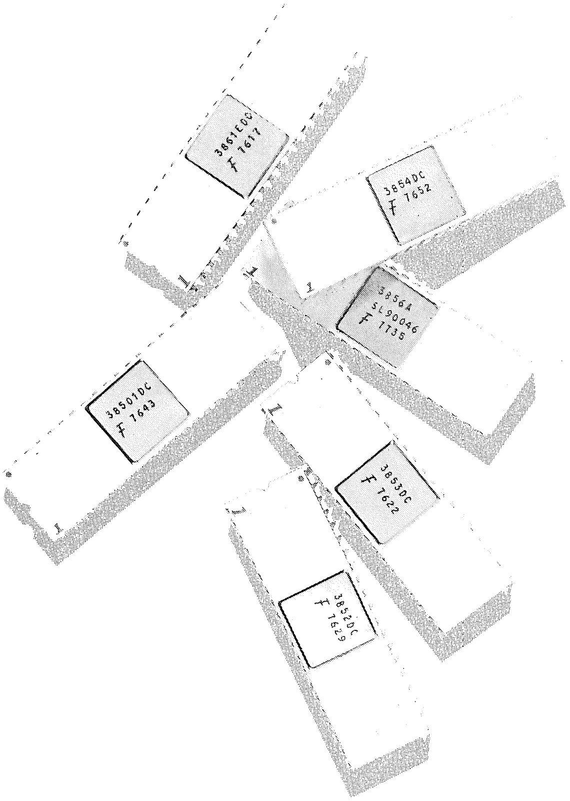
KD-BUG USER'S GUIDE

FAIRCHILD

KD-BUG USER'S GUIDE



FAIRCHILD

464 Ellis Street, Mountain View, California 94042

TABLE OF CONTENTS

Section **Page**

Appendices

1.0 General Description

A special F3856 PSU with a Keyboard Display Monitor (KDM) and a Debug monitor (KD-BUG) has been developed by Fairchild to provide the F8 user with a convenient and powerful programming debug facility to aid in the development of F8 programs. The debugging portion of the program (FAIR-BUG) provides the user with an interactive system via a teletype terminal. The KDM segment provides a simple alternative to an expensive terminal. FAIR-BUG and KDM will be discussed as separate subjects.

The KD-BUG PSU is assigned memory addresses ‘8000’-‘87FF’, with entry point being ‘8080’. The port assignments are as follows:

I/O Port A —	’08’
I/O Port B —	’09’
Local Int. Control —	’0A’
Timer —	’0B’

The Interrupt address vector for the timer is ‘0024’ and for an External Interrupt is ‘00A4’.

1.1 Electrical Specifications

The dc and ac electrical characteristics of the F3856 PSU are, since this is just a standard PSU with a special program on it, identical to those described in the F3856 PSU data sheet.

2.0 FAIR-BUG Overview

FAIR-BUG provides the following capabilities:

Display or Alter Memory Locations
Display or Alter Scratchpad Registers
Display or Alter I/O Ports
Display or Alter Accumulator, ISAR, Status (W Register)
Display or Alter PC0, DC0, DC1
Load Formatted Paper Tape (FAIR-BUG or Formulator Format)
Punch Formatted Paper Tape (Formulator Format)
Punch Paper Tape in PROM Format (4-Bit or 8-Bit Binary Format)
Entry from Keyboard or by Program Instruction
I/O Subroutines Available to User
Breakpoint
Block Memory Move
Hexadecimal Arithmetic

FAIR-BUG can be entered in several ways. Execution of program instructions such as PI ‘8080’; JMP ‘8080’; LR P0, Q, or PK (Q or K contains ‘8080’) can be used to achieve entry from another software module. Another technique is to use hardware to decode the RESET state on the ROMC lines and force the high order bit on the Data Bus to a ’1’ at this time.

FAIR-BUG will save the state of the machine upon entry and will restore it upon return to the users program. Register 8 and PC1 are destroyed by FAIR-BUG; however, under program control the user can save and restore these if necessary. The save area utilized by FAIR-BUG is scratchpad registers 3C to 3F and also the last 26 bytes of user RAM. FAIR-BUG tests locations BDF-BFF and if there is RAM there, it uses these locations for its save area; if it finds that these locations are not RAM, it assumes that locations 3DF-3FF of user memory are RAM and thus uses these locations for its save area. The interrupt is disabled by FAIR-BUG and may be re-enabled by the user if desired.

Two I/O ports (8 and 9) are available to the user when FAIR-BUG is not executing, as is the Timer and External Interrupt facilities (FAIR-BUG does not use either of these). During FAIR-BUG execution Port 8 is used for serial input/output and control functions while Port 9 is used for parallel input from a high-speed paper tape reader. Assignments for Port 8 are: (1 = GND, 0 = +5 V).

Bit	**Function**
7	Serial input (0 Volts = MARK)
6	Character Ready Input (Parallel Device) (+5 V = READY)
5	Enter Key/Display Monitor
4	Device Ready Input (Parallel Device) (+5 V = READY)
3	Step Reader Output (Parallel Device) (0 Volts = Step)
2-1	**Bit 2 Bit 1 Baud Rate**
	0 0 110 Baud for Serial Input/Output
	0 1 300 Baud for Serial Input/Output
	1 0 1200 Baud for Serial Input/Output
	1 1 Baud Delay Counter in Memory Loc. 3FF (or BFF)
0	Serial Output (0 Volts = MARK)

If Port 9 is not utilized by FAIR-BUG, then Pins 3, 4, and 6 of Port 8 are also available to the user. If Port 9 is used for parallel input, the parallel input data should have logic '1' correspond to +5 V electrical level.

Bits 1 and 2 of Port 8 are examined when FAIR-BUG is entered to initialize the Baud Delay Counter. If Bits 2 and 3 are at ground so that the Baud Delay count is being obtained from memory location 3FF (or BFF), then the total delay count (time between bits for a bit serial I/O device) can be adjusted as follows:

$$\text{Bit time interval} = (256 - \text{count}) \circ 36\ \mu s + 55\ \mu s \quad (\Phi = 2\ \text{MHz})$$

Thus, 110 Baud requires a count of 06
300 Baud requires a count of A4
1200 Baud requires a count of EA

2.1 FAIR-BUG Commands

When FAIR-BUG is entered a prompt character (?) is sent to the output device. The user then has the option of using any of the debug commands. After each DEBUG execution the user is again prompted with (?). All data and input parameters are in hexadecimal notation. (C/R) following a command indicates a carriage return.

COMMAND TYPE	**COMMAND**	**FUNCTION**
Display	A (C/R)	Display the contents of the Accumulator
	D0 (C/R)	Display the contents of DC0
	D1 (C/R)	Display the contents of DC1
	I (C/R)	Display the contents of ISAR
	M XXXX (C/R)	Display Memory Location XXXX
	M XXXX-YYYY (C/R)	Display Memory Location XXXX to YYYY
	O XX (C/R)	Display Port XX Data
	P0 (C/R)	Display the contents of PCO
	R XX (C/R)	Display the contents of Register XX
	R XX-YY (C/R)	Display the contents of Registers XX to YY
	S (C/R)	Display the contents of W Register, status
	W (C/R)	Display the contents of W Register, status

COMMAND TYPE	COMMAND	FUNCTION (Continued)
Change	C XX (C/R) (C/R)	Change the previously displayed memory location or port or register to XX
	C XX (C/R) YY (C/R)... (C/R) (C/R)	Change the sequential registers or memory locations to XX, YY...
	C XXXX (C/R) (C/R)	Change the previously displayed PC or DC to XXXX
Examine	E (C/R)	Display the last addressed register or memory location or port
Next	N (C/R)	Display the next register or memory location or port
Load	L (C/R)	Load formatted object paper tape. If (CK) prints, then checksum error has occurred on block last read
	H (C/R)	Load formatted object paper tape from high-speed reader
Hexadecimal	X aaaa ± bbbb =	Add or subtract value bbbb from value aaaa
Block Move	M dddd (C/R) V ssss-eeee (C/R)	Move memory block starting at ssss and ending at address eeee to memory destination address dddd
Punch	B XXXX-YYYY-Z	Binary Punch PROM format; XXXX is starting page address and YYYY is ending page address. Z is number of bytes per block; 0 = 256, 1 = 512. To Punch 0 to BFF then enter B0-C00-0. 4-bit wide data.
	J XXXX-YYYY-Z	Same as B, except 8-bit wide data.
	F XXXX-YYYY (C/R)	Formatted punch for future load, Formulator format.
Go To	G (C/R)	Go to address of PCO
	G AAAA (C/R)	Change PCO to address AAAA, then go to AAAA and continue execution
Delete Command	[	Delete command and start a new command input string
Breakpoint	UAAAA (C/R)	Set breakpoint (PK instruction) at address AAAA.
	T (C/R)	Clear breakpoint, restore user instruction.

NOTE: Clear Breakpoint must be issued prior to any IO port display or change commands or else instruction at AAAA will be incorrect.

2.2 FAIR-BUG Command Usage

A brief study of Appendix B will assist the reader in understanding FAIR-BUG command utilization. From these examples many advantages over console debugging are quite apparent. First and most important a history is recorded of the steps taken and the results displayed. Second is the accessability of the scratchpad registers as well as the other working registers of the system. Third is the simple access from the console to FAIR-BUG to the user.

One advantage that is not so apparent is that when a user wishes to end a debug session the user may save his modified program by punching the memory using the F command. In a subsequent debug session he may then load this "updated" program and continue to debug. If program patches were extensive much time and effort is saved. Of course the program may be re-assembled after editing the changes in order to obtain an updated program.

Another advantage and also not so apparent is the fact that Keyboard input is much less error prone. If the user uses a little care and examines his keyed input prior to the carriage return he can almost totally eliminate key-in errors. In addition, a debug session achieves more results in a quicker time.

2.3 FAIR-BUG Input/Output

The FAIR-BUG input/output routines assume that Port 8 of the PSU is available and is configured as shown in Appendix A. In order to communicate with FAIR-BUG an 11-bit serial type device such as a Teletype ASR 33 or compatible type TTY or CRT is required. FAIR-BUG has options to vary the Baud Rate by changing the counters for delay loops between bits. The timer is not utilized so the user is not deprived of this valuable resource. The counter value for 110 Baud is six which counts by incrementing an 8-bit register until zero; the six gives a loop count of 250. For 300 Baud the counter is initially H'A4' giving a loop count of H'5C' or 92 decimal. Other Baud Rates can be achieved by modifying the counter. These counts assumed a system clock of 2 MHz. For a faster clock, the counter must be changed to produce more delay loops, while for a slower clock the counter must be changed to produce less delay loops. This is due to the fact that each instruction time will change and the total loop time will change while the device speed is always constant.

When the Baud Rate is not set to default to either 110 or 300 or 1200 then the Baud Rate must be put into RAM location H'BFF' or if this location is a PROM address then into RAM location H'3FF'.

2.4 Load from Parallel Input Device

The loader in FAIR-BUG can load from either a teletype or from a parallel source. The usual parallel source would be a high-speed paper tape reader which reads 100 to 300 characters per second. The parallel device is controlled using a "handshaking" protocol. The handshaking eliminates synchronizing problems because it forces the device to wait for the microprocessor and forces the microprocessor to wait for the device.

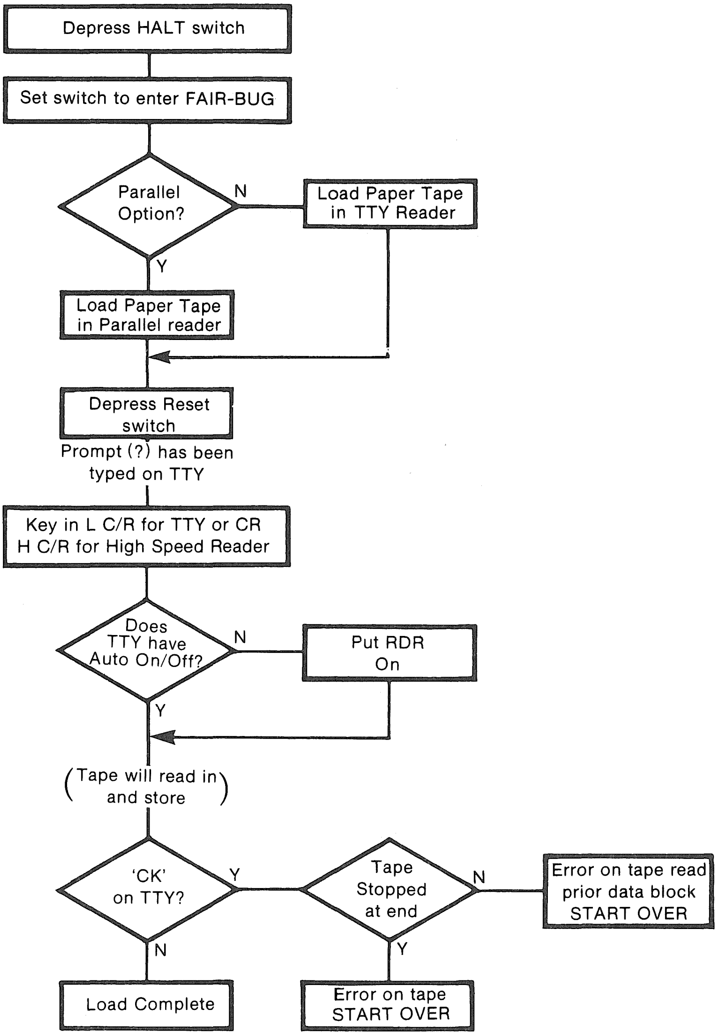

Depress HALT switch
Set switch to enter FAIR-BUG
Parallel Option?
N
Load Paper Tape in TTY Reader
Y
Load Paper Tape in Parallel reader
Depress Reset switch
Prompt (?) has been typed on TTY
Key in L C/R for TTY or CR H C/R for High Speed Reader
Does TTY have Auto On/Off?
N
Put RDR On
Y
(Tape will read in and store)
'CK' on TTY?
Y
Tape Stopped at end
N
Error on tape read prior data block START OVER
N
Load Complete
Y
Error on tape START OVER

FIGURE 1

FAIR-BUG parallel read routine examines DEVICE READY and waits for the ready signal, it then looks for Character Ready and delays 100 μs after detecting the ready, then it reads a character before the output of Step Reader. This sequence is repeated for each character. Only the formats shown in Appendix C can be read by FAIR-BUG with the Load command. However, the user may use the subroutine PINP to read other formats. Bits 1-2 are examined when FAIR-BUG is entered to initialize the baud delay counter. See Appendix A for the I/O port pin assignments. The flow chart in *Figure 1* indicates the steps necessary to load a paper tape program formatted as shown in Appendix C.

2.5 FAIR-BUG Subroutines

I/O Subroutines on the FAIR-BUG PSU are available to the user. These are listed below and documented in Appendix E.

NAME	ENTRY ADDRESS	FUNCTION
TTYI	8553	Input 1 byte from TTY type device (11 bits serial/character)
TTY0	8593	Output 1 byte to TTY type device (11 bits serial/character)
TTCR	8578	Output CR, LF, & 4 Null characters using TTY1 subroutine
PINP	853D	Input 1 byte from the parallel IP device (150μsec minimum delay between characters)
F0P1	811B	Output 1 or 2 hexidecimal digits in ASCII format from a memory location.
F0P2	811D	Output 1 or 2 hexidecimal digits in ASCII format from register QL.
BYTE	8515	Input 2 ASCII digits from a parallel or serial IP device; then convert them to one hexidecimal byte

2.6 Programming Examples

The following program linkages to FAIR-BUG are cited as examples as how to utilize FAIR-BUG as a "snap shot" display or breakpoint vehicle.

There are two general ways to enter FAIR-BUG; either through programmed instructions, or by manual intervention from the console. The manual entry is normally used for the following situations:

For initial program loading
For program punch and save
To start tracing a runaway program
To display or take further action following a BR*

The procedure to manually enter FAIR-BUG is to HALT, set DEBUG switch, push RESTART, then respond to prompt (?) with desired commands.

Programmed entry to FAIR-BUG can be achieved by building trace routines into the source program prior to assembly or else by patching the object program after loading it. In either case the following instructions provide the necessary linkage.

JMP H'8080 This is a 3 byte instruction that destroys the accumulator and does not save the program counter.

PI H'8080' This is a 3 byte instruction that destroys the accumulator and pushes the program counter (PC0) to the stack (PC1).

LR P0,Q This is a 1 byte instruction that does not destroy the accumulator and does not save the program counter.

PK This is a 1 byte instruction that does not destroy the accumulator and pushes the program counter (PC0) to the stack (PC1)

These instructions are explained in detail in the F8 USERS GUIDE. It should be noted that all except the PI are priviledged instructions and that an interrupt cannot be serviced following these instructions. Furthermore, the first instruction executed at H'8080' is a disable interrupt which inhibits interrupts until an Enable instruction is issued.

Examination of these instructions discloses that the PK instruction is the most desirable to use since it is 1 byte, saves the accumulator, and saves the PC0; however, it does require the K register (R12 and R13) be preset. If K is not used in the program being tested this is an ideal instruction. A recommended procedure is to code into the users program at location 0 the following instructions:

```
LI H'80'          ⎧ LI H'80'
LR KU, A       or ⎨ LR QU, A
LR KL, A          ⎩ LR QL, A
```

If one of the above housekeeping procedures are used patching the user's program for tracing or display purpose becomes an easier proposition. Instead of using a 2 byte BR* or 3 byte JMP or PI the simpler PK or LR P0,Q may be used and the prompt (?) will indicate that a desired point has been reached. Display of PC1 will then determine which (if more than one) of the trace points has been reached. The disadvantage of using PK or PI is that the object program PC1 has been lost. If this is detrimental for a particular section of code being debugged the JMP or LR P0,Q instructions are available.

It can be noted that the options are numerous and the user must decide for himself which one or which combination is most desirable for his purpose.

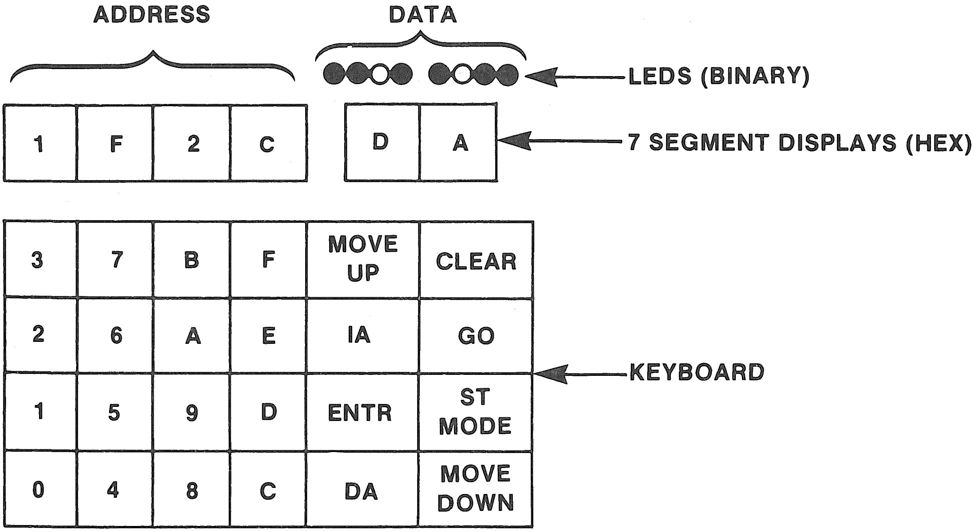
It is suggested that the user try a small program as shown in Appendix B and utilize all the options of FAIR-BUG until he feels confident that he understands all the options. This will save much time and effort in future debug sessions.

3.0 Keyboard and Display Monitor (KDM)

The Keyboard and Display Monitor (KDM) is a program that allows the use of a keyboard and display unit in conjunction with any F8 kit or microprocessor system that has I/O Ports 0 and 1 for use by the KDM. The keyboard and display are shown below and the schematic diagram in Appendix F.

The reset to FAIR-BUG switch enters a save routine which saves PC0, DC0, DC1, ACC, R0-R7, and R9-R15 in RAM as shown in Appendix A. After the save, Port 8 bit 5 is examined. If this pin is tied to V_{SS} then control goes to KDM. If this pin is left open then the FAIR-BUG monitor is in control.

KDM can now display or change any of the saved values or any other part of RAM memory. To restore these registers, prior to continuing execution of a program, the restore and continue subroutine must be executed. This is done by go to H'80E7'. The ISAR and status register will be destroyed. R16-R47 (decimal notation) will be unaltered.



3	7	B	F	MOVE UP	CLEAR
2	6	A	E	IA	GO
1	5	9	D	ENTR	ST MODE
0	4	8	C	DA	MOVE DOWN

3.1 Keyboard Instructions

0-9 and A-F Keys: When KDM is first entered the system will be in the addressing mode with zeros in the address display and the contents of memory location H'0000' in the data display. Use of any of these keys when in the addressing mode will cause the corresponding hex digit to be entered into the address display.

The data display will show the contents of the address being displayed. A maximum of four digits can be entered. If the system is in the store mode (see store mode key), then a maximum of two digits can be entered into the data display.

Store Mode Key

This key will cause the system to change from addressing mode to store mode or from store mode to addressing mode. When the system is in store mode the decimal point to the right of the right most display digit will be lit.

