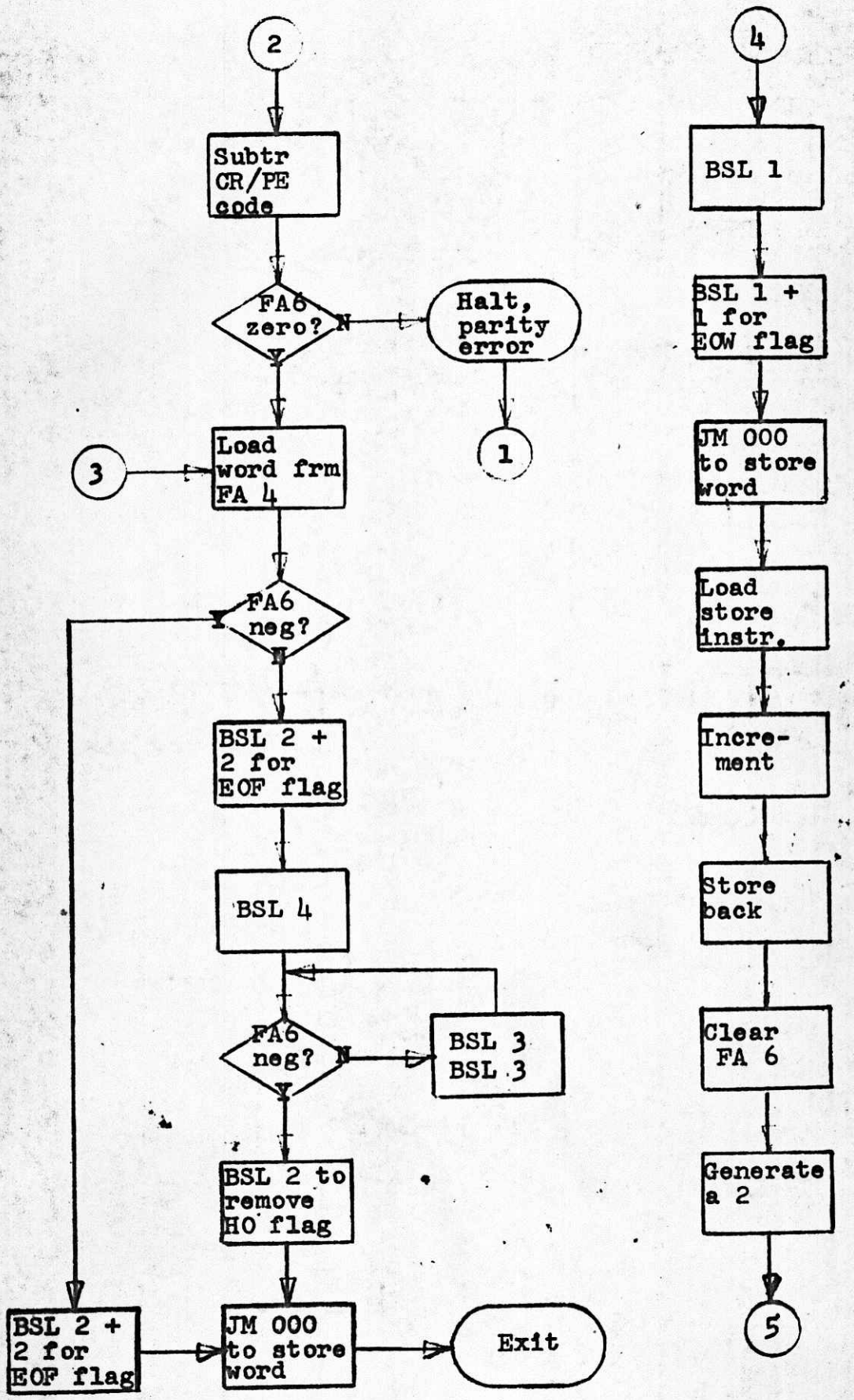
Five alphanumeric characters plus an end of word or end of field code may be stored in a register. An area of table registers large enough to hold the maximum number of input characters must be provided by the programmer.

At the option of the programmer input characters may be output to selected devices. Also, a value other than the tab code may be substituted to terminate the input field.

ALPHANUMERIC INPUT AND STORE WITH OUTPUT OPTION

PROBLEM #2




2
Subtr CR/PE code
FA6 zero?
N
Halt, parity error
1
3
Load word frm FA 4
FA6 neg?
BSL 2 + 2 for EOF flag
BSL 4
FA6 neg?
N
BSL 3 BSL 3
BSL 2 to remove HO flag
BSL 2 + 2 for EOF flag
JM 000 to store word
Exit
4
BSL 1
BSL 1 + 1 for EOW flag
JM 000 to store word
Load store instr.
Incre-ment
Store back
Clear FA 6
Generate a 2
5

