PHILIPS

Electronic components and materials

Technical note 132

The 2656 CP1002 system memory interface

The Signetics 2656 System Memory Interface (SMI), Fig. 1, is a mask programmable circuit with on-chip memory, I/O, and timing functions. It is usable either in 2-chip or multi-chip microcomputer systems. Used with the 2650 microprocessor, it provides a 2-chip microcomputer with 2Kx8 bits of ROM, 128x8 bits of RAM, and an 8-bit input-output port.

Used as a system interface in a multi-chip microcomputer, with larger memory and/or additional peripheral requirements, the programmable versatility of the SMI provides decoded chip enable outputs. These outputs connect directly to other memory or I/O functional blocks with few and often no requirement for additional interfacing chips. This reduces both chip count and cost in complex microcomputer systems.

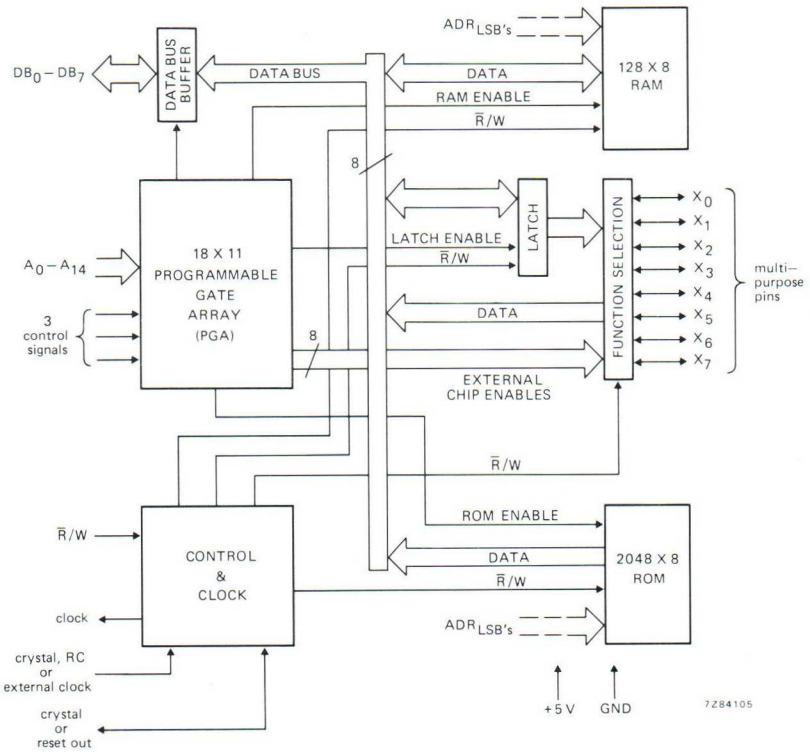

Fig. 1 Schematic diagram of the 2656 SMI.

signetics

This publication describes the 2656 CP1002 pre-programmed version of the SMI. This device contains two firmware packages: PIPBUG 2, an improved version of PIPBUG, the Signetics loader and debugging program, and PIPLA, an assembler that generates object code directly in memory in response to 2650A or 2650B assembly language instructions received from a terminal. The I/O pins of this device are preprogrammed for a variety of functions: direct input, direct output, memory chip select, and I/O chip select.

SMI resource allocation

A programmable gate array (PGA) in the SMI is used to specify the starting addresses of the internal ROM and RAM and the function of the eight multi-function I/O pins. In the CP1002 version, the ROM occupies memory locations H'0000' to H'07FF'. H'0000' to H'03FF' contain the PIPBUG2 program and H'0400' to H'07FF' contain the PIPLA assembler program. The RAM occupies addresses H'0800' to H'087F'. The first 98 bytes (addresses H'0800' – H'0861') are used by PIPBUG2 and PIPLA. The remaining 30 bytes (addresses H'0862' – H'087F') may be used as storage area by user programs.

TABLE 1 2656 CP1002 SMI I/O pin functions.

pin	function	address (HEX)	M/$\overline{IO}$*
X0	input/output	FF	0
X1	chip enable	0C00-0CFF	1
X2	chip enable	0D00-0DFF	1
X3	chip enable	0E00-0EFF	1
X4	chip enable	0F00-0FFF *or* 1F00-1FFF	1
X5	chip enable	00-03	0
X6	chip enable	04-07	0
X7	input/output	FF	0

* M/$\overline{IO}$ = 0 for pins activated using extended I/O instructions.
M/$\overline{IO}$ = 1 for pins activated using memory reference instructions.

Table 1 lists the use of the eight SMI I/O pins X0 to X7. Since only two of these pins (X0 and X7) provide I/O port functions, the remaining six bits of the SMI internal latch (bits 1 – 6) can be used for temporary data storage or software flags.

Pipbug

PIPBUG is a hexadecimal loader, editor, and debug monitor program intended for use in a 2650 based evaluation board or prototype system. PIPBUG provides the user with the necessary tools to develop and run small programs on the Signetics 2650 microprocessor. PIPBUG is described in Application Memo SS50.

PIPBUG 2 is an improved version of the original PIPBUG program. This new version permits the user to load memory from paper tape, dump the contents of memory onto paper tape, examine and alter the contents of memory, set breakpoints, and generate paper tape for programming of PROMs. PIPBUG 2 supports serial asynchronous communications via the 2650 SENSE input and FLAG output pins. It operates at either 110 or 300 baud and automatically synchronizes to the line rate.

PIPBUG 2 is entered by starting program execution at location zero. This is normally done with a RESET signal to the 2650. After some initial housekeeping, the program enters a wait loop looking for receipt of the character U, which is used to determine the terminal's baud rate (110 or 300 baud).

The PIPBUG 2 prompt character '*' is then issued and the monitor awaits an input command. It is not necessary to type in the character U each time the 2650 is reset since the proper baud rate is stored in SMI memory.

IMPORTANT NOTE

PIPBUG 2 uses three levels of the 2650 subroutine stack. Thus, programs being debugged by PIPBUG 2 are limited to five stack levels. Also note that the CPU registers are cleared after a reset.

Command entry

When the prompting character '*' is displayed on the terminal, PIPBUG 2 is ready to accept an input command. All commands have the same general format: a single alphabetic character followed, in some cases, by one or more hexadecimal values. The commands are summarized in Table 2 and described below. In the descriptions, the following conventions apply:

1. 'adr' represents a hexadecimal address value in the range 0 to 7FFF.
2. 'v' represents a hexadecimal data value in the range 0 to FF.
3. 'CR' means carriage return.
4. 'LF' means line feed.
5. '↑' is the character normally obtained by striking the 'N' key with 'SHIFT' depressed.
6. Parameters enclosed in parentheses ( ) are optional.
7. Leading zeros in numeric parameters are optional.
8. Numeric parameters are separated by blanks.

If PIPBUG 2 responds with a '?', the previous command line contained an error. The command line should be examined, corrected, and entered again. Errors can be corrected during editing by pressing the DEL (Delete) key on the terminal. The deleted character is then echoed to the terminal.

Command descriptions

I. Command: Examine and Alter Memory
Format: A adr CR.
Action: Outputs the specified memory address followed by its current content. The user responds with:
(v) CR, which terminates the command,
(v) LF, which displays the next memory location,
(v) ↑ , which displays the previous memory location.

If the optional parameter 'v' is entered, the current contents as shown are replaced by 'v'.

II. Command: Load Hex From Paper Tape
Format: L CR
Action: Starts reading blocks of data from paper tape in the hex object format defined in Application Memo SS51. In the case of illegal characters, a BCC error, or a length error the paper tape will be stopped and the command ended with the standard error message '?'.

At the end of a succesful load, control is passed to the address in the End of File (EOF) block. This would usually be back to the PIPBUG program or to the start of the user program.

III. Command: Dump Hex to Paper Tape
Format: D adr_1 adr_2 CR
Action: Punches a leader of 50 blanks and then outputs the contents of locations 'adr_1' to 'adr_2' inclusive, in hex object format. When done, the EOF block and a trailer of 50 blanks are punched. A starting address of '0000' is inserted in all dump tapes, which automatically transfers program control to PIPBUG after a load from paper tape.

IV. Command: See and Set Registers
Format: Sr CR
Action: The parameter 'r' is in the range 0 to 8 and selects a particular 2650 register:
0 = register 0
1 = register 1 bank 0
2 = register 2 bank 0
3 = register 3 bank 0
4 = register 1 bank 1
3 = register 2 bank 1
6 = register 3 bank 1
7 = PSW upper (PSU)
8 = PSW lower (PSL)

The contents of the specified register will be displayed. The user can respond with:
(v) CR, which ends the command,
(v) LF, which displays the contents of the next register.

If the optional parameter 'v' is entered, the current register contents are replaced by 'v'.

V. Command: Go To
Format: G adr CR
Action: Restores the 2650 register values to the values previously saved upon encountering a breakpoint or to the values specified by the S command and then begins program execution at 'adr'.

VI. Command: Set Breakpoint
Format: Bi adr CR
Action: Sets breakpoint one (i = 1) or breakpoint two (i = 2) at address 'adr'. See Breakpoints section.

VII. Command: Clear Breakpoint
Format: Ci CR
Action: Clears the first (i = 1) or second (i = 2) breakpoint. If the breakpoint is not set, a '?' is displayed. Breakpoints are cleared automatically during program execution.

VIII. Command: Punch PROM Tape
Format: P c adr L CR
Action: Punches a paper tape in the binary format needed to program PROMs on a DATA I/O PROM Programmer. The data to be programmed into the PROM starts at location 'adr' and the length of the PROM (i.e., the number of bytes) is 'L' + 1, where 'L' is in hex. The parameter 'c' determines which bits of each byte are punched:

0 = all 8 bits
1 = least-significant 4 bits
2 = most significant 4 bits

The first character punched is an 'FF' which indicates the start of tape. This is followed by the specified data. If options 1 or 2 are specified, the four bits of data are right justified and the upper four bits are zero.

IX. Command: Hex addition
Format: H a b CR
Action: Adds 'a' to 'b' and outputs the result in hex. Inputs and the result are in the range 0-FFFF.

The user can obtain a hex subtract by entering one of the parameters as the 2's complement of its real value. The 2's complement itself can be calculated by doing a 1's complement and using the H command to add 1. For example, 10A–D3 is the same as 10A+FF2D. FF2D is the 1's complement plus one of the original number D3. So 10A–D3 = 37.

TABLE 2 Pipbug 2 command summary

A	Alter Memory
B	Set Breakpoint
C	Clear Breakpoint
D	Dump Memory onto Paper Tape
G	Goto address
H	Hex Addition
L	Load Memory from Paper Tape
P	Punch PROM Programming Tape
S	See and Alter Registers

Breakpoints

Breakpoints enable the user to examine the program and the status of the microprocessor immediately prior to execution of the instruction at the breakpoint address. PIPBUG 2 allows two breakpoints to be set.

Setting a breakpoint at location 'adr' causes the two bytes of program at locations 'adr' and 'adr + 1' to be stored in a table in PIPBUG's RAM area. The original contents are replaced by a two-byte instruction which causes a jump to the appropriate breakpoint routine in the monitor. When the user program executes the instruction at location 'adr', the program jumps to the breakpoint routine. This routine saves the microprocessor registers, restores the two bytes of user program to locations 'adr' and 'adr + 1' and prints the breakpoint address. The See command can then be used to examine the processor registers.

Since the breakpoints are software implemented and are cleared when reached, there will not be another breakpoint when the user program is re-executed. The breakpoint must be explicity set again with the Breakpoint command. Breakpoints will remain in memory until executed or explicitly cleared with the Clear command. B1 and B2 may be set at the same address.

Important note

1. Care should be taken when inserting a breakpoint at a single-byte instruction, because B1 and B2 replace two bytes of program at locations 'adr' and 'adr + 1' by ZBRR *3 (9B83) and *5 (9B85) respectively. Therefore the opcode of the following instruction is replaced by the address of the breakpoint. Trying to execute this modified instruction (for example, by branching to this location from another point in the program) would result in an ADDZ,R3 (83) for breakpoint 1, or and ADDI,R1 'unknown value' (85??) for breakpoint 2.
2. If a RESET of the 2650 occurs while breakpoints are set, the modified code will remain at the set breakpoint locations. The original code should be restored with the Altere Memory command.

Subroutines available to the user

There are many subroutines within the PIPBUG 2 program that may be used as portions of programs for the 2650 microprocessor. These routines may be used by setting the 2650 registers as required and performing a branch to the desired subroutine. The most useful routines are described below together with their calling instruction in assembly language and hexadecimal formats. A complete listing of PIPBUG 2 is included at the end of this publication. (See appendix I.)

CHIN Inputs a byte to register 0 from the terminal. Call with ZBSR *9 (BB89).

COUT Outputs the byte in register 0 to the terminal. Call with a ZBSR *7 (BB87).

BIN Reads two hex characters from the terminal, and converts them into a binary byte value in R0. Call with ZBSR *D (BB8D).

BOUT Takes the binary value in register 1 and converts it into two hex characters which are printed on the terminal. Register zero information is lost. Call with a ZBSR *B (BB8B).

LKUP Converts the hex character in register 0 into a 4-bit binary value. Call with a ZBSR *26 (BBA6).

CHING Converts the binary value in register 0 into two hex characters which are returned to registers 1 and 2. Call with a BSTA,UN 028D (3F028D).

Pipla assembler

PIPLA is an assembly language program generator which facilitates quick coding of programs when using the PIPBUG 2 monitor. Its features are symbolic opcodes, register fields and conditions, and ten label operators for doing symbolic *forward* references. PIPLA generates and stores the object code in binary form directly in memory as the user enters the source code from his terminal. PIPLA supports both the 2650A and 2650B instruction sets.

PIPLA uses 1K of the SMI ROM and utilizes many of the I/O routines of the PIPBUG 2 monitor. The program includes a symbol table of all 2650 microprocessor opcodes and the set of symbolic register names and branch conditions defined in Table 3. A complete listing of PIPLA is given in Appendix II.

