B&C MICROSYSTEMS
6322 MOJAVE DRIVE
SAN JOSE, CA. 95120
PHONE (408) 997-7685
TELEX 4995363

1409 EPROM PROGRAMMER

VERSION 3.1

TECHNICAL MANUAL

SEPTEMBER 1984

INDEX

PAGE

READ FIRST

COMMANDS DESCRIPTION

ADVANCED COMMANDS

TECHNICAL HELP

IMPORTANT

THE I.C. PADS AT LOCATIONS 3C AND 4A IN THE LAY-OUT DIAGRAM HAVE 28 PIN HOLE PATTERNS. WHEN INSERTING A 24 PIN DEVICE (EITHER THE PROGRAM EPROM OR AN E(E)PROM TO BE PROGRAMMED), MAKE SURE THAT PIN 12 (OF 24) GOES TO WHERE PIN 14 (OF 28) SHOULD BE!!

INSERT AN E(E)PROM TO BE PROGRAMMED INTO THE SOCKET ONLY WITH THE 1409 IN THE COMMAND MODE AND WITH THE PROPER MENU SELECTED. THE PROPER PROMPT ON THE SCREEN SHOULD BE CHECKED BEFORE INSERTING THE CHIP. (see note on page 11).

IF YOU HAVE PURCHASED A 1409-1X KIT YOU MUST INSTALL SW2. FIRST CUT THE TRACE THAT SHORTS THE NO.1 POSITION. WHEN PROGRAMMING 27256, SW2, POSITION NO.1 (22V) MUST BE OFF AND POSITION NO.2 (13V) MUST BE ON. FOR ALL OTHER DEVICES POSITION NO.1 MUST BE ON AND NO.2 OFF. IF THIS PROCEDURE IS NOT FOLLOWED YOU MAY DAMAGE THE 27256.

INTRODUCTION

The 1409 E(E)PROM PROGRAMMER is a versatile tool which can program all the popular E(E)PROMS in a very convenient manner. The built-in menu allows easy selection for the desired DEVICE to be programmed. After selection, the programming voltages and the pin configuration are selected by the software. No programming voltages are applied during "read". This makes it possible to use the 1409 for reading the pin compatible ROMS or PROMS without the danger of destroying them.

The 1409 can be hooked up to any terminal or computer with a RS232 port.

Interfacing the 1409 to your computer should be very easy. The standard RS232 interface supports xon/xoff handshaking and monitors the CTS, RTS and DTR lines. You may choose to use any or none of these lines. After power up, the 1409 is always ready for a baud rate search. It will latch itself on any baud rate between 300 and 9600 baud by simply receiving a few "spaces".

The internal menu allows the following DEVICES to be read and programmed without the need for personality modules:

2508
2516
2532
2564

2758
2716
27C16
2732

27C32
2732A
2764
2764A*

27128
27128A*
27256

68732
68764
68766

52B13
52B23
52B33
52B43*

2816A*
2832A*
2864A*

8741
8748
8748H
8749H

*NOT IN THIS VERSION

HARDWARE

If you have purchased the 1409 KIT, use this section to build the board. It is a good idea to read through the hardware section first to minimize false starts and make assembly of the unit as painless as possible.

Start assembly with the power supply. The 7805 voltage regulartor has to be mounted on a heat sink. On the transformer side of the board, two holes are provided for an "L" shaped alluminum bracket. The same bracket could be used to support the power switch.

Solder all the power supply components, then check the following:

-D24, cathode = 4.8-5.1V.	-D16, cathode = 21.5-22.5V	-8085, pin 40 = 4.8-5.2V.
-D22, cathode = 28-31V.	-D15, cathode = 18.5-19.5V.	-1488, pin 14 = 5.2-8V.
-D17, cathode = 25.5-27V.	-D14, cathode = 12.5-13.5V.	-1488, pin 1 = 5.2-8V.

Please note that in the lay-out schematic all the diodes have the test point marked with a bar.

If you choose to program 8741 and 8748 instead of 8748H and 8749H, you have to make the modifications shown on sheet 3 of the schematic. In this case you will not be able to program 2732A, 2764 and 27128. Select the menu "8748H" to program both 8741 and 8748.

If all the voltages are O.K., you can start mounting the other components. We recommend that you use sockets for at least the EPROM (2764 in the schematic, location 3C) the microprocessor (8085, location 2A) and the support chips (8251, 2C and 8155, 3A). These components are relatively hard to replace if something goes wrong and you have soldered them directly on the P.C. Board.

The firmware resides in a 2732 (1409-0X) or 2764 (1409-1X) which is supplied with the kit.

If you do not intend to program 8748H or 8749H, you will not need to mount the X2 quartz crystal, capacitors C3 and C4, transistors T5-T7, T14, diodes D2,D4, and resistors R11-R13,R16,R17,R21. You may add these components at a later date, to allow programming of the 8748H or 8749H.

For both the 28 pin and the 40 pin programming sockets, we recommend that you use ZIF's (zero insertion force sockets), which could be soldered directly on the P.C. Board or inserted in regular sockets. Make sure that the regular sockets have wide enough space for each pin, to accept the ZIF's.

After all the components are inserted, you can start testing the 1409. Using a null-modem RS232 cable (see appendix I), connect your assembled programmer to your terminal or computer (DTE). If you wish to connect it directly to data communication equipment, a normal RS232 cable is required.

Start sending some spaces, and the 1409 should answer with a "signature" followed by a prompt. If the unit does not respond, use the first part of the "DIAGNOSTICS" sections for debug.

After establishing communication with the 1409, use the commands in the "COMMANDS DESCRIPTION" section (page 5) to read or program E(E)PROMS.

If the first attempt to read or program an E(E)PROM fails, use the second part of the "DIAGNOSTICS" section to debug the I/O drivers.

INTERFACING THE 1409

The 1409 programmer interfacing is supported by the 8251 UART chip. The standard RS-232 port supports three control signals: CTS, DTR and RTS. However, none of these are required to establish communication with the 1409. Simply sending and receiving data will be enough to support almost all of the 1409 commands. The only exception are the "p" commands (program), which requires some handshaking.

If you do not intend to use a computer link to send data to the 1409, you can ignore the handshaking recommendations. Otherwise, you have to make sure you do not overflow the internal FIFO which collects the data. In this case some bytes will be lost and an error message will be printed.

When typing data from your keyboard, there is very little danger to overflow the FIFO. No security measures are provided in this case. The only potential spot for trouble arises when using the "mp" command. If you type more than two characters every 50 ms, some will be lost. This adds up to a typing speed of about 400 characters per second, not too easy to obtain.

