B C C 1 8 0

Computer/Controller Technical Manual

Release 1.01
4/11/88

Micromint, Inc.
902 Waterway Place
Longwood, FL 32750
www.micromint.com

* *

Conditions of Sale and Product Warranty

Micromint Inc. and the Buyer agree to the following terms and conditions of Sale and Purchase:

1. Micromint Inc. extends the following warranty: a factory-manufactured circuit board or assembly carries with it a one-year warranty covering both parts and labor. Any unit which is found to have a defect in materials or workmanship will, at the discretion of Micromint Inc., be repaired or replaced.

2. A minimum inspection fee must be prepaid for the repair of units that are no longer under warranty. Call Micromint Inc. for a current list of fees.

3. Micromint Inc. will not be responsible for the repair or replacement of any unit damaged by user modification, negligence, abuse and mishandling, or improper installation.

4. Micromint Inc. is not responsible to the Buyer for any loss or claim of special or consequential damages.

5. All units returned for repair must first receive prior authorization from Micromint Inc. A return authorization number may be obtained by phone or letter. Please retain a record of the return authorization number since most subsequent correspondence will refer to this authorization. Under no circumstances should any product be returned to Micromint Inc. without such authorization. Micromint Inc. assumes no responsibility for returns unaccompanied by an authorization number. All returns must be shipped prepaid and ought to be insured. Micromint Inc. is not responsible for losses or damage during shipment. Repaired units will be returned with postage and insurance paid.

6. Micromint Inc. reserves the right to alter any feature or specification at any time. This right extends to fees, charges, and any other conditions or warranties contained herein.

* *

DISCLAIMER

While we have attempted to provide accurate and up-to-date information in this manual, Micromint Inc. makes no representations or warranties respecting its contents. We reserve the right to make periodic changes to the text and to issue new editions of this manual without notification.

Occasionally in this manual we refer to other manufacturers' products. Such references do not constitute an endorsement of these products, but are included for the purpose of illustration or clarification. We do not intend such technical information and interface data to supersede information provided by individual manufacturers.

Table of Contents

Introduction

The Micromint BCC180 is a single-board computer/controller featuring a new-generation 8-bit microprocessor which maintains software compatibility with the Zilog Z80 while incorporating advanced design features in a single 68-pin PLCC package. Configured primarily for process control, the BCC180 uses the same 44-pin BCC I/O expansion bus as Micromint's popular BCC52 controller board. All of Micromint's BCC-bus peripherals are compatible with the BCC180.

The BCC180 ROM monitor provides the system designer with a host of low-level development aids while the BASIC-180 multitasking BASIC compiler speeds high-level development of lightning-fast code right on board.

BCC180 Technical Specifications

* Hitachi HD64180 microprocessor running at 9.216 MHz Supports a superset of the Z80 instruction set
* Up to 384K bytes total memory on-board (128K EPROM or static RAM, 256K dynamic RAM)
* Two asynchronous serial ports (both support RS-232, one supports RS-422/485)
* One clocked (synchronous) serial port
* Six 8-bit parallel I/O ports (48 bits)
* 64K I/O space available through the BCC-bus edge connector
* On-board memory management unit, 2-channel DMA controller, 2 counter-timers, and 12 interrupt sources
* Requires just +5V (±12V necessary for RS-232 operation only)
* Accepts optional EPROM programmer daughter board for programming 27x256 EPROMs

Setting Up Your BCC180

Very few connections are required to bring up a functional BCC180 development system. Depending on your application, you may or may not want to connect a printer. Simply skip that step if it doesn't apply.

Power

The BCC180 requires the following voltages and current:

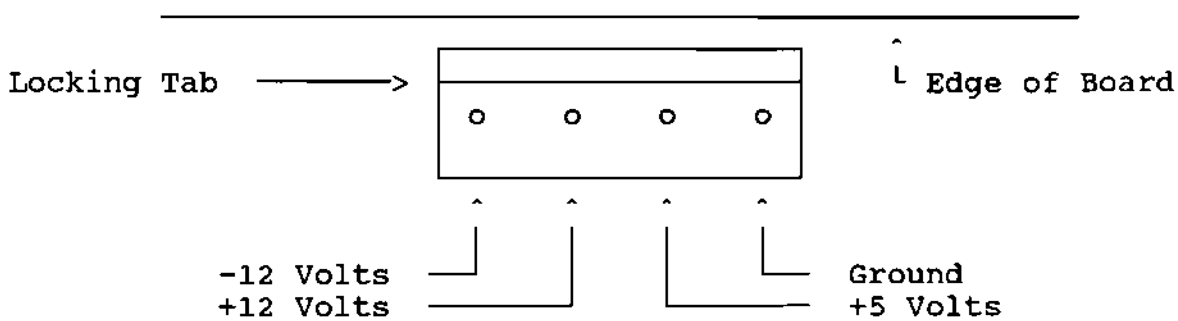
```
 +5V ±5% @ 750 mA (fully populated)
+12V ±20% @ 30 mA
-12V ±20% @ 30 mA
```

* -5V may be substituted for -12V if nonstandard RS-232 operation is acceptable and -12V isn't required for use by any other board on the bus.

* ±12V are used on the BCC180 for RS-232 only. If no serial communication is to be used or TTL serial is performed through the BCC bus, and ±12V aren't necessary for any other board on the bus, then the two supplies can be eliminated and the system can be run on +5V only.

Power can be connected to the BCC180 in one of two ways: through power connector J4 or through the BCC bus.