MONROBOT XI PROGRAM SHEET

PROGRAM

Alphanumeric Input and Storage with Print Option

PAGE	DATE	PROGRAMMER
050		

REGISTER		CONTENTS	NOTES
050	A	V06V	Load Input Dev. 1 and parity test instruction
	B	3053	Jump to continue
051	A	V06W	Load Input Dev. 2 and parity test instruction
	B	3053	Jump to continue
052	A	V06X	Load Input Dev. 3 and parity test instruction
	B	3053	No-op
053	A	T001	(050)(051) Store instruction in FA 1
	B	V002	Load exit
054	A	T06T	Save exit
	B	V06U	Load Constant 0000 001W
055	A	T003	Store end code
	B	U500	Clear FA 6 to 0
056	A	8201	Generate a binary 2
	B	T004	Store in FA 4
057	A	3401	(05U) Jump mark to enter 1 character (220, 705V)
	B	T005	Store character
058	A	W003	Subtract end code
	B	6064	0 = end code
059	A	V004	Load word
	B	705X	Jump negative if word is packed
05S	A	9020	(063) Shift word up 6 bits
	B	X005	Add latest character
05T	A	T004	Store new word
	B	6400	No-op (or print)
05U	A	3057	Jump to enter next character
	B	6400	Fill
05V	A	W070	Subtract Constant 8000 0081
	B	6064	0 = carriage return
05W	A	0041	Halt, parity error
	B	3001	Jump to enter character again
05X	A	9001	(059) Left 1 bit on packed word
	B	9401	Generate a binary 1

MONROBOT XI PROGRAM SHEET

PROGRAM

Alphanumeric Input and Storage with Print Option

PAGE	DATE	PROGRAMMER
060		

REGISTER		CONTENTS	NOTES
060	A	3400	Jump mark to store packed word
	B	V000	Load drum store instruction = TADR 3002
061	A	X071	Add Constant 0001 0000
	B	T000	Store incremented instruction
062	A	U500	Clear FA 6 to 0
	B	8201	Generate a binary 2
063	A	305S	Jump to continue
	B	6400	Fill
064	A	V004	(058)(05V) Load word
	B	7068	Jump negative if word is packed
065	A	8W02	Left 2 bits + 2
	B	9008	Shift word up 4 bits
066	A	706S	(067) Jump negative if final word is packed
	B	9004	Shift final word up 3 bits
067	A	9004	Shift final word up 3 bits
	B	3066	Jump to test for packed word
068	A	8W02	(064) Left 2 bits + 2 on word
	B	3400	Jump mark to store final packed word
069	A	306T	Jump to exit
	B	6400	Fill
06S	A	9002	(066) Left 2 bits on word
	B	3400	Jump mark to store final packed word
06T	A		Saved exit
	B		
06U	A	0000	Constant: tab end code
	B	001W	
06V	A	2201	Input Dev. 1 and parity test virgin instructions
	B	705V	
06W	A	2401	Input Dev. 2 and parity test virgin instructions
	B	705V	
06X	A	2801	Input Dev. 3 and parity test virgin instructions
	B	705V	

MONROBOT XI PROGRAM SHEET

PROGRAM

Alphanumeric Input and Storage with Print Option

PAGE	DATE	PROGRAMMER
070		

REGISTER		CONTENTS	NOTES
070	A	8000	Constant: Carriage Return with forced parity
	B	0081	
071	A	0001	Constant: Address increment
	B	0000	
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		

ALPHANUMERIC INPUT AND STORAGE WITH PRINT OPTION

```
ENTRY - FA 0  MUST CØNTAIN    T    NAME
                              J    002
        WHERE  NAME  EQUALS THE FIRST ADDRESS ØF STØRAGE TABLE.

EXIT -  TERMINATES ØN TAB ØR CARRIAGE RETURN.

ØP CØDE - JM  RA 1, RA 2, ØR RA 3.

CHANGES PRIØR TØ ENTRY -
TØ ØUTPUT ALL CHARACTERS EXCEPT END CØDE
        CA    SLASH H  T004 SD7X
        T     RAP

TØ CHANGE TAB END CØDE
        CA    SLASH  H ØR F      END CØDE VALUE
        T     RAEC


THE PRØGRAMMER MUST PRØVIDE A STØRAGE AREA FØR EACH ALPHANUMERIC
        FIELD TØ BE STØRED SUCH AS
              RAF1    GG          FIRST FIELD
                      GG          PRØVIDING FØR
                      GG          20 CHARACTERS
                      GG          WITH ENDING BITS.

THE CØNSTANT FØR FA 0  MAY BE WRITTEN
              RAK1    T    RAF1
                      J    002

REQUIRES ALL FAST ACCESS REGISTERS.


RA 1          CA   RAEC+1        INPUT VIRGINS
              J    RA 3+1

RA 2          CA   RAEC+2
              J    RA 3+1

RA 3          CA   RAEC+3
              J    RA 3+1

  1.          T    001           INPUT AND PARITY TEST
              CA   002
  2.          T    RAP+16        SAVE EXIT
              CA   RAEC          END CØDE
  3.          T    003
              C    006           CLEAR FA 6
  4.          G    8201          GENERATE A BINARY 2
              T    004

  5.          JM   001           ENTER 1 CHARACTER
              T    005           SAVE CHARACTER
```

```
      6.    S    003          SUBTRACT END CØDE
            JZ   RAP+9        END CØDE
      7.    CA   004          LØAD WØRD
            JN   RAP+4        TEST FØR PACKED WØRD

      8.    L    6            LEFT 6 BITS
            A    005          ADD LATEST CHARACTER

RAP         T    004          STØRE WØRD
            N
      1.    J    RA 3+5       TØ ENTER NEXT CHARACTER
            N

      2.    S    /H 8000 0081
            JZ   RAP+9        CARRIAGE RETURN ENTERED
      3.    G    41           HALT, PARITY ERRØR
            J    001          TØ ENTER AGAIN

      4.    L    1            LEFT 1 ØN PACKED WØRD
            G    9401         GENERATE A 1
      5.    JM   000          TØ STØRE PACKED WØRD
            CA   000          DRUM STØRE INSTRUCTIØN
      6.    A    /H 1 0000
            T    000          STØRE INCREMENTED INSTRUCTIØN
      7.    C    006          CLEAR FA 6
            G    8201         GENERATE A BINARY 2
      8.    J    RA 3+8       TØ CØNTINUE
            N

      9.    CA   004          LØAD WØRD
            JN   RAP+13       TEST FØR PACKED WØRD
     10.    G    8W02         LEFT 2 BITS +2
            L    4            LEFT 4 BITS

     11.    JN   RAP+15       TEST FØR PACKED WØRD
            L    3
     12.    L    3
            J    RAP+11       TØ PACK FINAL WØRD

     13.    G    8W02         LEFT 2 BITS + 2
            JM   000          TØ STØRE PACKED WØRD
     14.    J    RAP+16       EXIT
            N

     15.    L    2            LEFT 2 BITS
            JM   Q00          TØ STØRE FINAL WØRD

     16.    GG                SAVED EXIT

RAEC        GG   1W           END CØDE - TAB

      1.    G    2201         INPUT DEVICE 1
            JN   RAP+2

      2.    G    2401         INPUT DEVICE 2
            JN   RAP+2

      3.    G    2801         INPUT DEVICE 3
            JN   RAP+2
```