The internal FIFO of the 1409 can store up to 132 characters (64 data bytes) before requesting the first "break".

The two possible methods of handshaking are described here.

1. Hardware handshaking: From an RS-232 "point of view", the 1409 programmer is data terminal equipment. This means that the DTR (data terminal ready) and RTS (request to send) lines are outputs, and the CTS (clear to send) line is an input with respect to the 1409.

In order to permit the 1409 to start sending characters, the CTS line should be pulled high. If you do not monitor this line, use a pull-up resistor. If you pull the CTS line low at any time, the 1409 will end the transmission of the character presently being transmitted, and will stop the transmission until the CTS line is pulled high again.

The DTR and RTS lines are both outputs which indicate that an overflow condition is ready to occur. If you use hardware handshaking when sending data from a file to the 1409 and one of these lines goes low, you should stop sending data to the 1409 immediately. Probably, the easiest way to do it is to check the status of these lines (or at least one of them) before sending every character.

The 1409 communicates using standard ASCII upper orlower case characters. The UART sis up for 8 bit words, 2 stop bits, no parity.

2. Software handshaking: The 1409 programmer supports <xon> and <xoff> handshaking also. When receiving data from the 1409 (as in a "list" command), you can stop the flow of data on your screen by sending an <xoff>. The stream will resume only after you send an xon. This might be handy if you want to "freeze" the screen for a while. This can be accomplished by typing a "control S" <xoff> to stop and a "control Q" <xon> to restart the transmission. The same result will be obtained if you have a "no scroll" key.

Likewise, while in "p" command (automatic file dump) the 1409 expects your computer to stop sending data when an <xoff> is received. This occurs every 132 ASCII characters (64 data bytes) or at the end of each line, whichever is first.

Resume sending data only after receiving an <xon>.

If the above sequence is not followed, an overflow will occur and the proper message will be printed.

START—UP

After power up, the 1409 is ready to do a baud rate search. Start typing several "spaces" (ascii 20 in hex), until the 1409 finds your baud rate. If your keyboard has a "repeat" key (or repeat feature), use it. If not, hit the space bar successively. The programmer will latch on any baud rate between 300 baud and 9600 baud.

If you decide to change the baud rate in the middle of something, push the reset button to start a new baud rate search.

As soon as the baud rate is found, the 1409 will display the following message:

```
B&C Microsystems
UNIVERSAL E(E)PROM PROGRAMMER
model 1409 ver 3.1
copyright 1983

programming mode:      slow
current baud rate:     2400
current high address:  0fff
27(C)32->
```

From now on, any prompt appearing on the screen shows that the 1409 is in the command mode, i.e. any valid command will be accepted and executed.

COMMANDS DESCRIPTION

* GENERAL *

The E(E)PROM PROGRAMMER accepts any string of up to 14 characters as a command. If the string is a valid command as described in the following text, it is executed as soon as "carriage return" is typed.

If the command is not valid, the message "invalid command" is displayed, and the 1409 returns to the command mode.

When typing a command, any characters from a standard keyboard will be accepted. The successive characters are stored in an internal buffer. To prevent a buffer overflow, the typed in string is limited to 14 characters. If you type more than 14 characters before hitting "carriage return" (cr), the command will be simply ignored, and the command mode reentered.

After starting to type a command, you can back-up as many times as you want, using the "backspace" key. If you enter more backspaces than characters, the extra backspace will be ignored until you type in valid characters.

All the valid commands use standard upper or lower case ASCII format.

When an address has to be specified from your terminal, it can consist of one, two, three or four hex digits.

Any command can be interrupted at any time by typing "$" (dollar sign). This is also the normal terminator for "p" and "mp" commands ("$" must be used in order to end the execution).

* LIST COMMAND *

Mnemonic: l [address1] [address2]

This command will list the content of the currently selected E(E)PROM, from address1 to address2. Both addresses are optional. If neither of the addresses are specified, the whole content of the E(E)PROM is listed.

The letter "l" must be separated from the address1 (if present) by a "space". A space is also required between address1 and address2.

If only address1 is present, the list command has two interpretations: 1) If the first address is followed by a "space", the 1409 starts listing the E(E)PROM from address1 up. 2) If the command ends with address1, the 1409 lists the content of the E(E)PROM from zero to the typed address.

The output is formatted in 8 groups of 4 bytes, preceded by the address of the first byte in group. The address displayed is always an even multiple of 10 (in hex). If the start address is NOT an even multiple of 10, the first line will contain only those bytes found until the first address multiple of 10 is encountered.

If address2 is not specified, or if it is larger than the maximum physical address of the currently selected device, the list command will stop when the last existing byte has been displayed.

This is an example of the output of a list command:

```
27(C)32—>l 300 400
0300 cdc2 082a 1f40 2224 5021 ffff 221f 503e
0400 0d
```

* READ COMMAND *

Mnemonic: r [address1] [address2]

This command is similar to LIST, with the exception that the output is a contiguous string of characters. The main purpose of this command is to transmit the content of an EPROM "as is", for the purpose of being stored in a file for future use (i.e., copying one eprom to another, or verifying a programmed EPROM against the original data).

Example:

```
27(C)32—>r 300 400
cdc2082a1f4022245021ffff221f503e0d
```

* MENU COMMAND *

Mnemonic: m[a-4]

If only "m" is specified, the 1409 outputs the menu consisting of all the available E(E)PROMS and waits for the desired selection.

The letters "a" through "y" and numbers "1" through "4" which follow the letter "m" are optional. In case a valid character is specified, the proper E(E)PROM is selected and the programmer returns to the command mode with the new prompt. As soon as the selection is entered, the 1409 updates the parameters describing the current E(E)PROM, and returns to the command state.

This is the present menu of the 1409-1X:

```
27(C)32->m
SELECTION MENU:

a=2758      k=2508   o=68732   r=2816A*    1=SPARE
b=27(C)16   l=2516   p=68764   s=2832A*    2=SPARE
c=27(C)32   m=2532   q-68764   t=2864A*    3=8748H
d=2732A     n=2564             u=52B13     4=8749H
e=2764                         v=52B23
f=2764A*                       w=52B33
g=27128                        x=52B43*
h=27128A*                      y=SPARE
i=27256
j=SPARE
*:not supported in this version
```

* VERIFY BLANK COMMAND *

Mnemonic: vb

This command checks the content of the selected E(E)PROM for erased condition.