TABLE 3 Symbols for registers and conditions

* Register equates			
R0	EQU	0	register 0
R1	EQU	1	register 1
R2	EQU	2	register 2
R3	EQU	3	register 3
* Condition codes			
P	EQU	1	positive result
Z	EQU	0	zero result
N	EQU	2	negative result
LT	EQU	2	less than
EQ	EQU	0	equal to
GT	EQU	1	greater than
UN	EQU	3	unconditional

Using PIPLA

PIPLA is started by using the GO command in PIPBUG 2 and branching to location H'400'. PIPLA starts with the origin of the user program set to H'C00', the first available RAM location in the memory after the ABC1500 board is modified as described later. The user can then enter lines of 2650 microprocessor source instructions or any of the three available pseudo-op instructions.

When PIPLA is ready to assemble a source line, it prints the current memory address (program counter value) on the terminal. The source line is then entered. If PIPLA cannot assemble the line, it returns to PIPBUG 2 and prints the '?' error message. To continue, the user may continue by returning to PIPLA with a 'G 40E' command, which will save any forward references, or by re-starting with a 'G 400' command. The ORG pseudo-op can be used to reposition the program counter value to its proper location, thus saving any previously assembled code.

Source line format

The format for the source line input is:

LBL OPC R/C SYM OPND

LBL is optional label. If used, it must be one of the labels used to define forward references described later in this section.

OPC is the instruction or pseudo-op mnemonic. For instructions, the standard mnemonics defined in the 2650 manual are used.

R/C is the register or condition code field, if required. Either hex values or the symbols defined in Table 3 may be used.

SYM is a special indirect addressing and/or indexing symbol. See description below.

OPND is the operand for the instruction. This is a hex value or an address. An address may be in hex or it may be one of the labels for forward references. In the case of relative addressing, PIPLA expects an absolute address and will calculate the displacement for the user. For the ZBRR and ZBSR instructions, however, the address field must contain the actual displacement value.

Fields are separated by blanks. If the LBL field is *not* used, a leading blank is not required. However, if the LBL field *is* used *no* leading blank shall be typed, otherwise the label shall be treated as an instruction mnemonic and the standard error message '?' will be printed. The separator between the OPC and R/C fields may be a comma instead of a blank, and the blank between the SYM and OPND fields is optional. The DEL (Delete) key can be used to correct errors. When depressed, the deleted character is echoed to the terminal.

Label operators

The functions of the label operators is to aid the user in coding *forward* references; that is, a label can only be used in the LBL field *after* it has appeared in the OPND field. The label operators are intended for use in branch instructions and there are certain restrictions in their use. For example, they cannot be used with any of the other special operators or with any relative address instructions. Use Table 4 to find the second byte for the relative address.

There are ten label operators, @0 to @9. Each one can be used in the OPND field of instructions any number of times and finally defined in the LBL field once and then reused as a new label. All references must precede the definition, since a linked list of usage is built in RAM as part of the object code and loaded with the value only when the operator is defined.

Special operations

Indirect addressing and indexing are specified by a single character inserted into the SYM field of the source line. The characters and their functions are:

'*' Indirect addressing

'#' Normal indexing. The index register is given in the R/C field since register zero is the implied source/destination for the instruction. This is different from the normal assembler format.

'+' Auto-increment indexing. The index register is given in the R/C field.

'–' Auto-decrement indexing. The index register is given in the R/C field.

The indirect operator may be used with any of the index operators. For example:

LODA,R3 *+8AO

is a load indirect through address 8A0. Registor R3 is incremented and added to the indirect address generating the final effective address.

Comments

If the first character of the source line is an '*', the line is treated as a comment line. Maximum line length is 24 characters.

Assembler directives (pseudo-ops)

PIPLA recognizes three directives: ORG, ASCI, and END. The function of each of these is summarized below.

ORG nnnn

The hex number 'nnnn' becomes the new memory location (i.e., program counter).

ASCI < delimiter > < string > < delimiter >

PIPLA steps over all leading blanks, and takes the next character as the string delimiter. All of the following characters up to the next occurrence of the delimiter are part of the ASCII string that is stored into memory. The maximum length of the string is 34 characters with delimeters. Appendix III shows an ASCII conversion table.

END

PIPLA returns to PIPBUG 2.

Data constants that do not require conversion from ASCII may be stored directly in the memory using the PIPBUG 2 Alter Memory command.

Example

The example program (Figs 2 and 3) shows the features of PIPLA and how it is used with PIPBUG 2. The example shows the label operators @1 through @4 and the special operators for indirect, and normal indexing. The program reads a character from the terminal and if the SPACE bar is depressed, the message 'EXAMPLE OF PIPLA' will be printed. If the CR key is depressed, program control will be returned to PIPBUG 2 and an '*' will be printed. If any of the other keys is depressed, the terminal will output the standard error message '?' and an '*' and the program will wait for a new PIPBUG 2 command.

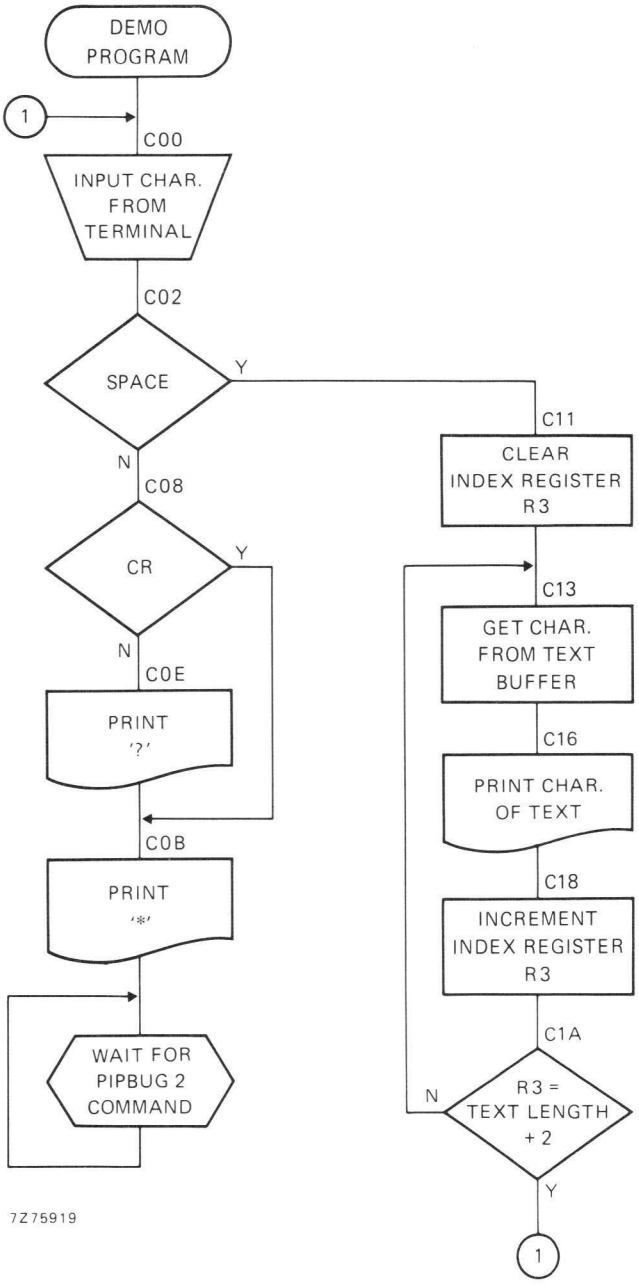

Fig. 2 Demonstration program – flow chart.

TABLE 4 Direct relative address displacement – second byte.

least sign. decimal digit	most significant decimal digit													
	0X		1X		2X		3X		4X		5X		6X	
X	+	–	+	–	+	–	+	–	+	–	+	–	+	–
0	00	–	0A	76	14	6C	1E	62	28	58	32	4E	3C	44
1	01	7F	0B	75	15	6B	1F	61	29	57	33	4D	3D	43
2	02	7E	0C	74	16	6A	20	60	2A	56	34	4C	3E	42
3	03	7D	0D	73	17	69	21	5F	2B	55	35	4B	3F	41
4	04	7C	0E	72	18	68	22	5E	2C	54	36	4A	–	40
5	05	7B	0F	71	19	67	23	5D	2D	53	37	49		
6	06	7A	10	70	1A	66	24	5C	2E	52	38	48		
7	07	79	11	6F	1B	65	25	5B	2F	51	39	47		
8	08	78	12	6E	1C	64	26	5A	30	50	3A	46		
9	09	77	13	6D	1D	63	27	59	31	4F	3B	45		

Indirect relative address: ADD $(80)_{16}$ to displacement.

```
*C400
0C00.*DEMONSTRATION PROGRAM
0C00.ZBSR *9
0C02.COMA,R0 @1
0C05.BCTA,EQ @2
0C08.COMA,R0 @3
0C0B.BCTA,EQ 3C
0C0E.*GO TO PIPBUG 2 MBUG ROUTINE
0C0F.BCTA,UN 38
0C11.*GO TO PIPBUG 2 EBUG ROUTINE
0C11.@2 LODI,R3 00
0C13.LODA,R3 #C50
0C16.ZBSR *7
0C18.ADDI,R3 1
0C1A.COMA,R3 @4
0C1D.BCTR,LT C13
0C1F.BCTR,UN C00
0C21.@1 NOP
0C22.*RESERVED FOR SPACE CODE 20
0C22.@3 NOP
0C23.*RESERVED FOR CR CODE 0D
0C23.@4 NOP
0C24.*RESERVED FOR TEXT LENGTH +2 IN HEX
0C24.ORG C50
0C50.ASCI 'EXAMPLE OF PIPLA'
0C60.END

*AC21
0C21    C0    20
0C22    C0    0D
0C23    C0    12

*AC60
0C60    D2    0L
0C61    89    0A

*GC00
EXAMPLE OF PIPLA
EXAMPLE OF PIPLA
EXAMPLE OF PIPLA
EXAMPLE OF PIPLA
```

Fig. 3.

Modifying the ABC1500

The sections below describe the modifications required to the ABC1500 to incorporate the 2656CP1002 and a 4 MHz crystal for crystal controlled timing. When this is done, the ABC1500 will contain the PIPBUG 2 and PIPLA capabilities described earlier.

Optional modifications to expand on-board RAM capacity to 1K bytes are also described. The ABC1500 is described in detail in Applications Memo SP55.

The SMI should be located in the wire wrap area near the 2608 ROM location to minimize wire lengths. The required wiring is shown in Fig. 4. Before this wiring is incorporated, the following changes must be made to the board:

Cut track between IC23, pin 5 and W11 at IC23 ($\overline{\text{RAMSEL 0}}$)
Cut track between IC23, pin 4 and W14 at IC23 ($\overline{\text{RAMSEL 1}}$)
Remove IC7 (2608) or IC5 and 6 (82S115).
Remove the jumper between W9 and W10 and connect the wire between SMI pin 10 and W10.

The optional changes to add an additional 512 bytes of RAM are also shown in Fig. 4. The 4 MHz crystal frequency is divided by four in the SMI to provide the 1 MHz required by the 2650.

The SMI is connected to the "on-board memory" data bus in order to obtain the correct number of inversions for the ROM data. Because the drivers for this bus are in the receive mode only for "read Page 0 memory" operations, the X0 and X7 pins of the SMI pins of the SMI cannot be used as inputs unless the MDE signal used to control the 8T26 bus drivers IC8 and IC13 is modified as shown in Fig. 5. However, this change should be made only if no other *extended* input ports are connected to the expansion data bus $\overline{\text{DBUS0}}$ to $\overline{\text{DBUS7}}$. Otherwise, bus contention will occur. Without modification of the MDE signal X0 and X7 can only be used as flags by an Extended Write instruction to address FF.

The X5 and X6 chip enable outputs of the SMI are active for extended I/O port addresses 0 – 3 and 4 – 7 respectively. They may be used to control multi-address devices such as the 2651 PCI or the 2655 PPI or for simple latches or input drivers. The data bus connections to these devices should be to $\overline{\text{DBUS0}}$ – $\overline{\text{DBUS7}}$, unless the previously described modification to the MDE signal has been made. In this case, connect the devices to the on-board non inverted data bus.

A memory map of the modified ABC1500 is shown in Fig. 6. Note that one block of RAM is enabled for two sets of addresses, H'0F00' to H'0FFF' or H'1F00 to H'1FFF'. This allows the use of ZBSR and ZBRR instructions with negative offsets in user programs. In addition, interrupts, if used, can provide address vectors in the negative direction from address 0.

Correct modification of the ABC1500 with the SMI and the MDE signal can be checked with the test program according to Figs 7 and 8. This program waits for a character from the terminal, and once received echoes this character in ASCII to the terminal and to the Non-Extended C and D ports. The data flow is through the SMI port located at Extended I/O address H'FF'.