Enter Key

When in addressing mode this key will terminate digit entry so that a new address can be entered, if desired. When in store mode this key will cause the digits that were entered into the data display from the keyboard to be stored in the memory location indicated by the address display. The address display will then be automatically incremented by one.

Increment Address Key

Each depression of this key will increment the address display by one. If held down for more than 1.5 seconds the address display will increment continuously. This key will operate in both addressing and store modes.

Decrement Address Key

Similar to increment key except that the address will be decremented.

Move Up Key

This key will operate in store mode only. It will cause the contents of all memory locations between the presently addressed location and the next higher XXFF location to be moved one memory location higher in address. The previous contents of the location just after XXFF will be lost. A NOP(2B) will be inserted in the presently addressed location.

Move Down Key

This key will operate in store mode only. It will cause the contents of all memory locations between the presently addressed location plus one and the next higher XX00 location to be moved one memory location lower in address. The previous contents of the presently addressed location will be lost.

Go Key

This key will cause a jump to the memory location contained in the address display.

Clear Key

This key allows for the clearance of any hex digit entries that have not yet been followed by a function key.

Reset Key

If the toggle switch is in run position, this key will cause a jump to "0000" address. If the toggle switch is in the debug position, the reset will cause a jump to the beginning of the FAIR-BUG program and the state of the machine will be saved according to the rules of FAIR-BUG. At this point Bit 5 of the Port 8 (Pin 36 of the F3856) is examined. If it is found to be at V_{SS} the program will jump to the KDM program. Otherwise, it will stay in FAIR-BUG.

3.2 Subroutines

There are three subroutines in the KDM program that can be utilized by a user's program:

Keyboard Read (Scan)
Display (Disp)
Song Generation (Song)

Scan

A subroutine that allows the user to use the keyboard as input device is available at address H'8731'. The following sequence of instructions is a suggested way of using the subroutine:

```
KDB         PI          H'8731'
            LI          H'FF'
            XS          0
            BNZ         KBD             BRANCH IF KEY STILL DOWN
LOOP        PI          H'8731'
            LI          H'FF'
            XS          0
            BZ          LOOP            BRANCH IF NO KEYS
```

The above routine assures that a key has been released before accepting a new key. An H'FF' in register 0 indicates that no key was detected. Otherwise register 0 will contain a number corresponding to the keys in the following manner:

KEY	REG 0
0-F	0-F
DA	10
ENT	11
IA	12
MOVE UP	13
MOVE DOWN	14
STM	15
GO	16
CLEAR	17
NO KEY	FF

The registers used by this subroutine are PC1, R0, R12, R13 and R48-R63. This subroutine also uses the display subroutine which takes the right hand four bit digits from R48-R53 and outputs them to the display, MSD to LSD respectively.

Disp

The display routine will display the six digits located in R48-R53.

ADDRESS:	H'8799'
ENTER:	R48-R53 (O'60'-O'65') DATA TO DISPLAY
EXIT:	R54-R55 (O'66'-'O67') DESTROYED

Song

A song generation subroutine can be utilized which outputs and audio signal (square wave) on Bit 6 of Port 8.

Load the note time value and pitch constants into memory in the following format:

```
DATA     DC        TIME VALUE CONSTANT (1)
         DC        PITCH CONSTANT (1)
          :          :       :
          :          :       :
         DC        TIME VALUE CONSTANT (n)
         DC        PITCH CONSTANT (n)
         DC        0
```

ADDRESS:	H'87BE'	
ENTER:	R7	TEMPO VALUE
	DC0	START OF DATA VALUES
EXIT:	R1-R9	DESTROYED

PITCH VALUES

	HEX CONSTANT
REST	7F
G_1	7E
G_1#	77
A_1	70
A_1#	6A
B_1	64
C	5E
C#	58
D	53
D#	4D
E	49
F	46
F#	42
G	3E
G#	3A
A	37
A#	33
B	30
C′	2D
C′#	2A
D′	28
D′#	25
E′	23
F′	22
F′#	20
G′	1E
G′#	1C
A′	1A
A′#	18
B′	17
C_2	16

TEMPO VALUES

d PER MINUTE	REGISTER 7 CONSTANT
40	6
48	5
60	4
80	3
120	2
240	1

TIME VALUE

Time Value of Note or Rest	Hex Constant
0	20
𝅗𝅥 .	18
𝅗𝅥	10
♩.	C
♩	8
♪.	6
♪	4
𝅘𝅥𝅯.	3
𝅘𝅥𝅯	2

APPENDIX A

FAIR-BUG PORTS ASSIGNMENT AND MEMORY UTILIZATION

ASSIGNMENTS FOR PORT 8

Bit	Function
7	Serial input
6	Character Ready (Parallel Device)
5	Enter Key/Display Monitor
4	Device Ready (Parallel Device)
3	Step Reader (Parallel Device)

bit 2	bit 1	Baud Rate
0	0	110 baud
0	1	300 baud
1	0	1200 baud
1	1	Baud delay counter in memory at "3FF" or "BFF"

Note: 0 = +5; 1 = GND, 0 V.

0	Serial Output

ASSIGNMENTS FOR PORT 9

Bit	Function
7	Parallel input byte — MSB
6	Parallel input byte
5	Parallel input byte
4	Parallel input byte
3	Parallel input byte
2	Parallel input byte
1	Parallel input byte
0	Parallel input byte — LSB

MAP OF FAIR-BUG MEMORY USE

RAM
Address

0XD8								Work 1
0XC0	Work 2	Work 3	Work 3	Work 4	Work 5	Work 7	Work 8	PC0$_U$
0XE8	PCO$_L$	PC1$_U$	PC1$_L$	DC0$_U$	DCO$_L$	DC1$_U$	DC1$_L$	R0
0XF0	R1	R2	R3	R4	R5	R6	R7	ACC
0XF8	R9	R10	R11	R12	R13	R14	R15	Baud Optn

NOTE: X = B or 3. If H'BEB' is RAM then H'BDF' to H'BFF' will be used as save area.

If H'BEB' is not RAM then H'3DF' to H'3FF' will be used as the save area.

NOTE: Work 1-8 is used to save:

1. Breakpoint address and user instruction.
2. Sequence of instructions to execute Outs, Out, Ins, or In sequence for Port Display or Change.

MAP OF FAIR-BUG SCRATCHPAD USE

Scratchpad
Address

3C-3F	Work A	Status	Work B	ISAR

NOTE: Work A & B are used during the entry and exit routines of FAIRBUG when saving or restoring the state of the user's program.

APPENDIX B
FAIR-BUG EXAMPLE 1

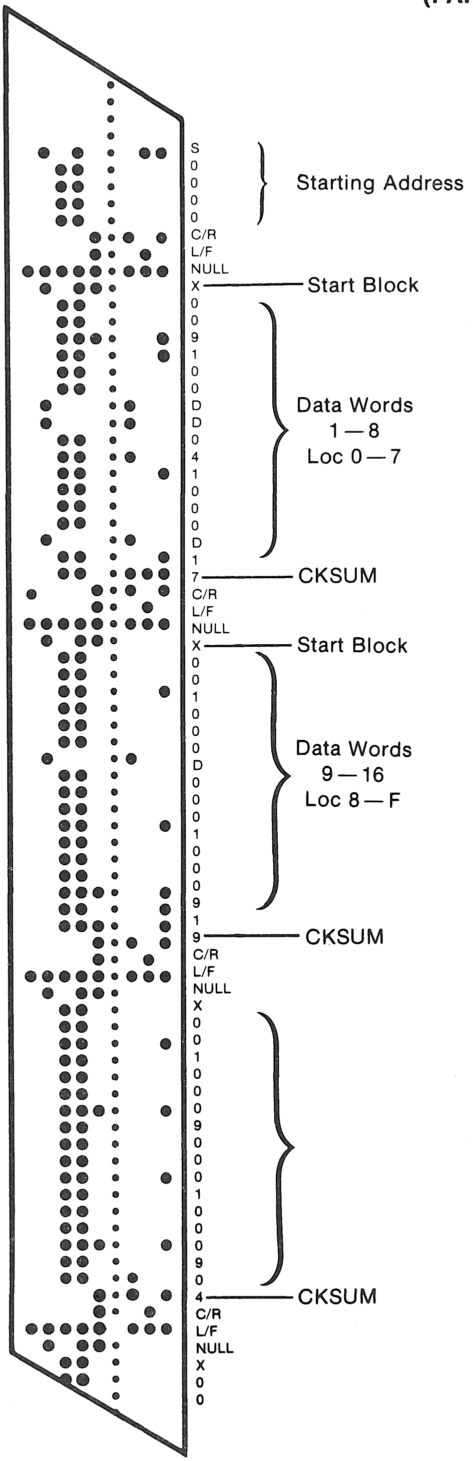
```
?M0
 M0000 = 00                                   Store a Program to Set
?C 70 (CR) (CR)                               Scratchpad to 0-3F. Note two methods to store
?NM0001 = 91                                                     data in memory. The same
?C0B (CR) (CR)                                                   may be done with
?NM0002 = 00                                                     registers also.
?C 5C (CR) (CR)                            LOOP        LIS 0
?NM0003 = DD                                           LR IS, A
?C 1F (CR) (CR)                                        LR S,A
?NM0004 = 04                                           INC
?C25 (CR) 3F (CR) 94 (CR) (CR)                         CI H'3F
?CF9 (CR) 29 (CR) 80 (CR) (CR)                         BNZ LOOP
?C80 (CR) (CR)                                         JMP H'8080'
?M0-A
 M0000 =  70  0B  5C  1F  25  3F  94  F9               Display Program
 M0008 =  29  80  80  D0  00  10  00  91
?R0-3F
 R0000 =  A4  FF  09  FF  00  00  FF  00
 R0008 =  83  0A  00  FF  81  97  03  FF
 R0010    0C  20  13  00  1E  00  BF  00
 R0018 =  9D  40  7D  01  DD  17  55  00               Display Scratchpad
 R0020 =  B7  F7  7F  A2  FF  0E  FF  22               Before Execution
 R0028 =  FF  76  FF  3C  FF  CE  5F  20
 R0030 =  18  04  02  00  D9  04  7F  00
 R0038 =  75  01  57  4A  0F  0A  0A  FF
?G0                                                    Go to loc 0 to Execute
(Program loop err.) Manual reset to FAIR-BUG
?M7
 M0007 = F9
?C FA                                                  Correct BNZ Instruction
?G0                                                    Go to 0
?R0-3F
 R0000 =  00  01  02  03  04  05  06  07
 R0008 =  80  09  0A  0B  0C  0D  0E  0F
 R0010 =  10  11  12  13  14  15  16  17               Display Registers after
 R0018 =  18  19  1A  1B  1C  1D  1E  1F               Execution.
 R0020 =  20  21  22  23  24  25  26  27               Note: R8, R3C-3F are used
 R0028 =  28  29  2A  2B  2C  2D  2E  2F               by FAIR-BUG
 R0030 =  30  31  32  33  34  35  36  37
 R0038 =  38  39  3A  3B  3E  09  09  0B
?P0 0000                                               PCO not saved by JMP
?PI EEEE
?M8
 M0008 = 29                                            Change JMP to PI
?C28
?G0                                                    Execute Again
?P0 000B                                               PCO Now saved!
?
```

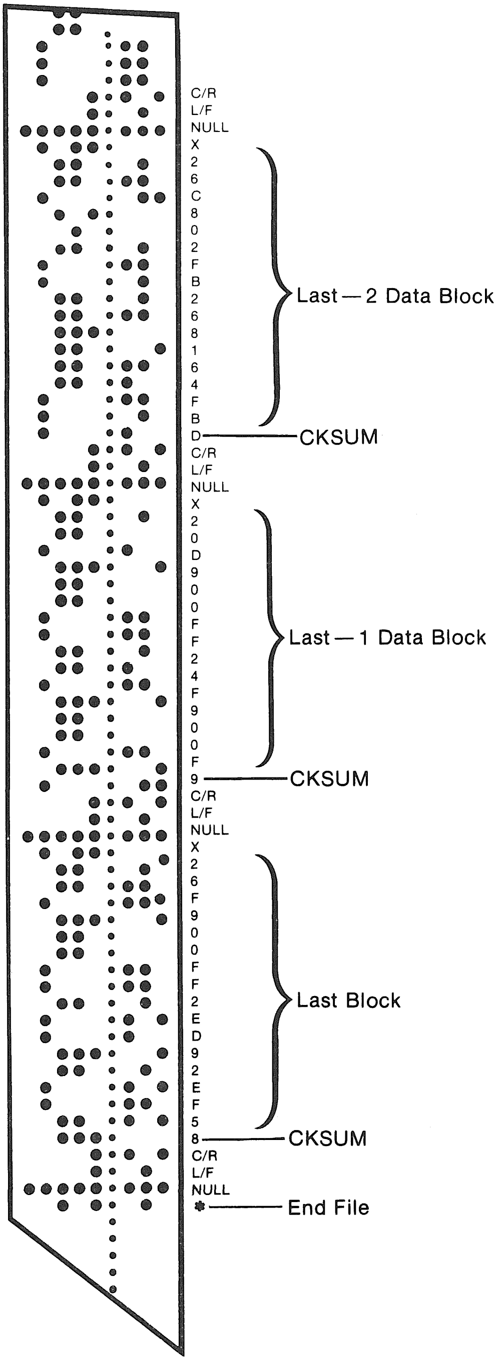
FAIR-BUG EXAMPLE 1

```
?L
?R5
 R0005 = 00                                      R1-2 Types R0-R7
?R1-2                                            Next = R8
 R0000 =  A4  FF  09  00  00  00  FF  00         Change R8 to 55
?N R0008 = 83                                    Examine R8
?C55
?E R0008 = 55
?R8
 R0008 = 55
?P0 83B0
?D0 807B
?D1 0000
?CFFFF                                           Display DC1, Change, Examine,
?E FFFF                                          Then Display Again
?D1 FFFF
?I = 0F                                          ISAR
?S = 0A                                          Status
?W = 0A
?M3E0-3FF
 M03E0 =  04  00  00  10  00  00  00  83
 M03E8 =  B0  DD  EE  80  7B  FF  FF  A4        Memory Dump
 M03F0 =  FF  09  00  00  00  FF  00  55
 M03F8 =  0A  00  FF  81  97  03  FF  00
?R0-F
 R0000 =  A4  FF  09  00  00  00  FF  00         Register Dump
 R0008 =  55  0A  00  FF  81  97  03  FF
?R10-40
 R0010 =  00  20  13  00  1E  00  BF  00
 R0018 =  9D  40  7D  01  DD 17   55  00
 R0020 =  B7  F7  7F  A2  FF  0E  FF  22
 R0028 =  FF  76  FF  3C  FF  CE  5F  20         Register Dump
 R0030 =  18  04  02  00  D9  04  7F  00         Note: R40-R47 is Actually R0-R7
 R0038 =  75  01  57  4A  0F  0A  0A  FF
 R0040 =  A4  FF  09  00  00  00  FF  00
?B0-100-00F0B0N0B0N0B0N0F0N0F0N0F000BNN0BMBM@0@0@M@0CM@M@M@0@MA0@MB0@0FM@NN0BL

                                     Punch Format (PROM Tape)
?F0-FF                               See Appendix D for Format

:18000000009100DD041000D1001000D000100091001000900010009OD4      Formulator
      .                  .                  .             .       Punch
      .                  .                  .             .       Format
      .                  .                  .             .       See
:1800E80026C802F3268164FB20D900FF24F900F926F900FF2ED92EF5C1      Appendix C
:00010000FF

?A = 83
?C00                                             Display, Change, Examine Accumulator
?E R0008 = 00
?N R0009 = 0F                                    NOTE: Accumulator is in R8 Therefore Next is R9
?G8080                                           Go to 8080

?P0 8080                                         PC0 Changed to 8080
?G                                               Go to PC0 (8080)
?
```

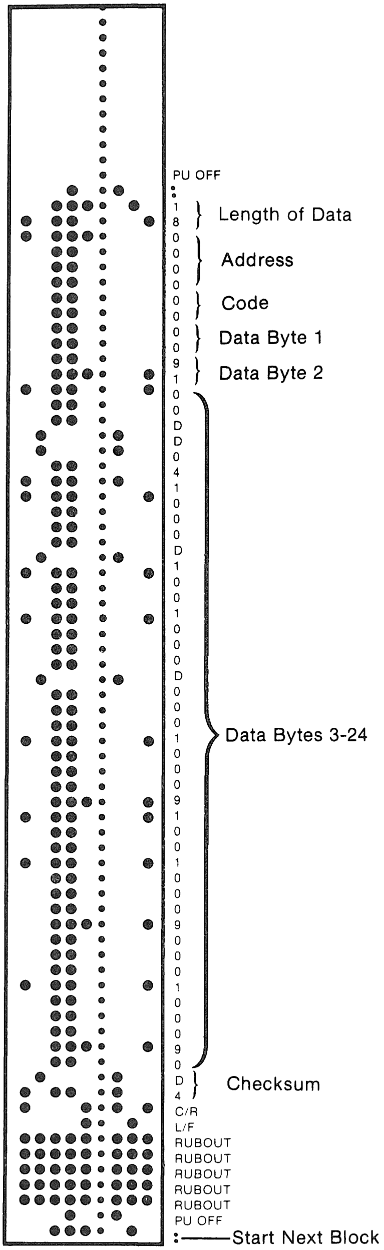
APPENDIX C
FORMATTED TAPE
(FAIRBUG FORMAT)

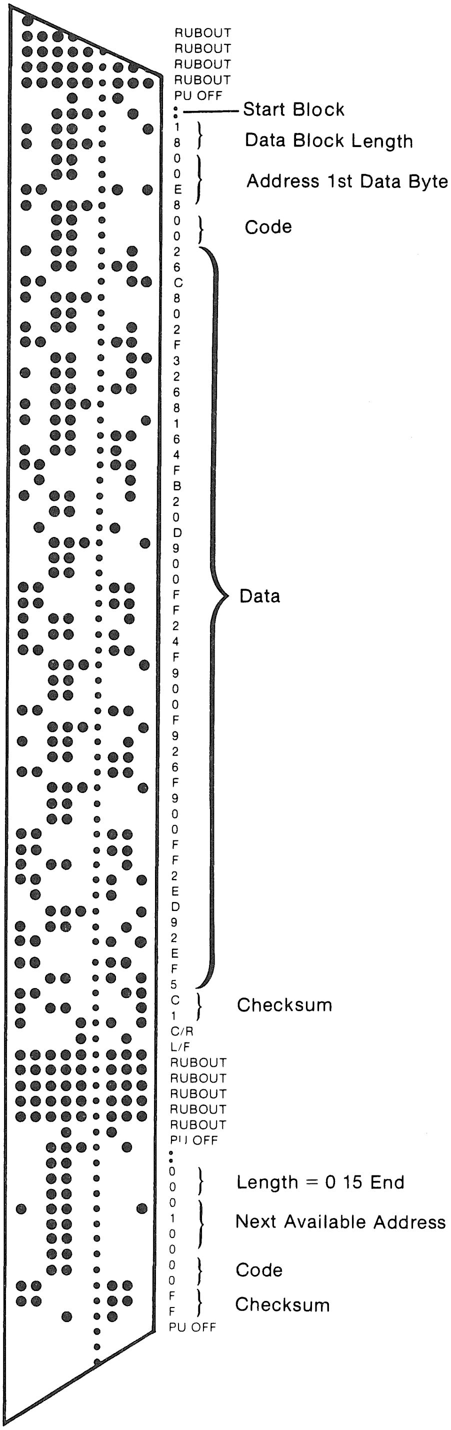
Note: This example was loaded
(L) by the instruction
shown in Appendix B. This is
ASCII 7 bit format.