SUBROUTINE FOR ALPHANUMERIC OUTPUT FROM STORAGE

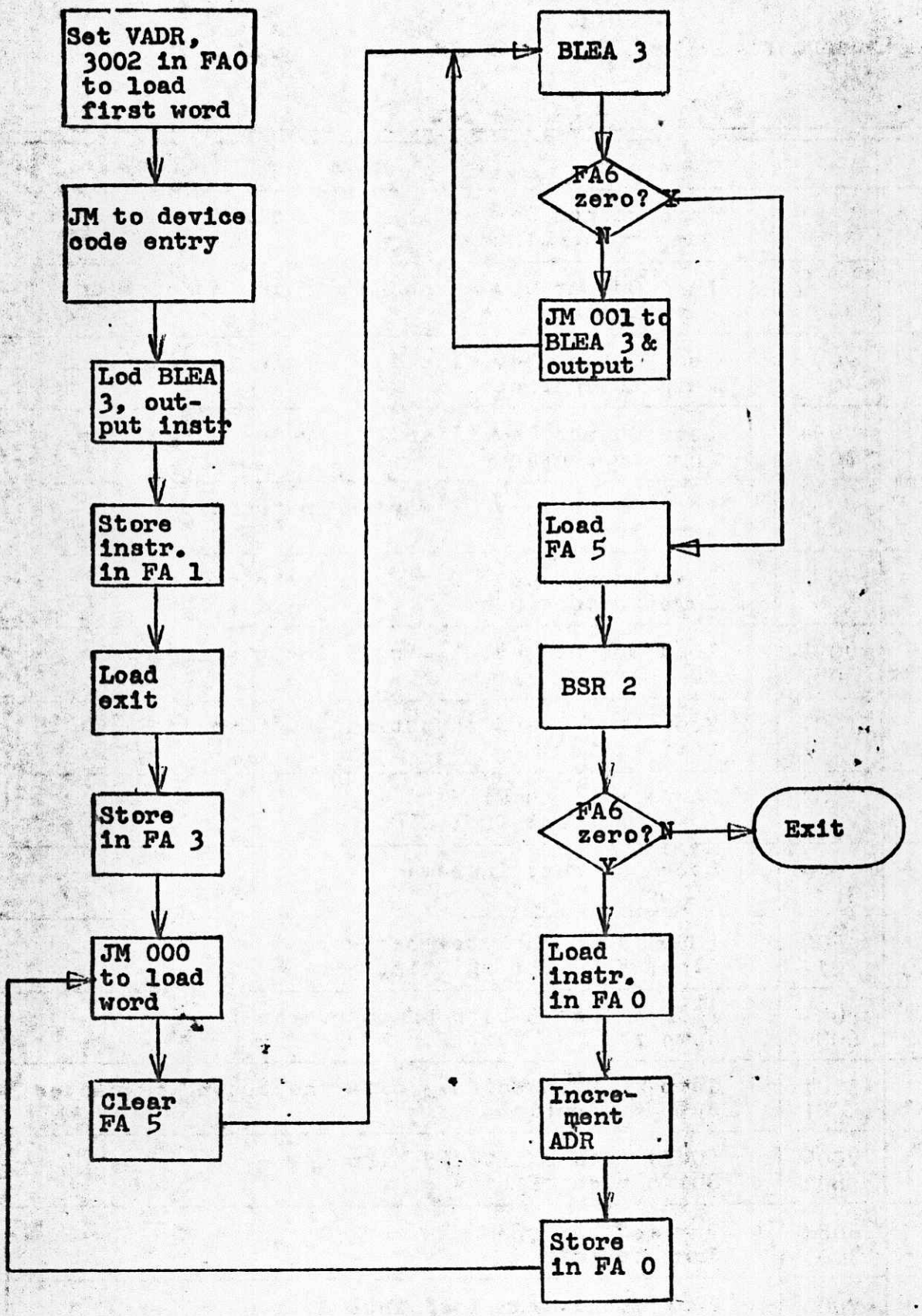
The flow charts and coding on the following pages operate in conjunction with the alpha input subroutine previously illustrated. It permits selection of any output device or combination of devices.