If all the cells are found to be in the high state (low state for 874x chips) the 1409 returns with the prompt. Otherwise, the addresses of the cells found to be different are displayed, preceded by the hex value of the data.

This is an example of an output for this command:

```
27(C)32->vb
c3 @ 0000
40 @ 0001
01 @ 0002, etc.
```

* MANUAL PROGRAM COMMAND *

Mnemonic: mp [address]

This command programs the content of the E(E)PROM, starting at the address specified. If no address is specified, programming starts at address zero.

The purpose of this command is to help you program an E(E)PROM without the use of an assembler. After typing "cr", the 1409 outputs the start address, and waits for data. As many as 64 characters can by typed before hitting "cr". You can change your mind about anything on the current line, backup and correct it. The data is not entered until you type "cr". If you completely dislike a line of data, type "$" to get rid of it without programming any location.

When you decide to exit the manual programming mode type "$". The "mp" command ignores the spaces, so you can format your data as you get it from your "manual" assembler. After typing "cr", the data is entered to program the successive locations, the next address to be programmed is displayed, and the process starts from the beginning. This is an example of the display when using this command:

```
27(C)32->mp 269
0269 cd c208
026c 2a 1d50, etc
```

* PROGRAM COMMAND *

Mnemonic: p [address]

This command programs the content of the E(E)PROM, starting at the address specified. If no address is specified, programming starts at address zero.

No echo is provided.

After typing "cr", the 1409 waits for the string of data to be programmed. The data is received from a computer link.

For a complete description of handshaking during "p" command refer to "INTERFACING" (page 3).

Once started, programming proceeds uninterrupted until a "$" is encountered in the string of data, which causes the 1409 to return to the command mode.

If an odd number of nibbles is sent, the last nibble is ignored.

* HELP COMMAND *

Mnemonic: h

This command lists a summary of the system commands. It might be useful in case you did not use the device for some time, forgot which letter stands for what, and did not want to search for this manual in your files.

This is the actual output of this command:

```
27(C)32->h

1409 COMMAND SYSTEM:

GENERAL COMMANDS    PROGRAM COMMANDS         LIST COMMANDS

h  = help           p  = program             l  = list
m  = menu           mp = manual program      il = INTEL format list
?  = status         fp = fast program        mol= MOTOROLA format list
e  = erase          sp = slow program        r  = read (unformated)
x  = identify       1p = low byte program
!  = monitor        2p = high byte program
ve = verify blank   vp = verify programming
```

* IDENTIFY COMMAND *

Mnemonic: x

This command causes the E(E)PROM programmer to identify itself, printing a "signature" including the version number and the original release date. Here is an example:

```
27(C)32->x
B&C Microsystems
UNIVERSAL E(E)PROM PROGRAMMER
model 1409 ver 3.1
copyright 1983
```

* BREAK COMMAND *

Mnemonic: $

This command interrupts any other command in progress and returns the 1409 to the command mode.

* INTEL/MOTOROLA FORMAT PROGRAMMING *

The INTEL/MOTOROLA source file formats are automaticly recognized and processed. See "PROGRAM COMMAND" for details.

* INTEL/MOTOROLA LIST COMMANDS *

Mnemonic: il/mol [address1] [address2]

These commands will list the currently selected E(E)PROM from address1 to address2 in INTEL/MOTOROLA format. If no address is present the total content of the device is listed.

example:

```
27(C)32->il 00 1f
:10000000fffffffffffffffffffffffffffffff00
:10001000ffffffffffffffffffffffffffffffff0
:0000000000

27(C)32->mol 00 1f
S1130000ffffffffffffffffffffffffffffffffC
S1130010fffffffffffffffffffffffffffffffec
S9
```

See APPENDIX II for description of the INTEL and MOTOROLA formats.

* FAST PROGRAM COMMAND *

Mnemonic: fp

This command enables the fast programming mode. In this mode, all the programming commands (p, mp, 1p, 2p) are executed aproximately 4 times faster (full 27128 in under 5 minutes). An attempt to program a device which does not support this mode will automatically reset the 1409 to the normal (slow) programming mode.

* SLOW PROGRAM COMMAND *

Mnemonic: sp

This command resets the 1409 to the normal (slow) programming mode. An attempt to program in normal mode a device which supports the fast mode only (i.e. 27256) will automatically set the fast mode. The 1409 defaults to the normal (slow) mode on power up or reset.

* LOW BYTE/HIGH BYTE PROGRAMMING *

Mnemonic: 1p/2p

These commands are used to split a 16 bit data path. The "1p" command programs all the odd bytes, ignoring the eve ones. The "2p" command does the opposite.

* VERIFY PROGRAM COMMAND *

Mnemonic: vp

This command compares the content of the E(E)PROM with a data file starting at the address specified. If no address is specified, verification starts at address zero.

No echo is provided.

After typing "cr", the 1409 waits for the string of data to be verified. The data is received from a computer link.

When all the data is verified the 1409 returns a prompt. If not the 1409 returns the error message:

"* can't verify @ [address]

* ERASE COMMAND *

Mnemonic: e [address1] [address2]

This command erases the contents of the selected EEPROM, from address1 to address2. If no address is stated the complete EEPROM is erased.

* STATUS COMMAND *

Mnemonic: ?

This command displays the ststus of the 1409.

Example:

```
27(C)32->?

programming mode:     slow
current baud rate:    2400
current high address  0fff
27(C)32->
```

* MONITOR COMMAND *

Mnemonic: !

This command enables the monitor. To return to the command mode type "p".

THE MONITOR

The 1409 includes a monitor which allows you to check every I/O driver or port. To run the monitor program, type "!". To return to the command mode of the 1409, type "p".

All the monitor commands start with a lower-case letter followed by a four-digit hex address. Unlike the 1409 executive program, the monitor always expects a four digit address. (ex. hex address 10 must by typed 0010).

The READ and WRITE commands must be interrupted with a "dollar" sign. ($).

The monitor is able to perform the following functions:

-READ
Mnemonic: r(address)(any key)$.
After typing the address, for every key stroke the monitor will print the data at the corresponding address.

-WRITE

Mnemonic: w(address) (data)$.

The hex data will be written starting at the selected address. Data can only be written to a RAM location or to an I/O port.

-GO

Mnemonic: g(address).

The microprocessor jumps at the indicated address.

If you want to use the monitor, here are some useful hardware addresses:

A000	-start of ram	A102	-8155 port b
A0ff	-end of ram	A103	-8155 port c
A100	-8155 control address	C000	-i/o port x
A101	-8155 port a	8000	-i/o port y

All the above addresses are in hex.