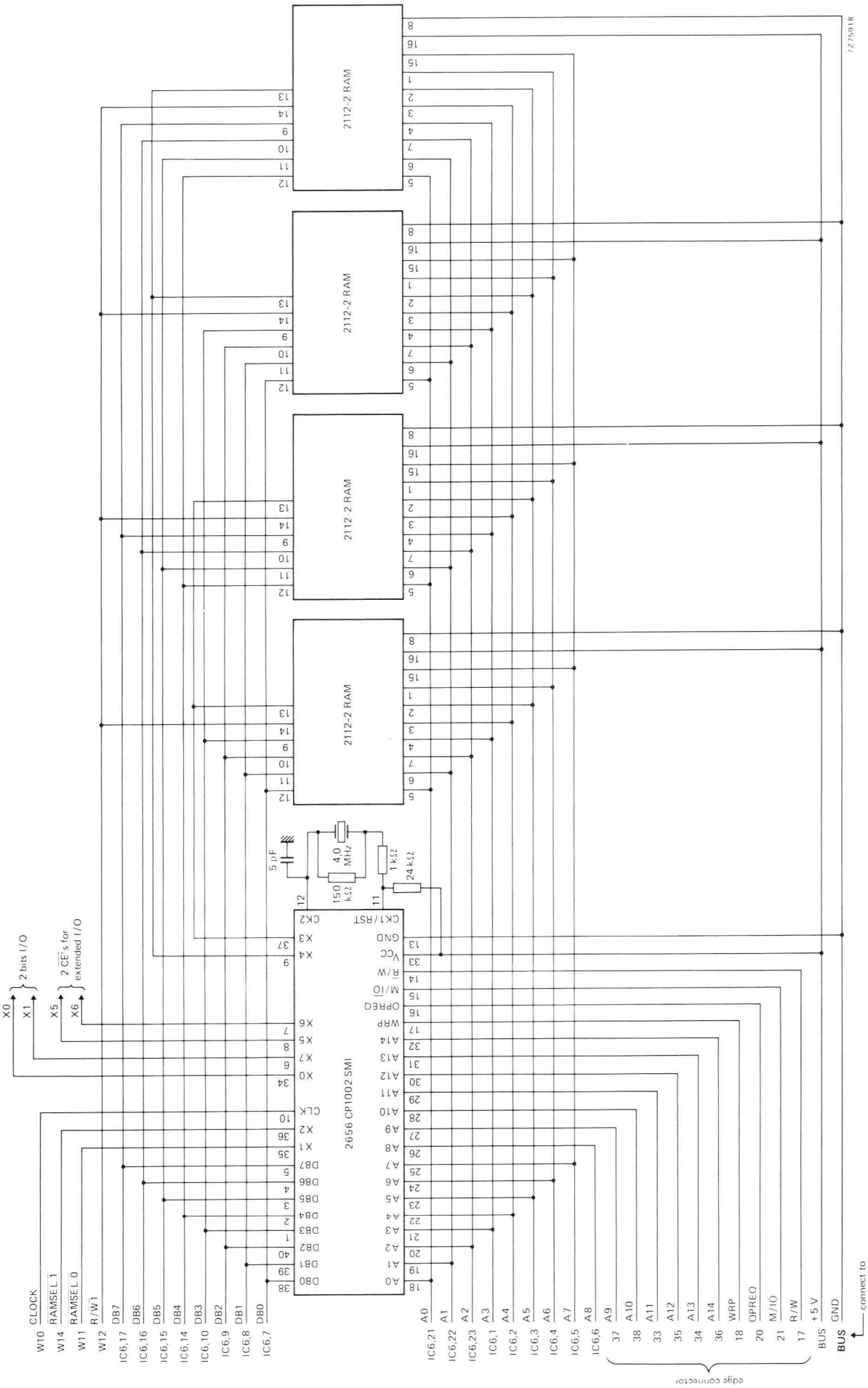

Fig. 4 ABC1500 modifications.

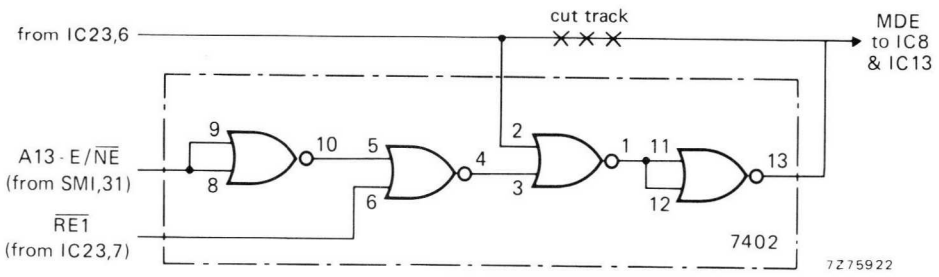

Fig. 5 Changing the MDE signal.

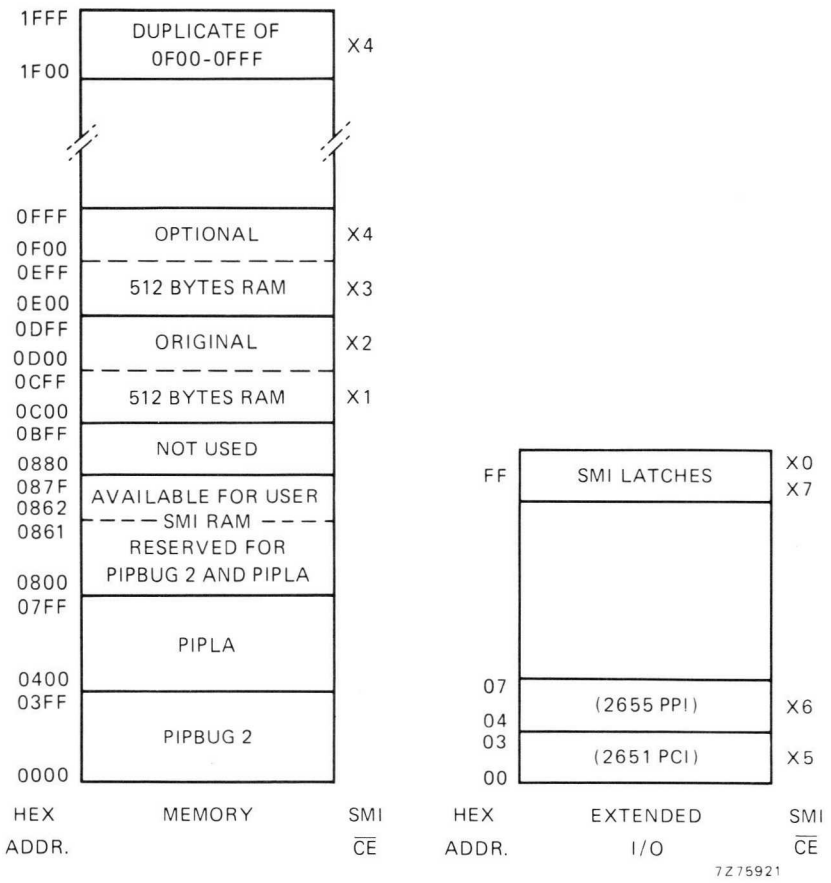

Fig. 6 Modified ABC1500 page 0 memory map.

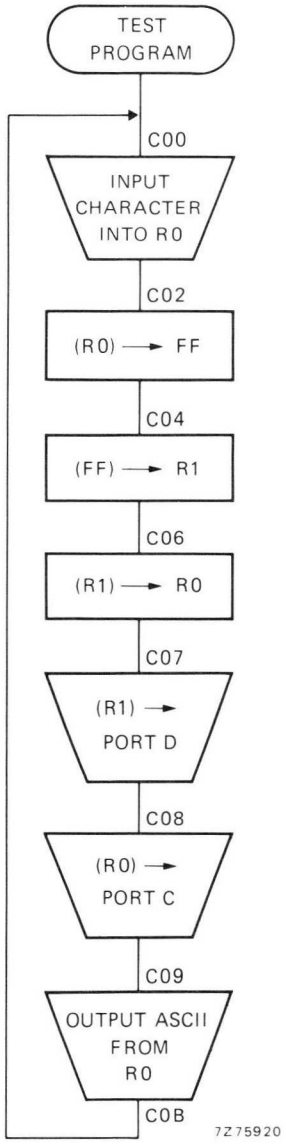

Fig. 7 ABC1500 modifications test program flow chart.

```
*G400
0C00.*TEST PROGRAM
0C00.ZBSR *9
0C02.WRTE,R0 FF
0C04.READE,R1 FF
0C06.LODZ,R1
0C07.WRTD,R1
0C08.WRTC,R0
0C09.ZBSR *B
0C0B.BSTA,UN C00
0C0E.END

*GC00
313233343536372041424344454647
```

Fig. 8.