APPENDIX C (Continued)
FORMATTED TAPE
(FORMULATOR FORMAT)

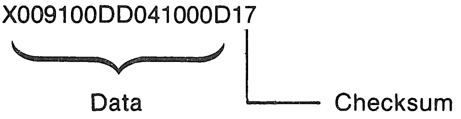




APPENDIX C (Continued)

FAIRBUG FORMAT AND CHECKSUM

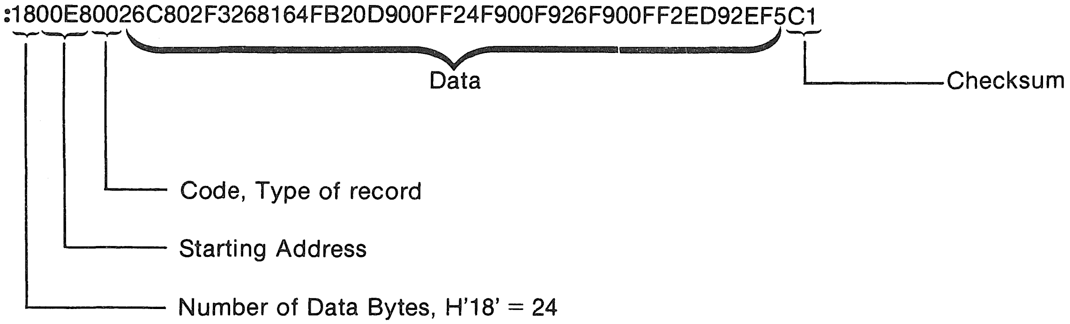
The following line of code is from Appendix C. The checksum is the sum of all the nibbles in the block.


```
X009100DD041000D17
```

```
0    0
0    4
9    1
1    0           4
0    0           3
0    0           —
D    D           7 = Checksum
D    1
—    —
4    3
```

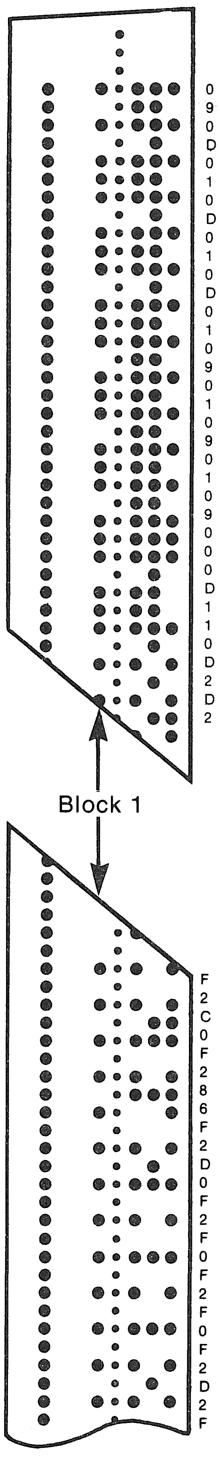
FORMULATOR FORMAT AND CHECKSUM

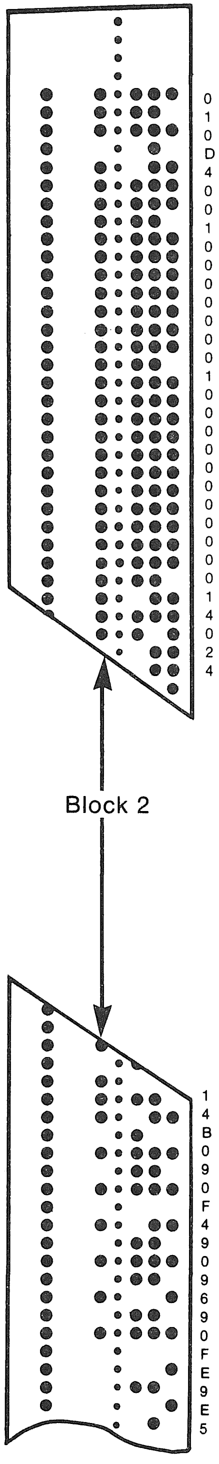
The following line of code is from example 2. The checksum is the 2's complement of the sum of all the bytes in the record. Notice that when the checksum is also added to the sum the total is always zero for the low 16 bits.


```
:1800E80026C802F3268164FB20D900FF24F900F926F900FF2ED92EF5C1
```

```
18    26    24    2E        E3
00    81    F9    D9        FE
E8    64    00    2E        34
00    FB    F9    F5        2A
26    20    26    —         —
C8    D9    F9    2A        3F = Total Sum
02    00    00              2's Complement (3̅F̅ +1 = C1)
F3    FF    FF
—     —     —
E3    FE    34
```

APPENDIX D
BINARY FORMAT
(PROM GENERATION FORMAT)

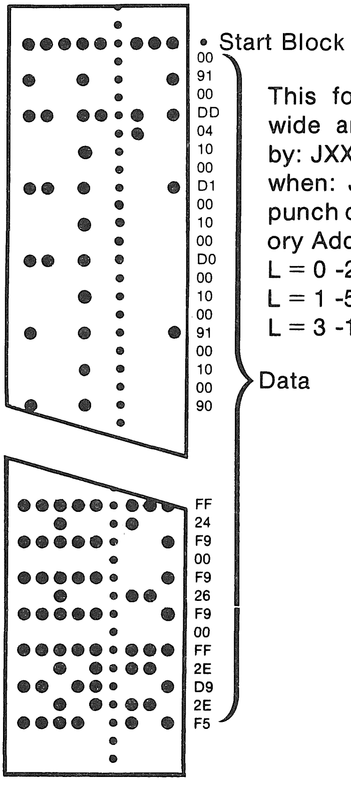




Note: This example was punched by the instruction shown in Appendix B. The first block is punched with 256 memory locations (high order bits). The next block has 256 memory locations (low order bits). Subsequent blocks then alternate high, low, high,low, etc. with a blank gap between. Only the low order 4 bits are significant and are punched in complement form. The table below shows examples of commands and the results produced.

Input Command B-XX00-YY0-L		Decimal Memory Addresses Punched	# Blocks	Block Length
B	0-100-0	0-255	2	256
B	0-400-0	0-1023	8	256
B	100-200-0	256-511	2	256
B	0-400-1	0-1023	4	512
B	0-1000-1	0-4095	16	512
B	200-400-1	512-1023	2	512

PROM 8 BIT FORMAT



This format is 8 bits wide and is punched by: JXX00-YY00-L when: J0-100-0 it will punch data from Memory Address 0 to H'FF'.
L = 0 -256 bytes
L = 1 -512 bytes
L = 3 -1024 bytes

APPENDIX E
FAIR-BUG SUBROUTINES

The following INPUT and OUTPUT subroutines exist in FAIR-BUG and should be called by the users program. All subroutines should be entered by: (PI Address).

TTYI — Input 1 byte from TTY type device, without echo. Data is 11 bits/character being received on Port 8 Pin 7.

Address: H'8553'
Enter: R0 — Delay Counter
Exit: W Reg — Destroyed
PC1 — User return address
Accum — Input byte
R0 — Unchanged
R1 — Input byte
R2 — -1

TTY0 — Output 1 byte to TTY type device. Data transmitted is 11 bits/character being output on Port 8 Pin 0.

Address: H'8593'
Enter: R0 — Delay Counter
R1 — Byte to output
Exit: W Reg — Destroyed
PC1 — User return address
Accum — 0
R0 — Unchanged
R1 — -1
R2 — 0

TTCR — Output CR/LF/NULL to TTY type device; subroutine TTY0 is called.

Address: H'8578'
Enter: R0 — Delay Counter
Exit: W Reg — Destroyed
PC1 —
Accum — 0
K Reg — User return address
R0 — Unchanged
R1 — -1
R2 — 0

PINP — Input 1 byte from parallel input device; minimum delay between characters is 150 μs. Byte is received on Port 9 with control bits on Port 8, pins 3, 4, and 6.

Address: H'853D'
Enter: No setup
Exit: W Reg — Destroyed
PC1 — User return address
Accum — Input byte
R1 — Input byte

BYTE — Input 2 ASCII hexidecimal characters and convert to 1 byte; also accumulate the checksum. If input is not ASCII characters 0-9 or A-F meaningless results will be returned. Either TTYI or PINP is called as input routine.

Address: H'8515'
Enter: Q — H'853D'(for parallel input)
Q — H'8553' (serial input) RO = Delay Counter
R7 — Previously accumulated checksum
Exit: W Reg — Destroyed
PC1 — Destroyed
Accum — Input byte
K — User return address
Q — Unchanged
R0 — Unchanged
R1 — Destroyed
R2 — -1 (if serial IP), unchanged for parallel IP
R7 — Checksum
R8 — 0
R11 — Input byte

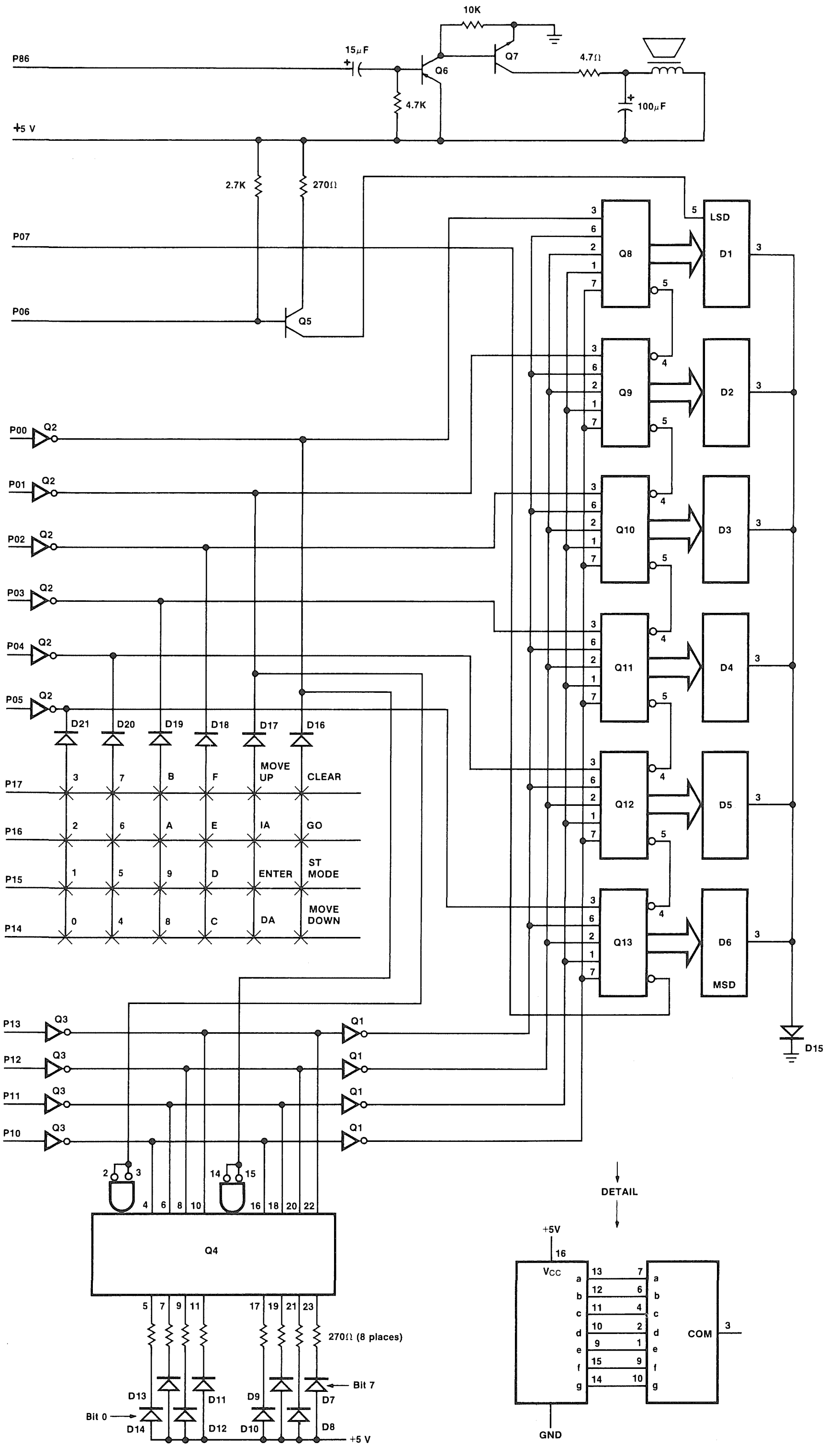
F0P1 — Output byte of data from memory to TTY type device using TTY0 subroutine. Byte is converted to 1 or 2 ASCII hexidecimal characters.

Address: H'811B'
Enter: R0 — Delay Counter
R8 — Flag Pos# = 0P Hi 4 bits, then Lo 4 as ASCII
Neg# = 0P Lo 4 bits as ASCII
DC0 — Memory address of data
Exit: W Reg — Destroyed
PC1 — Destroyed
Accum — Destroyed
DC0 — DC0 + 1
K Reg — User return address
QL — Data byte
R0 — Unchanged
R1 — -1
R2 — 0
R7 — Checksum (low 4 bits significant)

F0P2 — Output byte of data from QL. Same routine as F0P1 except DC0 is not used.

Address H'811D'
Enter: R0 — Delay Counter
R8 — Same as F0P1
QL — Data byte to output
Exit: Same as F0P1

APPENDIX F
KEYBOARD/DISPLAY SCHEMATIC DIAGRAM



APPENDIX F (Continued)

PARTS DESCRIPTION

Q1, Q2, Q3	9N04	HEX Inverter
Q5, Q7	2N2222	NPN Transistor
Q8 - Q13	9368	7 Segment Decoding Latch
Q4	9308	Dual 4 bit latch
D1 - D6	FND500	7 Segment LED Display
D7 - D14	FLV110	LED Lamps
Q6	2N3638	PNP Transistor
D16 - D21	1N461	Diode
D15	1N4001	Diode

PINOUT DESCRIPTION

P86	Output Pin for Song
P07	Output Blanking Signal
P06	Output Signal for Decimal Point
P00 - P05	Output Signal for Digit Select
P14 - P17	Input Signals, Keyboard Read
P10 - P13	Output Signals, Digit to Display

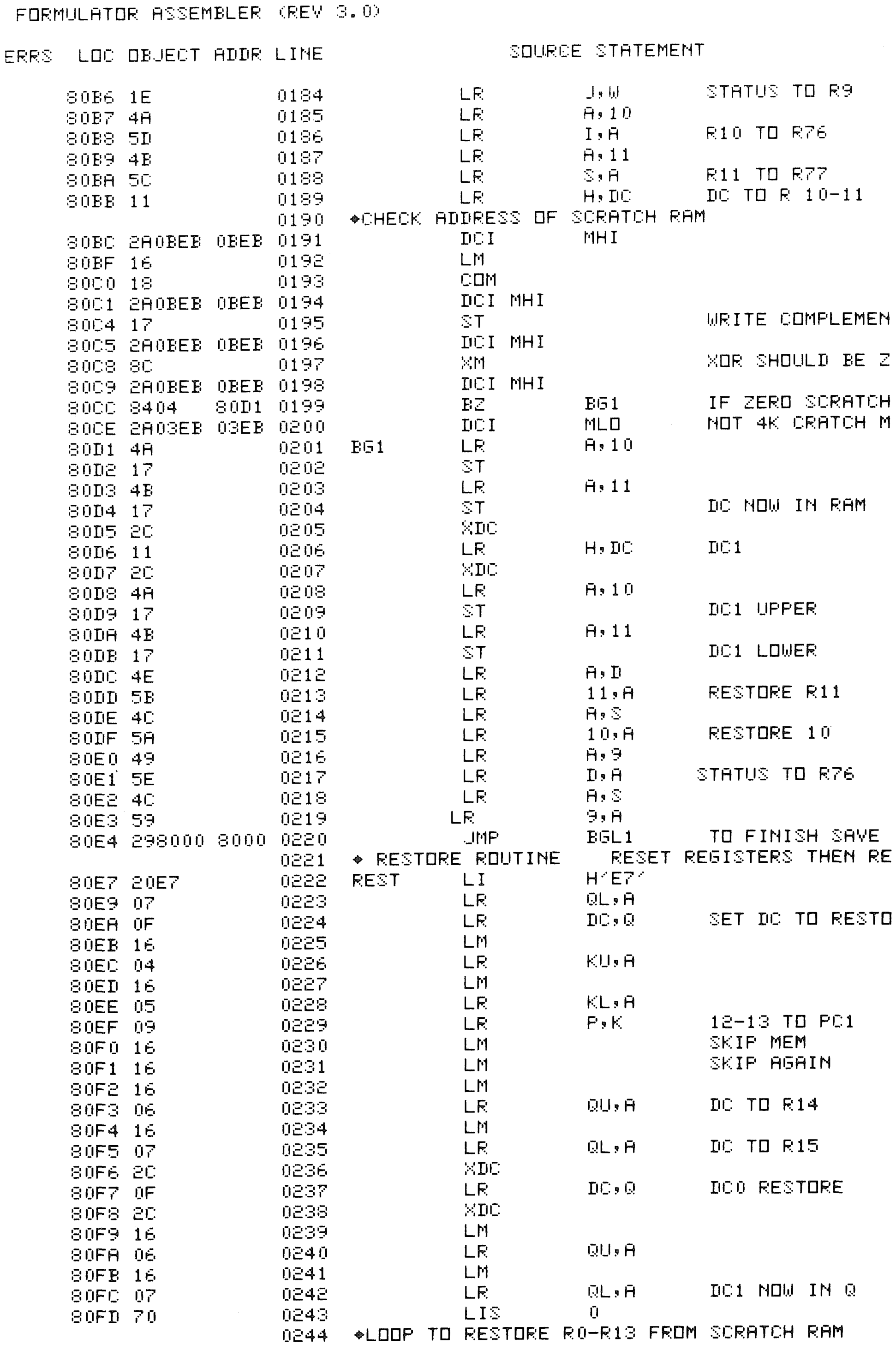
NOTE: P86 Designates Port 8, Bit 6.

APPENDIX G
ASCII CHARACTER CODES

Character	7 Bit Hex Code	Character	7 Bit Hex Code	Character	7 Bit Hex Code
(Space)	20	0	30	H	48
!	21	1	31	I	49
”	22	2	32	J	4A
#	23	3	33	K	4B
$	24	4	34	L	4C
%	25	5	35	M	4D
&	26	6	36	N	4E
’ (Quote)	27	7	37	O	4F
(	28	8	38	P	50
)	29	9	39	Q	51
*	2A	:	3A	R	52
+	2B	;	3B	S	53
, (Comma)	2C	>	3C	T	54
-	2D	=	3D	U	55
.	2E	<	3E	V	56
/	2F	?	3F	W	57
		@	40	X	58
Line Feed	0A	A	41	Y	59
Carriage RTN	0D	B	42	Z	5A
Bell	87	C	43	[	5B
Punch ON	92	D	44	\	5C
Punch OFF	94	E	45	]	5D
Reader ON	91	F	46	↑	5E
Reader OFF	93	G	47	←	5F
Null	7F				
Null	FF				

```
ERRS  LOC OBJECT ADDR LINE                SOURCE STATEMENT

                      0001  *FAIRBUG      +       EDUCATOR
                      0002  *R.VATERT             J.EFSTATHIOU
                      0003  *FOR 2K PSU
                      0004  *9/28/77
                 8000 0005  XX        RORG      H'8000'
                 82B9 0006  A         EQU       (A1-1)
                 0082 0007  HIBY      EQU       H'82'          HI 8 BIT ADD
                 00B9 0008  LOBY      EQU       H'B9'          LO 8 BIT ADD
                 0000 0009  BAUD      EQU       0
                 0001 0010  CHRS      EQU       1
                 0002 0011  BCNT      EQU       2
                 0003 0012  TEMP      EQU       3
                 0008 0013  HFLG      EQU       8
                 0005 0014  CCNT      EQU       5
                 0006 0015  XFLG      EQU       6
                 0006 0016  SAVE      EQU       XFLG
                 0007 0017  CKSM      EQU       7
                 0008 0018  FCNT      EQU       8
                 0009 0019  FLG       EQU       9
                 0007 0020  SIZE      EQU       CKSM
                 0003 0021  SA        EQU       3
                 0005 0022  EA        EQU       5
                 0006 0023  EALO      EQU       XFLG
                 0005 0024  EAHI      EQU       CCNT
                 03EB 0025  MLO       EQU       H'3EB'     END RAM 0-1K
                 0BEB 0026  MHI       EQU       H'BEB'     END RAM 3-4K
                 000C 0027  DD        EQU       12         ID DISPLAY PORT
                 000D 0028  PP        EQU       13
                 000F 0029  RR        EQU       15         FLG+H'43'=REGIS
                 000A 0030  MM        EQU       10         FLG+H'43'=MEMOR
                 00F8 0031  HI        EQU       H'F8'
                 000B 0032  CC        EQU       11
                 000B 0033  CHR1      EQU       11
                 000D 0034  CR        EQU       H'0D'
                 000A 0035  LF        EQU       H'0A'
                 000B 0036  NUSE      EQU       H'0B'      UNUSED PORT ADD
                 0008 0037  OPOR      EQU       8          SERIAL PORT,OUT
                 0008 0038  IPOR      EQU       8          SERIAL PORT,INP
                 0008 0039  PSTS      EQU       8          PARALLEL PORT,S
                 0009 0040  PPRT      EQU       9          PARALLEL PORT,D
                      0041  *LOOP TO SAVE R0-R15 IN MEMORY
      8000 70         0042  BGL1      LIS       0
      8001 0B         0043  BGLP      LR        IS,A
      8002 4C         0044            LR        A,S
      8003 17         0045            ST
      8004 0A         0046            LR        A,IS
      8005 1F         0047            INC
      8006 2510       0048            CI        16
      8008 94F8  8001 0049            BNZ       BGLP       NOT 16,NOT DONE
      800A 16         0050            LM                   BAUD
      800B 50         0051            LR        BAUD,A     SET TEMP AT LEA
      800C 20E7       0052            LI        -25
      800E 8E         0053            ADC                  SET DC
      800F 08         0054            LR        K,P        PC TO R12-13
      8010 00         0055            LR        A,KU
      8011 17         0056            ST                   SAVE PC HI
      8012 01         0057            LR        A,KL
      8013 17         0058            ST                   SAVE PC LO
      8014 71         0059            LIS       1
      8015 B8         0060            OUTS      OPOR       MARK BIT TO STA
      8016 A8         0061            INS       IPOR
```