ERROR MESSAGES

The 1409 prints several diagnostic messages to help the user who writes his own communications software. All error messages are preceded by a "*".

- "* invalid command" -printed when a command not included in the command summary is typed. Also printed if an E(E)PROM not specified in the menu is selected.

- "* can't program @ [address]" -printed when the 1409 attempts to program a specific address, and fails. This may be caused by a nonerased E(E)PROM, a bad E(E)PROM or a wrong menu selection.

- "* can't program fast @ [address]" -printed when the 1409 attempts to program a specific address, in fast mode and fails. This may be caused by a nonerased E(E)PROM, a bad E(E)PROM or a wrong menu selection.

NOTE -a programming attempt with a wrong menu selection will most likely damage the E(E)PROM, due to the fact that programming voltage (21 or 25 V) is applied to a pin which probably accepts only TTL input.

- "* can't verify @ [address]" -printed when the 1409 attempts to verify a specific address, and fails.

- "* use u.v. to erase!" -printed when the "e" command is entered on an EPROM prompt.

- "* error @ [address]" -printed when an invalid hex data is sent to the specified location during programming.

- "* overrun error" -printed when, during a "p" command, the 1409 requests a stop in the string of data being sent by your computer, and the request is ignored.

After any of these messages, the 1409 returns to the command mode.

DIAGNOSTICS

```
                    ____________________________
                   !          START:            !
                   ! -all voltages OK           !
                   ! -all components mounted    !
                   ! -1409 does not "respond"   !
                   ------------------------------
                                  !
        _______                   !
       DO A1>--------------------->!
        ^                          !
        !                     _____!_______
        !<---------NO-<  TEST 1 OK ?
                                   !
                                  YES
        _______                    !
       DO A2>--------------------->!
        ^                          !
        !                     _____!_______
        !<---------NO-<  TEST 2 OK ?
                                   !
                                  YES
        ________                   !
       DO A10>-------------------->!
        ^                          !
        !                     _____!_______
        !<---------NO-<  TEST 10 OK ?
                                   !
                                  YES
                                   !
                         ____________________
                          1409 FUNCTIONAL
```

TEST#	TEST TYPE	TEST POINT	VALUE	ACTION	ACTION TYPE
T1	measure	8085 pin 1	4 MHZ sine wave	A1	-check C1, X1 -check 8085
T2	measure	8085 pin 36	-low if SW1 ON -high if SW1 OFF	A2	-check D1, R5 -check C2, SW1
T3	measure	8085 pin 3	-low if SW1 OFF -high if SW1 ON	A3	check 8085
T4	measure	8085 pin 37	2 MHZ sine wave	A4	check 8085
T5	measure	8085 pin 30	aprox. 750 KHZ TTL pulses	A5	check 8085
T6	measure	8085 pin 6	aprox. 150 KHZ square wave	A6	-check 1A,3C,4C -replace 8155
T7	-conn. RS232 -type spaces	1489 pin 1	low freq. square wave pulses	A7	check RS232 link
T8	-type spaces	8251 pin 3	low freq. square wave pulses	A8	check 1489
T9	-type spaces	8251 pin 19	low freq. square wave pulses	A9	check 8251
T10	-type spaces	1488 pin 3	train of random pulses	A10	-check RS232 -check 1488

-IF THE 1409 RESPONDS WHEN YOU TYPE SPACES BUT DOES NOT READ OR PROGRAM EPROMS OR EEPROMS, TYPE IN CONSECUTIVE ORDER THE COMMANDS LISTED IN THE TABLE ON THE FOLLOWING PAGE AND CHECK FOR THE APPRIOATE VOLTAGE LEVELS.
-IF A TEST DOES NOT LEAD TO THE INDICATED RESULT, CHECK THE DIODES, TRANSISTORS, RESISTORS, AND BUFFERS (7407) ASSOCIATED WITH THE PIN.
-TYPE ! TO GET INTO THE MONITOR.