Appendix 1

PIPBUG 2 Source listing

```
:WIN ASSEMBLER VER 2.X1   PIPBUG VERSION 2.0                    PAGE 0001

LINE ADDR  OBJECT  E SOURCE

0002                ********************
0003                *  DEFINITIONS OF SYMBOLS
0004                *   REGISTER EQUATES
0005 0000           R0      EQU     0       REGISTER 0
0006 0001           R1      EQU     1       REGISTER 1
0007 0002           R2      EQU     2       REGISTER 2
0008 0003           R3      EQU     3       REGISTER 3
0009                *  CONDITION CODES
0010 0001           P       EQU     1       POSITIVE RESULT
0011 0000           Z       EQU     0       ZERO RESULT
0012 0002           N       EQU     2       NEGATIVE RESULT
0013 0002           LT      EQU     2       LESS THAN
0014 0000           EQ      EQU     0       EQUAL TO
0015 0001           GT      EQU     1       GREATER THAN
0016 0003           UN      EQU     3       UNCONDITIONAL
0017                *  PSW LOWER EQUATES
0018 0000           CC      EQU     H'00'   CONDITIONAL CODES
0019 0020           IDC     EQU     H'20'   INTERDIGIT CARRY
0020 0010           RS      EQU     H'10'   REGISTER BANK
0021 0008           WC      EQU     H'08'   1=WITH 0=WITHOUT CARRY
0022 0004           OVF     EQU     H'04'   OVERFLOW
0023 0002           COM     EQU     H'02'   1=LOGIC 0=ARITHMETIC COMPAR
0024 0001           C       EQU     H'01'   CARRY/BORROW
0025                *  PSW UPPER EQUATES
0026 0080           SENS    EQU     H'80'   SENSE BIT
0027 0040           FLAG    EQU     H'40'   FLAG BIT
0028 0020           II      EQU     H'20'   INTERRUPT INHIBIT
0029 0007           SP      EQU     H'07'   STACK POINTER
0030                * END OF EQUATES
```

```
LINE ADDR  OBJECT  E SOURCE

0032 0020          SPAC  EQU   H'20'
0033 0001          BMAX  EQU   1       NO. BKPTS-1
0034 007F          DELE  EQU   H'7F'
0035 000D          CR    EQU   13
0036 000A          LF    EQU   10
0037 0028          BLEN  EQU   40
0038 003A          STAR  EQU   A':'
0039               *
0040 0000                ORG   0
0041 0000 1F03CC         BCTA,UN INIT
0042               *ADDRESS VECTOR FOR THE ZBRR AND ZBSR INSTRUCTIONS
0043 0003 01AB     VEC   ACON    BK01    BREAKPOINT ONE
0044 0005 01B9           ACON    BK02    BREAKPOINT TWO
0045 0007 02CE     COUT  ACON    ICOU    CHAR OUTPUT
0046 0009 02A6     CHIN  ACON    ICHI    CHAR INPUT
0047 000B 0281     BOUT  ACON    IBOU    BYTE OUTPUT
0048 000D 0267     BIN   ACON    IBIN    BYTE INPUT
0049 000F 02EA     GNUM  ACON    IGNU    GET A NUMBER SUBROUTINE
0050 0011 00DA     LINE  ACON    ILIN    GET A CMD LINE
0051 0013 00D0     ADOUT ACON    IADO    ADDRESS OUT
0052 0015 00C9     GPAR  ACON    IGPAR   GET PARAMETERS
0053               *
0054               *THESE SUBROUTINES ARE CALLED BY ZBSR TO SAVE CODE
0055               * SUBR THAT INCREMENTS TEMP  ---     MUST PRECEED
0056 0017 0D080D   INCRT LODA,R1 TEMP
0057 0018          ATEMP SET     $-2
0058 001A 0E080E         LODA,R2 TEMP+1
0059 001B          ATEMP1 SET    $-2
0060 001D DA02           BIRR,R2 STRT
0061 001F D900           BIRR,R1 STRT
0062               * SUBR THAT STORES DOUBLE PRECISION (R1,R2) INTO T
0063 0021 C9F5     STRT  STRR,R1 *ATEMP
0064 0023 CAF6           STRR,R2 *ATEMP1
0065 0025 17             RETC,UN
0066               *SUBR TO VERIFY AND TRANSLATE HEX CHARACTER TO BIN
0067               * INPUT: ASCII HEX CHARACTER IN R0
0068               * OUTPUT
0069               * OUTPUT: 4 BIT BINARY VALUE IN R0, RIGHT JUSTIFIE
0070               * AN ILLEGAL CHAR CAUSES RETURN TO PIPBUG
0071               * USES R0
0072 0026 A430     LKUP  SUBI,R0 H'30'
0073 0028 1A0A           BCTR,N  ABRT
0074 002A E40A           COMI,R0 H'A'
0075 002C 16             RETC,LT
0076 002D A407           SUBI,R0 7
0077 002F 1A03           BCTR,N  ABRT
0078 0031 E410           COMI,R0 H'10'
0079 0033 16             RETC,LT
0080               *THIS ROUTINE WILL ABORT THE CURRENT CMD
0081               * THE STACK IS RESET TO THE USER PROG LEVEL
0082 0034 0C0807   ABRT  LODA,R0 COMM+7   RESET STACK
0083 0037 92             LPSU
0084               *
0085               * COMMAND HANDLER
0086 0038 043F     EBUG  LODI,R0 A'?'    ERROR RETURN FOR ALL ROUTIN
```

```
LINE ADDR  OBJECT  E SOURCE

0087 003A BB87            ZBSR    *COUT
0088 003C 75FF      MBUG  CPSL    H'FF'   START OF CMD LOOP,RESET PSW
0089 003E 3F0109          BSTA,UN CRLF
0090 0041 042A            LODI,R0 A'*'
0091 0043 BB87            ZBSR    *COUT
0092 0045 BB91            ZBSR    *LINE   GET COMMAND LINE
0093 0047 0C0814          LODA,R0 CNT
0094 004A 1870            BCTR,Z  MBUG
0095 004C 8701            ADDI,R3 1       INCREMENT POINTER PAST COMM
0096 004E 0C0822          LODA,R0 BUFF    DECODE CMD
0097 0051 E441            COMI,R0 A'A'
0098 0053 1C0115          BCTA,EQ ALTE
0099 0056 E442            COMI,R0 A'B'
0100 0058 1C022C          BCTA,EQ BKPT
0101 005B E443            COMI,R0 A'C'
0102 005D 1C0214          BCTA,EQ CLR
0103 0060 E444            COMI,R0 A'D'
0104 0062 1C031F          BCTA,EQ DUMP
0105 0065 E447            COMI,R0 A'G'
0106 0067 1C0187          BCTA,EQ GOTO
0107 006A E448            COMI,R0 A'H'
0108 006C 1810            BCTR,EQ HADD
0109 006E E44C            COMI,R0 A'L'
0110 0070 1C0399          BCTA,EQ LOAD
0111 0073 E453            COMI,R0 A'S'
0112 0075 1C014E          BCTA,EQ SREG
0113 0078 E450            COMI,R0 A'P'
0114 007A 1814            BCTR,EQ PROM
0115 007C 9B38            ZBRR    EBUG    FAILED TO FIND CMD
0116                *
0117                * PIPBUG COMMANDS
0118                *
0119                * HEX 16-BIT ADD
0120                * INPUT IS TWO PARAMETERS, GIVES RESULT IN TEMP
0121                * USES R1,R2 AND TEMP
0122 007E BB95      HADD  ZBSR    *GPAR
0123 0080 7708            PPSL    WC
0124 0082 7501            CPSL    C
0125 0084 8E080E          ADDA,R2 TEMP+1
0126 0087 8D080D          ADDA,R1 TEMP
0127 008A 7509            CPSL    WC+C
0128 008C BB93            ZBSR    *ADOUT
0129 008E 9B3C            ZBRR    MBUG
0130                * PROM BINARY TAPE: 0=8 BIT,1=LSN,2=MSN
0131 0090 BB8F      PROM  ZBSR    *GNUM
0132 0092 CE0811          STRA,R2 TEMR
0133 0093           ATEMR SET     $-2
0134 0095 BB95            ZBSR    *GPAR   LENGTH IN R1,R2
0135 0097 3F0364          BSTA,UN GAP
0136 0098           AGAP  SET     $-2
0137 009A 04FF            LODI,R0 H'FF'   MARKER CHAR
0138 009C BB87             ZBSR    *COUT
0139 009E 01        DPROM  LODZ    R1
0140 009F 62               IORZ    R2
0141 00A0 5804             BRNR,R0 APROM
0142 00A2 3BF4             BSTR,UN *AGAP   DONE
```

```
LINE ADDR  OBJECT  E SOURCE

0143 00A4 9B3C            ZBRR    MBUG
0144 00A6 A601      APROM SUBI,R2 1
0145 00A8 7708            PPSL    WC
0146 00AA A500            SUBI,R1 0
0147 00AC 7508            CPSL    WC
0148 00AE 0C880D          LODA,R0 *TEMP
0149 00B1 0BE0            LODR,R3 *ATEMR
0150 00B3 E701            COMI,R3 1
0151 00B5 1806            BCTR,EQ BPROM
0152 00B7 1A06            BCTR,LT CPROM
0153 00B9 50              RRR,R0  2=MOST SIGNIF NIBBLE
0154 00BA 50              RRR,R0
0155 00BB 50              RRR,R0
0156 00BC 50              RRR,R0
0157 00BD 440F      BPROM ANDI,R0 H'0F'   1=LEAST SIGNIF NIBBLE
0158 00BF BB87      CPROM ZBSR    *COUT
0159 00C1 7710            PPSL    RS
0160 00C3 BB17            ZBSR    INCRT
0161 00C5 7510            CPSL    RS
0162 00C7 1B55            BCTR,UN DPROM
0163                * GET PARAMETERS
0164 00C9 BB8F      IGPAR ZBSR    *GNUM
0165 00CB BB21            ZBSR    STRT
0166 00CD BB8F            ZBSR    *GNUM
0167 00CF 17              RETC,UN
0168                * ADDRESS OUT - TYPES OUT ADDRESS IN R1 AND R2
0169 00D0 BB21      IADO  ZBSR    STRT
0170 00D2 BB8B            ZBSR    *BOUT
0171 00D4 0D080E          LODA,R1 TEMP+1
0172 00D7 BB8B            ZBSR    *BOUT
0173 00D9 17              RETC,UN
0174                * INPUT A CMD LINE INTO BUFFER
0175                *CODE IS THE END OF LINE CHARACTER LF,CR,OR UP-ARR
0176 00DA 0700      ILIN  LODI,R3 0
0177 00DC E728      LLIN  COMI,R3 BLEN
0178 00DE 1C0038          BCTA,EQ EBUG    ON BUFFER OVERFLOW HAVE ERR
0179 00E1 BB89            ZBSR    *CHIN
0180 00E3 E47F            COMI,R0 DELE
0181 00E5 980A            BCFR,EQ ALIN
0182 00E7 03              LODZ    R3      ECHO AND BACK PTR
0183 00E8 1872            BCTR,Z  LLIN
0184 00EA 0F4822          LODA,R0 BUFF,R3,-
0185 00ED BB87            ZBSR    *COUT
0186 00EF 1B6B            BCTR,UN LLIN
0187 00F1 0503      ALIN  LODI,R1 3       NUMBER OF EOL'S
0188 00F3 ED6111    BLIN  COMA,R0 EOLTAB-1,R1
0189 00F6 1809            BCTR,EQ CLIN    HAVE END OF LINE
0190 00F8 F979            BDRR,R1 BLIN
0191 00FA CF6822          STRA,R0 BUFF,R3 STORE DATA CHAR INTO BUFFER
0192 00FD BB87            ZBSR    *COUT   ECHO IT
0193 00FF DB5B            BIRR,R3 LLIN
0194 0101 CF0814    CLIN  STRA,R3 CNT     SAVE LAST INPUT INDEX
0195 0104 CD0815          STRA,R1 CODE
0196 0107 0700            LODI,R3 0
0197 0109 040D      CRLF  LODI,R0 CR
0198 010B BB87            ZBSR    *COUT
```

```
LINE ADDR  OBJECT  E SOURCE

0199 010D 040A          LODI,R0 LF
0200 010F BB87          ZBSR    *COUT
0201 0111 17            RETC,UN
0202 0112 0A0D5E    EOLTAB DATA    H'0A,0D,5E'     END OF LINE CHARS
0203                    *
0204                    * DISPLAY AND ALTER MEMORY
0205 0115 BB8F      ALTE   ZBSR   *GNUM
0206 0117 BB93      LALT   ZBSR    *ADOUT
0207 0119 3F0360           BSTA,UN FORM
0208 011C 0D880D           LODA,R1 *TEMP   DISPLAY CONTENTS
0209 011F BB8B             ZBSR   *BOUT
0210 0121 3F0360           BSTA,UN FORM
0211 0124 BB91             ZBSR    *LINE
0212 0126 0C0814           LODA,R0 CNT
0213 0129 1805             BCTR,Z  DALT    NO NEW CONTENTS
0214 012B BB8F             ZBSR   *GNUM
0215 012D CE880D           STRA,R2 *TEMP   UPDATE CONTENTS,CHK FOR CR
0216 0130 0C0815    DALT   LODA,R0 CODE    CHECK FOR DONE,UP,OR DOWN
0217 0133 E402             COMI,R0 2       CHECK FOR CR
0218 0135 1C003C           BCTA,EQ MBUG    CR
0219 0138 1904             BCTR,GT BALT
0220 013A BB17             ZBSR    INCRT   LINE FEED
0221 013C 1B59             BCTR,UN LALT
0222 013E 0D080D    BALT   LODA,R1 TEMP
0223 0141 0E080E           LODA,R2 TEMP+1
0224 0144 5A02             BRNR,R2 CALT
0225 0146 A501             SUBI,R1 1
0226 0148 A601      CALT   SUBI,R2 1
0227 014A BB21             ZBSR    STRT
0228 014C 1B49             BCTR,UN LALT
0229                    * SELECTIVELY DISPLAY AND ALTER REGISTERS
0230 014E BB8F      SREG   ZBSR   *GNUM    GET INDEX OF REG
0231 0150 E608      LSRE   COMI,R2 8       CHECK RANGE
0232 0152 1D0038           BCTA,GT EBUG
0233 0155 CE0811           STRA,R2 TEMR
0234 0158 0E6800           LODA,R0 COMM,R2  DISPLAY CONTENTS
0235 015B C1               STRZ    R1
0236 015C BB8B             ZBSR   *BOUT
0237 015E 3F0360           BSTA,UN FORM
0238 0161 BB91             ZBSR   *LINE
0239 0163 0C0814           LODA,R0 CNT
0240 0166 1810             BCTR,Z  BSRE
0241 0168 BB8F      ASRE   ZBSR   *GNUM
0242 016A 02               LODZ    R2
0243 016B 0E0811           LODA,R2 TEMR
0244 016E CE6800           STRA,R0 COMM,R2
0245 0171 E608             COMI,R2 8       MUST UPDATE PSW LOWER
0246 0173 9803             BCFR,EQ BSRE
0247 0175 CC080A           STRA,R0 XGOT+1
0248 0178 0C0815    BSRE   LODA,R0 CODE
0249 017B E401             COMI,R0 1       CHECK FOR LF
0250 017D 9C003C           BCFA,EQ MBUG
0251 0180 0E0811    CSRE   LODA,R2 TEMR
0252 0183 8601             ADDI,R2 1
0253 0185 1B49             BCTR,UN LSRE
0254                    * GOTO ADDRESS
```

```
LINE ADDR  OBJECT  E SOURCE

0255 0187 BB8F       GOTO   ZBSR   *GNUM
0256 0189 BB21              ZBSR    STRT    PUT ADDR IN RAM
0257 018B 0C0807            LODA,R0 COMM+7
0258 018E 92                LPSU
0259 018F 0D0801            LODA,R1 COMM+1   BANK ZERO
0260 0192 0E0802            LODA,R2 COMM+2
0261 0195 0F0803            LODA,R3 COMM+3
0262 0198 7710              PPSL    RS      BANK ONE
0263 019A 0D0804            LODA,R1 COMM+4
0264 019D 0E0805            LODA,R2 COMM+5
0265 01A0 0F0806            LODA,R3 COMM+6
0266 01A3 0C0800            LODA,R0 COMM
0267 01A6 75FF              CPSL    H'FF'
0268 01A8 1F0809            BCTA,UN XGOT    AND BCTA,UN $TEMP
0269                 *
0270                 * BREAKPOINT RUNTIME CODE
0271 01AB CC0800     BK01   STRA,R0 COMM     ENTRY FOR BKPT-1 VIA VECTO
0272 01AE 13                SPSL
0273 01AF CC0808            STRA,R0 COMM+8
0274 01B2 CC080A            STRA,R0 XGOT+1  IN RAM FOR REG RESTORE
0275 01B5 0400              LODI,R0 0       BKPT INDEX
0276 01B7 1B0C              BCTR,UN BKEN
0277 01B9 CC0800     BK02   STRA,R0 COMM     ENTRY FOR BKPT-2
0278 01BC 13                SPSL
0279 01BD CC0808            STRA,R0 COMM+8
0280 01C0 CC080A            STRA,R0 XGOT+1  IN RAM FOR REG RESTORE
0281 01C3 0401              LODI,R0 1
0282 01C5 CC0811     BKEN   STRA,R0 TEMR
0283 01C8 12                SPSU
0284 01C9 CC0807            STRA,R0 COMM+7
0285 01CC 7710              PPSL    RS
0286 01CE CD0804            STRA,R1 COMM+4
0287 01D1 CE0805            STRA,R2 COMM+5
0288 01D4 CF0806            STRA,R3 COMM+6
0289 01D7 7518              CPSL    RS+WC   FORCE TO B0,CLEAR WC
0290 01D9 CD0801            STRA,R1 COMM+1
0291 01DC CE0802            STRA,R2 COMM+2
0292 01DF CF0803            STRA,R3 COMM+3
0293 01E2 0E0811            LODA,R2 TEMR
0294 01E5 3B0E              BSTR,UN CLBK
0295 01E7 7640              PPSU    FLAG    OUTPUT STOP BITS TO TTY
0296 01E9 0D080D            LODA,R1 TEMP    PRINT BKPT ADDRESS
0297 01EC BB8B              ZBSR   *BOUT
0298 01EE 0D080E            LODA,R1 TEMP+1
0299 01F1 BB8B              ZBSR   *BOUT
0300 01F3 9B3C              ZBRR    MBUG
0301                 * SUBR TO CLEAR A BKPT     LIKE MANY SUBR HAS REL
0302 01F5 20         CLBK   EORZ    R0
0303 01F6 CE6818            STRA,R0 MARK,R2
0304 01F9 0E681E            LODA,R0 HADR,R2
0305 01FC CC080D            STRA,R0 TEMP
0306 01FF 0E6820            LODA,R0 LADR,R2
0307 0202 CC080E            STRA,R0 TEMP+1
0308 0205 0E681A            LODA,R0 HDAT,R2