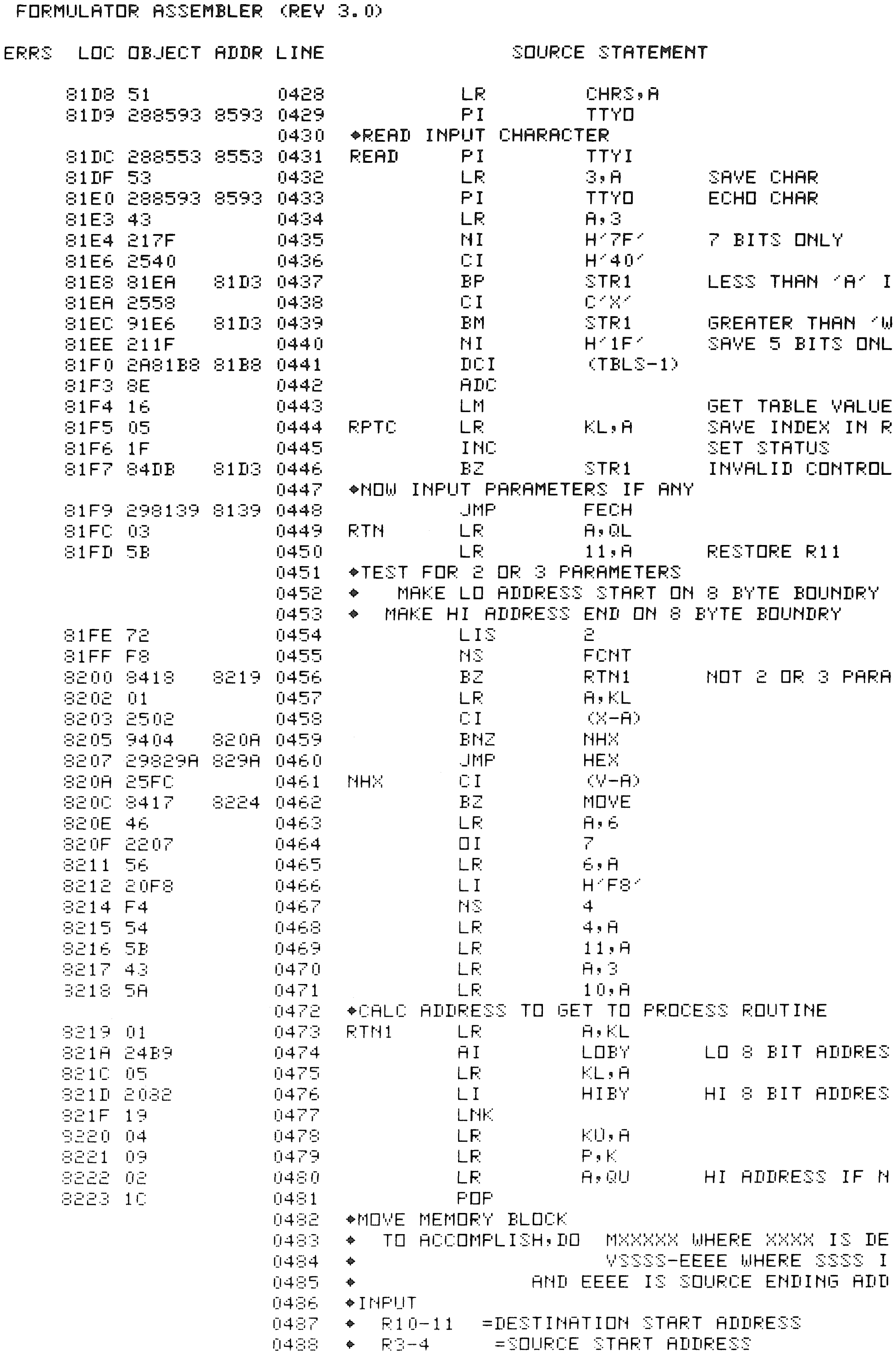
```
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                  SOURCE STATEMENT

      8017 2120        0062           NI         H'20
      8019 9419   8033 0063           BNZ        EDUL
      801B 0E          0064           LR         Q,DC       QU WILL NOW HAV
      801C A8          0065           INS        IPOR       GET BAUD OPTION
      801D 18          0066           COM
      801E 2106        0067           NI         6
      8020 12          0068           SR         1
      8021 840A   802C 0069           BZ         BGND
      8023 2A802F 802F 0070           DCI        (TBX-1)
      8026 8E          0071           ADC
      8027 16          0072           LM
      8028 50          0073           LR         BAUD,A      BAUD = TABLE VA
      8029 288560 8560 0074           PI         TTD        DELAY FOR KEY BO
      802C 2981D1 81D1 0075 BGND      JMP        STRT       TO CONVERSATIONA
      802F 2B          0076           NOP
                       0077 *
      8030 EA          0078 TBX       DC         H'EA'       1200 BAUD
      8031 A4          0079           DC         H'A4'       300 BAUD
      8032 06          0080           DC         6           110 BAUD
      8033 2985AD 85AD 0081 EDUL      JMP        (EDUX+2)
                       0082 *BOOTSTRAP FORMAT PUNCH SUBROUTINE
                       0083 *    START ADDRESS INDC ON ENTRY
                       0084 *    END ADDRESS IN R6-7 ON ENTRY
                       0085 *WRITE BLANK LEADER
      8036 45          0086 FPUN      LR         A,EAHI
      8037 18          0087           COM
      8038 55          0088           LR         EAHI,A     NOW 1'S COMPLIM
      8039 46          0089           LR         A,EALO
      803A 18          0090           COM                   NEGATIVE
      803B 1F          0091           INC                   2'S COMPLIMENT
      803C 56          0092           LR         EALO,A
      803D 45          0093           LR         A,EAHI
      803E 19          0094           LNK
      803F 55          0095           LR         EAHI,A     NOW 2'S COMPLIM
      8040 2092        0096           LI         H'92'      NUMBER OF NULLS
      8042 57          0097           LR         CKSM,A
      8043 9002   8046 0098           BR         (FPN1+1)
      8045 70          0099 FPN1      LIS        0
      8046 51          0100           LR         CHRS,A
      8047 288593 8593 0101           PI         TTYO       WRITE BLANKS
      804A 37          0102           DS         CKSM
      804B 94F9   8045 0103           BNZ        FPN1       CONTINUE BLANKI
      804D E8          0104           XS         HFLG
      804E 8109   8058 0105           BP         FCON       THIS WAS LEADER
                       0106 *TRAILER PUNCHED NOW FINISH
      8050 2094        0107           LI         H'94'
      8052 51          0108           LR         CHRS,A
      8053 288593 8593 0109           PI         TTYO
      8056 9057   80AE 0110           BR         BGSV
                       0111 *
                       0112 *WRITE STARTING ADDRESS
      8058 20FF        0113 FCON      LI         H'FF'
      805A 51          0114           LR         CHRS,A
      805B 288593 8593 0115           PI         TTYO
      805E 203A        0116           LI         H'3A'      COLON FORMULATO
      8060 51          0117           LR         CHRS,A
      8061 288593 8593 0118           PI         TTYO
      8064 2018        0119           LI         24         LENGTH=24
      8066 58          0120           LR         HFLG,A
      8067 07          0121           LR         QL,A       OUTPUT BYTE
      8068 28811D 811D 0122           PI         FOP2       WRITE LENGTH
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                SOURCE STATEMENT

     806B 4A           0123           LR          A,10
     806C 07           0124           LR          QL,A
     806D 28811D 811D  0125           PI          FOP2        OUTPUT HI ADDRE
     8070 4B           0126           LR          A,11
     8071 07           0127           LR          QL,A
     8072 28811D 811D  0128           PI          FOP2        OUTPUT LO ADDRE
     8075 70           0129           LIS         0
     8076 07           0130           LR          QL,A        RECORD TYPE=0
     8077 28811D 811D  0131           PI          FOP2
     807A 10           0132           LR          DC,H
     807B 2018         0133           LI          24
     807D 58           0134           LR          HFLG,A      SET TO 24
     807E 9003    8082 0135           BR          FLOP
                       0136 *****************
     8080 902D    80AE 0137           BR          BGSV        START FAIRBUG
                       0138 *****************
     8082 28811B 811B  0139 FLOP      PI          FOP1        WRITE 2 CHAR FR
     8085 11           0140           LR          H,DC
     8086 38           0141           DS          HFLG        COUNT-1
     8087 94FA    8082 0142           BNZ         FLOP        NOT 24 YET
                       0143 *NOW CHECKSUM
     8089 47           0144           LR          A,CKSM
     808A 18           0145           COM
     808B 1F           0146           INC
     808C 07           0147           LR          QL,A
     808D 78           0148           LIS         8
     808E 58           0149           LR          HFLG,A      FLAG PLUS FOR 2
     808F 28811D 811D  0150           PI          FOP2        WRITE 2 CHAR CK
     8092 288578 8578  0151           PI          TTCR        TYPE CR/LF
                       0152 *NOW CHECK IF ALL MEMORY IS PUNCHED,HI MUST
     8095 4B           0153           LR          A,11        LAST LOW ADDRES
     8096 C6           0154           AS          EALO        ENDING LOW ADDR
     8097 4A           0155           LR          A,10        LAST HI ADDRESS
     8098 19           0156           LNK
     8099 C5           0157           AS          EAHI
     809A 92BD    8058 0158           BNC         FCON        NOT DONE YET
     809C 07           0159           LR          QL,A        ZERO
                       0160 *
                       0161 *NOW WROTE ZERO RECORD PRIOR TO TRAILER
                       0162 *
     809D 203A         0163           LI          H'3A'       COLON
     809F 51           0164           LR          CHRS,A
     80A0 288593 8593  0165           PI          TTYO        PUNCH VCOLON
     80A3 78           0166           LIS         8
     80A4 58           0167           LR          HFLG,A
     80A5 28811D 811D  0168 HERE      PI          FOP2    PUNCH 2 ZERO CHAR
     80A8 38           0169           DS          HFLG
     80A9 94FB    80A5 0170           BNZ         HERE        REPEAT IF NOT 0
     80AB 38           0171           DS          HFLG        SET TO -1
     80AC 908B    8038 0172           BR          (FPUN+2)    TRAILER
                       0173 *TABLE OF DEFAULT DELAY COUNTERS
                       0174 *SAVE ROUTINE   TO SAVE THE INITIAL STATE OF
                       0175 *     RESTORE SUBROUTINE WILL PUT IT BACK
     80AE 1A           0176 BGSV      DI                      DISABLE INTERRU
     80AF 58           0177           LR          8,A         A SAVED IN R8
     80B0 0A           0178           LR          A,IS        ISAR TO A
     80B1 67           0179           LISU        7
     80B2 6C           0180           LISL        4
     80B3 5D           0181           LR          I,A         ISAR TO R74
     80B4 49           0182           LR          A,9
     80B5 5D           0183           LR          I,A         R9 TO R75
```



```
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE              SOURCE STATEMENT

      80B6 1E          0184            LR          J,W        STATUS TO R9
      80B7 4A          0185            LR          A,10
      80B8 5D          0186            LR          I,A        R10 TO R76
      80B9 4B          0187            LR          A,11
      80BA 5C          0188            LR          S,A        R11 TO R77
      80BB 11          0189            LR          H,DC       DC TO R 10-11
                       0190  *CHECK ADDRESS OF SCRATCH RAM
      80BC 2A0BEB 0BEB 0191            DCI         MHI
      80BF 16          0192            LM
      80C0 18          0193            COM
      80C1 2A0BEB 0BEB 0194            DCI MHI
      80C4 17          0195            ST                     WRITE COMPLEMEN
      80C5 2A0BEB 0BEB 0196            DCI MHI
      80C8 8C          0197            XM                     XOR SHOULD BE Z
      80C9 2A0BEB 0BEB 0198            DCI MHI
      80CC 8404   80D1 0199            BZ          BG1        IF ZERO SCRATCH
      80CE 2A03EB 03EB 0200            DCI         MLO        NOT 4K CRATCH M
      80D1 4A          0201  BG1       LR          A,10
      80D2 17          0202            ST
      80D3 4B          0203            LR          A,11
      80D4 17          0204            ST                     DC NOW IN RAM
      80D5 2C          0205            XDC
      80D6 11          0206            LR          H,DC       DC1
      80D7 2C          0207            XDC
      80D8 4A          0208            LR          A,10
      80D9 17          0209            ST                     DC1 UPPER
      80DA 4B          0210            LR          A,11
      80DB 17          0211            ST                     DC1 LOWER
      80DC 4E          0212            LR          A,D
      80DD 5B          0213            LR          11,A       RESTORE R11
      80DE 4C          0214            LR          A,S
      80DF 5A          0215            LR          10,A       RESTORE 10
      80E0 49          0216            LR          A,9
      80E1 5E          0217            LR          D,A       STATUS TO R76
      80E2 4C          0218            LR          A,S
      80E3 59          0219           LR           9,A
      80E4 298000 8000 0220            JMP         BGL1       TO FINISH SAVE
                       0221  * RESTORE ROUTINE      RESET REGISTERS THEN RE
      80E7 20E7        0222  REST      LI          H'E7'
      80E9 07          0223            LR          QL,A
      80EA 0F          0224            LR          DC,Q       SET DC TO RESTO
      80EB 16          0225            LM
      80EC 04          0226            LR          KU,A
      80ED 16          0227            LM
      80EE 05          0228            LR          KL,A
      80EF 09          0229            LR          P,K        12-13 TO PC1
      80F0 16          0230            LM                     SKIP MEM
      80F1 16          0231            LM                     SKIP AGAIN
      80F2 16          0232            LM
      80F3 06          0233            LR          QU,A       DC TO R14
      80F4 16          0234            LM
      80F5 07          0235            LR          QL,A       DC TO R15
      80F6 2C          0236            XDC
      80F7 0F          0237            LR          DC,Q       DC0 RESTORE
      80F8 2C          0238            XDC
      80F9 16          0239            LM
      80FA 06          0240            LR          QU,A
      80FB 16          0241            LM
      80FC 07          0242            LR          QL,A       DC1 NOW IN Q
      80FD 70          0243            LIS         0
                       0244  *LOOP TO RESTORE R0-R13 FROM SCRATCH RAM
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE               SOURCE STATEMENT

      80FE 0B          0245  LOUP     LR        IS,A
      80FF 16          0246           LM
      8100 5C          0247           LR        S,A        R0-R13 RESTORE
      8101 0A          0248           LR        A,IS
      8102 1F          0249           INC
      8103 250E        0250           CI        14
      8105 94F8   80FE 0251           BNZ       LOUP       NOT DONE YET
      8107 67          0252           LISU      7
      8108 49          0253           LR        A,9
      8109 5D          0254           LR        I,A
      810A 16          0255           LM
      810B 59          0256           LR        9,A        R14 TO R9
      810C 16          0257           LM
      810D 0F          0258           LR        DC,Q       RESTORE DC
      810E 2C          0259           XDC                  DC1 FROM DC0 TO
      810F 07          0260           LR        QL,A
      8110 49          0261           LR        A,9        RESTORE R14-15
      8111 06          0262           LR        QU,A
                       0263  *NOW DO SR,ISAR,9,A
      8112 4E          0264           LR        A,D        GET STATUS
      8113 59          0265           LR        9,A        SR TO J
      8114 1D          0266           LR        W,J        STATUS RESTORED
      8115 4E          0267           LR        A,D        GET R9
      8116 59          0268           LR        9,A        R9 RESTORED
      8117 4E          0269           LR        A,D        GET ISAR
      8118 0B          0270           LR        IS,A       ISAR RESTORED
      8119 48          0271           LR        A,8        RESTORE A
      811A 1C          0272           POP                  TO USER
                       0273  *SUBROUTINE TO OUTBYTE 1 BYTE AS 2  4 BIT AS
                       0274  *CKSM IS ACCUMULATED
                       0275  *HFLG MUST BE POSITIVE NUMBER
      811B 16          0276  FOP1     LM
      811C 07          0277           LR        QL,A
                       0278  *ENTRY WITH CHARACTER IN REG SAVE
      811D 08          0279  FOP2     LR        K,P        SAVE RETURN INK
      811E 48          0280  FLP1     LR        A,HFLG
      811F 18          0281           COM                  SET STATUS
      8120 58          0282           LR        HFLG,A     FLIP-FLOP FLAG
      8121 03          0283           LR        A,QL
      8122 9102   8125 0284           BM        FHI        DO HI 4 BITS IF
      8124 15          0285  FLO      SL        4          LOSE HI 4 BITS
      8125 14          0286  FHI      SR        4          MOVE TO LO 4 BI
                       0287  *CONVERT TO ASC 0='30',9='39',A='41',F='46'
      8126 2430        0288           AI        H'30'
      8128 2539        0289           CI        H'39'
      812A 8103   812E 0290           BP        FINT       NOT A-F
      812C 2407        0291           AI        7
      812E 51          0292  FINT     LR        CHRS,A
      812F 288593 8593 0293           PI        TTYO       TYPE FRAME
      8132 38          0294           DS        HFLG
      8133 91EA   811E 0295           BM        FLP1       DO LOW HALF WOR
                       0296  *DO CHECKSUM
      8135 03          0297           LR        A,QL       LAST BYTE OUTPU
      8136 C7          0298           AS        CKSM       ADD TO PROIR CK
      8137 57          0299           LR        CKSM,A     SAVE UPDATED CK
      8138 0C          0300           PK                   RETURN FROM K R
                       0301  *FETCH PARAMETERS AFTER LEGIT CODE IS IN
                       0302  *MAX FIELD 1 AND 2 IS 4 HEX DIGITS
                       0303  *MAX FIELD 3 IS 1 HEX DIGIT
      8139 73          0304  FECH     LIS       3
      813A 58          0305           LR        FCNT,A     PARAMETER COUNT
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE               SOURCE STATEMENT

      813B 0B          0306          LR         IS,A       ISAR=3
      813C 74          0307  FA      LIS        4
      813D 5B          0308          LR         CC,A       CHAR CNT=4
                       0309  *READ CHARACTER
      813E 288553 8553 0310  FB      PI         TTYI       GET CHAR
      8141 217F        0311          NI         H'7F'
      8143 57          0312          LR         7,A
      8144 250D        0313          CI         CR
      8146 843E   8185 0314          BZ         CORE       CORRECT FIELD B
      8148 288593 8593 0315          PI         TTYO       ECHO INPUT CHAR
      814B 47          0316          LR         A,7
      814C 252D        0317          CI         C'-'       FIELD SEPERATOR
      814E 8433   8182 0318          BZ         CORN
      8150 252B        0319          CI         C'+'
      8152 842F   8182 0320          BZ         CORN
      8154 253D        0321          CI         C'='
      8156 8430   8187 0322          BZ         CORT
      8158 255B        0323          CI         H'5B'      KILL CHAR?
      815A 8478   81D3 0324          BZ         STR1       KILL INPUT
                       0325  *MUST BE HEX 0-9, OR A-F
      815C 252F        0326          CI         H'2F'      LESS THAN ZERO
      815E 81DF   813E 0327          BP         FB         ERROR IGNORE
      8160 2546        0328          CI         H'46'      F?
      8162 91DB   813E 0329          BM         FB         ERROR IGNORE
      8164 24D0        0330          AI         H'D0'
      8166 2509        0331          CI         H'9'
      8168 8107   8170 0332          BP         FOK
      816A 2510        0333          CI         H'10'
      816C 81D1   813E 0334          BP         FB         ERROR,IGNORE IT
      816E 24F9        0335          AI         H'F9'
      8170 3B          0336  FOK     DS         CC
      8171 91CC   813E 0337          BM         FB         FULL FIELD IGNO
                       0338  *ZERO LO 4 BITS,THEN ADD THIS DIGIT
      8173 52          0339          LR         2,A
      8174 4C          0340          LR         A,S
      8175 15          0341          SL         4
      8176 C2          0342          AS         2
      8177 5D          0343          LR         I,A        TO PARAMETER LO
      8178 71          0344          LIS        1
      8179 FB          0345          NS         CC
      817A 84C3   813E 0346          BZ         FB         EVEN CHAR,2 DIG
      817C 4E          0347          LR         A,D        SET ISAR BACK 1
      817D 8FC0   813E 0348          BR7        FB         GET NEXT DIGIT
      817F 73          0349          LIS        3
      8180 9035   81B6 0350          BR         COR7       PARAMETERS FULL
      8182 59          0351  CORN    LR         9,A
      8183 9003   8187 0352          BR         CORT
                       0353  *CORRECT HEX FIELD ,RIGHT JUSTIFY
                       0354  *    IF ABOC THEN OABC
                       0355  *IF ABCD THEN IT IS OK
                       0356  *    IF A00 THEN 000A
                       0357  *IF    AB00 THEN 00AB
      8185 7D          0358  CORE    LIS        H'D'
      8186 57          0359          LR         7,A
      8187 74          0360  CORT    LIS        4
      8188 EB          0361          XS         CC
      8189 841F   81A9 0362          BZ         COR6       NO PARAMETERS
      818B 3B          0363          DS         CC
      818C 911B   81A8 0364          BM         COR5       FIELD ABCD,AOK
      818E 940C   819B 0365          BNZ        COR1
      8190 7F          0366          LIS        15
```