Operating Instructions

On entry to the subroutine FA 0 must contain VADR, 3002 where ADR is the first location of the alphanumeric storage table. Entry to the subroutine is by a jump mark instruction to one of five entry points for output device selection.

ALPHANUMERIC OUTPUT FROM STORAGE

PROBLEM #3



MONROBOT XI PROGRAM SHEET

PROGRAM

Alphanumeric Output from Storage

PAGE	DATE	PROGRAMMER
080		

REGISTER		CONTENTS	NOTES
080	A	V097	Load Output Dev. 1, 2, and 3 virgin instruction
	B	3087	Jump to continue
081	A	V096	Load Output Dev. 2 and 3 virgin instruction
	B	3087	Jump to continue
082	A	V095	Load Output Dev. 1 and 3 virgin instruction
	B	3087	Jump to continue
083	A	V094	Load Output Dev. 1 and 2 virgin instruction
	B	3087	Jump to continue
084	A	V093	Load Output Dev. 3 virgin instruction
	B	3087	Jump to continue
085	A	V092	Load Output Dev. 2 virgin instruction
	B	3087	Jump to continue
086	A	V091	Load Output Dev. 1 virgin instruction
	B	3087	No-op
087	A	T001	Store shift and output instruction in FA 1
	B	V002	Load exit
088	A	T003	Store exit in FA 3
	B	V098	Load Constant 0001 0000
089	A	T004	Store address increment
	B	308S	No-op
08S	A	3400	(090) Jump mark to load word
	B	S07X	Clear FA 5 for shifting
08T	A	8U04	(08U) First 3 bits of character to FA 5
	B	608V	Jump zero to test for end
08U	A	3401	Jump mark to shift 3 bits and output character
	B	308T	Jump to continue
08V	A	V005	(08T) Load shifted 3 bits
	B	9802	Shift right 2 bits
08W	A	608X	0 = end of word
	B	3003	Jump to exit
08X	A	V000	(08W) Load drum load instruction
	B	X004	Add Constant 0001 0000

MONROBOT XI PROGRAM SHEET

PROGRAM

Alphanumeric Output from Storage

PAGE	DATE	PROGRAMMER
090		

REGISTER		CONTENTS	NOTES
090	A	T000	Store incremented drum load instruction
	B	308S	Jump to load next word
091	A	8U04	Output Dev. 1 virgin instruction
	B	S27X	
092	A	8U04	Output Dev. 2 virgin instruction
	B	S47X	
093	A	8U04	Output Dev. 3 virgin instruction
	B	S87X	
094	A	8U04	Output Dev. 1 and 2 virgin instruction
	B	S67X	
095	A	8U04	Output Dev. 1 and 3 virgin instruction
	B	SS7X	
096	A	8U04	Output Dev. 2 and 3 virgin instruction
	B	SU7X	
097	A	8U04	Output Dev. 1, 2 and 3 virgin instruction
	B	SW7X	
098	A	0001	Constant: address increment
	B	0000	
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		

ALPHANUMERIC OUTPUT FROM STORAGE

```
ENTRY -  FA 0  MUST CØNTAIN   CA   NAME
                              J    002
         WHERE  NAME  EQUALS THE FIRST ADDRESS ØF STØRAGE TABLE.

ØP CØDE -  JM  WA 1, WA 2, WA 3, WA 12, WA 13,
               WA 23, ØR WA 123.

THE CØNSTANT FØR  FA 0  MAY BE WRITTEN
          WAK1        CA   RAF1
                      J    002

REQUIRES ALL FAST ACCESS REGISTERS.

WA 123        CA   WA 1+17      ØUTPUT VIRGINS
              J    WA 1+1

WA 23         CA   WA 1+16
              J    WA 1+1

WA 13         CA   WA 1+15
              J    WA 1+1

WA 12         CA   WA 1+14
              J    WA 1+1

WA 3          CA   WA 1+13
              J    WA 1+1

WA 2          CA   WA 1+12
              J    WA 1+1

WA 1          CA   WA 1+11
              J    WA 1+1

  1.          T    001          STØRE  LC 3  ØR D
              CA   002
  2.          T    003          SAVE EXIT
              CA   /H 1 0000
  3.          T    004          ADDRESS INCREMENT
              J    WA 1+4

  4.          JM   000          TØ LØAD WØRD
              £    005          CLEAR FØR SHIFT

  5.          LC   3            BINARY LEFT END ARØUND 3
              JZ   WA 1+7       TØ TEST FØR END ØF WØRD
  6.          JM   001          TØ ØUTPUT.
              J    WA 1+5

  7.          CA   005
              R    2            RIGHT 2 BITS
```

```
8.      JZ    WA 1+9      0 EQUALS END ØF WØRD
        J     003         EXIT

9.      CA    000         DRUM LØAD INSTRUCTIØN
        A     004         ADD 0001 0000
10.     T     000         STØRE INCREMENTED INSTRUCTIØN
        J     WA 1+4      TØ LØAD NEXT WØRD

11.     LC    3
        ØR    2           ØUTPUT DEVICE 1

12.     LC    3
        ØR    4           ØUTPUT DEVICE 2

13.     LC    3
        ØR    8           ØUTPUT DEVICE 3

14.     LC    3
        ØR    6           ØUTPUT DEVICE 1 AND 2

15.     LC    3
        ØR    S           ØUTPUT DEVICE 1 AND 3

16.     LC    3
        ØR    U           ØUTPUT DEVICE 2 AND 3

17.     LC    3
        ØR    W           ØUTPUT DEVICE 1, 2, AND 3
```