0309 0208 CC880D            STRA,R0 *TEMP
0310 020B 0E681C            LODA,R0 LDAT,R2
```

```
LINE ADDR  OBJECT  E SOURCE

0311 020E 0501            LODI,R1 1
0312 0210 CDE80D          STRA,R0 *TEMP,R1
0313 0213 17              RETC,UN
0314                 * BREAK POINT  MARK INDICATES IF SET
0315                 * HADR +LADR IS BKPT ADDR,     HDAT+LDAT IS TWO BY
0316 0214 3B0A       CLR  BSTR,UN NOK
0317 0216 0E6818          LODA,R0 MARK,R2 CLEAR IT IF SET
0318 0219 1C0038          BCTA,Z  EBUG
0319 021C 3B57            BSTR,UN CLBK
0320 021E 9B3C            ZBRR    MBUG
0321 0220 BB8F       NOK  ZBSR   *GNUM    CHECK RANGE ON BKPT NUMBER
0322 0222 A601            SUBI,R2 1
0323 0224 1E0034          BCTA,N  ABRT
0324 0225            AABRT SET     $-2
0325 0227 E601            COMI,R2 BMAX
0326 0229 19FA            BCTR,GT *AABRT
0327 022B 17              RETC,UN
0328 022C 3B72       BKPT BSTR,UN NOK     SET BKPT AND CLR NY EXISTIN
0329 022E 0E6818          LODA,R0 MARK,R2
0330 0231 BC01F5          BSFA,Z  CLBK    CLEAR EXISTING ONE
0331 0234 CE0811          STRA,R2 TEMR
0332 0235            ATEMR SET     $-2
0333 0237 BB8F            ZBSR   *GNUM    GET BKPT ADDR
0334 0239 BB21            ZBSR    STRT    SUBR TO STORE R1-R2 IN TEMP
0335 023B 0BF8            LODR,R3 *ATEMR
0336 023D 02              LODZ    R2
0337 023E CF6820          STRA,R0 LADR,R3
0338 0241 01              LODZ    R1
0339 0242 CF681E          STRA,R0 HADR,R3
0340 0245 0C880D          LODA,R0 *TEMP   SAVE CONTENTS
0341 0248 CF681A          STRA,R0 HDAT,R3
0342 024B 059B            LODI,R1 H'9B'   -ZBRR
0343 024D CD880D          STRA,R1 *TEMP
0344 0250 0601            LODI,R2 1
0345 0252 0EE80D          LODA,R0 *TEMP,R2
0346 0255 CF681C          STRA,R0 LDAT,R3
0347 0258 0F6265          LODA,R0 DISP,R3
0348 025B CEE80D          STRA,R0 *TEMP,R2
0349 025E 04FF            LODI,R0 -1
0350 0260 CF6818          STRA,R0 MARK,R3
0351 0263 9B3C            ZBRR    MBUG
0352 0265 83         DISP DATA    VEC+H'80'
0353 0266 85              DATA    VEC+H'80'+2
0354                 *
0355                 * INPUT TWO HEX CHARS AND FORM AS BYTE IN R1
0356 0267 BB89       IBIN ZBSR    *CHIN
0357 0269 BB26            ZBSR    LKUP
0358 026B D0              RRL,R0
0359 026C D0              RRL,R0
0360 026D D0              RRL,R0
0361 026E D0              RRL,R0
0362 026F C1              STRZ    R1
0363 0270 BB89            ZBSR    *CHIN
0364 0272 BB26            ZBSR    LKUP
0365 0274 61              IORZ    R1
0366 0275 C1              STRZ    R1
```

```
LINE ADDR  OBJECT  E SOURCE

0367 0276 3B01            BSTR,UN CBCC
0368 0278 17              RETC,UN
0369                 * CALCULATE THE BCC CHAR, EOR AND THEN ROTATE
0370 0279 01         CBCC LODZ    R1
0371 027A 2C0817          EORA,R0 BCC
0372 027B            ABCC SET     $-2
0373 027D D0              RRL,R0
0374 027E C8FB            STRR,R0 *ABCC
0375 0280 17              RETC,UN
0376                 * BYTE IN R1 OUTPUT IN HEX
0377 0281 3B76       IBOU BSTR,UN CBCC
0378 0283 01              LODZ    R1
0379 0284 3B07            BSTR,UN CHNG
0380 0286 01              LODZ    R1
0381 0287 BB87            ZBSR    *COUT
0382 0289 02              LODZ    R2
0383 028A BB87            ZBSR    *COUT
0384 028C 17              RETC,UN
0385                 *SUBR TO CONVERT BINARY TO HEX
0386                 *      BINARY IN R0, RETURN CHARS IN R1,R2
0387 028D C1         CHNG STRZ    R1
0388 028E 440F            ANDI,R0 H'0F'
0389 0290 3B0C            BSTR,UN ACHN
0390 0292 C2              STRZ    R2
0391 0293 01              LODZ    R1
0392 0294 44F0            ANDI,R0 H'F0'
0393 0296 50              RRR,R0
0394 0297 50              RRR,R0
0395 0298 50              RRR,R0
0396 0299 50              RRR,R0
0397 029A 3B02            BSTR,UN ACHN
0398 029C C1              STRZ    R1
0399 029D 17              RETC,UN
0400 029E 8430       ACHN ADDI,R0 H'30'
0401 02A0 E43A            COMI,R0 H'3A'
0402 02A2 16              RETC,LT
0403 02A3 8407            ADDI,R0 7
0404 02A5 17              RETC,UN
0405                 *
0406                 * SUBR TO INPUT A CHARACTER INTO R0, USING R1 AND
0407 02A6 7718       ICHI PPSL    RS+WC
0408 02A8 12         ACHI SPSU    LOOK FOR START BIT
0409 02A9 1A7D            BCTR,LT ACHI
0410 02AB 3B1B            BSTR,UN HDLY
0411 02AD 7718            PPSL    RS+WC
0412 02AF 0500            LODI,R1 0
0413 02B1 0608            LODI,R2 8
0414 02B3 3B0D       BCHI BSTR,UN DELAY   WAIT TO MIDDLE OF DATA
0415 02B5 12              SPSU
0416 02B6 D0              RRL,R0          MOVE BIT 7 OF R0 INTO R1, B
0417 02B7 51              RRR,R1
0418 02B8 FA79            BDRR,R2 BCHI
0419 02BA 3B06            BSTR,UN DELAY
0420 02BC 457F            ANDI,R1 H'7F'   DELETE PARITY BIT
0421 02BE 01              LODZ    R1
0422 02BF 7518            CPSL    RS+WC
```

```
LINE ADDR  OBJECT  E SOURCE

0423 02C1 17              RETC,UN
0424                 * DELAY LOOP FOR ONE BIT TIME, USING R3
0425                 *      BASED ON A 1 MHZ CLOCK, USING THE COUNT IN
0426 02C2 0F0816     DELAY  LODA,R3 BAUD    AVOID SUBR SO NOT OVERFLOW
0427 02C3            ABAUD  SET     $-2
0428 02C5 14         H1     RETC,Z  USED AS 3 CYCLE NOP
0429 02C6 FB7D              BDRR,R3 H1
0430 02C8 0BF9       HDLY   LODR,R3 *ABAUD
0431 02CA 14         H2     RETC,Z
0432 02CB FB7D              BDRR,R3 H2
0433 02CD 17                RETC,UN
0434                 *
0435                 *OUTPUT A CHAR FROM R0, USING R1
0436                 *ASSUME CARRY CAN BE DESTROYED
0437 02CE 7718       ICOU   PPSL    RS+WC
0438 02D0 7501              CPSL    C       USE CARRY AS THE START BIT
0439 02D2 0509              LODI,R1 9
0440 02D4 50         ACOU   RRR,R0  FIRST ROTATE GETS C AS THE START BI
0441 02D5 1A04              BCTR,LT ONE
0442 02D7 7440       ZERO   CPSU    FLAG
0443 02D9 1B02              BCTR,UN BCOU
0444 02DB 7640       ONE    PPSU    FLAG
0445 02DD 3B63       BCOU   BSTR,UN DELAY
0446 02DF F973              BDRR,R1 ACOU
0447 02E1 7640              PPSU    FLAG
0448 02E3 3B5D              BSTR,UN DELAY
0449 02E5 3B5B              BSTR,UN DELAY
0450 02E7 7518              CPSL    RS+WC
0451 02E9 17                RETC,UN
0452                 *
0453                 * GET A NUMBER FROM THE BUFFER INTO R1-R2
0454                 * STAR WITH THE BINARY NIBBLES R1=AB,R2=CD
0455                 * READ NEXT HEX CHAR AND CONVERT TO NIBBLE E
0456                 * SHIFT INTO ADDR BEING BUILD R1=BC,R2=DE
0457 02EA 20         IGNU   EORZ    R0
0458 02EB C1                STRZ    R1
0459 02EC C2                STRZ    R2
0460 02ED CC0812            STRA,R0 TEMS
0461 02EE            ATEMS  SET     $-2
0462 02F0 08FC       DNUM   LODR,R0 *ATEMS  SKIP SPACES UNTIL NUMERIC C
0463 02F2 15                RETC,P
0464 02F3 EF0814     LNUM   COMA,R3 CNT
0465 02F6 14                RETC,EQ
0466 02F7 0F6822            LODA,R0 BUFF,R3         GET CHAR
0467 02FA E420              COMI,R0 SPAC
0468 02FC 9802              BCFR,EQ BNUM
0469 02FE DB70              BIRR,R3 DNUM
0470 0300 BB26       BNUM   ZBSR    LKUP
0471 0302 D2                RRL,R2  R0=0E,R1=AB,R2=CD
0472 0303 D2                RRL,R2
0473 0304 D2                RRL,R2
0474 0305 D2                RRL,R2
0475 0306 CE0813            STRA,R2 MCNT    R2=DC
0476 0307            AMCNT  SET     $-2
0477 0309 46F0              ANDI,R2 H'F0'
0478 030B 62                IORZ    R2      R0=DE
```

```
LINE ADDR  OBJECT  E SOURCE

0479 030C C2              STRZ    R2
0480 030D D1              RRL,R1
0481 030E D1              RRL,R1
0482 030F D1              RRL,R1
0483 0310 D1              RRL,R1
0484 0311 45F0            ANDI,R1 H'F0'   R1=B0
0485 0313 08F2            LODR,R0 *AMCNT
0486 0315 440F            ANDI,R0 H'0F'
0487 0317 61              IORZ    R1      R0=BC
0488 0318 C1              STRZ    R1      R1=BC,R2=DE
0489 0319 0401            LODI,R0 1
0490 031B C8D1            STRR,R0 *ATEMS
0491 031D DB54            BIRR,R3 LNUM
0492                * DUMP TO PAPER TAPE IN OBJECT FORMAT
0493 031F BB95      DUMP  ZBSR    *GPAR   GET START ADDRESS
0494 0321 3F0364          BSTA,UN GAP     PUNCH LEADER TAPE
0495 0324 DA02            BIRR,R2 KDUM
0496 0326 8501            ADDI,R1 1
0497 0328 CD080F    KDUM  STRA,R1 TEMQ
0498 0329           ATEMQ SET     $-2
0499 032B CE0810          STRA,R2 TEMQ+1
0500 032C           ATEMQ1 SET    $-2
0501 032E 04FF      FDUM  LODI,R0 -1      OUTPUT A RECORD
0502 0330 CC0814          STRA,R0 CNT
0503 0333 3F0109          BSTA,UN CRLF    PUNCH FOR CR/LF AND START C
0504 0336 043A            LODI,R0 STAR
0505 0338 BB87            ZBSR    *COUT
0506 033A 20              EORZ    R0
0507 033B CC0817          STRA,R0 BCC
0508 033E 09E9            LODR,R1 *ATEMQ
0509 0340 0E832C          LODA,R2 *ATEMQ1
0510 0343 AE080E          SUBA,R2 TEMP+1  GET BYTE COUNT
0511 0344           ATEMP1 SET    $-2
0512 0346 770A            PPSL    WC+COM
0513 0348 AD080D          SUBA,R1 TEMP
0514 0349           ATEMP SET     $-2
0515 034B 7508            CPSL    WC
0516 034D 1E0038          BCTA,N  EBUG    START > END ADDR
0517                *CALC DIFFERENCE, AND USE 1E IF ITS LESS THAN REMA
0518 0350 591F            BRNR,R1 ADUM    REMAINDER > 255
0519 0352 5A19            BRNR,R2 GDUM
0520 0354 0704            LODI,R3 4       EOF, PUNCH ZERO BLK
0521 0356 BB8B      CDUM  ZBSR   *BOUT
0522 0358 0500            LODI,R1 0
0523 035A FB7A            BDRR,R3 CDUM
0524 035C 3B06            BSTR,UN GAP
0525 035E 9B3C            ZBRR    MBUG
0526                * SUBRS FOR OUTPUTTING BLANKS
0527 0360 0703      FORM  LODI,R3 3
0528 0362 1B02            BCTR,UN GAPA
0529 0364 0732      GAP   LODI,R3 50
0530 0366 0420      GAPA  LODI,R0 SPAC
0531 0368 BB87            ZBSR    *COUT
0532 036A FB7A            BDRR,R3 GAPA
0533 036C 17              RETC,UN
0534 036D E61E      GDUM  COMI,R2 H'1E'
```

```
LINE ADDR  OBJECT  E SOURCE

0535 036F 1A02          BCTR,LT BDUM
0536 0371 061E      ADUM  LODI,R2 H'1E'
0537 0373 CE0814    BDUM  STRA,R2 CNT
0538 0374           ACNT  SET     $-2
0539 0376 09D1          LODR,R1 *ATEMP  STARTING ADDRESS
0540 0378 BB8B          ZBSR   *BOUT
0541 037A 09C8          LODR,R1 *ATEMP1
0542 037C BB8B          ZBSR   *BOUT
0543 037E 09F4          LODR,R1 *ACNT   COUNT OF DATA BYTES IN BLOC
0544 0380 BB8B          ZBSR   *BOUT
0545 0382 0D0817        LODA,R1 BCC
0546 0383           ABCC  SET     $-2
0547 0385 BB8B          ZBSR   *BOUT
0548                *NOW OUTPUT THE DATA PART
0549 0387 0BEB          LODR,R3 *ACNT
0550 0389 0D880D    DDUM  LODA,R1 *TEMP
0551 038C BB8B          ZBSR    *BOUT
0552 038E BB17          ZBSR    INCRT
0553 0390 FB77          BDRR,R3 DDUM
0554 0392 09EF      EDUM  LODR,R1 *ABCC
0555 0394 BB8B          ZBSR   *BOUT
0556 0396 1F032E        BCTA,UN FDUM
0557                * LOAD FROM PAPER TAPE IN OBJECT FORMAT
0558 0399 BB89      LOAD  ZBSR    *CHIN   LOOK FOR START CHAR
0559 039B E43A          COMI,R0 STAR
0560 039D 987A          BCFR,EQ LOAD
0561 039F 20            EORZ    R0
0562 03A0 CC0817        STRA,R0 BCC
0563 03A1           ABCC  SET     $-2
0564 03A3 BB8D          ZBSR    *BIN    READ ADDR AND COUNT INFO
0565 03A5 CD080D        STRA,R1 TEMP
0566 03A8 BB8D          ZBSR    *BIN
0567 03AA CD080E        STRA,R1 TEMP+1
0568 03AD BB8D          ZBSR    *BIN
0569 03AF 01            LODZ    R1
0570 03B0 1C003C        BCTA,Z  MBUG
0571 03B3 C3        ALOA  STRZ    R3
0572 03B4 BB8D          ZBSR    *BIN    CHECK BCC ON DATA
0573 03B6 08E9          LODR,R0 *ABCC
0574 03B8 9C0038        BCFA,Z  EBUG
0575 03BB BB8D      BLOA  ZBSR    *BIN
0576 03BD CD880D        STRA,R1 *TEMP       STORE DATA
0577 03C0 BB17          ZBSR    INCRT
0578 03C2 FB77          BDRR,R3 BLOA
0579 03C4 BB8D          ZBSR    *BIN
0580 03C6 08D9          LODR,R0 *ABCC
0581 03C8 184F          BCTR,Z  LOAD
0582 03CA 9B38          ZBRR    EBUG
0583                *
0584 03CC 073F      INIT  LODI,R3 63      ZERO DATA AREA
0585 03CE 20            EORZ    R0
0586 03CF CF4800    AINI  STRA,R0 COMM,R3,-
0587 03D2 5B7B          BRNR,R3 AINI
0588 03D4 0477          LODI,R0 H'77'
0589 03D6 CC0807        STRA,R0 COMM+7  SET FLAG PIN TO STOP BIT L
0590 03D9 CC0809        STRA,R0 XGOT    LOAD CODE TO SET PSU
```

```
LINE ADDR  OBJECT  E SOURCE

0591 03DC 041B            LODI,R0 H'1B'
0592 03DE CC080B          STRA,R0 XGOT+2
0593 03E1 0480            LODI,R0 H'80'
0594 03E3 CC080C          STRA,R0 XGOT+3
0595                *THIS CODE WILL AUTO-SYNCH ITS BAUD RATE
0596                *ON START-UP OR RESET OF PIPBUG, TYPE A 'U'
0597                *BAUD CONTAINS THE DELAY COUNT
0598 00F9           D110   EQU     249     DELAY FOR 110 BAUD
0599 0059           D300   EQU     89      DELAY FOR 300 BAUD
0600 03E6 7640      SYNCH  PPSU    FLAG
0601 03E8 0459            LODI,R0 D300
0602 03EA CC0816          STRA,R0 BAUD
0603 03EB           ABAUD  SET     $-2
0604 03ED 12        ASYN   SPSU
0605 03EE 1A7D            BCTR,N  ASYN
0606 03F0 3F02C8          BSTA,UN HDLY
0607 03F3 3F02C2          BSTA,UN DELAY
0608 03F6 12              SPSU    IF LOW THEN 110,ELSE 300
0609 03F7 1E003C          BCTA,N  MBUG
0610 03FA 04F9            LODI,R0 D110
0611 03FC C8ED            STRR,R0 *ABAUD
0612 03FE 9B3C            ZBRR    MBUG
0613 0400                 ORG     H'800'
0614                ********        RAM DEFINITIONS
0615 0800           COMM    RES     9
0616 0809 7700      XGOT   PPSL    0
0617 080B 1B80             BCTR,UN *$+2    MUST PRECEED THE TEMP LOC
0618 080D           TEMP   RES     2
0619 080F           TEMQ   RES     2
0620 0811           TEMR   RES     1
0621 0812           TEMS   RES     1
0622 0813           MCNT   RES     1
0623 0814           CNT    RES     1
0624 0815           CODE   RES     1
0625 0816           BAUD   RES     1
0626 0817           BCC    RES     1
0627 0818           MARK   RES     BMAX+1
0628 081A           HDAT   RES     BMAX+1
0629 081C           LDAT   RES     BMAX+1
0630 081E           HADR   RES     BMAX+1
0631 0820           LADR   RES     BMAX+1
0632 0822           BUFF   RES     1
0633 03CC                  END     INIT

 TOTAL ASSEMBLY ERRORS = 0000
```