APPENDIX H (Continued)

```
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                SOURCE STATEMENT

      8191 FC          0367           NS         S
      8192 5E          0368           LR         D,A
      8193 4D          0369           LR         A,I
      8194 15          0370           SL         4
      8195 EC          0371           XS         S
      8196 5E          0372           LR         D,A
      8197 4C          0373           LR         A,S
      8198 14          0374           SR         4
      8199 900C   81A6 0375           BR         COR4
      819B 3B          0376  COR1     DS         CC
      819C 8405   81A2 0377           BZ         COR2
      819E 7F          0378           LIS        15
      819F FD          0379           NS         I
      81A0 9003   81A4 0380           BR         COR3
      81A2 4E          0381  COR2     LR         A,D
      81A3 4D          0382           LR         A,I
      81A4 5E          0383  COR3     LR         D,A
      81A5 70          0384           LIS        0
      81A6 5D          0385  COR4     LR         I,A
      81A7 4D          0386           LR         A,I
      81A8 38          0387  COR5     DS         FCNT
      81A9 47          0388  COR6     LR         A,7
      81AA 250D        0389           CI         CR
      81AC 8405   81B2 0390           BZ         COR8
      81AE 253D        0391           CI         C'='
      81B0 948B   813C 0392           BNZ        FA          LAST CHAR NOT C
      81B2 48          0393  COR8     LR         A,FCNT
      81B3 18          0394           COM
      81B4 2404        0395           AI         4
      81B6 58          0396  COR7     LR         FCNT,A
      81B7 9044   81FC 0397           BR         RTN
                       0398  *ROUTINE TO FETCH DIRECTIVE,THEN CALL PARAME
                       0399  *    THEN GO TO CORRECT PROCESS ROUTINE THRU
      81B9 01          0400  TBLS     DC      AL1(A1-A)      ACCUMULATOR DIS
      81BA 9E          0401           DC      AL1(B-A)       BINARY DUMP
      81BB 06          0402           DC      AL1(C-A)       CHANGE FIELD
      81BC 42          0403           DC      AL1(D-A)       DC DISPLAY
      81BD 47          0404           DC      AL1(E-A)       EXAMINE
      81BE 4D          0405           DC      AL1(F-A)       FORMATTED PUNCH
      81BF 50          0406           DC      AL1(G-A)       GO TO
      81C0 6A          0407           DC      AL1(H-A)       HIGH SPEED LOAD
      81C1 5F          0408           DC      AL1(I-A)       ISAR DISPLAY
      81C2 A2          0409           DC      AL1(J-A)       8 BIT FROM PUNC
      81C3 FF          0410           DC      AL1(-1)          K
      81C4 70          0411           DC      AL1(L-A)       LOAD
      81C5 73          0412           DC      AL1(M-A)       MEMORY DISPLAY
      81C6 76          0413           DC      AL1(N-A)       NEXT DISPLAY
      81C7 6D          0414           DC      AL1(O-A)       PORT DISPLAY
      81C8 82          0415           DC      AL1(P-A)       PC DISPLAY
      81C9 FF          0416           DC      AL1(-1)          Q
      81CA A8          0417           DC      AL1(R-A)       REGISTER DISPLA
      81CB 9A          0418           DC      AL1(S-A)       STATUS DISPLAY
      81CC 04          0419           DC      AL1(T-A)       CLEAR BREAK POI
      81CD A5          0420           DC      AL1(U-A)       BREAKPOINT
      81CE FC          0421           DC      AL1(V-A)       MOVE BLOCK
      81CF 9A          0422           DC      AL1(S-A)       W (STATUS) DISP
      81D0 02          0423  HXX      DC      AL1(X-A)       HEX ARITHMETIC
      81D1 4B          0424  STRT     LR        A,11         FREE R11
      81D2 07          0425           LR        QL,A
      81D3 288578 8578 0426  STR1     PI        TTCR         WRITE CR/LF
      81D6 203F        0427           LI        C'?'         PROMPT CHARACTE
```

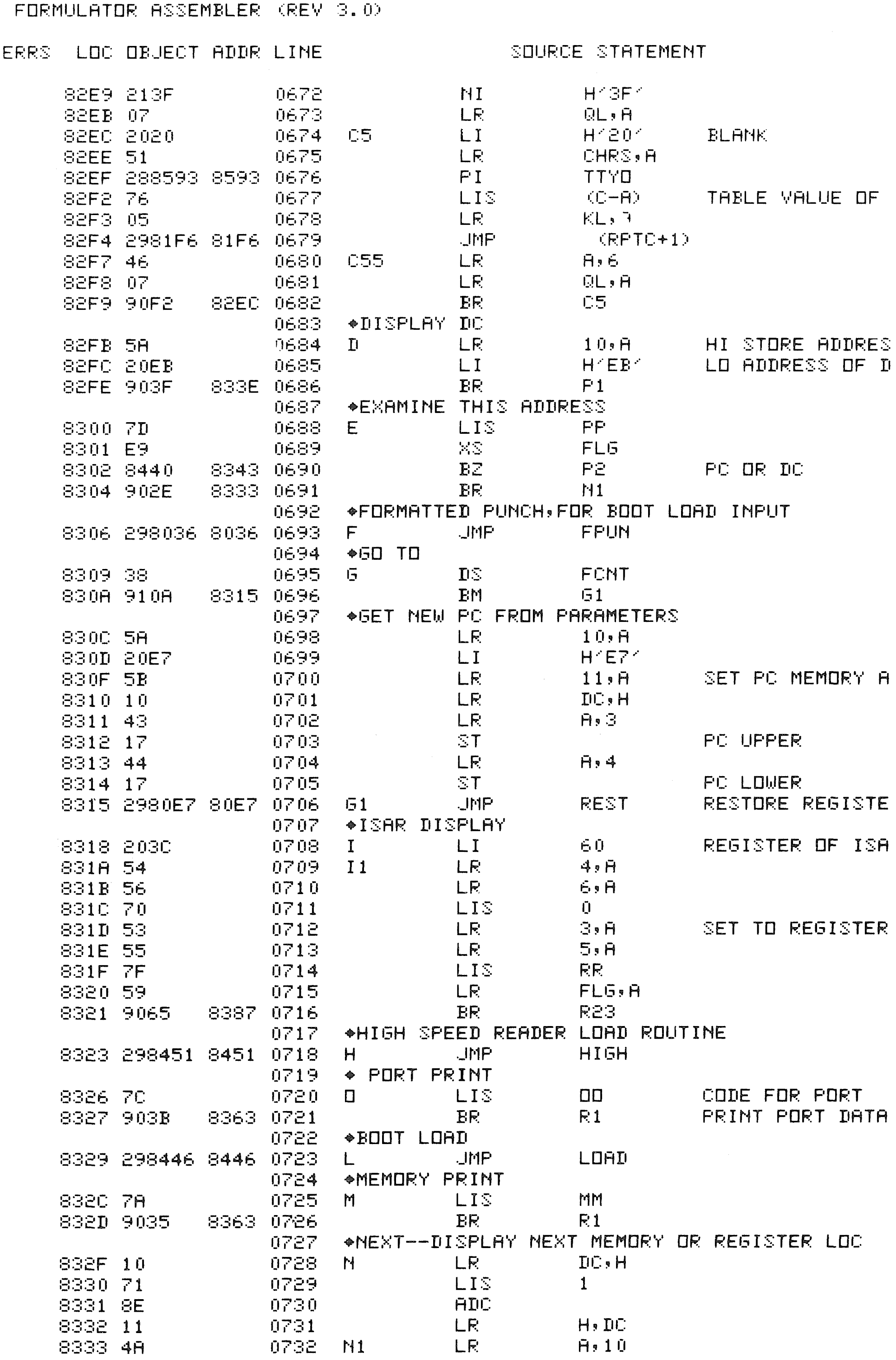


```
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE              SOURCE STATEMENT

      81D8 51          0428          LR       CHRS,A
      81D9 288593 8593 0429          PI       TTYO
                       0430  ◆READ INPUT CHARACTER
      81DC 288553 8553 0431  READ    PI       TTYI
      81DF 53          0432          LR       3,A       SAVE CHAR
      81E0 288593 8593 0433          PI       TTYO      ECHO CHAR
      81E3 43          0434          LR       A,3
      81E4 217F        0435          NI       H'7F'     7 BITS ONLY
      81E6 2540        0436          CI       H'40'
      81E8 81EA   81D3 0437          BP       STR1      LESS THAN 'A' I
      81EA 2558        0438          CI       C'X'
      81EC 91E6   81D3 0439          BM       STR1      GREATER THAN 'W
      81EE 211F        0440          NI       H'1F'     SAVE 5 BITS ONL
      81F0 2A81B8 81B8 0441          DCI      (TBLS-1)
      81F3 8E          0442          ADC
      81F4 16          0443          LM                 GET TABLE VALUE
      81F5 05          0444  RPTC    LR       KL,A      SAVE INDEX IN R
      81F6 1F          0445          INC                SET STATUS
      81F7 84DB   81D3 0446          BZ       STR1      INVALID CONTROL
                       0447  ◆NOW INPUT PARAMETERS IF ANY
      81F9 298139 8139 0448          JMP      FECH
      81FC 03          0449  RTN     LR       A,QL
      81FD 5B          0450          LR       11,A      RESTORE R11
                       0451  ◆TEST FOR 2 OR 3 PARAMETERS
                       0452  ◆   MAKE LO ADDRESS START ON 8 BYTE BOUNDRY
                       0453  ◆  MAKE HI ADDRESS END ON 8 BYTE BOUNDRY
      81FE 72          0454          LIS      2
      81FF F8          0455          NS       FCNT
      8200 8418   8219 0456          BZ       RTN1      NOT 2 OR 3 PARA
      8202 01          0457          LR       A,KL
      8203 2502        0458          CI       (X-A)
      8205 9404   820A 0459          BNZ      NHX
      8207 29829A 829A 0460          JMP      HEX
      820A 25FC        0461  NHX     CI       (V-A)
      820C 8417   8224 0462          BZ       MOVE
      820E 46          0463          LR       A,6
      820F 2207        0464          OI       7
      8211 56          0465          LR       6,A
      8212 20F8        0466          LI       H'F8'
      8214 F4          0467          NS       4
      8215 54          0468          LR       4,A
      8216 5B          0469          LR       11,A
      8217 43          0470          LR       A,3
      8218 5A          0471          LR       10,A
                       0472  ◆CALC ADDRESS TO GET TO PROCESS ROUTINE
      8219 01          0473  RTN1    LR       A,KL
      821A 24B9        0474          AI       LOBY      LO 8 BIT ADDRES
      821C 05          0475          LR       KL,A
      821D 2082        0476          LI       HIBY      HI 8 BIT ADDRES
      821F 19          0477          LNK
      8220 04          0478          LR       KU,A
      8221 09          0479          LR       P,K
      8222 02          0480          LR       A,QU      HI ADDRESS IF N
      8223 1C          0481          POP
                       0482  ◆MOVE MEMORY BLOCK
                       0483  ◆  TO ACCOMPLISH,DO  MXXXXX WHERE XXXX IS DE
                       0484  ◆                        VSSSS-EEEE WHERE SSSS I
                       0485  ◆                   AND EEEE IS SOURCE ENDING ADD
                       0486  ◆INPUT
                       0487  ◆  R10-11  =DESTINATION START ADDRESS
                       0488  ◆  R3-4      =SOURCE START ADDRESS
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                SOURCE STATEMENT

                      0489  *  R5-6     =SOURCE END ADDRESS
                      0490  *WORK REGISTERS
                      0491  *  R10-11    =CORRENT SOURCE BYTE ADDRESS
                      0492  *  R5-6      =SOURCE START IF MOVING FROM END
                      0493  *            =SOURCE END ADDRESS IF MOVE STAR
      8224 4A         0494  MOVE      LR          A,10        DESTINATION HI
      8225 18         0495            COM
      8226 1F         0496            INC
      8227 C3         0497            AS          3          SOURCE START HI
      8228 840B  8234 0498            BZ          MBY2       EQUAL,TEST LO B
      822A 910F  823A 0499            BM          MEND       MO V E BACKWARD
      822C 10         0500  MBGN      LR          DC,H
      822D 43         0501            LR          A,3
      822E 5A         0502            LR          10,A
      822F 44         0503            LR          A,4
      8230 5B         0504            LR          11,A
      8231 70         0505            LIS  0
      8232 9026  8259 0506            BR          MCOM
      8234 4B         0507  MBY2      LR          A,11
      8235 18         0508            COM
      8236 1F         0509            INC
      8237 C4         0510            AS          4
      8238 81F3  822C 0511            BP         MBGN
                      0512  *CALC LENGTH OF BLOCK
      823A 43         0513  MEND      LR          A,3
      823B 18         0514            COM
      823C 51         0515            LR          1,A
      823D 44         0516            LR          A,4
      823E 18         0517            COM
      823F 1F         0518            INC
      8240 1E         0519            LR          J,W
      8241 C6         0520            AS          6
      8242 9202  8245 0521            BNC         MXX
      8244 1E         0522            LR          J,W
      8245 CB         0523  MXX       AS          11
      8246 5B         0524            LR          11,A
      8247 41         0525            LR          A,1
      8248 19         0526            LNK
      8249 1D         0527            LR          W,J
      824A 19         0528            LNK
      824B C5         0529            AS          5
      824C CA         0530            AS          10
      824D 5A         0531            LR          10,A
                      0532  *MOVE TO DC THEN IONCREMENT TO ADD 1 TO LENG
      824E 10         0533            LR          DC,H
                      0534  *MOVE SOURCE START ADDRESS TO R10-11
      824F 45         0535            LR          A,5
      8250 5A         0536            LR          10,A
      8251 46         0537            LR          A,6
      8252 5B         0538            LR          11,A
                      0539  *NOW MOVE SOURCE START TO R5-6 FOR END COMPA
      8253 43         0540            LR          A,3
      8254 55         0541            LR          5,A
      8255 44         0542            LR          A,4
      8256 56         0543            LR          6,A
      8257 20FE       0544            LI          -2         MEMORY ADDRESS
      8259 57         0545  MCOM      LR          7,A
      825A 2C         0546            XDC
      825B 10         0547            LR          DC,H
      825C 11         0548  MLOP      LR          H,DC       SAVE ADDRESS FO
      825D 16         0549            LM
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE              SOURCE STATEMENT

      825E 2C          0550           XDC
      825F 17          0551           ST
      8260 47          0552           LR        A,7
      8261 8E          0553           ADC
      8262 2C          0554           XDC
      8263 8E          0555           ADC
                       0556 *COMPARE FOR ADDRESS END
      8264 4A          0557           LR        A,10
      8265 E5          0558           XS        5
      8266 94F5   825C 0559           BNZ       MLOP
      8268 4B          0560           LR        A,11
      8269 E6          0561           XS        6
      826A 94F1   825C 0562           BNZ       MLOP
      826C 2981D1 81D1 0563 BXX       JMP       STRT      ALL DONE
                       0564 *BREAKPOINT----STORE PK IN USER MEMORY +CHAN
                       0565 *               ADDRESS IN FAIRBUG
                       0566 *GET BYTE FROM MEMORY AAND SAVE ADDRESS AND
      826F 20E0        0567 BRAK      LI        H'E0'     SCRATCH MEM LO
      8271 07          0568           LR        QL,A
      8272 0F          0569           LR        DC,Q
      8273 43          0570           LR        A,3
      8274 5A          0571           LR        10,A
      8275 17          0572           ST
      8276 44          0573           LR        A,4
      8277 5B          0574           LR        11,A
      8278 17          0575           ST
                       0576 *NOW FETCH USER BYTE
      8279 2C          0577           XDC       LR        DC,H
      827A 10          0578           LR        DC,H
      827B 16          0579           LM
      827C 2C          0580           XDC
      827D 17          0581           ST
                       0582 *NOW STORE PK IN MEMORY
      827E 10          0583           LR        DC,H
      827F 7C          0584           LIS       12
      8280 17          0585           ST
                       0586 *NOW CHANGE USER K TO RESTORE ADDRESS
      8281 20FB        0587           LI        H'FB'
      8283 07          0588           LR        QL,A
      8284 0F          0589           LR        DC,Q
      8285 2080        0590           LI        H'80'
      8287 17          0591           ST
      8288 20AF        0592           LI        H'AF'
      828A 17          0593           ST
      828B 90E0   826C 0594           BR        BXX       TO START
                       0595 *RESTORE BYTE TO USER MEMORY
      828D 20E0        0596 TT        LI        H'E0'
      828F 07          0597           LR        QL,A
      8290 0F          0598           LR        DC,Q
      8291 16          0599           LM
      8292 5A          0600           LR        10,A
      8293 16          0601           LM
      8294 5B          0602           LR        11,A
      8295 16          0603           LM
      8296 10          0604·          LR        DC,H      USER ADDRESS
      8297 17          0605           ST
      8298 90D3   826C 0606           BR        BXX       TO START
                       0607 *HEX ARITHMETIC
      829A 49          0608 HEX       LR        A,9
      829B 252B        0609           CI        C'+'
      829D 840B   82A9 0610           BZ        PLUS
```

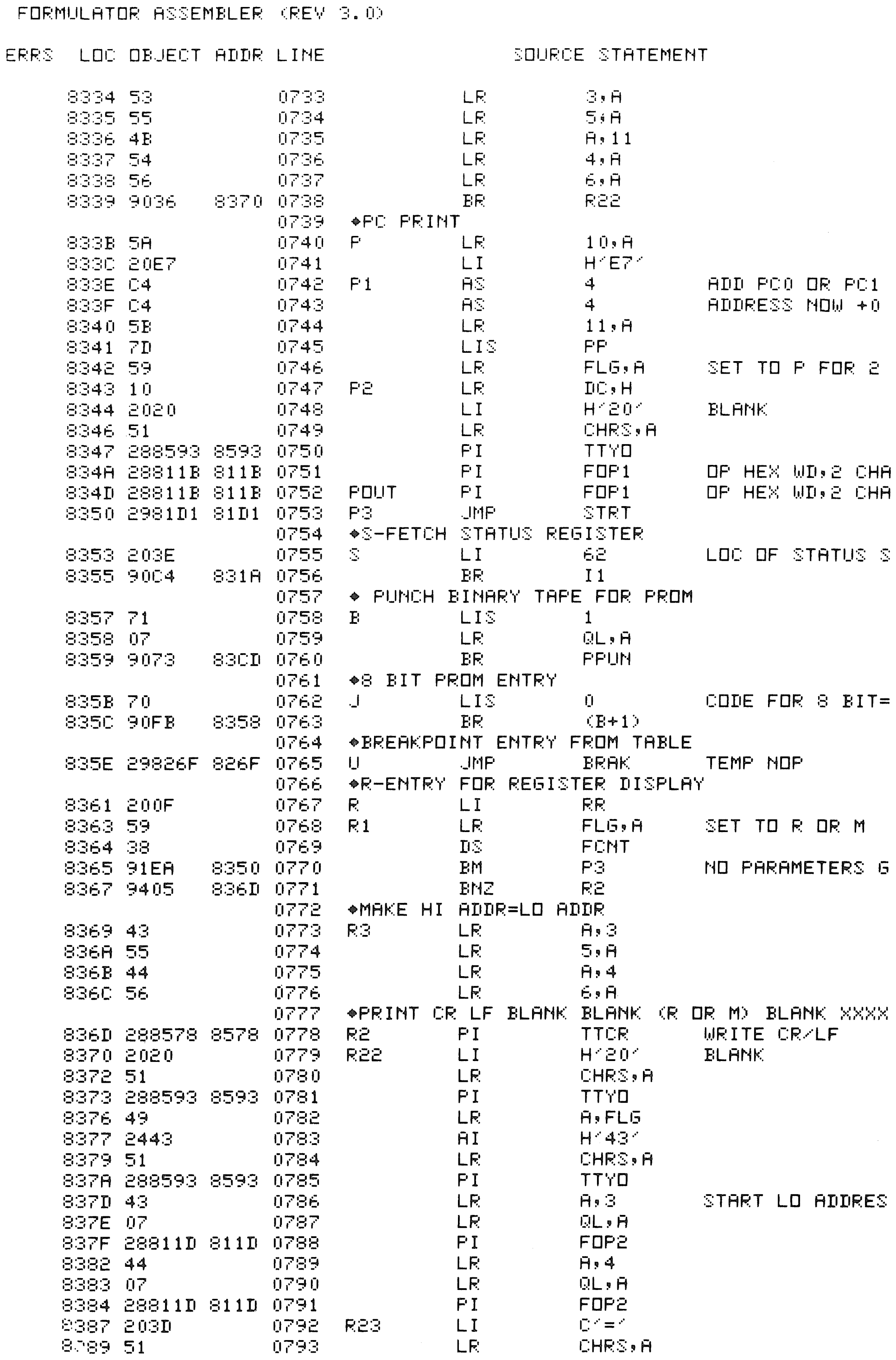
APPENDIX H (Continued)

```
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE              SOURCE STATEMENT

                      0611  *COMPLIMENT BEFORE ADD=SUBTRACT
      829F 46         0612            LR        A,6
      82A0 18         0613            COM
      82A1 1F         0614            INC                 2'S COMPLIMENT
      82A2 1E         0615            LR        J,W
      82A3 56         0616            LR        6,A
      82A4 45         0617            LR        A,5
      82A5 18         0618            COM
      82A6 1D         0619            LR        W,J
      82A7 19         0620            LNK
      82A8 55         0621            LR        5,A
                      0622  *ADDITION
      82A9 44         0623  PLUS      LR        A,4
      82AA C6         0624            AS        6
      82AB 54         0625            LR        4,A
      82AC 43         0626            LR        A,3
      82AD 19         0627            LNK
      82AE C5         0628            AS        5
                      0629  *NOW WRITE TO DISPLAY RESULTS
      82AF 07         0630            LR        QL,A
      82B0 28811D 811D 0631           PI        FOP2
      82B3 44         0632            LR        A,4
      82B4 07         0633            LR        QL,A
      82B5 28811D 811D 0634  V        PI        FOP2
      82B8 90B3   826C 0635           BR        BXX
                      0636  *PRINT ACCUMULATOR
      82BA 78         0637  A1        LIS       8
      82BB 905E   831A 0638  X        BR        I1
                      0639  *CLEAR BRAKE POINT ENTRY
      82BD 90CF   828D 0640  T        BR        TT
                      0641  *CHANGE LAST USED FIELD
      82BF 38         0642  C         DS        FCNT
      82C0 91F7   82B8 0643           BM        (V+3)       NO PARAMETERS S
      82C2 49         0644            LR        A,FLG
      82C3 10         0645            LR        DC,H
      82C4 250C       0646            CI        OO          CK FOR PORT
      82C6 9404   82CB 0647           BNZ       C0          NOT PORT CHANGE
      82C8 2984E6 84E6 0648           JMP       PORT        GO CHANGE PORT
      82CB 250D       0649  C0        CI        PP
      82CD 910B   82D9 0650           BM        C2          REGISTER CHANGE
      82CF 9403   82D3 0651           BNZ       C1          MEMORY CHANGE 1
      82D1 43         0652            LR        A,3
      82D2 17         0653            ST                    PC OR DC 1ST HA
      82D3 44         0654  C1        LR        A,4
      82D4 17         0655            ST
      82D5 11         0656            LR        H,DC
      82D6 4B         0657            LR        A,11
      82D7 9013   82EB 0658           BR        (C3+4)
      82D9 4B         0659  C2        LR        A,11
      82DA 0B         0660            LR        IS,A
      82DB 24EF       0661            AI        H'EF'
      82DD 07         0662            LR        QL,A
      82DE 0F         0663            LR        DC,Q
      82DF 1F         0664            INC
      82E0 44         0665            LR        A,4
      82E1 8104   82E6 0666           BP        C4
      82E3 17         0667            ST
      82E4 9002   82E7 0668           BR        C3
      82E6 5C         0669  C4        LR        S,A
      82E7 4B         0670  C3        LR        A,11
      82E8 1F         0671            INC
```