MONROBOT XI PROGRAM SHEET

PROGRAM

Alphanumeric Input and Output, Combined Subroutines

PAGE	DATE	PROGRAMMER
050		

REGISTER		CONTENTS	NOTES
050	A	T001	Store virgin instruction X004 TADR
	B	V002	Load exit
051	A	T05W	Save exit
	B	V06T	Load Constant XXXX XXXU
052	A	T005	Store count
	B	V06U	Load Constant V000 X070
053	A	U001	V000 X070 to FA 1/ X004 TADR to FA 6
	B	3454	Jump mark to enter alphanumeric data
054	A	T000	Store X004 TADR in FA 0
	B	V06S	Load Constant 0000 0080
055	A	T004	(058)(061) Store word in FA 4
	B	2201	Input one character
056	A	705X	Jump negative if parity error or carriage return
	B	X004	Add shifted word
057	A	705U	Jump negative if register is packed
	B	9004	Shift word up 3 bits
058	A	9004	Shift word up 3 bits
	B	3055	Jump to store word and input next character
059	A	V004	(05X) Load word
	B	X068	Add Constant 0000 0010
05S	A	7062	(05T) Jump negative if final word is packed
	B	9004	Shift final word up 3 bits
05T	A	9004	Shift final word up 3 bits
	B	305S	Jump to test if final word is packed
05U	A	9402	(057) Left 2 bits +*2
	B	1406	Word to FA 4/Add 1 to FA 5
05V	A	6000	Jump zero to store packed word and continue
	B	3400	Jump mark to store final word
05W	A		Saved exit
	B		
05X	A	W069	(056) Subtract Constant 8000 0081
	B	6059	0 = carriage return

MONROBOT XI PROGRAM SHEET

PROGRAM

Alphanumeric Input and Output, Combined Subroutines

PAGE	DATE	PROGRAMMER
060		

REGISTER		CONTENTS	NOTES
060	A	0045	Halt, parity error
	B	V004	Load word
061	A	3055	Jump to re-enter character
	B	6400	Fill
062	A	9001	(05S) Left 1 on final word
	B	T004	Store word
063	A	3400	Jump mark to store final word
	B	305W	Jump to exit
064	A	T066	Store Constant T066 VADR in program
	B	3065	No-op
065	A	V066	(06V) Load variable
	B	X070	Add Constant 0000 0001
066	A	(	[T066 VADR] Store self
	B	)	Load word
067	A	S07X	Clear FA 5 to 0
	B	306V	Jump to output
068	A	Q000	Constant: final flag
	B	0010	
069	A	8000	Constant: carriage return with forced parity
	B	0081	
06S	A	0000	Constant: flag for packed word
	B	0080	
06T	A	XXXX	Constant: count for 20 characters
	B	XXXU	
06U	A	V000	Constant: Load drum load instruction
	B	X070	Add Constant 0000 0001
06V	A	8U01	(06X) 1 bit to FA 5
	B	6065	0 = end of register
06W	A	8U10	5 bits to FA 5
	B	6002	Jump zero to exit
06X	A	S27X	Print character
	B	306V	Jump to continue

MONROBOT XI PROGRAM SHEET

PROGRAM

Alphanumeric Input and Output, Combined Subroutines

PAGE	DATE	PROGRAMMER
070		

REGISTER		CONTENTS	NOTES
070	A	0000	Constant: address increment
	B	0001	
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		

ALPHANUMERIC INPUT AND OUTPUT - COMBINED SUBROUTINES

THESE SUBRØUTINES CAN BE FULLY ØPTIMIZED BY KEEPING THE SECTIONS 4 SECTØRS DISTANT FRØM A TØ B, B TØ C, C TØ D, AND D TØ A.

INPUT SUBRØUTINE

ENTRY - XOO4 TADR IN FA 6, WHERE ADR IS THE FIRST ADDRESS ØF THE STØRAGE TABLE.

INPUT SET FØR DEVICE NØ. 1.

ØP CØDE - JM RAA

EXIT ØN CARRIAGE RETURN ØR A MAXIMUM INPUT ØF 20 CHARACTERS.