Appendix 2

PIPLA Source listing

```
TWIN ASSEMBLER VER 2650  1K ASSEMBLER VER 1.0                          PAGE     2

 LOC OBJECT   ADDR E STMT   SOURCE LINE

                      2   ********************
                      3   *  DEFINITIONS OF SYMBOLS
                      4   *   REGISTER EQUATES
                 0000     5   R0      EQU     0       REGISTER 0
                 0001     6   R1      EQU     1       REGISTER 1
                 0002     7   R2      EQU     2       REGISTER 2
                 0003     8   R3      EQU     3       REGISTER 3
                      9   *  CONDITION CODES
                 0001    10   P       EQU     1       POSITIVE RESULT
                 0000    11   Z       EQU     0       ZERO RESULT
                 0002    12   N       EQU     2       NEGATIVE RESULT
                 0002    13   LT      EQU     2       LESS THAN
                 0000    14   EQ      EQU     0       EQUAL TO
                 0001    15   GT      EQU     1       GREATER THAN
                 0003    16   UN      EQU     3       UNCONDITIONAL
                     17   *  PSW LOWER EQUATES
                 0000    18   CC      EQU     H'00'   CONDITIONAL CODES
                 0020    19   IDC     EQU     H'20'   INTERDIGIT CARRY
                 0010    20   RS      EQU     H'10'   REGISTER BANK
                 0008    21   WC      EQU     H'08'   1=WITH 0=WITHOUT CARRY
                 0004    22   OVF     EQU     H'04'   OVERFLOW
                 0002    23   COM     EQU     H'02'   1=LOGIC 0=ARITHMETIC COMPARE
                 0001    24   C       EQU     H'01'   CARRY/BORROW
                     25   *  PSW UPPER EQUATES
                 0080    26   SENS    EQU     H'80'   SENSE BIT
                 0040    27   FLAG    EQU     H'40'   FLAG BIT
                 0020    28   II      EQU     H'20'   INTERRUPT INHIBIT
                 0007    29   SP      EQU     H'07'   STACK POINTER
                     30   *
                     31   *   PIPBUG ROUTINES USED
                     32   *
                 0007    33   COUT    EQU     H'0007' CHAR OUTPUT
                 000B    34   BOUT    EQU     H'000B' BYTE OUTPUT
                 000F    35   GNUM    EQU     H'000F' GET A NUMBER
                 0011    36   LINE    EQU     H'0011' GET A LINE
                 0017    37   INCRT   EQU     H'0017' INCREMENT TEMP
                 0021    38   STRT    EQU     H'0021' STORE INTO TEMP
                 0036    39   ABRT    EQU     H'0036' ABORT
                 003E    40   MBUG    EQU     H'003E' START OF COMMAND LINE
                     41   *
                     42   *   MISC. EQUATES
                     43   *
                 0040    44   LABEL   EQU     A'@'
                 0020    45   SPAC    EQU     H'20'
                 002A    46   CMNT    EQU     A'*'
                 002E    47   PROMPT  EQU     A'.'
                 000A    48   LBLS    EQU     10      NUMBER OF LABELS ALLOWED
                     49   * END OF EQUATES
```

```
 LOC OBJECT   ADDR E STMT   SOURCE LINE

0400                 51          ORG     H'0400'
0400 20              52  LASM    EORZ    R0
0401 0714            53          LODI,R3 LBLS*2
0403 CF484E  084E    54  ILP     STRA,R0 LABL,R3,- INITIALIZE LABEL POINTER
0406 5B7B    0403    55          BRNR,R3 ILP
0408 050C            56          LODI,R1 <LABLND
040A 0600            57          LODI,R2 >LABLND
040C BB21    0021    58  IMAIN   ZBSR    STRT    STORE PC
                     59  * RESTART AFTER ERROR HERE
040E 0D080D  080D    60  MAIN    LODA,R1   TEMP
0411 BB8B    000B    61          ZBSR    *BOUT
0413 0D080E  080E    62          LODA,R1 TEMP+1
0416 BB8B    000B    63          ZBSR    *BOUT
0418 042E            64          LODI,R0 PROMPT
041A BB87    0007    65          ZBSR    *COUT   OUTPUT PROMPT
041C BB91    0011    66          ZBSR    *LINE   GET A LINE
041E 0C0822  0822    67          LODA,R0 BUFF
0421 E42A            68          COMI,R0 CMNT
0423 1869    040E    69          BCTR,EQ MAIN    BYPASS COMMENTS
0425 E440            70          COMI,R0 LABEL   CHECK FOR LABEL
0427 983C    0465    71          BCFR,EQ MN1
                     72  *
                     73  *   FIX ALL REFS TO LABEL
                     74  *
0429 0F2822  0822    75          LODA,R0 BUFF,R3,+
042C A430            76          SUBI,R0 A'0'
042E 1E0036  0036    77          BCTA,N  ABRT
             042F    78  AABRT   SET     $-2
0431 C3              79          STRZ    R3
0432 E709            80          COMI,R3 LBLS-1
0434 19F9    042F    81          BCTR,GT *AABRT
0436 D3              82          RRL,R3
0437 0601            83          LODI,R2 1
0439 0F684E  084E    84  LLOOP   LODA,R0 LABL,R3 SAVE ADDRESS OF CURRENT INST. TO FIX
043C CC080F  080F    85          STRA,R0 TEMQ
043F C1              86          STRZ    R1
0440 0F684F  084F    87          LODA,R0 LABL+1,R3
0443 CC0810  0810    88          STRA,R0 TEMQ+1
0446 61              89          IORZ    R1      CHECK IF DONE
0447 181A    0463    90          BCTR,Z  LLEND
0449 0C880F  080F    91          LODA,R0 *TEMQ   GET ADDRESS OF NEXT INST. TO FIX
044C CF684E  084E    92          STRA,R0 LABL,R3
044F 0EE80F  080F    93          LODA,R0 *TEMQ,R2
0452 CF684F  084F    94          STRA,R0 LABL+1,R3
0455 0C080D  080D    95          LODA,R0 TEMP    FIX ADDRESS IN CURRENT INST.
0458 CC880F  080F    96          STRA,R0 *TEMQ
045B 0C080E  080E    97          LODA,R0 TEMP+1
045E CEE80F  080F    98          STRA,R0 *TEMQ,R2
0461 1B56    0439    99          BCTR,UN LLOOP   GO FIX NEXT INSTRUCTION
0463 0702           100  LLEND   LODI,R3 2       BYPASS LABEL
                    101  *
0465 20             102  MN1     EORZ    R0
0466 CC0815  0815   103          STRA,R0 CODE
0469 3F0569  0569   104          BSTA,UN DEBLK   BYPASS BLANKS
046C 3F0579  0579   105          BSTA,UN FNDS    FIND MNEMONIC IN TABLE
```

```
 LOC OBJECT   ADDR E STMT   SOURCE LINE

046F CC0811  0811    106          STRA,R0 TEMR
0472 60              107          IORZ    R0
0473 9A2A    049F    108          BCFR,N  OPCD
0475 440F            109          ANDI,R0 H'0F'
0477 1C003E  003E    110          BCTA,Z  MBUG    GO TO PIPBUG IF END
                     111  *
                     112  *   CHECK FOR ASCII STRING
                     113  *
047A F402            114          TMI,R0  H'02'   CHECK FOR ASCI
047C 981C    049A    115          BCFR,EQ MN2
047E 3F0569  0569    116          BSTA,UN DEBLK
0481 0F6822  0822    117          LODA,R0 BUFF,R3 GET DELIMITTER
0484 C2              118          STRZ    R2
0485 0F2822  0822    119  ASCLP   LODA,R0 BUFF,R3,+
0488 EF0814  0814    120          COMA,R3 CNT     CHECK FOR END OF LINE
048B 9E040E  040E    121          BCFA,LT MAIN
             048C    122  AMAIN   SET     $-2
048E E2              123          COMZ    R2      CHECK FOR END DELIMITTER
048F 18FB    048C    124          BCTR,EQ *AMAIN
0491 CC880D  080D    125          STRA,R0 *TEMP   STORE CHARACTER
0494 02              126          LODZ    R2
0495 BB17    0017    127          ZBSR    INCRT   INCREMENT PC
0497 C2              128          STRZ    R2
0498 1B6B    0485    129          BCTR,UN ASCLP
                     130  *
049A BB8F    000F    131  MN2     ZBSR    *GNUM   MUST BE ORG - GET ADDRESS
049C 1F040C  040C    132          BCTA,UN IMAIN   GO STORE IT IN PC
                     133  *
                     134  *   HAVE INSTRUCTION - NOT PSEUDO OP
                     135  *
049F CE880D  080D    136  OPCD    STRA,R2 *TEMP   STORE OPCODE IN RAM
04A2 E410            137          COMI,R0 H'10'
04A4 9A20    04C6    138          BCFR,LT AFTREG
04A6 8701            139          ADDI,R3 1       INCREMENT PAST , OR SPACE
04A8 3F0569  0569    140          BSTA,UN DEBLK   BYPASS BLANKS
             04A9    141  ADEBLK  SET     $-2
04AB 0F6822  0822    142          LODA,R0 BUFF,R3
04AE E440            143          COMI,R0 A'@'    CHECK FOR SYMBOL
04B0 9909    04BB    144          BCFR,GT MN3
04B2 3F0579  0579    145          BSTA,UN FNDS    GET VALUE OF SYMBOL
04B5 8410            146          ADDI,R0 H'10'
04B7 1804    04BD    147          BCTR,Z  MN4
04B9 9B36    0036    148          ZBRR    ABRT    ABORT IF FOUND OTHER THAN SYMBOL
04BB BB8F    000F    149  MN3     ZBSR    *GNUM   GET REG OR CONDITION
04BD 4603            150  MN4     ANDI,R2 H'03'
04BF 0C880D  080D    151          LODA,R0 *TEMP   GET CURRENT FIRST BYTE
04C2 62              152          IORZ    R2      OR ON REG OR CONDITION
04C3 CC880D  080D    153          STRA,R0 *TEMP   STORE COMPLETE FIRST BYTE
04C6 BB17    0017    154  AFTREG  ZBSR    INCRT   INCREMENT PC
04C8 0C0811  0811    155          LODA,R0 TEMR
04CB F401            156          TMI,R0  1
04CD 1C040E  040E    157          BCTA,EQ MAIN    DONE IF TYPE E OR Z
04D0 F402            158          TMI,R0  2
04D2 1C0522  0522    159          BCTA,EQ GTBYT2  NO * ALLOWED IF TYPE I OR EI
                     160  *
```

```
 LOC OBJECT   ADDR E STMT   SOURCE LINE

                     161   *   CHECK FOR INDEX AND/OR INDIRECT
                     162   *
04D5 3BD2      04A9  163   ARLOOP BSTR,UN *ADEBLK
04D7 0F6822    0822  164          LODA,R0 BUFF,R3
04DA E430            165          COMI,R0 H'30'   CHECK FOR SPECIAL CHARACTER
04DC 9A1B      04F9  166          BCFR,LT CHKBYT  NO - GO GET NUMBER
04DE 8701            167          ADDI,R3 1
04E0 05FF            168          LODI,R1 -1
04E2 ED25CF    05CF  169   AR2    COMA,R0 STBL,R1,+       FIND SPECIAL CHAR IN TABLE
04E5 1806      04ED  170          BCTR,EQ AR3
04E7 E504            171          COMI,R1 4
04E9 1A77      04E2  172          BCTR,LT AR2
04EB 9B36      0036  173          ZBRR    ABRT
04ED D1              174   AR3    RRL,R1
04EE D1              175          RRL,R1
04EF D1              176          RRL,R1
04F0 D1              177          RRL,R1
04F1 D1              178          RRL,R1
04F2 6D0815    0815  179          IORA,R1 CODE
               04F3  180   ACODE  SET     $-2
04F5 C9FC      04F3  181          STRR,R1 *ACODE
04F7 1B5C      04D5  182          BCTR,UN ARLOOP  GO CHECK FO ANOTHER SPECIAL CHAR
                     183   *
                     184   *   CHECK FOR LABEL AS ADDRESS
                     185   *
04F9 E440            186   CHKBYT COMI,R0 LABEL   CHECK FOR A LABEL
04FB 9825      0522  187          BCFR,EQ GTBYT2
04FD 0F2822    0822  188          LODA,R0 BUFF,R3,+
0500 A430            189          SUBI,R0 A'0'
0502 1E0036    0036  190          BCTA,N  ABRT
0505 C3              191          STRZ    R3
0506 E709            192          COMI,R3 LBLS-1
0508 1D0036    0036  193          BCTA,GT ABRT
050B D3              194          RRL,R3
050C 0F684E    084E  195          LODA,R0 LABL,R3
050F C1              196          STRZ    R1
0510 0F684F    084F  197          LODA,R0 LABL+1,R3
0513 C2              198          STRZ    R2
0514 0C080D    080D  199          LODA,R0 TEMP
0517 CF684E    084E  200          STRA,R0 LABL,R3
051A 0C080E    080E  201          LODA,R0 TEMP+1
051D CF684F    084F  202          STRA,R0 LABL+1,R3
0520 1B36      0558  203          BCTR,UN TWOBYT
                     204   *
0522 BB8F      000F  205   GTBYT2 ZBSR    *GNUM
0524 0C0811    0811  206          LODA,R0 TEMR
0527 F408            207          TMI,R0  H'08'   CHECK IF ADDRESS OR BYTE
0529 182D      0558  208          BCTR,EQ TWOBYT
052B F404            209          TMI,R0  H'04'
052D 9824      0553  210          BCFR,EQ ONEBYT
052F 7709            211          PPSL    WC+C
0531 A601            212          SUBI,R2 1
0533 A500            213          SUBI,R1 0
0535 7701            214          PPSL    C
0537 AE080E    080E  215          SUBA,R2 TEMP+1
```

```
 LOC OBJECT   ADDR E STMT   SOURCE LINE

053A AD080D   080D    216          SUBA,R1 TEMP
053D 7508             217          CPSL    WC
053F 180D     054E    218          BCTR,Z  POS
0541 8501             219          ADDI,R1 1
0543 9C0036   0036    220          BCFA,Z  ABRT
              0544    221  AABRT   SET     $-2
0546 F6C0             222          TMI,R2  H'C0'
0548 98FA     0544    223          BCFR,EQ *AABRT
054A 467F             224          ANDI,R2 H'7F'
054C 1B05     0553    225          BCTR,UN ONEBYT
054E 04C0             226  POS     LODI,R0 H'C0'
0550 42               227          ANDZ    R2
0551 98F1     0544    228          BCFR,Z  *AABRT
0553 6E0815   0815    229  ONEBYT  IORA,R2 CODE
              0554    230  ACODE   SET     $-2
0556 1B09     0561    231          BCTR,UN PUTSEC
0558 69FA     0554    232  TWOBYT  IORR,R1 *ACODE
055A CD880D   080D    233          STRA,R1 *TEMP
055D 02               234          LODZ    R2
055E BB17     0017    235          ZBSR    INCRT
0560 C2               236          STRZ    R2
0561 CE880D   080D    237  PUTSEC: STRA,R2 *TEMP
0564 BB17     0017    238          ZBSR    INCRT
0566 1F040E   040E    239          BCTA,UN MAIN
```

```
 LOC OBJECT   ADDR E STMT   SOURCE LINE

                     241   *
                     242   *   BYPASS BLANKS IN BUFF
                     243   *