APPENDIX H (Continued)

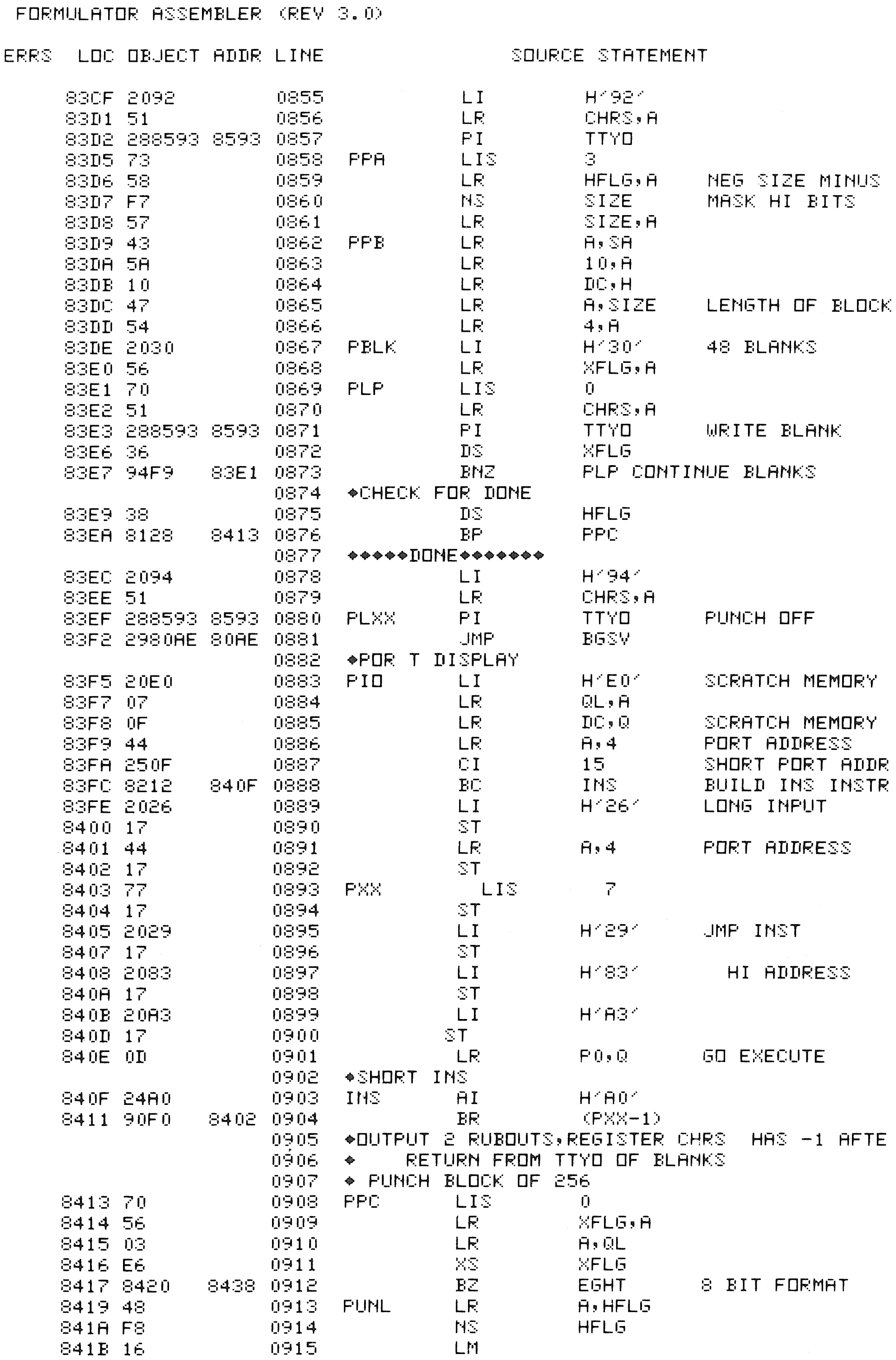


```
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE              SOURCE STATEMENT

      82E9 213F        0672         NI        H'3F'
      82EB 07          0673         LR        QL,A
      82EC 2020        0674  C5     LI        H'20'     BLANK
      82EE 51          0675         LR        CHRS,A
      82EF 288593 8593 0676         PI        TTYO
      82F2 76          0677         LIS       (C-A)     TABLE VALUE OF
      82F3 05          0678         LR        KL,A
      82F4 2981F6 81F6 0679         JMP        (RPTC+1)
      82F7 46          0680  C55    LR        A,6
      82F8 07          0681         LR        QL,A
      82F9 90F2   82EC 0682         BR        C5
                       0683  *DISPLAY DC
      82FB 5A          0684  D      LR        10,A      HI STORE ADDRES
      82FC 20EB        0685         LI        H'EB'     LO ADDRESS OF D
      82FE 903F   833E 0686         BR        P1
                       0687  *EXAMINE THIS ADDRESS
      8300 7D          0688  E      LIS       PP
      8301 E9          0689         XS        FLG
      8302 8440   8343 0690         BZ        P2        PC OR DC
      8304 902E   8333 0691         BR        M1
                       0692  *FORMATTED PUNCH,FOR BOOT LOAD INPUT
      8306 298036 8036 0693  F      JMP       FPUN
                       0694  *GO TO
      8309 38          0695  G      DS        FCNT
      830A 910A   8315 0696         BM        G1
                       0697  *GET NEW PC FROM PARAMETERS
      830C 5A          0698         LR        10,A
      830D 20E7        0699         LI        H'E7'
      830F 5B          0700         LR        11,A      SET PC MEMORY A
      8310 10          0701         LR        DC,H
      8311 43          0702         LR        A,3
      8312 17          0703         ST                  PC UPPER
      8313 44          0704         LR        A,4
      8314 17          0705         ST                  PC LOWER
      8315 2980E7 80E7 0706  G1     JMP       REST      RESTORE REGISTE
                       0707  *ISAR DISPLAY
      8318 203C        0708  I      LI        60        REGISTER OF ISA
      831A 54          0709  I1     LR        4,A
      831B 56          0710         LR        6,A
      831C 70          0711         LIS       0
      831D 53          0712         LR        3,A       SET TO REGISTER
      831E 55          0713         LR        5,A
      831F 7F          0714         LIS       RR
      8320 59          0715         LR        FLG,A
      8321 9065   8387 0716         BR        R23
                       0717  *HIGH SPEED READER LOAD ROUTINE
      8323 298451 8451 0718  H      JMP       HIGH
                       0719  * PORT PRINT
      8326 7C          0720  O      LIS       OO        CODE FOR PORT
      8327 903B   8363 0721         BR        R1        PRINT PORT DATA
                       0722  *BOOT LOAD
      8329 298446 8446 0723  L      JMP       LOAD
                       0724  *MEMORY PRINT
      832C 7A          0725  M      LIS       MM
      832D 9035   8363 0726         BR        R1
                       0727  *NEXT--DISPLAY NEXT MEMORY OR REGISTER LOC
      832F 10          0728  N      LR        DC,H
      8330 71          0729         LIS       1
      8331 8E          0730         ADC
      8332 11          0731         LR        H,DC
      8333 4A          0732  N1     LR        A,10
```



```
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                 SOURCE STATEMENT

      8334 53           0733          LR         3,A
      8335 55           0734          LR         5,A
      8336 4B           0735          LR         A,11
      8337 54           0736          LR         4,A
      8338 56           0737          LR         6,A
      8339 9036    8370 0738          BR         R22
                        0739    *PC PRINT
      833B 5A           0740    P     LR         10,A
      833C 20E7         0741          LI         H'E7'
      833E C4           0742    P1    AS         4          ADD PC0 OR PC1
      833F C4           0743          AS         4          ADDRESS NOW +0
      8340 5B           0744          LR         11,A
      8341 7D           0745          LIS        PP
      8342 59           0746          LR         FLG,A      SET TO P FOR 2
      8343 10           0747    P2    LR         DC,H
      8344 2020         0748          LI         H'20'      BLANK
      8346 51           0749          LR         CHRS,A
      8347 288593 8593  0750          PI         TTYO
      834A 28811B 811B  0751          PI         FOP1       OP HEX WD,2 CHA
      834D 28811B 811B  0752    POUT  PI         FOP1       OP HEX WD,2 CHA
      8350 2981D1 81D1  0753    P3    JMP        STRT
                        0754    *S-FETCH STATUS REGISTER
      8353 203E         0755    S     LI         62         LOC OF STATUS S
      8355 90C4    831A 0756          BR         I1
                        0757    * PUNCH BINARY TAPE FOR PROM
      8357 71           0758    B     LIS        1
      8358 07           0759          LR         QL,A
      8359 9073    83CD 0760          BR         PPUN
                        0761    *8 BIT PROM ENTRY
      835B 70           0762    J     LIS        0          CODE FOR 8 BIT=
      835C 90FB    8358 0763          BR         (B+1)
                        0764    *BREAKPOINT ENTRY FROM TABLE
      835E 29826F 826F  0765    U     JMP        BRAK       TEMP NOP
                        0766    *R-ENTRY FOR REGISTER DISPLAY
      8361 200F         0767    R     LI         RR
      8363 59           0768    R1    LR         FLG,A      SET TO R OR M
      8364 38           0769          DS         FCNT
      8365 91EA    8350 0770          BM         P3         NO PARAMETERS G
      8367 9405    836D 0771          BNZ        R2
                        0772    *MAKE HI ADDR=LO ADDR
      8369 43           0773    R3    LR         A,3
      836A 55           0774          LR         5,A
      836B 44           0775          LR         A,4
      836C 56           0776          LR         6,A
                        0777    *PRINT CR LF BLANK BLANK (R OR M) BLANK XXXX
      836D 288578 8578  0778    R2    PI         TTCR       WRITE CR/LF
      8370 2020         0779    R22   LI         H'20'      BLANK
      8372 51           0780          LR         CHRS,A
      8373 288593 8593  0781          PI         TTYO
      8376 49           0782          LR         A,FLG
      8377 2443         0783          AI         H'43'
      8379 51           0784          LR         CHRS,A
      837A 288593 8593  0785          PI         TTYO
      837D 43           0786          LR         A,3        START LO ADDRES
      837E 07           0787          LR         QL,A
      837F 28811D 811D  0788          PI         FOP2
      8382 44           0789          LR         A,4
      8383 07           0790          LR         QL,A
      8384 28811D 811D  0791          PI         FOP2
      8387 203D         0792    R23   LI         C'='
      8389 51           0793          LR         CHRS,A
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE               SOURCE STATEMENT

      838A 288593 8593 0794            PI        TTYO
                       0795 *NOW PRINT DATA
      838D 43          0796 R4         LR        A,3
      838E 5A          0797            LR        10,A
      838F 44          0798            LR        A,4
      8390 5B          0799            LR        11,A
      8391 49          0800            LR        A,FLG
      8392 250C        0801            CI        00        PORT?
      8394 8460   83F5 0802            BZ        PIO       GO TO PORT ROUT
      8396 250F        0803            CI        RR        TEST FOR REGIST
      8398 9414   83AD 0804            BNZ       R5        MEMORY LOC
                       0805 *
                       0806 *REGISTER FETCH IS IT IN SCRATCH OR TEMP MEM
      839A 44          0807            LR        A,4
      839B 0B          0808            LR        IS,A
      839C 0A          0809            LR        A,IS
      839D 250F        0810            CI        15
      839F 8108   83A8 0811            BP        R44       NOT REGISTER AD
                       0812 *LOAD ISAR AND FETCH
      83A1 4C          0813            LR        A,S
      83A2 07          0814            LR        QL,A
      83A3 28811D 811D 0815 R76          PI        FOP2
      83A6 900A   83B1 0816            BR        R77
                       0817 *UPDATE REG 0-15 ADDRESS TO MEM LOC
      83A8 24EF        0818 R44        AI        H'EF'     LO RAM-START OF
      83AA 5B          0819            LR        11,A
      83AB 02          0820            LR        A,QU      HI ADDRESS
      83AC 5A          0821            LR        10,A      HI RAM ADDRESS
                       0822 *FETCH DATA FROM MEMORY
      83AD 10          0823 R5         LR        DC,H
                       0824 *DATA NOW IN NOW PRINT
      83AE 28811B 811B 0825 R6         PI        FOP1      TYPE 2 CHAR FRO
      83B1 2020        0826 R77        LI        H'20'     BLANK
      83B3 51          0827            LR        CHRS,A
      83B4 288593 8593 0828            PI        TTYO      BLANK
                       0829 *CHECK FOR END
      83B7 44          0830            LR        A,4
      83B8 5B          0831            LR        11,A      SAVE FOR REENTR
      83B9 E6          0832            XS        6
      83BA 43          0833            LR        A,3
      83BB 5A          0834            LR        10,A      SAVE FOR REENTR
      83BC 9404   83C1 0835            BNZ       R7
      83BE E5          0836            XS        5
      83BF 8490   8350 0837            BZ        P3
      83C1 44          0838 R7         LR        A,4
                       0839 *UPDATE FOR NEXT WORD,LINE HAS MAX OF 8
      83C2 1F          0840            INC
      83C3 54          0841            LR        4,A       INCREMENT ADDRE
      83C4 43          0842            LR        A,3
      83C5 19          0843            LNK
      83C6 53          0844            LR        3,A       INCR HI ADDRESS
      83C7 77          0845            LIS       7
      83C8 F4          0846            NS        4
      83C9 84A3   836D 0847            BZ        R2        NEW LINE
      83CB 90C1   838D 0848            BR        R4        CONTINUE THIS L
                       0849 *PUNCH HEADER,THEN 256 X SIZE. HI 4 BITS,
                       0850 *  THEN HEADER 256 X SIZE LO BITS.
                       0851 *  CONTINUE ALTERNATING AS ABOVE UNTIL LAST
                       0852 *  HFLG INITIALLY=SIZE,WHEN=0 PUNCH LEADER,W
      83CD 70          0853 PPUN       LIS       0
      83CE 5B          0854            LR        11,A      LOW DC ADDRESS
```