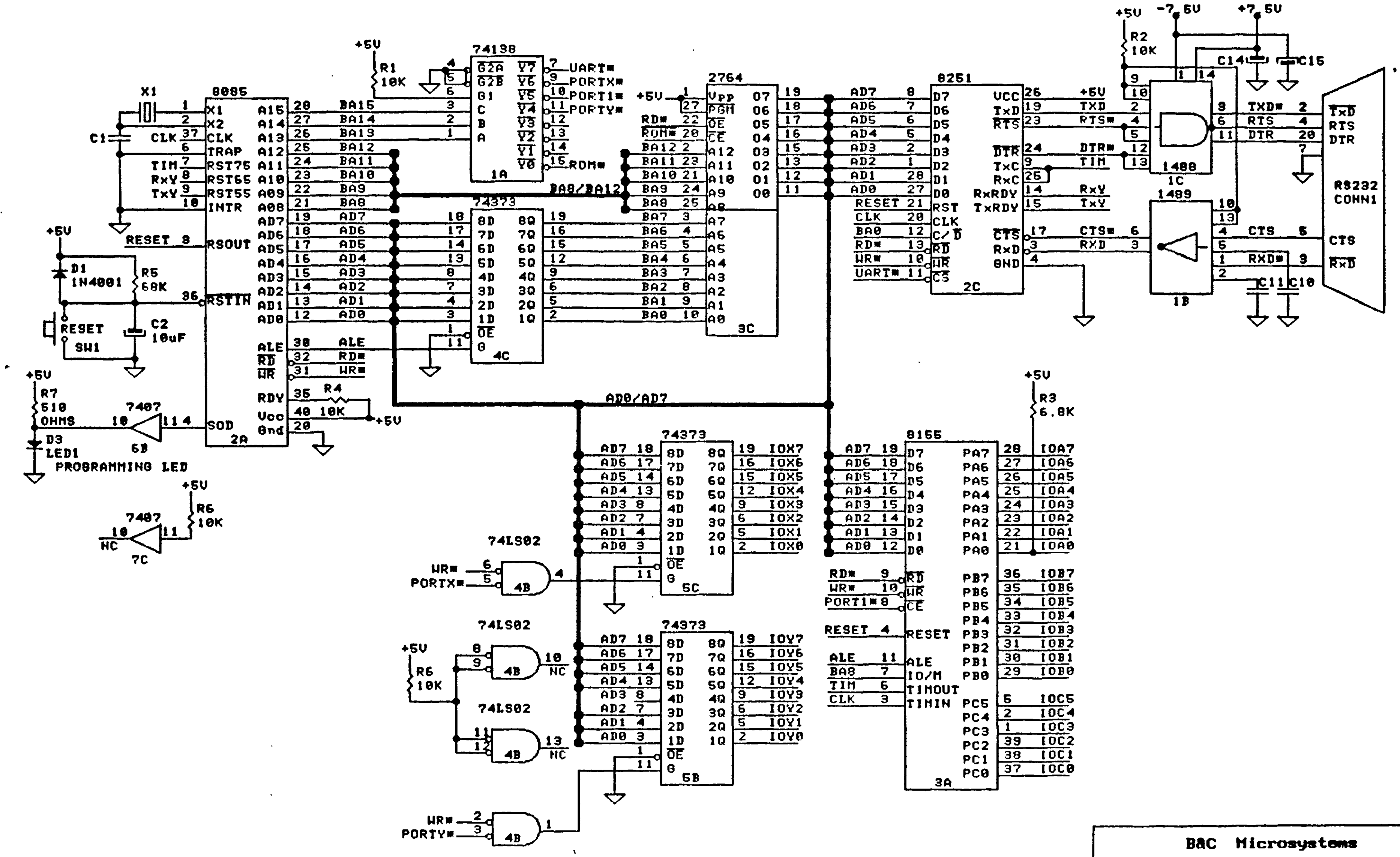
	VOLTAGE ON 4A PIN:			VOLTAGE ON 6A PIN:					
COMMAND	1	22	23	1	4	7	25	26	40
wc000 00	low	low	low	–	–	–	–	–	–
01	+5v	low	low	–	–	–	–	–	–
02	+21v	low	low	–	–	–	–	–	–
03	+25v	low	low	–	–	–	–	–	–
08	low	+5v	low	–	–	–	–	–	–
10	low	+21v	low	–	–	–	–	–	–
20	low	+25v	low	–	–	–	–	–	–
40	low	low	+5V	–	–	–	–	–	–
80$	low	low	+25V	–	–	–	–	–	–
	–	–	–	–	–	–	–	–	–
wal00 0f$	–	–	–	–	–	–	–	–	–
wal03 00$	–	–	–	low	low	+5v	low	+5v	low
wal03 01$	–	–	–	+5v	low	+5v	low	+5v	low
wal03 02$	–	–	–	low	+5v	+5v	low	+5v	low
wal03 04$	–	–	–	low	low	+18v	low	+5v	low
wal03 08$	–	–	–	low	low	+5v	+18v	+5v	low
wal03 10$	–	–	–	low	low	+5v	low	+21v	low
wal03 20$	–	–	–	low	low	+5v	low	+5v	+5v

	VOLTAGE ON 4A PIN:							
COMMAND	2	20	21	24	25	26*	27	28
wa8000 00	low	low	low	low	low	low	low	low
01	high	low	low	low	low	low	low	low
02	low	high	low	low	low	low	low	low
04	low	low	high	low	low	low	low	low
08	low	low	low	high	low	low	low	low
10	low	low	low	low	high	low	low	low
20	low	low	low	low	low	+5v	low	low
40	low	low	low	low	low	low	high	low
80$	low	low	low	low	low	low	low	+5v

	VOLTAGE ON 4A PIN:							
COMMAND	10	9	8	7	6	5	4	3
wal00 0f$	–	–	–	–	–	–	–	–
wal02 00$	low	low	low	low	low	low	low	low
wa002 01$	high	low	low	low	low	low	low	low
wa002 02$	low	high	low	low	low	low	low	low
wa002 04$	low	low	high	low	low	low	low	low
wa002 08$	low	low	low	high	low	low	low	low
wa002 10$	low	low	low	low	high	low	low	low
wa002 20$	low	low	low	low	low	high	low	low
wa002 40$	low	low	low	low	low	low	high	low
wa002 80$	low	low	low	low	low	low	low	high

	VOLTAGE ON 4A PIN:							
COMMAND	11	12	13	15	16	17	18	19
wal01 00$	low	low	low	low	low	low	low	low
wal01 01$	high	low	low	low	low	low	low	low
wal01 02$	low	high	low	low	low	low	low	low
wal01 04$	low	low	high	low	low	low	low	low
wal01 08$	low	low	low	high	low	low	low	low
wal01 10$	low	low	low	low	high	low	low	low
wal01 20$	low	low	low	low	low	high	low	low
wal01 40$	low	low	low	low	low	low	high	low
wal01 80$	low	low	low	low	low	low	low	high

NOTE: low = TTL low level. high = TTL high level.
* solder a 10k ohm pull-down resistor to ground.