0569 0420            244   DEBLK  LODI,R0 SPAC
056B FB00    056D    245          BDRR,R3 DBLOOP
056D EF2822  0822    246   DBLOOP COMA,R0 BUFF,R3,+
0570 187B    056D    247          BCTR,EQ DBLOOP
0572 EF0814  0814    248          COMA,R3 CNT
0575 9E0036  0036    249          BCFA,LT ABRT
0578 17              250          RETC,UN
                     251   *
                     252   *   FIND SYMBOL POINTED TO BY R3
                     253   *
0579 06FC            254   FNDS   LODI,R2 -4      GET 4 BYTES TO MNEM
057B A701            255          SUBI,R3 1
057D 0F2822  0822    256   FMLOOP LODA,R0 BUFF,R3,+
0580 EF0814  0814    257          COMA,R3 CNT
0583 9E0593  0593    258          BCFA,LT FML2
0586 E430            259          COMI,R0 H'30'
0588 1A09    0593    260          BCTR,LT FML2
058A CE674E  074E    261          STRA,R0 (MNEM-252).AND.H'1FFF',R2
058D DA6E    057D    262          BIRR,R2 FMLOOP
058F 8701            263          ADDI,R3 1
0591 1B07    059A    264          BCTR,UN FS
0593 0420            265   FML2   LODI,R0 H'20'
0595 CE674E  074E    266   FML3   STRA,R0 (MNEM-252).AND.H'1FFF',R2
0598 DA7B    0595    267          BIRR,R2 FML3
059A CF0813  0813    268   FS     STRA,R3 MCNT
059D 7501            269          CPSL    C
059F 7708            270          PPSL    WC
05A1 0505            271          LODI,R1 <TABL
05A3 06D4            272          LODI,R2 >TABL
05A5 CD080F  080F    273   FSLOOP STRA,R1 TEM0
05A8 CE0810  0810    274          STRA,R2 TEM0+1
05AB 07FF            275          LODI,R3 -1
05AD 0FA80F  080F    276   FSL2   LODA,R0 *TEM0,R3,+
05B0 1C0036  0036    277          BCTA,Z  ABRT
05B3 EF684A  084A    278          COMA,R0 MNEM,R3
05B6 1806    05BE    279          BCTR,EQ FSL3
05B8 8606            280          ADDI,R2 6       INCREMENT TO NEXT MNEMONIC
05BA 8500            281          ADDI,R1 0
05BC 1B67    05A5    282          BCTR,UN FSLOOP
05BE E703            283   FSL3   COMI,R3 3
05C0 1A6B    05AD    284          BCTR,LT FSL2
05C2 0FA80F  080F    285          LODA,R0 *TEM0,R3,+      GET VALUE OF MNEMONIC
05C5 C2              286          STRZ    R2
05C6 0FA80F  080F    287          LODA,R0 *TEM0,R3,+      GET MNEMONIC TYPE
05C9 0F0813  0813    288          LODA,R3 MCNT
05CC 7508            289          CPSL    WC
05CE 17              290          RETC,UN
```

```
 LOC OBJECT   ADDR E STMT   SOURCE LINE

                     292   *
                     293   *   TABLES
                     294   *
05CF 2C2B2D23        295   STBL   DATA    A',+-#*'
     2A
                     296   *
             05D4    297   TABL   EQU     $
05D4 52302020        298          DATA    A'R0  '
05D8 00F0            299          DATA    H'00,F0'
05DA 52312020        300          DATA    A'R1  '
05DE 01F0            301          DATA    H'01,F0'
05E0 52322020        302          DATA    A'R2  '
05E4 02F0            303          DATA    H'02,F0'
05E6 52332020        304          DATA    A'R3  '
05EA 03F0            305          DATA    H'03,F0'
05EC 50202020        306          DATA    A'P   '
05F0 01F0            307          DATA    H'01,F0'
05F2 5A202020        308          DATA    A'Z   '
05F6 00F0            309          DATA    H'00,F0'
05F8 4E202020        310          DATA    A'N   '
05FC 02F0            311          DATA    H'02,F0'
05FE 4C542020        312          DATA    A'LT  '
0602 02F0            313          DATA    H'02,F0'
0604 45512020        314          DATA    A'EQ  '
0608 00F0            315          DATA    H'00,F0'
060A 47542020        316          DATA    A'GT  '
060E 01F0            317          DATA    H'01,F0'
0610 554E2020        318          DATA    A'UN  '
0614 03F0            319          DATA    H'03,F0'
0616 454E4420        320          DATA    A'END '
061A 0080            321          DATA    H'00,80'
061C 4F524720        322          DATA    A'ORG '
0620 0081            323          DATA    H'00,81'
0622 41534349        324          DATA    A'ASCI'
0626 0082            325          DATA    H'00,82'
0628 4C4F445A        326          DATA    A'LODZ'
062C 0001            327          DATA    H'00,01'
062E 4C4F4449        328          DATA    A'LODI'
0632 0402            329          DATA    H'04,02'
0634 4C4F4452        330          DATA    A'LODR'
0638 0804            331          DATA    H'08,04'
063A 4C4F4441        332          DATA    A'LODA'
063E 0C08            333          DATA    H'0C,08'
0640 5354525A        334          DATA    A'STRZ'
0644 C001            335          DATA    H'C0,01'
0646 53545252        336          DATA    A'STRR'
064A C804            337          DATA    H'C8,04'
064C 53545241        338          DATA    A'STRA'
0650 CC08            339          DATA    H'CC,08'
0652 494F525A        340          DATA    A'IORZ'
0656 6001            341          DATA    H'60,01'
0658 494F5249        342          DATA    A'IORI'
065C 6402            343          DATA    H'64,02'
065E 494F5252        344          DATA    A'IORR'
0662 6804            345          DATA    H'68,04'
```

```
 LOC OBJECT   ADDR E STMT   SOURCE LINE

0664 494F5241          346          DATA    A'IORA'
0668 6C08              347          DATA    H'6C,08'
066A 414E445A          348          DATA    A'ANDZ'
066E 4001              349          DATA    H'40,01'
0670 414E4449          350          DATA    A'ANDI'
0674 4402              351          DATA    H'44,02'
0676 414E4452          352          DATA    A'ANDR'
067A 4804              353          DATA    H'48,04'
067C 414E4441          354          DATA    A'ANDA'
0680 4C08              355          DATA    H'4C,08'
0682 454F525A          356          DATA    A'EORZ'
0686 2001              357          DATA    H'20,01'
0688 454F5249          358          DATA    A'EORI'
068C 2402              359          DATA    H'24,02'
068E 454F5252          360          DATA    A'EORR'
0692 2804              361          DATA    H'28,04'
0694 454F5241          362          DATA    A'EORA'
0698 2C08              363          DATA    H'2C,08'
069A 42435452          364          DATA    A'BCTR'
069E 1804              365          DATA    H'18,04'
06A0 42435441          366          DATA    A'BCTA'
06A4 1C0C              367          DATA    H'1C,0C'
06A6 42434652          368          DATA    A'BCFR'
06AA 9804              369          DATA    H'98,04'
06AC 42434641          370          DATA    A'BCFA'
06B0 9C0C              371          DATA    H'9C,0C'
06B2 434F4D5A          372          DATA    A'COMZ'
06B6 E001              373          DATA    H'E0,01'
06B8 434F4D49          374          DATA    A'COMI'
06BC E402              375          DATA    H'E4,02'
06BE 434F4D52          376          DATA    A'COMR'
06C2 E804              377          DATA    H'E8,04'
06C4 434F4D41          378          DATA    A'COMA'
06C8 EC08              379          DATA    H'EC,08'
06CA 4144445A          380          DATA    A'ADDZ'
06CE 8001              381          DATA    H'80,01'
06D0 41444449          382          DATA    A'ADDI'
06D4 8402              383          DATA    H'84,02'
06D6 41444452          384          DATA    A'ADDR'
06DA 8804              385          DATA    H'88,04'
06DC 41444441          386          DATA    A'ADDA'
06E0 8C08              387          DATA    H'8C,08'
06E2 5355425A          388          DATA    A'SUBZ'
06E6 A001              389          DATA    H'A0,01'
06E8 53554249          390          DATA    A'SUBI'
06EC A402              391          DATA    H'A4,02'
06EE 53554252          392          DATA    A'SUBR'
06F2 A804              393          DATA    H'A8,04'
06F4 53554241          394          DATA    A'SUBA'
06F8 AC08              395          DATA    H'AC,08'
06FA 52455443          396          DATA    A'RETC'
06FE 1401              397          DATA    H'14,01'
0700 52455445          398          DATA    A'RETE'
0704 3401              399          DATA    H'34,01'
0706 42535452          400          DATA    A'BSTR'
```

```
 LOC OBJECT   ADDR E STMT   SOURCE LINE

070A 3804            401          DATA   H'38,04'
070C 42535441        402          DATA   A'BSTA'
0710 3C0C            403          DATA   H'3C,0C'
0712 42534652        404          DATA   A'BSFR'
0716 B804            405          DATA   H'B8,04'
0718 42534641        406          DATA   A'BSFA'
071C BC0C            407          DATA   H'BC,0C'
071E 52525220        408          DATA   A'RRR '
0722 5001            409          DATA   H'50,01'
0724 52524C20        410          DATA   A'RRL '
0728 D001            411          DATA   H'D0,01'
072A 43505355        412          DATA   A'CPSU'
072E 7412            413          DATA   H'74,12'
0730 4350534C        414          DATA   A'CPSL'
0734 7512            415          DATA   H'75,12'
0736 50505355        416          DATA   A'PPSU'
073A 7612            417          DATA   H'76,12'
073C 5050534C        418          DATA   A'PPSL'
0740 7712            419          DATA   H'77,12'
0742 42524E52        420          DATA   A'BRNR'
0746 5804            421          DATA   H'58,04'
0748 42524E41        422          DATA   A'BRNA'
074C 5C0C            423          DATA   H'5C,0C'
074E 42495252        424          DATA   A'BIRR'
0752 D804            425          DATA   H'D8,04'
0754 42495241        426          DATA   A'BIRA'
0758 DC0C            427          DATA   H'DC,0C'
075A 42445252        428          DATA   A'BDRR'
075E F804            429          DATA   H'F8,04'
0760 42445241        430          DATA   A'BDRA'
0764 FC0C            431          DATA   H'FC,0C'
0766 42534E52        432          DATA   A'BSNR'
076A 7804            433          DATA   H'78,04'
076C 42534E41        434          DATA   A'BSNA'
0770 7C0C            435          DATA   H'7C,0C'
0772 4E4F5020        436          DATA   A'NOP '
0776 C011            437          DATA   H'C0,11'
0778 48414C54        438          DATA   A'HALT'
077C 4011            439          DATA   H'40,11'
077E 544D4920        440          DATA   A'TMI '
0782 F402            441          DATA   H'F4,02'
0784 57525444        442          DATA   A'WRTD'
0788 F001            443          DATA   H'F0,01'
078A 52454444        444          DATA   A'REDD'
078E 7001            445          DATA   H'70,01'
0790 57525443        446          DATA   A'WRTC'
0794 B001            447          DATA   H'B0,01'
0796 52454443        448          DATA   A'REDC'
079A 3001            449          DATA   H'30,01'
079C 57525445        450          DATA   A'WRTE'
07A0 D402            451          DATA   H'D4,02'
07A2 52454445        452          DATA   A'REDE'
07A6 5402            453          DATA   H'54,02'
07A8 5A425352        454          DATA   A'ZBSR'
07AC BB10            455          DATA   H'BB,10'
```

```
 LOC OBJECT   ADDR E STMT   SOURCE LINE

07AE 5A425252          456          DATA    A'ZBRR'
07B2 9B10              457          DATA    H'9B,10'
07B4 54505355          458          DATA    A'TPSU'
07B8 B412              459          DATA    H'B4,12'
07BA 5450534C          460          DATA    A'TPSL'
07BE B512              461          DATA    H'B5,12'
07C0 4C505355          462          DATA    A'LPSU'
07C4 9211              463          DATA    H'92,11'
07C6 4C50534C          464          DATA    A'LPSL'
07CA 9311              465          DATA    H'93,11'
07CC 53505355          466          DATA    A'SPSU'
07D0 1211              467          DATA    H'12,11'
07D2 5350534C          468          DATA    A'SPSL'
07D6 1311              469          DATA    H'13,11'
07D8 42535841          470          DATA    A'BSXA'
07DC BF1C              471          DATA    H'BF,1C'
07DE 42584120          472          DATA    A'BXA '
07E2 9F1C              473          DATA    H'9F,1C'
07E4 44415220          474          DATA    A'DAR '
07E8 9401              475          DATA    H'94,01'
07EA 4C44504C          476          DATA    A'LDPL'
07EE 101C              477          DATA    H'10,1C'
07F0 5354504C          478          DATA    A'STPL'
07F4 111C              479          DATA    H'11,1C'
07F6 00000000          480          DATA    H'00,00,00,00,00,00'
     0000
                       482  *
                       483  *   RAM DEFINITIONS
                       484  *
0800                   485          ORG     H'0800'
0800                   486          RES     13
080D                   487  TEMP    RES     2
080F                   488  TEMQ    RES     2
0811                   489  TEMR    RES     1
0812                   490  TEMS    RES     1
0813                   491  MCNT    RES     1
0814                   492  CNT     RES     1
0815                   493  CODE    RES     1
0816                   494          RES     12
0822                   495  BUFF    RES     40
084A                   496  MNEM    RES     4
084E                   497  LABL    RES     LBLS*2
0C00                   498          ORG     H'0C00'
              0C00     499  LABLND  EQU     $
                       500  *
              0400     501          END     LASM

  TOTAL NUMBER OF ERRORS    0
```