APPENDIX H (Continued)



```
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE              SOURCE STATEMENT

      83CF 2092        0855          LI         H'92'
      83D1 51          0856          LR         CHRS,A
      83D2 288593 8593 0857          PI         TTYO
      83D5 73          0858  PPA     LIS        3
      83D6 58          0859          LR         HFLG,A    NEG SIZE MINUS
      83D7 F7          0860          NS         SIZE      MASK HI BITS
      83D8 57          0861          LR         SIZE,A
      83D9 43          0862  PPB     LR         A,SA
      83DA 5A          0863          LR         10,A
      83DB 10          0864          LR         DC,H
      83DC 47          0865          LR         A,SIZE    LENGTH OF BLOCK
      83DD 54          0866          LR         4,A
      83DE 2030        0867  PBLK    LI         H'30'     48 BLANKS
      83E0 56          0868          LR         XFLG,A
      83E1 70          0869  PLP     LIS        0
      83E2 51          0870          LR         CHRS,A
      83E3 288593 8593 0871          PI         TTYO      WRITE BLANK
      83E6 36          0872          DS         XFLG
      83E7 94F9   83E1 0873          BNZ        PLP CONTINUE BLANKS
                       0874  *CHECK FOR DONE
      83E9 38          0875          DS         HFLG
      83EA 8128   8413 0876          BP         PPC
                       0877  *****DONE*******
      83EC 2094        0878          LI         H'94'
      83EE 51          0879          LR         CHRS,A
      83EF 288593 8593 0880  PLXX    PI         TTYO      PUNCH OFF
      83F2 2980AE 80AE 0881          JMP        BGSV
                       0882  *POR T DISPLAY
      83F5 20E0        0883  PIO     LI         H'E0'     SCRATCH MEMORY
      83F7 07          0884          LR         QL,A
      83F8 0F          0885          LR         DC,Q      SCRATCH MEMORY
      83F9 44          0886          LR         A,4       PORT ADDRESS
      83FA 250F        0887          CI         15        SHORT PORT ADDR
      83FC 8212   840F 0888          BC         INS       BUILD INS INSTR
      83FE 2026        0889          LI         H'26'     LONG INPUT
      8400 17          0890          ST
      8401 44          0891          LR         A,4       PORT ADDRESS
      8402 17          0892          ST
      8403 77          0893  PXX        LIS        7
      8404 17          0894          ST
      8405 2029        0895          LI         H'29'     JMP INST
      8407 17          0896          ST
      8408 2083        0897          LI         H'83'       HI ADDRESS
      840A 17          0898          ST
      840B 20A3        0899          LI         H'A3'
      840D 17          0900         ST
      840E 0D          0901          LR         P0,Q      GO EXECUTE
                       0902  *SHORT INS
      840F 24A0        0903  INS     AI         H'A0'
      8411 90F0   8402 0904          BR         (PXX-1)
                       0905  *OUTPUT 2 RUBOUTS,REGISTER CHRS  HAS -1 AFTE
                       0906  *    RETURN FROM TTYO OF BLANKS
                       0907  * PUNCH BLOCK OF 256
      8413 70          0908  PPC     LIS        0
      8414 56          0909          LR         XFLG,A
      8415 03          0910          LR         A,QL
      8416 E6          0911          XS         XFLG
      8417 8420   8438 0912          BZ         EGHT      8 BIT FORMAT
      8419 48          0913  PUNL    LR         A,HFLG
      841A F8          0914          NS         HFLG
      841B 16          0915          LM
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE               SOURCE STATEMENT

      841C 9402   841F 0916            BNZ       *+3
      841E 15          0917            SL        4        TO HIY 4 BITS
      841F 14          0918            SR        4        TO LO 4 BITS
      8420 234F        0919            XI        H'4F'
      8422 51          0920            LR        CHRS,A
      8423 288593 8593 0921            PI        TTYO
      8426 36          0922            DS        XFLG
      8427 94F1   8419 0923            BNZ       PUNL
                       0924  *CHECK FOR LENGTH GREATER THAN 256
      8429 34          0925            DS        4        NO PAGES/BLOCK
      842A 81E8   8413 0926            BP        PPC      NEXT 256
                       0927  *CHECK IF LOW BYTES ALREADY DONE
      842C 38          0928            DS        HFLG
      842D 81AB   83D9 0929            BP        PPB      GO DO LOW BYTES
                       0930  *UPDATE TO NEXT PAGE
      842F 43          0931  PONT      LR        A,SA
      8430 1F          0932            INC
      8431 C7          0933            AS        SIZE
      8432 53          0934            LR        SA,A
      8433 E5          0935            XS        EA       END PAGE
      8434 84A9   83DE 0936            BZ        PBLK     DO TRAILER
      8436 909E   83D5 0937            BR        PPA      START NEXT HI B
                       0938  *8 BIT FORMAT
      8438 16          0939  EGHT      LM
      8439 51          0940            LR        CHRS,A
      843A 288593 8593 0941            PI        TTYO     OUTPUT BYTE
      843D 36          0942            DS        XFLG
      843E 94F9   8438 0943            BNZ       EGHT     NEXT BYTE
      8440 34          0944            DS        4        PAGES PER BLOCK
      8441 81D1   8413 0945            BP        PPC      START NEXT BLOC
      8443 58          0946            LR        HFLG,A
      8444 90EA   842F 0947            BR        PONT     NEXT PAGE
                       0948  *  START OF BOOT LOAD
      8446 2091        0949  LOAD      LI        H'91'    READER ON COMMA
      8448 51          0950            LR        CHRS,A   PASS IT
      8449 288593 8593 0951            PI        TTYO     AND TYPE IT
      844C 2A8553 8553 0952            DCI       TTYI     SERIAL INPUT PR
      844F 900C   845C 0953            BR        BOT1     SKIP OVER PARAL
      8451 2A853D 853D 0954  HIGH      DCI       PINP
      8454 70          0955            LIS       0
      8455 B9          0956  SLF1      OUTS      PPRT     INITIALIZE PARA
      8456 B8          0957            OUTS      PSTS     INIT CONTROL PO
      8457 A8          0958            INS       PSTS
      8458 12          0959            SR        1
      8459 15          0960            SL        4
      845A 91FA   8455 0961            BM        SLF1     LOCK IN LOOP TI
      845C 70          0962  BOT1      LIS       0
      845D 56          0963            LR        XFLG,A   CLEAR FIRST X D
      845E 0E          0964            LR        Q,DC     SET TO TTYI OR
      845F 57          0965            LR        CKSM,A
      8460 28853C 853C 0966  IDLE      PI        CHAR     GET HEADER CHAR
      8463 13          0967            SL        1        CLEAR PARITY BI
      8464 12          0968            SR        1
      8465 253A        0969            CI        H'3A'    COLON
      8467 844F   84B7 0970            BZ        FORM     FORMULATOR FORM
      8469 2553        0971            CI        C'S'     IS IT AN LOAD A
      846B 8431   849D 0972            BZ        SETA
      846D 252A        0973            CI        C'*'     IS IT THE END O
      846F 8438   84A8 0974            BZ        ENDX
      8471 2358        0975            XI        C'X'     WELL, IF IT ISN
      8473 94EC   8460 0976            BNZ       IDLE
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                SOURCE STATEMENT

                      0977  ******  HAVE THE START OF A DATA LINE   ***
     8475 57          0978  DATA    LR       CKSM,A    INITIALIZE CHK
     8476 78          0979          LIS      H'08'
     8477 55          0980          LR       CCNT,A    INITIALIZE BYTE
     8478 56          0981          LR       XFLG,A    SHOW THAT X HAS
     8479 288515 8515 0982  CONT    PI       BYTE
     847C 17          0983          ST                 STORE THE BYTE
     847D 35          0984          DS       CCNT
     847E 94FA   8479 0985          BNZ      CONT
     8480 28853C 853C 0986          PI       CHAR      GET CHK CHAR FR
     8483 213F        0987          NI       H'3F'     MASK TO SIX BIT
     8485 24D0        0988          AI       H'D0'     ASCII CONVERT--
     8487 8203   848B 0989          BC       *+4       CARRY IF IT WAS
     8489 2439        0990          AI       H'39'     FINISH CONVERSI
     848B E7          0991          XS       CKSM      MAKE THE COMPAR
     848C 15          0992          SL       4         LO 4 BITS TO HI
     848D 84D2   8460 0993          BZ       IDLE     IF OK , LET'S GE
                      0994  **********     CHK SUM ERROR HALT    ********
     848F 2043        0995  SLF2    LI       C'C'
     8491 51          0996          LR       CHRS,A
     8492 288593 8593 0997          PI       TTYO
     8495 204B        0998          LI       C'K'
     8497 51          0999          LR       CHRS,A
     8498 288593 8593 1000          PI       TTYO
     849B 900F   84AB 1001          BR       STPP
     849D 288515 8515 1002  SETA    PI       BYTE      GET NEW LOAD AD
     84A0 5A          1003          LR       10,A
     84A1 288515 8515 1004          PI       BYTE
     84A4 5B          1005          LR       11,A
     84A5 10          1006          LR       DC,H       SET THE ADDRES
     84A6 90B9   8460 1007          BR       IDLE
     84A8 F6          1008  ENDX    NS       XFLG      HAVE AN *, BUT
     84A9 84B6   8460 1009          BZ       IDLE      MUST BE ONLY AT
                      1010                             WAS 2A+0  OR 2A
                      1011  *
                      1011  *********    HALT LOOP FOR WHEN FINISHED  **
     84AB 2093        1012  STPP    LI       H'93'     READER OFF COMM
     84AD 51          1013          LR       CHRS,A               PASS
     84AE 2983EF 83EF 1014  SLF3    JMP      PLXX
     84B1 2B          1015       NOP
     84B2 2B          1016          NOP
     84B3 2B          1017          NOP
     84B4 2B          1018          NOP
     84B5 2B          1019          NOP
     84B6 2B          1020          NOP
     84B7 70          1021  FORM    LIS    0
     84B8 57          1022          LR     CKSM,A  ZERO CKSM TO START
     84B9 18          1023          COM
     84BA 56          1024          LR XFLG,A
     84BB 288515 8515 1025          PI     BYTE      LENGTH OF BLOCK
     84BE 55          1026          LR       CCNT,A
     84BF F5          1027  XS      NS       CCNT
     84C0 84EA   84AB 1028          BZ       STPP      ZERO RECORD IS
     84C2 288515 8515 1029          PI       BYTE      ADDRESS HI
     84C5 5A          1030          LR       10,A
     84C6 288515 8515 1031          PI       BYTE      ADDRESS LO
     84C9 5B          1032          LR       11,A
     84CA 10          1033          LR       DC,H
     84CB 288515 8515 1034          PI       BYTE      CODE BYTE
     84CE FB          1035          NS       CHR1
     84CF 9490   8460 1036          BNZ      IDLE      NOT DATA RECORD
     84D1 288515 8515 1037  FMLP    PI       BYTE      DATA FETCH
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                 SOURCE STATEMENT

      84D4 17           1038            ST
      84D5 35           1039            DS       CCNT        LENGTH
      84D6 94FA   84D1  1040            BNZ       FMLP       LOOP FOR MORE D
      84D8 2B           1041            NOP
      84D9 2B           1042            NOP
      84DA 2B           1043            NOP
      84DB 2B           1044            NOP
      84DC 288515 8515  1045            PI     BYTE       FETCH CKSM
      84DF 47           1046            LR     A,CKSM
      84E0 F7           1047            NS     CKSM       SHOULD BE ZERO
      84E1 94AD   848F  1048            BNZ        SLF2       CKSM ERROR
      84E3 298460 8460  1049            JMP        IDLE       GET NEXT BLOCK
                        1050   *PORT CHANGE ROUTINE
      84E6 20DF         1051   PORT     LI         H'DF'
      84E8 07           1052            LR         QL,A
      84E9 0F           1053            LR         DC,Q
      84EA 46           1054            LR         A,6
      84EB 5B           1055            LR         11,A
      84EC 250F         1056            CI         15
      84EE 821A   8509  1057            BC         OUTS       DO SHORT OUTS
      84F0 2027         1058            LI         H'27'
      84F2 17           1059            ST
      84F3 46           1060            LR         A,6
      84F4 17           1061            ST
      84F5 2044         1062            LI         H'44'
      84F7 17           1063            ST
      84F8 2027         1064            LI         H'27'
      84FA 17           1065            ST
      84FB 46           1066            LR         A,6
      84FC 17           1067   JMP      ST
      84FD 2029         1068            LI         H'29'
      84FF 17           1069            ST
      8500 2082         1070            LI         H'82'
      8502 17           1071            ST
      8503 20F7         1072            LI         H'F7'      TO C55'
      8505 17           1073            ST
      8506 2000         1074            LI         0
      8508 0D           1075            LR         P0,Q       NOW EXECUTE
      8509 24B0         1076   OUTS     AI         H'B0'
      850B 17           1077            ST
      850C 2044         1078            LI         H'44'
      850E 17           1079            ST
      850F 20B0         1080            LI         H'B0'
      8511 C6           1081            AS         6
      8512 90E9   84FC  1082            BR         JMP
      8514 2B           1083            NOP
                        1084   ***GET A BYTE****
                        1085   ***      GETS THE BYTE,CONVERTS,AND ADDS TO
      8515 08           1086   BYTE     LR         K,P        SAVE PC1
      8516 72           1087            LIS        2
      8517 58           1088            LR         HFLG,A     SET THE HALF FL
      8518 4B           1089   AGAN     LR         A,CHR1
      8519 15           1090            SL         4
      851A 5B           1091            LR         CHR1,A
      851B 28853C 853C  1092            PI         CHAR
      851E 213F         1093            NI         H'3F'      MASK TO 6 BITS
      8520 24D0         1094            AI         H'D0'      ASCII CONVERT--
      8522 8203   8526  1095            BC         *+4
      8524 2439         1096            AI         H'39'      NEXT CLEAN UP A
      8526 CB           1097            AS         CHR1
      8527 5B           1098            LR         CHR1,A     TEMP STORE
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                   SOURCE STATEMENT

      8528 46          1099            LR        A,XFLG
      8529 F6          1100            NS        XFLG        TEST FOR NEGATI
      852A 9104   852F 1101            BM        AEND        NEGATIVE=FORMUL
      852C 4B          1102            LR        A,CHR1
      852D C7          1103            AS        CKSM        ADD NEW CHAR TO
      852E 57          1104            LR        CKSM,A
      852F 38          1105 AEND       DS        HFLG        DECREMENT HALF
      8530 94E7   8518 1106            BNZ       AGAN        GET 2ND HALF
      8532 46          1107            LR        A,XFLG
      8533 F6          1108            NS        XFLG
      8534 8104   8539 1109            BP        ADON        IF FORMULATOR
      8536 47          1110            LR        A,CKSM
      8537 CB          1111            AS        CHR1
      8538 57          1112            LR        CKSM,A
      8539 4B          1113 ADON       LR        A,CHR1      ONLY HAVE UPPER
      853A 09          1114            LR        P,K         RESTORE PC1
      853B 1C          1115            POP
                       1116 *********       PARALLEL AND SERIAL INPUT ROUTI
                       1117 **       COMMON CALL IS A PUSH TO CHAR
                       1118 **       CHAR USES Q TO JUMP TO APPROPRIATE R
      853C 0D          1119 CHAR       LR       P0,Q         JUMP TO INPUT R
                       1120 **    PINP:  GET A CHARACTER FROM PORT 4
                       1121 **     TYPICALLY USED WITH TAPE READER, BUT H
                       1122 **     DISCIPLINE THAT IS APPLICABLE TO OTHER
                       1123 **     LOOKS FOR A CHARACTER READY INPUT, AND
                       1124 **     CHARACTER. NEXT CPU GIVES AN ADVANCE P
                       1125 **     AFTER THE DEVICE READY INPUT GOES NOT
      853D A8          1126 PINP       INS       PSTS        GET A CHAR FROM
      853E 13          1127            SL        1
      853F 91FD   853D 1128            BM        PINP        LOOK FOR SPOCKE
      8541 7F          1129            LIS       H'0F'       100 US DELAY AF
      8542 18          1130            COM
      8543 1F          1131 PDLY       INC
      8544 94FE   8543 1132            BNZ       PDLY
      8546 A9          1133            INS       PPRT        AND NOW GET DAT
      8547 18          1134            COM
      8548 51          1135            LR        CHRS,A      TEMP STORE CNEW
      8549 79          1136            LIS       9           ADVANCE READER
      854A B8          1137            OUTS      PSTS
      854B A8          1138 NOSP       INS       PSTS        GET READER STAT
      854C 13          1139            SL        1
      854D 81FD   854B 1140            BP        NOSP               AND LOOK
      854F 71          1141            LIS       1
      8550 B8          1142            OUTS      PSTS        REMOVE DRIVE PU
      8551 41          1143            LR        A,CHRS      PICK BACK UP TH
      8552 1C          1144            POP
                       1145 ***TTYI: SERIAL I/P               CHARACTER RETU
                       1146 **    REG BCNT HOLDS BIT COUNT   REG CHRS HOL
      8553 A8          1147 TTYI       INS       IPOR
      8554 91FE   8553 1148            BM        TTYI        LOOK FOR START
      8556 40          1149            LR        A,BAUD      GET DELAY COUNT
      8557 9001   8559 1150 DLY3       BR        *+2         SILLY BRANCH FO
      8559 2401        1151            AI        H'01'
      855B 94FB   8557 1152            BNZ       DLY3        THIS LOOP IS HA
      855D A8          1153            INS       IPOR        CHECK START BIT
      855E 91F4   8553 1154            BM        TTYI
      8560 79          1155 TTD      LIS       9
      8561 52          1156          LR        BCNT,A        SET BIT COUNT,9
      8562 2180        1157 LOOP       NI        H'80'       MASK TO GET INP
      8564 C1          1158            AS        CHRS        (LD'G START BIT
      8565 32          1159            DS        BCNT        DROP BIT CNT: 0
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                SOURCE STATEMENT

     8566 9110   8577 1160           BM        STOP       NEG IF LOOKING
     8568 8402   856B 1161           BZ        LOP2       IF LAST BIT, DO
     856A 12          1162 LOP3      SR        1          SHIFT ASSEMBLED
     856B 51          1163 LOP2      LR        CHRS,A     STORE ASSEMBLED
     856C 40          1164           LR        A,BAUD     START OF FULL B
     856D BB          1165 DLY4      OUTS      NUSE       NOP FOR DELAY
     856E BB          1166           OUTS      NUSE
     856F BB          1167           OUTS      NUSE
     8570 2401        1168           AI        H'01'      INCR WITH A 5US
     8572 94FA   856D 1169           BNZ       DLY4
     8574 A8          1170           INS       IPOR       GET NEW BIT
     8575 90EC   8562 1171           BR        LOOP
     8577 1C          1172 STOP      POP
                      1173 ******* SERIAL OUTPUT ROUTINE  ***********
                      1174 **  HAS 1 START, 8 DATA, 2 STOP. USES PORT 0
                      1175 **  BAUD RATE IS SET BY A DELAY COUNT IN REG
                      1176 **   CALL BY PUTTING CHAR IN REG CHRS, AND S
                      1177 **   ROUTINE RETURNS WITH ALL 1'S IN REG CHR
     8578 08          1178 TTCR      LR        K,P
     8579 7D          1179           LIS       CR
     857A 51          1180           LR        CHRS,A
     857B 288593 8593 1181           PI        TTYO
     857E 7A          1182           LIS       LF
     857F 51          1183           LR        CHRS,A
     8580 288593 8593 1184           PI        TTYO
     8583 288593 8593 1185           PI        TTYO
     8586 288593 8593 1186           PI        TTYO
     8589 288593 8593 1187           PI        TTYO
     858C 288593 8593 1188           PI        TTYO
     858F 288593 8593 1189           PI        TTYO
     8592 0C          1190           PK                   RETURN
     8593 7B          1191 TTYO      LIS       H'0B'
     8594 52          1192           LR        BCNT,A     SET BIT COUNT F
     8595 70          1193           LIS       0
     8596 B8          1194           OUTS      OPOR       OUTPUT START BI
                      1195 ***   DELAY ROUTINE-- 3.3MS FOR 300 BAUD, 9
     8597 40          1196 DLY1      LR        A,BAUD     GET DELAY COUNT
     8598 BB          1197 DLY2      OUTS      NUSE       NOP FOR DELAY
     8599 BB          1198           OUTS      NUSE
     859A BB          1199           OUTS      NUSE
     859B 2401        1200           AI        H'01'      INCR WITH A 5 U
     859D 94FA   8598 1201           BNZ       DLY2
     859F 32          1202           DS        BCNT
     85A0 9402   85A3 1203           BNZ       DLY5       NOW NEXT BIT
     85A2 1C          1204           POP                  ALL FINI,BACK T
     85A3 71          1205 DLY5      LIS       1          GET CHARACTER
     85A4 F1          1206           NS        CHRS       MASK OFF ALL BU
     85A5 B8          1207           OUTS      OPOR       OUTPUT THE NEW
     85A6 41          1208           LR        A,CHRS     NOW SHIFT THE C
     85A7 12          1209           SR        1
     85A8 2480        1210           AI        H'80'      FILL,WITH 1'S F
     85AA 51          1211           LR        CHRS,A
                      1212 *
     85AB 90EB   8597 1213 EDUX      BR        DLY1       NOW DELAY,TNEN
                      1214 *
                      1215 *
                      1216 * EDUCATOR INITIALIZATION
                      1217 *
     85AD 70          1218 EDUC LIS   0       CLEAR INTERRUPT PORTS
     85AE 52          1219      LR    2,A     CLEAR STORE MODE FLAG
     85AF 207F        1220      LI    H'7F'
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                 SOURCE STATEMENT

      85B1 B0           1221        OUTS 0
      85B2 67           1222        LISU 7
      85B3 68           1223        LISL 0
      85B4 1F           1224        INC
      85B5 5D           1225        LR   I,A
      85B6 02           1226        LR   A,QU   SAVE QU FOR RETURN TO FAIR-
      85B7 5E           1227        LR   D,A
                        1228   * LOAD ZEROES INTO ADDRESS DISPLAY AND DC
      85B8 66           1229   A9   LISU 6
      85B9 6B           1230        LISL 3
      85BA 70           1231        LIS  0
      85BB 5E           1232   A20  LR   D,A
      85BC 8FFE    85BB 1233        BR7  A20
      85BE 74           1234        LIS  4
      85BF 57           1235        LR   7,A
                        1236   * LOAD MEMORY TO DATA DISPLAY
      85C0 2A0000  0000 1237        DCI  0
      85C3 0E           1238        LR   Q,DC
      85C4 288713  8713 1239        PI   MTDD
                        1240   * SCAN KEYBOARD