B&C Microsystems

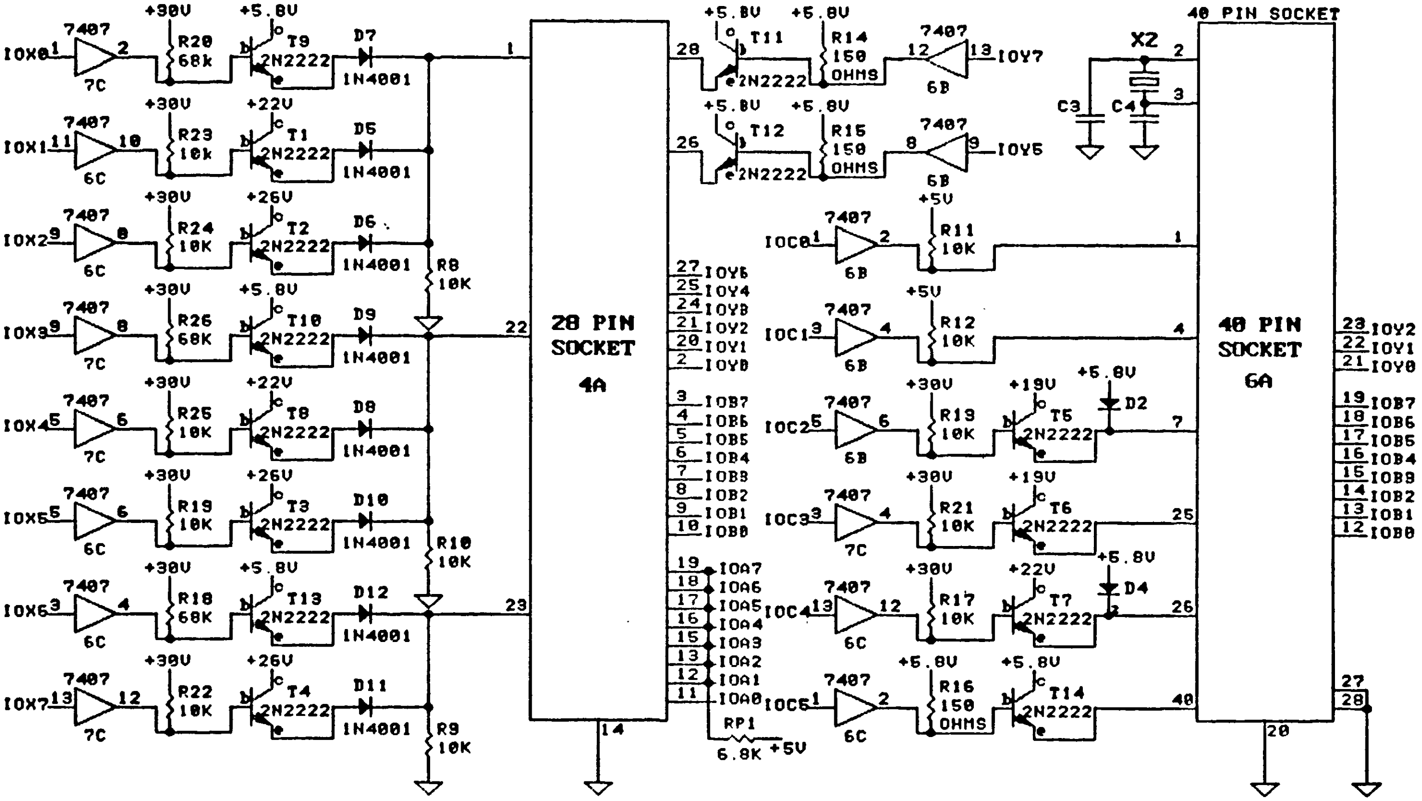
20917 Greenleaf Dr.Cupertino, CA 95014

TITLE

PROCESSOR

SIZE	CODE	MODEL	VER
A	IU	1489 PROGRAMMER	3.0

DATE FEB 1, 1984 SHEET 1 OF 5



B&C Microsystems

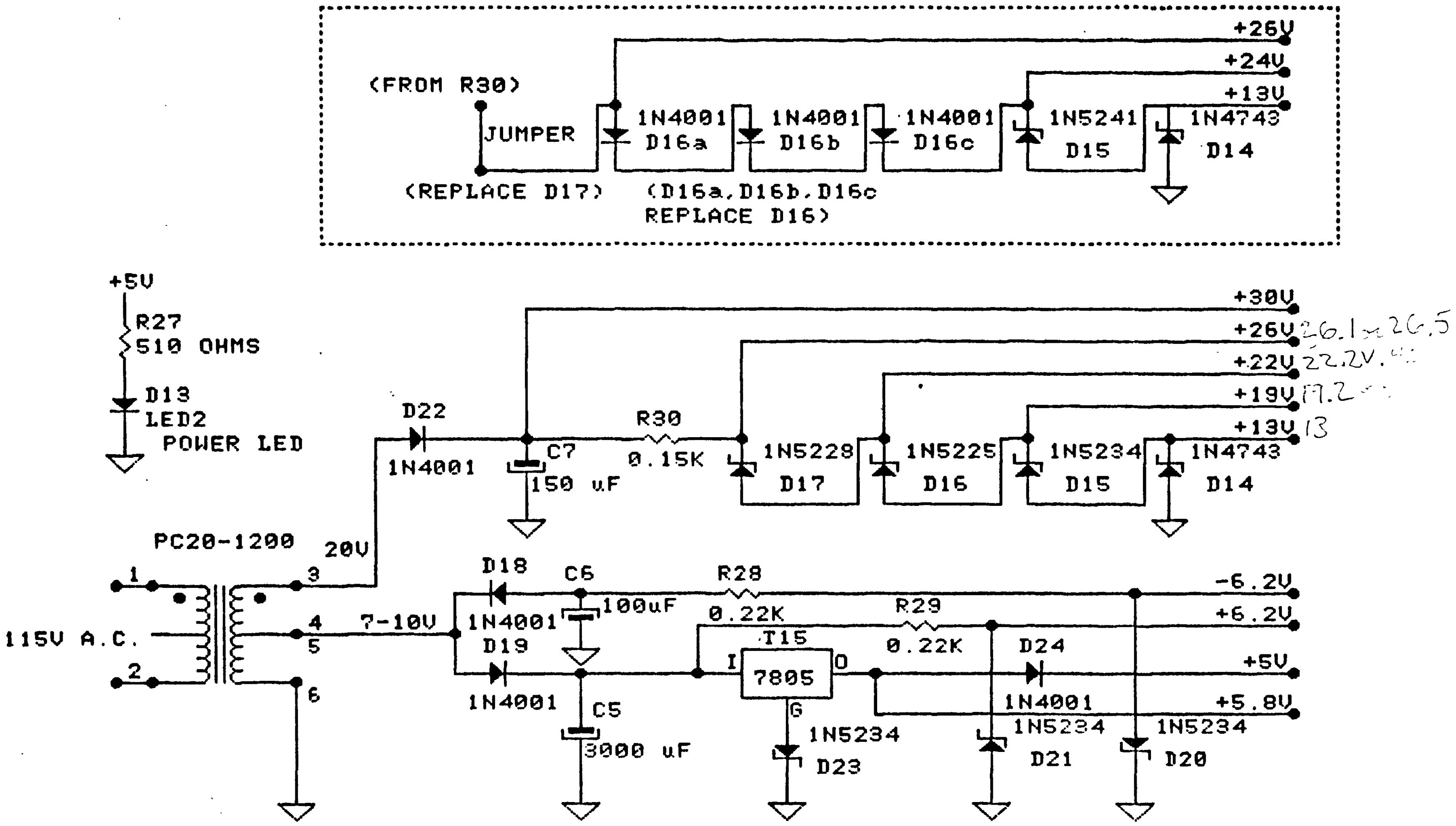
20917 Greenleaf Dr.Cupertino, CA 95014

TITLE

OUTPUT DRIVERS

SIZE	CODE	MODEL	VER
A	IV	1409 PROGRAMMER	3.0

DATE FEB 1, 1984 | SHEET 2 OF 5



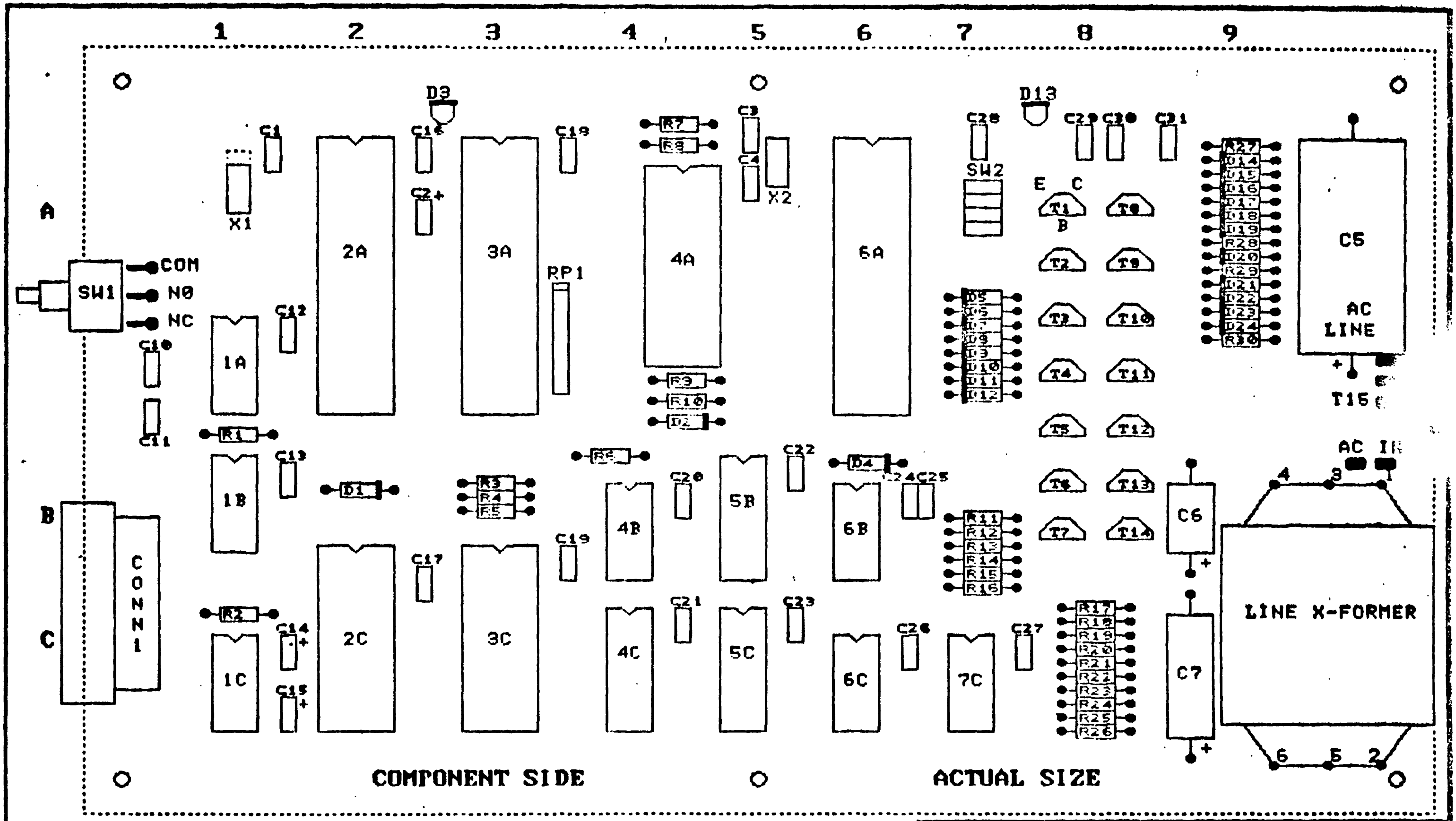
NOTE: REPLACE D15, D16, D17 AS SHOWN IN THE DOTTED AREA TO PROGRAM 8741 AND 8748 (INSTEAD OF 8748H, 8749H)

B&C Microsystems
20917 Greenleaf Dr.Cupertino, CA 95014

TITLE			
POWER SUPPLY			
SIZE A	CODE IV	MODEL 1409 PROGRAMMER	VER 3.0
DATE FEB 1, 1984		SHEET 3 OF 5	

1409 PARTS LIST

TYPE	DESCRIPTION	LOCATION	EACH
8085	MICROPROSSER	2A	1
8155	RAM-I/O-TIMER	3A	1
8251	UART	2C	1
2732	EPROM (1409-0X)	3C	1
2764	EPROM (1409-1X)	3C	1
74LS02	QUAD 2 INPUT NOR	4B	1
7407	HEX OPEN COLL DRIVER	6B, 6C, 7C	3
74LS138	3 TO 8 MULTIPLEXER	1A	1
74LS373	OCTAL LATCH	4C, 5B, 5C	3
1488	RS232 DRIVER	1C	1
1489	RS232 RECEIVER	1B	1
-	-	-	-
-	LED DIODES	D3, D13	2