Appendix 3

ASCII conversion table

ACSII CHARACTER SET (7-BIT CODE)									
L.S. CHAR \ M.S. CHAR		0 000	1 001	2 010	3 011	4 100	5 101	6 110	7 111
0	0000	NUL	DLE	SP	0	@	P		p
1	0001	SOH	DC1	!	1	A	Q	a	q
2	0010	STX	DC2	"	2	B	R	b	r
3	0011	ETX	DC3	#	3	C	S	c	s
4	0100	EOT	DC4	$	4	D	T	d	t
5	0101	ENQ	NAK	%	5	E	U	e	u
6	0110	ACK	SYN	&	6	F	V	f	v
7	0111	BEL	ETB	'	7	G	W	g	w
8	1000	BS	CAN	(	8	H	X	h	x
9	1001	HT	EM	)	9	I	Y	i	y
A	1010	LF	SUB	*	:	J	Z	j	z
B	1011	VT	ESC	+	;	K	[	k	{
C	1100	FF	FS	,	<	L	\	l	|
D	1101	CR	GS	–	=	M	]	m	}
E	1110	SO	RS	•	>	N	↑	n	~
F	1111	SI	US	/	?	O	← or _	o	DEL

Related 2650 Applications Memos

2650 Family Publications

no.	title	summary	ordering code
AS50	Serial Input/Output	Using the Sense/Flag capability of the 2650 for serial I/O interfaces	9398 906 00611
AS51	Bit & Byte Testing Procedures	Several methods of testing the contents of the internal registers in the 2650	9399 509 53661
AS52	General Delay Routines	Several time delay routines for the 2650, incl. formulas for calculating the delay time.	9399 509 53761
AS53	Binary Arithmetic Routines	Examples for processing binary arithmetic addition, subtraction, multiplication and division with the 2650.	9399 509 53861
AS54	Conversion Routines	• Eight-bit unsigned binary to BCD • Sixteen-bit signed binary to BCD • Signed BCD to binary • Signed BCD to ASCII • ASCII to BCD • Hexadecimal to ASCII • ASCII to Hexadecimal	9399 509 53961
AS55	Fixed Point Decimal Arithmetic	Methods of performing addition, subtraction, multiplication and division of BCD numbers with the 2650.	9398 907 80011
SP50	2650 Evaluation Printed Circuit Board (PC1001)	Detailed description of the PC1001, an evaluation and design tool for the 2650.	9398 905 80611
SP51	2650 Demo System	Detailed description of the Demo System, a hardware base for use with the 2650 CPU prototyping board (PC1001 or PC1500).	9399 509 54061
SP52	Support Software for use with the NCSS Timesharing System	Step-by-step procedures for generating, editing, assembling, punching, and simulating Signetics 2650 programs using the NCSS timesharing service.	9398 906 40611
SP53	Simulator, Version 1.2	Features and characteristics of version 1.2 of the 2650 simulator.	9398 906 10611
SP54	Support Software for use with the General Electric Mark III Timesharing System	Step-by-step procedures for generating, editing, assembling, simulator and punching Signetics 2650 programs using General Electric's Mark III timesharing system.	9398 906 50611
SP55	The ABC 1500 Adaptable Board Computer	Describes the components and applications of the ABC 1500 system development card.	9398 907 60011
SS50	PIPBUG	Detailed description of PIPBUG, a monitor program designed for use with the 2650.	9398 905 90611
SS51	Absolute Object Format	Describes the absolute object code format for the 2650.	9399 509 54361
MP51	Initialization	Procedures for initializing the 2650 microprocessor memory, and I/O devices to their required initial states.	9398 905 70611
MP52	Low-Cost Clock Generator Circuits	Several clock generator circuits, based on 7400 series TTL, that may be used with the 2650. They include RC, LC and crystal oscillator types.	9399 509 54461
MP53	Address and Data Bus Interfacing Techniques	Examples of interfacing the 2650 address and data busses with ROMs and RAMs, such as the 2608, 2606 and 2602.	9398 907 90011
MP54	2650 Input/Output Structures and Interfaces	Examines the use of the 2650's versatile set of I/O instructions and the interface between the 2650 and I/O ports. A number of application examples for both serial and parallel I/O are given.	9398 907 70011
TN 064	Digital cassette interface for a 2650 microprocessor system	Interface hardware and software for the Philips DCR digital cassette drive.	9398 006 40011
TN 069	2650 Microprocessor keyboard interfaces	Simple interfaces for low-cost keyboard systems.	9398 006 90011
TN 072	Introducing the Signetics 2651 PCI Terminology and operation modes	Description of the 2651 Programmable Communications Interface.	9398 007 20011
TN 083	Using the Signetics 2651 PCI with popular microprocessors	Simple hardware interfaces to use the 2651 Programmable Communications Interface with various microprocessors.	9398 008 30011
TN 084	Using seven-segment LED display with the 2650 microprocessor	Interfaces for single and multi-digit LED displays.	9398 008 40011
TN 085	Cyclic redundancy check by software	A short routine to encode and decode CRC check characters for the 2650.	9398 008 50011
TN 086	Introducing the Signetics 2655 PPI	Description of the 2655 Programmable Peripheral Interface	9398 008 60011
TN 087	Audio cassette recorder interface for the 2650 microprocessor	Economical alternatives to the digital cassette recorder.	9398 008 70011
TN 089	CRT display using a standard TV monitor for 2650-based microcomputers	Describes hardware and software to use a standard TV monitor as an I/O peripheral for a 2650 microcomputer system.	9398 008 90011
TN 092	2650 sorting routines	Sorting routines for single and multiple byte numbers.	9398 009 20011
TN 093	2650 binary floating point routines	Arithmetic routines for binary floating point numbers.	9398 009 30011
TN 094	2650 BCD floating point routines	Arithmetic routines for BCD floating point numbers.	9398 009 40011

Related 2650 Data sheets

product	description	ordering code
2650A	8-bit N-Channel MOS microprocessor (standard version)	9398 209 50011
2650A-1	8-bit N-Channel MOS microprocessor (high speed standard version)	9398 209 50011
2650B	8-bit N-Channel MOS microprocessor (enhanced version)	9398 209 50011
2650B-1	8-bit N-Channel MOS microprocessor (high speed enhanced version)	9398 209 50011
2651	Programmable Communications Interface (PCI)	9398 208 20011
2652	Multi-Protocol Communications Controller (MPCC)	9398 209 80011
2652-1	Multi-Protocol Communications Controller (MPCC – high speed version)	9398 209 80011
2653	Polynomial Generator Checker (PGC)	9398 209 70011
2655	Programmable Peripheral Interface (PPI)	9398 208 00011
2656	System Memory Interface (SMI)	9398 208 70011
2661-1	Enhanced Programmable Communications Interface (EPCI – Baud rate: set 1)	Oct. 1979
2661-2	Enhanced Programmable Communications Interface (EPCI – Baud rate: set 2)	Oct. 1979
2661-3	Enhanced Programmable Communications Interface (EPCI – Baud rate: set 3)	Oct. 1979
2621	TV Sync Generator (PAL)	9398 207 70011
2622	TV Sync Generator (NTSC)	9398 207 80011
2636	Programmable Video Interface (PVI)	9398 207 90011
2637	Universal Video Interface (UVI)	Sept. 1979
KT 9500	Adaptable Board Computer, ABC Kit	see SP 55
PC 1001	Prototyping Board	see SP 50
PC 1500	Adaptable Board Computer, ABC Board	see SP 55
PC 1600	Resident Assembler Board	see TI 52
PC 2000	4K Bytes Static Memory Board	see TP 1604
PC 4000	2656 SMI Emulator Board	9398 208 80011
DS 2000	Prototyping Base	see SP 51
TWIN	Microcomputer Development System	9398 200 10011

Related 2650 Technical Manuals

2650 Microprocessor User's Manual (bound manual)

Contains the complete specification for the 2650 microprocessor. Describes the instruction set, interface signals, the internal organization, and the electrical characteristics.

Includes user guides to the 2650 Assembly Language and the 2650 Simulator. Ordering code 9398 602 50011.

2650 Register Microprocessor User's Manual (loose-leaf)

Same as above with the addition of all 2650 microprocessor application memos with automatic updating service.
Ordering code 9398 602 40011.

Prometheus Manual

A user's guide for the 2650PC1600 resident assembler board. The PC1600 must be used together with the PC1001 or PC1500 board, because it uses the PIPBUG loader and debugger program resident on the prototyping board. Ordering code TP1604.

TWIN 2650 Assembly Language Manual

A user's guide to the 2650 Assembly Language for the TWIN Microcomputer Development System. Ordering code TW09005000JJ.

TWIN Relocatable Assembler Language Manual

A user's guide to the 2650 relocatable assembler for the TWIN microcomputer development system. Ordering code TW09007000JJ.

TWIN Operator's Guide

Describes all aspects of TWIN system operation, from unpacking, through switches and indicators, to the use of the various system development programs. Ordering code TW09003000JJ.

TWIN System Reference Manual

Describes each board in the TWIN system, with functional description and a theory of operation at the block diagram level. A knowledge of microcomputer development systems and the 2650 microprocessor is assumed. Ordering code TW09004000JJ.

TWIN Maintenance Manual

Detailed guidelines for the TWIN system maintenance and the diagnostic software. The manual contains circuit diagrams of all boards used in the System. Ordering code TW09006000JJ.

Data Communications Handbook

The main objective of the book is to aid in guiding the reader to understand the specific Data Communications terms, protocols and formats. Ordering code 9398 606 80011.

Designing with Microcomputers

An introductory text on microcomputer fundamentals for electronic circuit and system designers and managers.
Ordering code 9398 906 60611.

Introduction to the Instructor 50 Desktop Computer

An extensive introduction into microcomputer fundamentals and how these have been implemented and used in the Instructor 50.
Ordering code 9398 915 60011.

The Instructor 50 Desktop Computer Users' Guide

This manual provides tutorial and reference information on the Signetics Instructor 50. Detailed descriptions illustrated with many examples are given on the function of all controls. The manual also contains the full circuit diagrams of the Instructor 50.
Ordering code 9398 606 90011.

Related 2650 Leaflets

Signetics Testware Instrument (TWIN)

A Glossy leaflet on the TWIN Microcomputer Development System which outlines what the TWIN system is, and which options there are. Ordering code 9399 503 18901.

Instructor 50 Desktop Computer, the complete, ready-to-use microprocessor learning package

A short introduction into the Instructor 50 which describes what it does and which the key feature are.
Ordering code 9398 302 80011.

IMS – Industrial Microcomputer System

Reviews the 2650 microcomputer cardset on single Eurocard format. A short description is given of the CPU card, memory cards, Input/Output cards, Teletype card, Debug card, Display panel and Back panel. Ordering code 9398 304 50011.

2650 Reference Guide

This handy card gives the 2650 programmer a quick overview of all instructions, instruction formats, assembler formats and relative address displacements. Ordering code 9398 301 90011.

Printed in The Netherlands Date of release: 6 August 1979 (60) W.E.M. 9398 013 20011