      85C7 288731  8731 1241   A3   PI   SCAN
      85CA 2012         1242        LI   H'12'
      85CC E0           1243        XS   0
      85CD 8417    85E5 1244        BZ   A38    BRANCH IF IA KEY
      85CF 24FE         1245        AI   H'FE'  WAS IT H'10'?
      85D1 8413    85E5 1246        BZ   A38    BRANCH IF DA KEY
      85D3 76           1247        LIS  6
      85D4 56           1248        LR   6,A    LOAD DELAY TIME FOR REPEAT
      85D5 2020         1249        LI   H'20'
      85D7 55           1250        LR   5,A    LOAD FIRST DELAY TIME FOR R
      85D8 40           1251        LR   A,0
      85D9 1F           1252        INC
      85DA 94EC    85C7 1253        BNZ  A3     BRANCH IF KEY STILL DOWN
      85DC 288731  8731 1254   A5   PI   SCAN
      85DF 40           1255        LR   A,0
      85E0 1F           1256        INC
      85E1 84FA    85DC 1257        BZ   A5     BRANCH IF NO KEYS
      85E3 900E    85F2 1258        BR   A54
      85E5 E5           1259   A38  XS   5
      85E6 8404    85EB 1260        BZ   A53    BRANCH IF FIRST DELAY IS OV
      85E8 35           1261        DS   5
      85E9 90DD    85C7 1262        BR   A3
      85EB 36           1263   A53  DS   6
      85EC 94DA    85C7 1264        BNZ  A3     BRANCH IF DELAY TIME IS OVE
      85EE 76           1265        LIS  6
      85EF 56           1266        LR   6,A    RELOAD DELAY TIME FOR REPEA
      85F0 90EB    85DC 1267        BR   A5
                        1268   * DETERMINE WHICH KEY WAS DETECTED
      85F2 11           1269   A54  LR   H,DC
      85F3 2010         1270        LI   H'10'
      85F5 F0           1271        NS   0
      85F6 9404    85FB 1272        BNZ  A6     BRANCH IF NOT HEX DIGIT KEY
      85F8 298697  8697 1273        JMP  A27
      85FB 2A860B  860B 1274   A6   DCI  FUNC
      85FE 7F           1275        LIS  15
      85FF F0           1276        NS   0
      8600 50           1277        LR   0,A
      8601 8406    8608 1278        BZ   A19
      8603 73           1279   A4   LIS  3
      8604 8E           1280        ADC
      8605 30           1281        DS   0
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS   LOC OBJECT ADDR LINE                 SOURCE STATEMENT

      8606 94FC   8603 1282        BNZ  A4
      8608 0E          1283   A19  LR   Q,DC
      8609 10          1284        LR   DC,H
      860A 0D          1285        LR   P0,Q
      860B 29865C 865C 1286   FUNC JMP  DA
      860E 298623 8623 1287        JMP  ENT
      8611 298650 8650 1288        JMP  IA
      8614 2986D2 86D2 1289        JMP  MOVU
      8617 2986EF 86EF 1290        JMP  MOVD
      861A 298664 8664 1291        JMP  STM
      861D 298640 8640 1292        JMP  GO
      8620 298683 8683 1293        JMP  CLR
                       1294   * 'ENTER' KEY DETECTED
      8623 71          1295   ENT  LIS  1
      8624 F2          1296        NS   2
      8625 8417   863D 1297        BZ   LA      BRANCH IF NOT STORE MODE
      8627 72          1298        LIS  2
      8628 E7          1299        XS   7
      8629 849D   85C7 1300        BZ   A3      BRANCH IF NO DIGITS ENTERED
      862B 20FF        1301   A14  LI   H'FF'   ADDRESS DISPLAY TO DC
      862D 8E          1302        ADC
      862E 6C          1303        LISL 4       DATA DISPLAY TO MEMORY
      862F 4D          1304        LR   A,I
      8630 15          1305        SL   4
      8631 EC          1306        XS   S
      8632 17          1307        ST
      8633 28871C 871C 1308        PI   INAD
      8636 288713 8713 1309   A25  PI   MTDD
      8639 72          1310   A23  LIS  2       SET DIGIT COUNT TO 2
      863A 57          1311   A16  LR   7,A
      863B 908B   85C7 1312        BR   A3
      863D 74          1313   LA   LIS  4       SET DIGIT COUNT TO 4
      863E 90FB   863A 1314        BR   A16
                       1315   * 'GO' KEY DETECTED
      8640 20FF        1316   GO   LI   H'FF'   DISPLAY OFF
      8642 B1          1317        OUTS 1
      8643 20C0        1318        LI   H'C0'
      8645 B0          1319        OUTS 0
      8646 70          1320        LIS  0
      8647 BE          1321        OUTS 14      INTERRUPT OFF
      8648 288702 8702 1322        PI   ATDC
      864B 67          1323        LISU 7
      864C 69          1324        LISL 1
      864D 4C          1325        LR   A,S
      864E 06          1326        LR   QU,A    RESTORE QU FOR FAIR-BUG
      864F 0C          1327        PK
                       1328   * 'INCREMENT ADDRESS' KEY DETECTED
      8650 28871C 871C 1329   IA   PI   INAD    INCREMENT ADDRESS DISPLAY
      8653 288713 8713 1330   A34  PI   MTDD    MEMORY TO DATA DISPLAY
      8656 71          1331        LIS  1
      8657 F2          1332        NS   2
      8658 94E0   8639 1333        BNZ  A23     BRANCH IF STORE MODE FLAG
      865A 90E2   863D 1334        BR   LA
                       1335   * 'DECREMENT ADDRESS' KEY DETECTED
      865C 288727 8727 1336   DA   PI   DEAD
      865F 288702 8702 1337        PI   ATDC
      8662 90F0   8653 1338        BR   A34
                       1339   * 'STORE MODE' KEY DETECTED
      8664 71          1340   STM  LIS  1
      8665 F2          1341        NS   2
      8666 940C   8673 1342        BNZ  A26     BRANCH IF STORE MODE FLAG
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE               SOURCE STATEMENT

      8668 71          1343       LIS  1       SET STORE MODE FLAG
      8669 E2          1344       XS   2
      866A 52          1345       LR   2,A
      866B 67          1346       LISU 7
      866C 68          1347       LISL 0
      866D 4C          1348       LR   A,S
      866E 2240        1349       OI   H'40'
      8670 5C          1350       LR   S,A     SET STORE MODE LIGHT FLAG
      8671 90C7   8639 1351       BR   A23
      8673 E2          1352  A26  XS   2
      8674 52          1353       LR   2,A     RESET STORE MODE FLAG
      8675 67          1354       LISU 7
      8676 68          1355       LISL 0
      8677 2040        1356       LI   H'40'
      8679 EC          1357       XS   S
      867A 5C          1358       LR   S,A     RESET STORE MODE LIGHT FLAG
      867B 288702 8702 1359  A18  PI   ATDC
      867E 288713 8713 1360       PI   MTDD
      8681 90BB   863D 1361       BR   LA
                       1362  ◆ 'CLEAR' KEY DETECTED
      8683 71          1363  CLR  LIS  1
      8684 F2          1364       NS   2
      8685 9408   868E 1365       BNZ  A13     BRANCH IF STORE MODE FLAG
      8687 74          1366       LIS  4
      8688 E7          1367       XS   7
      8689 8431   86BB 1368       BZ   A24     BRANCH IF DIGIT COUNT IS 4
      868B 2985B8 85B8 1369       JMP  A9
      868E 72          1370  A13  LIS  2
      868F F7          1371       NS   7
      8690 942A   86BB 1372       BNZ  A24     BRANCH IF DIGIT COUNT IS 2
      8692 20FF        1373       LI   H'FF'
      8694 8E          1374       ADC          DECREMENT DATA COUNTER
      8695 90A0   8636 1375       BR   A25
                       1376  ◆ CHECK IF FUNCTION FLAG SET BEFORE ACCEPTIN
      8697 40          1377  A27  LR   A,0
      8698 53          1378       LR   3,A
      8699 70          1379       LIS  0
      869A E7          1380       XS   7
      869B 841F   86BB 1381       BZ   A24     BRANCH IF DIGIT COUNT IS 0
      869D 71          1382       LIS  1
      869E F2          1383       NS   2
      869F 941E   86BE 1384       BNZ  A33     BRANCH IF STORE MODE FLAG
      86A1 74          1385       LIS  4
      86A2 E7          1386       XS   7
      86A3 9405   86A9 1387       BNZ  A28     BRANCH IF DIGIT COUNT IS NO
      86A5 6B          1388       LISL 3
      86A6 5E          1389  A30  LR   D,A
      86A7 8FFE   86A6 1390       BR7  A30
                       1391  ◆ LOAD DIGIT TO ADDRESS DISPLAY
      86A9 69          1392  A28  LISL 1
      86AA 4E          1393       LR   A,D
      86AB 5D          1394       LR   I,A
      86AC 5D          1395       LR   I,A
      86AD 4E          1396       LR   A,D
      86AE 5D          1397       LR   I,A
      86AF 5D          1398       LR   I,A
      86B0 4E          1399       LR   A,D
      86B1 5D          1400       LR   I,A
      86B2 43          1401       LR   A,3
      86B3 5C          1402       LR   S,A
      86B4 288702 8702 1403       PI   ATDC    ADDRESS DISPLAY TO DC
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE                SOURCE STATEMENT

     86B7 288713 8713 1404       PI   MTDD    MEMORY TO DATA DISPLAY
     86BA 37          1405  A29  DS   7       DECREMENT DIGIT COUNT
     86BB 2985C7 85C7 1406  A24  JMP  A3
                      1407  * LOAD DIGIT TO DATA DISPLAY
     86BE 72          1408  A33  LIS  2
     86BF F7          1409       NS   7
     86C0 8406   86C7 1410       BZ   A35     BRANCH IF DIGIT COUNT NOT 2
     86C2 66          1411       LISU 6
     86C3 6C          1412       LISL 4       CLEAR DATA DISPLAY
     86C4 70          1413       LIS  0
     86C5 5D          1414       LR   I,A
     86C6 5C          1415       LR   S,A
     86C7 E7          1416  A35  XS   7
     86C8 84F2   86BB 1417       BZ   A24     BRANCH IF DIGIT COUNT IS ZE
     86CA 6D          1418       LISL 5       DIGIT TO DATA DISPLAY
     86CB 4E          1419       LR   A,D
     86CC 5D          1420       LR   I,A
     86CD 43          1421       LR   A,3
     86CE 5C          1422       LR   S,A
     86CF 37          1423       DS   7
     86D0 90EA   86BB 1424       BR   A24
                      1425  * 'MOVE UP' KEY DETECTED
     86D2 71          1426  MOVU LIS  1
     86D3 F2          1427       NS   2
     86D4 84E6   86BB 1428       BZ   A24     BRANCH IF NOT IN STORE MODE
     86D6 20FF        1429       LI   H'FF'
     86D8 53          1430       LR   3,A
     86D9 8E          1431       ADC
     86DA 0E          1432       LR   Q,DC
     86DB 8E          1433       ADC
     86DC 11          1434       LR   H,DC
     86DD 07          1435       LR   QL,A
     86DE 0F          1436       LR   DC,Q
     86DF 16          1437  A12  LM
     86E0 17          1438       ST
     86E1 20FD        1439       LI   H'FD'
     86E3 8E          1440       ADC
     86E4 33          1441       DS   3
     86E5 43          1442       LR   A,3
     86E6 EB          1443       XS   11
     86E7 94F7   86DF 1444       BNZ  A12
     86E9 16          1445       LM
     86EA 202B        1446       LI   H'2B'
     86EC 17          1447       ST
     86ED 908D   867B 1448  A17  BR   A18
                      1449  * 'MOVE DOWN' KEY DETECTED
     86EF 71          1450  MOVD LIS  1
     86F0 F2          1451       NS   2
     86F1 84C9   86BB 1452       BZ   A24     BRANCH IF NOT IN STORE MODE
     86F3 16          1453       LM
     86F4 50          1454       LR   0,A
     86F5 20FE        1455       LI   H'FE'
     86F7 8E          1456       ADC
     86F8 40          1457       LR   A,0
     86F9 17          1458       ST
     86FA 16          1459       LM
     86FB 11          1460       LR   H,DC
     86FC 71          1461       LIS  1
     86FD EB          1462       XS   11
     86FE 94F0   86EF 1463       BNZ  MOVD
     8700 90EC   86ED 1464       BR   A17
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS   LOC OBJECT ADDR LINE                SOURCE STATEMENT

                       1465  ◆
                       1466  ◆ ADDRESS DISPLAY TO DC SUBROUTINE
                       1467  ◆
       8702 66         1468  ATDC LISU 6
       8703 6B         1469       LISL 3
       8704 4E         1470  A15  LR   A,D
       8705 54         1471       LR   4,A
       8706 4E         1472       LR   A,D
       8707 15         1473       SL   4
       8708 E4         1474       XS   4
       8709 8F05  870F 1475       BR7  A36
       870B 06         1476       LR   QU,A
       870C 04         1477       LR   KU,A
       870D 0F         1478       LR   DC,Q
       870E 1C         1479       POP
       870F 07         1480  A36  LR   QL,A
       8710 05         1481       LR   KL,A
       8711 90F2  8704 1482       BR   A15
                       1483  ◆
                       1484  ◆ MEMORY TO DATA DISPLAY SUBROUTINE
                       1485  ◆
       8713 6C         1486  MTDD LISL 4
       8714 16         1487       LM
       8715 54         1488       LR   4,A
       8716 14         1489       SR   4
       8717 5D         1490       LR   I,A
       8718 7F         1491       LIS  15
       8719 F4         1492       NS   4
       871A 5C         1493       LR   S,A
       871B 1C         1494       POP
                       1495  ◆
                       1496  ◆ INCREMENT ADDRESS DISPLAY SUBROUTINE
                       1497  ◆
       871C 6B         1498  INAD LISL 3
       871D 4C         1499  A21  LR   A,S
       871E 1F         1500       INC
       871F 15         1501       SL   4
       8720 14         1502       SR   4
       8721 5E         1503       LR   D,A
       8722 9403  8726 1504       BNZ  A22     BRANCH IF DIGIT WAS NOT 'F'
       8724 8FF8  871D 1505       BR7  A21     BRANCH IF NOT MSD
       8726 1C         1506  A22  POP
                       1507  ◆
                       1508  ◆ DECREMENT ADDRESS DISPLAY SUBROUTINE
                       1509  ◆
       8727 6B         1510  DEAD LISL 3
       8728 3C         1511  A31  DS   S
       8729 4C         1512       LR   A,S
       872A 14         1513       SR   4
       872B 8404  8730 1514       BZ   A32     BRANCH IF DIGIT WAS NOT ZER
       872D 5E         1515       LR   D,A     15 TO DISPLAY REG.
       872E 8FF9  8728 1516       BR7  A31     BRANCH IF NOT MSD
       8730 1C         1517  A32  POP
                       1518  ◆
                       1519  ◆ SCAN KEYBOARD SUBROUTINE
                       1520  ◆
       8731 08         1521  SCAN LR   K,P
       8732 0A         1522       LR   A,IS
       8733 65         1523       LISU 5
       8734 6F         1524       LISL 7
       8735 5E         1525       LR   D,A
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS  LOC OBJECT ADDR LINE               SOURCE STATEMENT

                      1526  * LOAD KEY CODE
      8736 2017       1527  A39  LI    H'17'
      8738 50         1528       LR    0,A
                      1529  * LOAD FIRST STROBE
      8739 20FE       1530       LI    H'FE'
      873B 5C         1531       LR    S,A
                      1532  * OUTPUT STROBE
      873C 7F         1533  A40  LIS   15      BLANK DISPLAY
      873D B1         1534       OUTS  1
      873E 20C0       1535       LI    H'C0'
      8740 B0         1536       OUTS  0
      8741 4E         1537       LR    A,D
      8742 B0         1538       OUTS  0
                      1539  * READ KEYS
      8743 74         1540       LIS   4
      8744 5C         1541       LR    S,A
      8745 A1         1542       INS   1
      8746 14         1543       SR    4
      8747 8417  875F 1544       BZ    A43     BRANCH IF NO KEY IN THIS CO
      8749 15         1545       SL    4
      874A 9108  8753 1546  A41  BM    A42     BRANCH IF KEY IS IN THIS RO
                      1547  * GO TO NEXT ROW IF NOT LAST ROW
      874C 30         1548       DS    0
      874D 3C         1549       DS    S
      874E 842B  877A 1550       BZ    A47     BRANCH IF LAST ROW IN THIS
      8750 13         1551       SL    1
      8751 90F8  874A 1552       BR    A41
                      1553  * IS THIS FIRST SCAN
      8753 6C         1554  A42  LISL  4
      8754 71         1555       LIS   1
      8755 EC         1556       XS    S
      8756 9426  877D 1557       BNZ   A48     BRANCH IF KBD FLAG NOT SET
                      1558  * RESET KBD FLAG
      8758 5E         1559       LR    D,A
                      1560  * COMPARE TWO SCANS
      8759 4C         1561       LR    A,S
      875A E0         1562       XS    0
      875B 8419  8775 1563       BZ    A46     BRANCH IF SCANS ARE THE SAM
      875D 900F  876D 1564       BR    A45
                      1565  * CHANGE KEY CODE FOR NEXT COLUMN
      875F 6E         1566  A43  LISL  6
      8760 20FC       1567       LI    H'FC'
      8762 C0         1568       AS    0
      8763 50         1569       LR    0,A
                      1570  * SHIFT STROBE ONE POSITION
      8764 4C         1571  A44  LR    A,S
      8765 13         1572       SL    1
      8766 2301       1573       XI    1
      8768 5C         1574       LR    S,A
                      1575  * HAS LAST COLUMN BEEN SCANNED ALREADY
      8769 2140       1576       NI    H'40'
      876B 94D0  873C 1577       BNZ   A40     BRANCH IF NOT LAST STROBE
                      1578  * STORE 'FF' IN KEY CODE REGISTER
      876D 20FF       1579  A45  LI    H'FF'
      876F 50         1580       LR    0,A
                      1581  * RESET KBD FLAG
      8770 6C         1582       LISL  4
      8771 70         1583       LIS   0
      8772 5C         1584       LR    S,A
      8773 900C  8780 1585       BR    A51
                      1586  * RESTORE ISAR
 FORMULATOR ASSEMBLER (REV 3.0)

ERRS   LOC OBJECT ADDR LINE                 SOURCE STATEMENT

      8775 6F           1587  A46  LISL 7
      8776 4C           1588       LR   A,S
      8777 0B           1589       LR   IS,A
      8778 09           1590       LR   P,K
      8779 1C           1591       POP
                        1592  ◆ GO TO NEXT ROW
      877A 6E           1593  A47  LISL 6
      877B 90E8    8764 1594       BR   A44
                        1595  ◆ SET KBD FLAG
      877D 5E           1596  A48  LR   D,A
                        1597  ◆ STORE KEY CODE OF FIRST SCAN
      877E 40           1598       LR   A,0
      877F 5E           1599       LR   D,A
                        1600  ◆ LOAD COUNTERS FOR BOUNCE TIME OF 28MS
      8780 288799 8799 1601  A51  PI   DISP    RESTORE DISPLAY
      8783 65           1602       LISU 5
      8784 6A           1603       LISL 2
      8785 7B           1604       LIS  11
      8786 5E           1605       LR   D,A
      8787 20FF         1606  A49  LI   H'FF'
      8789 5C           1607       LR   S,A
      878A 3C           1608  A50  DS   S
      878B 94FE    878A 1609       BNZ  A50     BRANCH IF COUNTER 1 NOT ZER
      878D 6A           1610       LISL 2
      878E 3E           1611       DS   D
      878F 94F7    8787 1612       BNZ  A49     BRANCH IF COUNTER 2 NOT ZER
      8791 6C           1613       LISL 4
      8792 71           1614       LIS  1
      8793 FC           1615       NS   S
      8794 84E0    8775 1616       BZ   A46     BRANCH IF NOT KBD FLAG
      8796 6E           1617       LISL 6
      8797 909E    8736 1618       BR   A39
                        1619  ◆
                        1620  ◆ DISPLAY SUBROUTINE
                        1621  ◆
      8799 66           1622  DISP LISU 6
      879A 6F           1623       LISL 7
      879B 41           1624       LR   A,1
      879C 5E           1625       LR   D,A     SAVE REG. 1
      879D 40           1626       LR   A,0
      879E 5C           1627       LR   S,A     SAVE REG. 0
      879F 67           1628       LISU 7
      87A0 68           1629       LISL 0
      87A1 71           1630       LIS  1
      87A2 50           1631       LR   0,A
      87A3 4C           1632       LR   A,S
      87A4 51           1633       LR   1,A
      87A5 66           1634       LISU 6
      87A6 6D           1635       LISL 5
      87A7 41           1636  A52  LR   A,1
      87A8 23FF         1637       XI   H'FF'
      87AA B0           1638       OUTS 0
      87AB 4E           1639       LR   A,D
      87AC 18           1640       COM
      87AD 15           1641       SL   4
      87AE 14           1642       SR   4
      87AF B1           1643       OUTS 1
      87B0 A0           1644       INS  0
      87B1 E0           1645       XS   0
      87B2 B0           1646       OUTS 0
      87B3 40           1647       LR   A,0
```

```
87B4 13                     SL    1
87B5 50                     LR    0,A
87B6 8FF0      87A7         BR7   A52
87B8 6E                     LISL  6
87B9 4D                     LR    A,I
87BA 50                     LR    0,A       RESTORE REG. 0
87BB 4C                     LR    A,S
87BC 51                     LR    1,A       RESTORE REG. 1
87BD 1C                     POP
                      *
                      * SONG SUBROUTINE
                      *      LOAD STARTING LOCATION OF SONG CONSTAN
                      *      LOAD TEMPO INTO R7.
                      *
87BE 2040             SONG  LI    H'40'
87C0 56                     LR    6,A       SET NOTE FLAG
87C1 47                     LR    A,7
87C2 55                     LR    5,A       STORE TEMPO CONSTANT IN WOR
87C3 16                     LM
87C4 54                     LR    4,A       STORE NOTE DURATION CONSTAN
87C5 51                     LR    1,A       RETAIN NOTE DURATION CONSTA
87C6 F1                     NS    1
87C7 8437      87FF         BZ    FINS      END OF SONG IF NOTE DURATIO
87C9 16                     LM
87CA 53                     LR    3,A       RETAIN NOTE PITCH CONSTANT
87CB 237F                   XI    H'7F'
87CD 9402      87D0         BNZ   A2        BRANCH IF NOT A REST
87CF 56                     LR    6,A       CLEAR NOTE FLAG
87D0 20FF             A2    LI    H'FF'
87D2 59                     LR    9,A       LOAD PITCH COUNTER A
87D3 76                     LIS   6
87D4 58                     LR    8,A       LOAD PITCH COUNTER B
87D5 43               A55   LR    A,3
87D6 52                     LR    2,A       STORE NOTE PITCH CONSTANT I
87D7 A8                     INS   8
87D8 E6                     XS    6
87D9 B8                     OUTS  8         TOGGLE OUTPUT PORT
87DA 39               A11   DS    9
87DB 9404      87E0         BNZ   A8        BRANCH IF PITCH COUNTER A N
87DD 38                     DS    8
87DE 8406      87E5         BZ    A7        BRANCH IF PITCH COUNTER B I
87E0 32               A8    DS    2         DECREMENT NOTE PITCH CONSTA
87E1 81F8      87DA         BP    A11       BRANCH IF NOTE PITCH CONSTA
87E3 90F1      87D5         BR    A55       GO TO TOGGLING OF OUTPUT
87E5 32               A7    DS    2         DECR. NOTE PITCH CONSTANT T
87E6 32                     DS    2
87E7 76                     LIS   6
87E8 58                     LR    8,A       RELOAD PITCH COUNTER B
87E9 34                     DS    4         DECREMENT NOTE DURATION CON
87EA 9408      87F3         BNZ   A10       BRANCH IF NOTE DURATION CON
87EC 41                     LR    A,1
87ED 54                     LR    4,A       RELOAD NOTE DURATION CONSTA
87EE 35                     DS    5         DECREMENT TEMPO CONSTANT
87EF 94EA      87DA         BNZ   A11       BRANCH IF TEMPO CONSTANT NO
87F1 90CC      87BE         BR    SONG      GO TO NEXT NOTE
87F3 71               A10   LIS   1
87F4 E5                     XS    5
87F5 94E4      87DA         BNZ   A11       BRANCH IF TEMPO CONSTANT NO
87F7 32                     DS    2         DECR. NOTE PITCH CONSTANT T
87F8 71                     LIS   1
87F9 E4                     XS    4
87FA 94DF      87DA         BNZ   A11       BRANCH IF NOTE DURATION CON
87FC 56                     LR    6,A       CLEAR NOTE FLAG TO PROVIDE
87FD 90DC      87DA         BR    A11
87FF 1C               FINS  POP
                            END
```

FAIRCHILD

Fairchild reserves the right to make changes in the circuitry or specifications in this book at any time without notice.

Fairchild cannot assume responsibility for use of any circuitry described other than circuitry entirely embodied in a Fairchild product. No other circuit patent licenses are implied.

Printed in U.S.A./331-70-0002-028 15M