CHANGES PRIØR TØ ENTRY -
TØ CHANGE INPUT DEVICE

```
        CA      SLASH H     TOO4 2DO1
        T       RAI
```

TØ CHANGE MAXIMUM INPUT CØUNT

```
        CA      SLASH H     CØMPLEMENT ØF THE NUMBER ØF REGISTERS TØ
                            BE FILLED.  THE MAXIMUM MUST ALWAYS BE A
                            MULTIPLE ØF 5.
        T       RAR
```

REQUIRES FAST ACCESS 0, 1, 2, 4, 5 AND 6.

ØUTPUT SUBRØUTINE

ENTRY - THE CØNTENTS ØF REGISTER INWA IN FA 6, WHERE INWA CØNTAINS

```
                    T     WA+2
                    CA    ADR
```

AND ADR IS THE FIRST ADDRESS ØF THE STØRAGE TABLE.

ØP CØDE - JM WA

ØUTPUT SET FØR DEVICE NØ. 1.

TØ CHANGE ØUTPUT DEVICE -
CHANGE PRIØR TØ ENTRY

```
        CA      WACD
        T       WAØ
```

WHERE WACD CØNTAINS

```
                    ØR      Ø
                    J       RAD
```

REQUIRES FAST ACCESS 2, 5 AND 6.

```
SECTØR A          INPUT

RAA               T    001          STØRE  X004 TADR
                  CA   002
   1.             T    RAB+2        SAVE EXIT
                  CA   RAR
   2.             T    005          STØRE CØUNT
                  CA   RAR+1        CA 000  A WAØ+1
   3.             XC   001          CØNSTANT TØ 1/VARIABLE TØ 6
                  JM   RAA+4        TØ READ AND STØRE
   4.             T    000          STØRE  X004 TADR
                  CA   RAC+2        80

RAI               T    004          STØRE IN FA 4
                  G    2201         INPUT 1 CHARACTER
   1.             JN   RAB+3        PARITY ØR C/R
                  A    004          ADD WØRD
   2.             JN   RAB          REGISTER PACKED
                  L    3
   3.             L    3
                  J    RAI          TØ ENTER NEXT CHARACTER

   4.             CA   004          LØAD WØRD
                  A    RAC          ADD 0000 0010

   5.             JN   RAB+6        REGISTER LEFT ADJUSTED
                  L    3
   6.             L    3
                  J    RAI+5        TØ PACK WØRD

SECTØR B          INPUT

RAB               G    9402         L 2 + 2
                  G    1406         WØRD TØ 4/ ADD 1 TØ 5
   1.             JZ   000          TØ STØRE
                  JM   000          TØ STØRE LAST WØRD

   2.             GG                SAVED EXIT

   3.             S    RAC+1        SUBTRACT 8000 0081
                  JZ   RAI+4        CARRIAGE RETURN
   4.             G    45           HALT, PARITY ERRØR
                  CA   004          LØAD WØRD
   5.             J    RAI          TØ ENTER CHARACTER AGAIN
                  N

   6.             L    1
                  T    004          SAVE WORD
   7.             JM   000          TØ STØRE LAST WØRD
                  J    RAB+2        EXIT
```

SYMBO C OUTPUT OF SHORT TITLES

The Symbo C programmer has the advantage of packed word output short titles or captions as shown in the example below. If more than a little alphanumeric output is required this technique is more economical of storage than a series of control register output instructions. It also eliminates devious methods of setting up alpha packed registers or the necessity of writing one-time input routines to provide them.

Control portion coding:

```
TITLE       CA   /F ALPHA       FIRST 5 CHARACTERS ØF TITLE
            JM   WRAN           ØUTPUT
  1         CA   /F NUMERIC     NEXT 5 CHARACTERS
            JM   WRAN           ØUTPUT
  2         CA   /F IC ØU       NEXT 5 CHARACTERS
            JM   WRAN           ØUTPUT
  3         CA   /F TPUT.       LAST 5 CHARACTERS
            JM   WRAN
```

Subroutine coding:

```
WRAN        C    005            FØR ØUTPUT
            G    8W02           BLEA 2 + 2S BIT FLAG
  1.        LC   3              SHIFT CHARACTER 3 BITS
            JZ   002            TEST FØR SHIFT ØF FLAG
  2.        LC   3              SHIFT CHARACTER 3 BITS
            ØR   D              ØUTPUT
  3.        J    WRAN + 1       TØ REPEAT SHIFT
            N                   NØP
```

SPECIAL APPLICATIONS

VISIBLE CHARACTERS IN PUNCHED TAPE

A program is available to produce large alphanumeric characters on eight channel tape. This provides a means of permanently identifying a program tape in English to the computer operator. As the heading is entered on the typewriter a program is assembled. Upon recognition of the end code the heading will be output to tape in readable block letters. See "Recreational Programs" (MO-147).

Visible numeric digits can be output from the computer by special subroutines for binary to decimal conversion. This is a valuable means of identifying output data tapes to avoid the possibility of incorrect labeling. Such identification may be in the form of month, day and year, route number, or any other numeric field input to the program for control purposes. See "Symbo Subroutine Library, Section 3: Numeric Output" (MO-293).

CONVERSION OF ALPHANUMERIC VALUES FOR SEQUENCING OR COMPARISON TESTING

Occasionally the programmer must deal with a mixed alpha and numeric field which he is not permitted to alter into a straight numeric code. Product codes which have been established in a pre-data-processing era are often of this nature. Comparison testing or sequencing of such codes is often a necessity.