1N5243A	0.5W/13V ZENER	D14	1
1N5234B	0.5W/6.2V ZENER	D15, D20, D21, D23	4
1N5228B	0.5W/3.9V ZENER	D17	1
1N5226B	0.5W/3.3V ZENER	D16	1
1N4148	100MA/50V DIODE	D1, D2, D4 THRU D12, D18	12
1N4001-5	1A/50V RECTIFIER	D19, D22, D24	3
2N2222	TRANSISTOR	T1 THRU T14	14
7805(K)	5V REGULATOR	T15	1
4 MHZ	QUARTZ CRYSTAL	X1, X2	2
-	-	-	-
150 OHMS	RESISTOR	R14 THRU R16, R30	4
220 OHMS	RESISTOR	R28, R29	2
510 OHMS	RESISTOR	R7, R27	2
10K OHMS	RESISTOR	R1 THRU R4, R6, R8 THRU R13	11
10K OHMS	RESISTOR	R17, R19, R21 THRU R25	7
68K OHMS	RESISTOR	R5, R18, R20, R26	4
4.7-10K	8 PIN SIP RESISTOR	RP1	1
-	-	-	-
3000uF/16V	CAPACITOR	C5	1
150uF/35V	CAPACITOR	C7	1
100uF/16V	CAPACITOR	C6	1
10uF/15V	CAPACITOR	C2, C14, C15	3
0.01uF/50	CAPACITOR	C12, C13, C16 THRU C31	18
20pF/50V	CAPACITOR	C1, C4, C10, C11	4
5pF/50V	CAPACITOR	C3	1
-	-	-	-
DB25	RS232 PC MOUNT MALE CONN.	CONN 1	1
28 PIN ZIF	SOCKET	4A	1
40 PIN ZIF	SOCKET	6A	1
TP12H	RESET SWITCH	SW1	1
4PST	4 POS. DIP SWITCH	SW2 (1409-1X)	1
PC20-1200	110VAC LINE TRANSFORMER	TRX 1	1
1409PROG	PRINTED CIRCUIT BOARD	-	1
1409DOC	26 PAGE INSTRUCTION SET	-	1