This problem is solved by input/output subroutines which convert the alphabetic characters to the proper sequence of ascending base 16 values on input, and re-convert these values on output.

Conversion Table

Character (external)	0, 1 - 9	IBMZ	A - I	J - R	S - Z	Inpt Outpt (*)
True base 16 value	0, 1 - 9	10	31 - 39	21 - 29	12 - 19	
Base 16 modifier	00	-10	-20	00	+20	
Converted base 16 value (internal)	0, 1 - 9	0	11 - 19	21 - 29	32 - 39	

During input (external to internal) each character is tested: a BCD numeric code and the alpha codes J through R are not modified; the IBM zero code and alpha codes A through I and S through Z are modified according to the table above.

During output (internal to external) each character is tested: a zero value will be output with an IBM zero code. The sign of the base 16 modifier is reversed for output to re-convert to the correct external code. (*)

The input field is packed 6 bits per character and is right justified in the register without flag bits. Input of more than 5 characters prior to the end code will produce incorrect results.

Alphanumeric fields input and converted in this manner can be tested by subtraction to compare for less than, equal to, or greater than a previously input value.

Sequencing

An alphanumeric field input and converted as above may be used as the keyword in the Easy Sort Program (see MO-257). The field size for such a keyword must not exceed 3 alphanumeric characters or 1 numeric followed by 3 alphanumerics. The "sort control word" must be set for sorting of a 6 decimal digit field.

Operating Instructions: Alphanumeric Conversion, Input

Entry to the subroutine is by jump mark instruction. Input from device 2 will be terminated by a tab code. The input field is left in FA 6 and is right justified. The input device code and the end code character may be changed prior to entry to the subroutine.

Operating Instructions: Converted Alphanumeric Word Output

The alphanumeric word to be reconverted must be in FA 6 on entry to the subroutine. Jump mark entry to the subroutine will cause output to device 1 with non-significant postions appearing as IBM zeros. The output device code may be changed prior to entry to the subroutine.

MONROBOT XI PROGRAM SHEET

PROGRAM: Alphanumeric Conversion, Input

PAGE: 050 | DATE: | PROGRAMMER:

REGISTER		CONTENTS	NOTES
050	A	V063	Load Constant 0000 001W
	B	T003	Store tab end code
051	A	S07X	Clear FA 5 to 0
	B	3052	No-op
052	A	2401	(05V)(062) Input one character
	B	T004	Save character in FA 4
053	A	7061	Jump negative if parity error or carriage return
	B	W064	Subtract Constant 0000 000S
054	A	705T	Jump negative = character 1-9, space
	B	W065	Subtract Constant 0000 0006
055	A	605X	0 = IBM zero
	B	V004	Load character
056	A	W003	Subtract Constant 0000 001W
	B	6060	0 = end code
057	A	V004	Load character
	B	X466	Extract Constant 0000 0030, alpha portion
058	A	W067	Subtract Constant 0000 0020
	B	605T	0 = character J-R
059	A	705W	Jump negative = character S-Z
	B	V004	Load character , A-I
05S	A	W067	Subtract Constant 0000 0020
	B	T004	Store in FA 4
05T	A	V005	(054)(058)(05X) Load word
	B	9020	Shift word left 6 bits for latest character
05U	A	X004	Add latest character
	B	T005	Store new word
05V	A	3052	Jump to input next character
	B	6400	Fill
05W	A	V004	(059) Load character
	B	X067	Modify S-Z
05X	A	T004	(055) Store character
	B	305T	Jump to build up word

MONROBOT XI PROGRAM SHEET

PROGRAM

Alphanumeric Conversion, Input

PAGE	DATE	PROGRAMMER
060		

REGISTER		CONTENTS	NOTES
060	A	V005	(056)(061) Load word
	B	3002	Exit
061	A	W068	(053) Subtract Constant 8000 0081
	B	6060	0 = carriage return
062	A	0043	Halt, parity error
	B	3052	Jump to enter character again
063	A	0000	Constant: tab end code
	B	001W	
064	A	0000	Constant: character 1-9, space
	B	000S	
065	A	0000	Constant: IBM zero
	B	0006	
066	A	0000	Constant: extractor for alpha portion
	B	0030	
067	A	0000	Constant: modifier
	B	0020	
068	A	8000	Constant: carriage return with forced parity
	B	0081	
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		
	A		
	B		

TØ CØNVERT ALPHA VALUES FØR SØRTING, SEE MØ-253, PAGE 23.

ENTRY - NØ CØNDITIØNS

ØP CØDE - JM RCA

EXIT - CØNVERTED ALPHANUMERIC WØRD UP TØ 5 CHARACTERS IN FA 6.
INPUT TERMINATES ØN TAB ØR CARRIAGE RETURN.

INPUT ØF MØRE THAN 5 ALPHANUMERIC CHARACTERS IS NØT CHECKED.

INPUT SET FØR DEVICE NØ. 2.