RESISTOR	R1	R2	R3-R5	R6	R7-10	R11-16	R17-26	R27-30	
POSITION	1B	1C	3B	4B	4A	7B	8C	9A	
CAPACITOR	C1	C2,16	C3,C4	C5	C6,C7	C10-12	C13	C14,15	
POSITION	1A	2A	5A	9A	9BC	1A	1B	1C	
CAPACITOR	C17	C18	C19	C20	C21	C22	C23	C24,25	C26,27
POSITION	2B	3A	3B	4B	4C	5B	5C	6B	7C
DIODE	D1	D2	D3	D4	D5-12	D13	D14-22		
POSITION	2B	4B	3A	6B	7A	8A	9A		

B&C Microsystems

20917 Greenleaf Dr.Cupertino, CA 95014

TITLE

BOARD LAY-OUT

SIZE	CODE	MODEL	VER
A	IV	1409 PROGRAMMER	3.0

DATE FEB 1984 SHEET 5 OF 5

APENDIX I

NULL-MODEM CABLE SPECIFICATIONS

THE 1409 IS CONFIGURED AS DATA TERMINAL EQUIPTMENT. IN ORDER TO COMMUNICATE WITH A TERMINAL, A NULL-MODEM CABLE IS REQUIRED.

THESE ARE THE CONNECTIONS REQUIRED FOR THE NULL-MODEM CABLE:

```
CONNECTOR  DB-25S

(BA)  TXD (pin 2) >----------►------------< (BB) RXD (pin 3)
(BB)  RXD (pin 3) >----------◄------------< (BA) TXD (pin 2)
(CA)  RTS (pin 4) >----------►------------< (CB) CTS (pin 5)
(CB)  CTS (pin 5) >----------◄------------< (CA) RTS (pin 4)
(CC)  DSR (pin 6) >-----------------------< (CD) DTR (pin 20)
(CD)  DTR (pin 20)>----------►------------< (CC) DSR (pin 6)
(AB)  GND (pin 7) >-----------------------< (AB) GND (pin 7)
```

APPENDIX II

INTEL HEX FORMAT

DATA RECORD:

BYTE 1	COLON (:).
2..3	NUMBER OF DATA BYTES IN THIS RECORD.
4..5	LOAD ADDRESS FOR THIS RECORD, HIGH BYTE.
6..7	LOAD ADDRESS FOR THIS RECORD, LOW BYTE.
8..9	RECORD TYPE, MUST BE "00".
10..X	DATA BYTES, TWO ASCII-HEX CHARACTERS EACH.
X+1..X+2	CHECKSUM, TWO ASCII-HEX CHARACTERS.
X+3..X+4	CR,LF.

END RECORD:

BYTE 1	COLON (:).
2..3	RECORD LENGTH, MUST BE "00".
4..7	START ADDRESS, "0000" IN MCS-86 END RECORD.
8..9	RECORD TYPE, "01" OR "00" IS ACCEPTABLE.
10..11	CHECK SUM.
12..13	CR,LF.

THE CHECKSUM IS THE TWO'S COMPLIMENT OF THE 8-BIT SUM, WITHOUT CARRY, OF ALL THE DATA BYTES, THE TWO BYTES IN THE LOAD ADDRESS, AND THE BYTE COUNT.

MOTOROLA HEX FORMAT

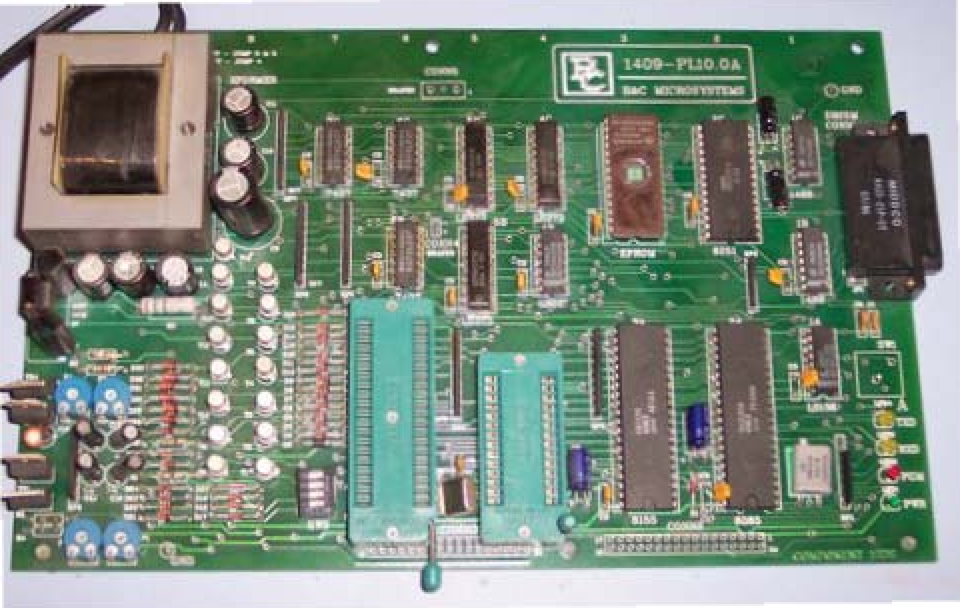
DATA RECORD:

BYTE 1..2	"S1".
3..4	NUMBER OF DATA BYTES IN THIS RECORD.
5..6	LOAD ADDRESS, HIGH BYTES.
7..8	LOAD ADDRESS, LOW BYTES.
9..X	DATA BYTES, TWO CHARACTERS EACH.
X+1..X+2	CHECKSUM.
X+3..X+4	CR,LF.

THE BYTE COUNT IS 3 GREATER THAN THE NUMBER OF DATA BYTES. THE CHECKSUM IS THE ONE'S COMPLIMENT OF THE 8-BIT SUM, WITHOUT CARRY, OF THE BYTE COUNT, THE TWO BYTES OF THE LOAD ADDRESS, AND THE DATA BYTES.

END RECORD:

BYTE 1..2	"S9".
3..4	CR,LF.


1409-PL10.0A
B&C MICROSYSTEMS
EPROM
8155
8085