CHANGES PRIØR TØ ENTRY -
TØ CHANGE INPUT DEVICE

```
        CA      SLASH H    2D01 T004
        T       RCAD
```

TØ CHANGE TAB END CØDE

```
        CA      SLASH  H ØR F     END CØDE VALUE
        T       RCAEC
```

REQUIRES FAST ACCESS 2, 3, 4, 5 AND 6.

```
RCA             CA      RCAEC
                T       003             END CØDE
  1.            C       005             CLEAR FA 5
                J       RCAD

RCAD            G       2401            INPUT CHARACTER
                T       004             SAVE CHARACTER
  1.            JN      RCAD+15         PARITY ØR C/R
                S       /D 10
  2.            JN      RCAD+9          SPACE ØR 1-9
                S       /D 6
  3.            JZ      RCAD+13         IBM ZERØ
                CA      004             CHARACTER
  4.            S       003             SUBTRACT END CØDE
                JZ      RCAD+14         0 EQUALS END CØDE
  5.            CA      004             CHARACTER
                XT      /H 30           ALPHA PØRTIØN
  6.            S       /H 20
                JZ      RCAD+9          J - R
  7.            JN      RCAD+12         S - Z
                CA      004             CHARACTER
  8.            S       /H 20           A - I
                T       004

  9.            CA      005             LØAD WØRD
                L       6
 10.            A       004             ADD LATEST CHARACTER
                T       005             STØRE WØRD
```

```
11.             J     RCAD          TØ INPUT NEXT CHARACTER
                N

12.             CA    004
                A     /H 20         MØDIFY S - Z

13.             T     004
                J     RCAD+9        TØ BUILD UP WØRD

14.             CA    005           LØAD WØRD
                J     002           EXIT

15.             S     /H 8000 0081
                JZ    RCAD+14       CARRIAGE RETURN CØDE
16.             G     43            HALT, PARITY ERRØR
                J     RCAD          TØ ENTER CHARACTER AGAIN

RCAEC           GG    1W            END CØDE - TAB
```

MONROBOT XI PROGRAM SHEET

PROGRAM

Converted Alphanumeric Word Output

PAGE	DATE	PROGRAMMER
080		

REGISTER		CONTENTS	NOTES
080	A	S07X	Clear FA 5 for shift
	B	8W02	Left 2 bits on word + 2
081	A	8U04	Binary left end around shift 3 bits
	B	6002	0 = exit, end of word
082	A	8U04	Binary left end around shift 3 bits, character to FA 5
	B	U005	Word to FA 5/ character to FA 6
083	A	608S	0 = IBM zero
	B	T004	Store character in FA 4
084	A	X48W	Extract Constant 0000 0030, alpha portion
	B	608T	0 = character 1-9
085	A	W08X	Subtract Constant 0000 0020
	B	608T	0 = character J-R
086	A	708U	Negative = character A-I
	B	V004	Load character S-Z
087	A	W08X	Subtract Constant 0000 0020
	B	3088	No-op
088	A	U005	(08S)(08T)(08V) Character to FA 5/Word to FA 6
	B	S27X	Print character
089	A	3081	Jump to continue output
	B	6400	Fill
08S	A	9410	(083) Generate an IBM zero
	B	3088	Jump to print
08T	A	V004	(084)(085) Load character
	B	3088	Jump to print
08U	A	V004	(086) Load character
	B	X08X	Add Constant 0000 0020 to restore value
08V	A	3088	Jump to print
	B	6400	Fill
08W	A	0000	Constant: extractor for alpha portion
	B	0030	
08X	A	0000	Constant: modifier
	B	0020	

CONVERTED ALPHANUMERIC WORD OUTPUT

TØ RECØNVERT ALPHA VALUES, SEE MØ-253, PAGE 23.

ENTRY - CØNVERTED ALPHANUMERIC WØRD IN FA 6.

ØP CØDE - JM CWA

WØRDS ØF LESS THAN 5 ALPHANUMERIC CHARACTERS WILL BE ØUTPUT WITH NØNSIGNIFICANT IBM ZERØS.

ØUTPUT SET FØR DEVICE NØ. 1.

TØ CHANGE ØUTPUT DEVICE -
CHANGE PRIØR TØ ENTRY

```
        CA    SLASH H    U005 SD7X
        T     CWAD
```

REQUIRES FAST ACCESS 2, 3, 4, 5 AND 6.

```
CWA             C     005          CLEAR FØR SHIFT
                G     8W02         LC 2 + 2 FØR FLAG

  1.            LC    3            BINARY END ARØUND 3 BITS
                JZ    002          EXIT
  2.            LC    3
                XC    005          WØRD TØ 5/CHARACTER TØ 6
  3.            JZ    CWAD+2       IBM ZERØ
                T     004          SAVE CHARACTER
  4.            XT    /H 30
                JZ    CWAD+3       CHARACTER 1-9
  5.            S     /H 20
                JZ    CWAD+3       CHARACTER J - R
  6.            JN    CWAD+4       CHARACTER A - I
                CA    004          CHARACTER S - Z
  7.            S     /H 20
                J     CWAD

CWAD            XC    005          CHARACTER TØ 5/WØRD TØ 6
                ØR    2            PRINT CHARACTER
  1.            J     CWA+1        TØ CØNTINUE
                N

  2.            G     9410         GENERATE AN IBM ZERØ
                J     CWAD         TØ PRINT

  3.            CA    004          LØAD CHARACTER
                J     CWAD         TØ PRINT

  4.            CA    004          LØAD CHARACTER
                A     /H 20        RESTØRE VALUE
  5.            J     CWAD         TØ PRINT
                N
```