A 4-pin Molex-type plug that mates with J4 has been included with your BCC180. The following top view of J4 should be used to properly connect the plug to your power supply. **Be sure to double check your connections before applying power to the BCC180! Boards damaged by incorrect power connections are not covered under warranty.**



J4

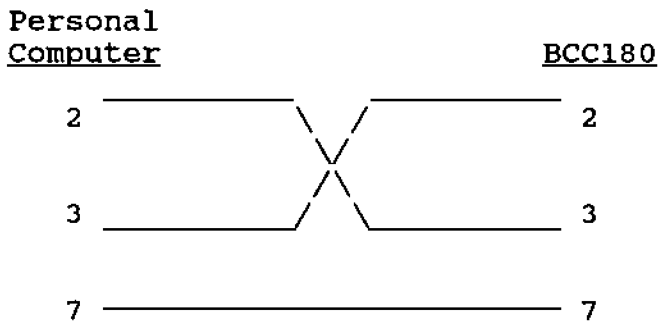
If you plan to plug your BCC180 into a backplane, power should be supplied to the board (and the rest of the boards on the bus) through the power pins on the bus. Motherboards available from Micromint use the same type of power connector as that shown above. The same procedure and cautions should be used to connect your power supply to the backplane.

When plugging the BCC180 and other BCC boards into the backplane, be sure the "A" and "Z" markings on the component side of the board match up with the "A" and "Z" markings on the motherboard. **Plugging a board into the bus backwards may damage the board. A board damaged by plugging it into the bus backwards is not covered under warranty, so exercise appropriate care.**

Terminal

Any standard RS-232 serial terminal can used as a console device for the BCC180. Alternatively, a personal computer using communication software and a serial port can be used as a console device.

The BCC180 has a standard DB-25S connector used to connect with the console device. If a terminal is being used, a straight-through DB25-to-DB25 cable can be used to connect the terminal to the BCC180. If a personal computer is to be used, a swap or "null modem" cable often must be used. The minimum requirements for this cable are that pins 2 and 3 should be swapped and pin 7 should pass straight through. The following diagram shows this minimum cable:



Please refer to the "Serial I/O" section later in this manual if your terminal requires additional handshaking lines.

The BCC180 monitor will automatically detect the baud rate being used by the terminal as long as it's one of the following: 19200, 9600, 2400, 1200, or 300. We recommend setting your terminal for 9600 bps until your system is set up and working. Additionally, be sure your terminal (or communication software) is set up with the following parameters: 8 data bits, 1 stop bit, no parity, and CR (not CR/LF) generated when "Return" is pressed. If your terminal doesn't allow parity to be turned off, 7 data bits and space (clear) parity should also work.

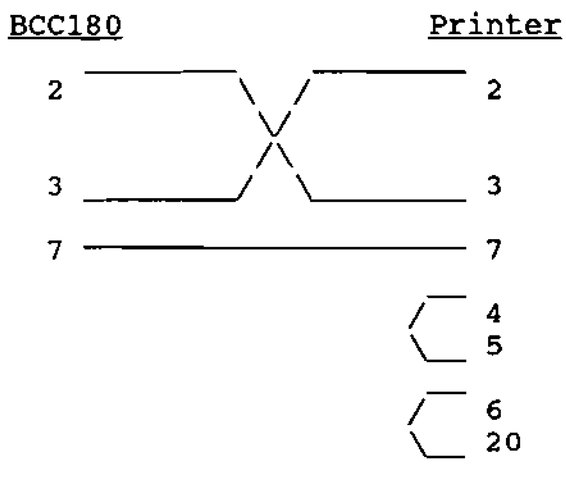
Printer

The BCC180 supports a serial printer at baud rates up to 19200 bps and will support software (XON/XOFF) handshaking. The default printer baud rate used by the BCC180 monitor is 9600 bps. See the BCC180 monitor manual for information on how to change this to another baud rate (both temporarily and permanently).

Serial printers are all different in the type of RS-232 connection they require. Some need just one data line plus ground, while others may need elaborate handshaking connections. We will attempt to show one common setup here. However, it is inevitable that this won't work with some printers. Consult the printer's manual and do some experimenting before giving up. A serial breakout box is invaluable in getting a stubborn serial connection to work. Alternatively, there are smart cables available from many computer stores and mail-order companies that look at both ends of the connection and automatically decide which pins should be connected.

Micromint has available a short cable (P/N SB180-AUX-MOD) that goes from the 20-pin Berg-type header (J3) on the BCC180 to a standard DB-25P connector. The connections shown below assume this DB-25P connector is present. If you don't have the proper adapter cable, please refer to the "Serial I/O" section later in this manual for the pinout of J3.

The following is a recommended cable configuration for connecting a serial printer to the BCC180:



Remember that hardware handshaking is not supported, so the printer's Busy output (normally pin 11) doesn't have to be connected to anything.

Powering it up

Once everything has been connected, double check all the connections. Turn on the power to the terminal and the printer (if connected). Once the terminal is warmed up, turn on the power to the BCC180. If the terminal is set up for 9600 bps and everything is connected properly, the string "BCC180" should be displayed on the terminal screen. Press "Return" once or twice and the monitor banner should come up on the screen. Your BCC180 system is now working. You should now refer to either the BCC180 monitor manual or (if BASIC-180 is installed) the BASIC-180 manual for instructions on how to proceed.

If your terminal is set for something other than 9600 bps, you should get a few garbage characters on the screen when you first apply power to the BCC180. This is what "BCC180" transmitted at 9600 bps looks like at whatever baud rate you're using. Pressing "Return" once or twice tells the BCC180 what baud rate you're really using and the monitor banner should come up as described above, but at the baud rate you're using.

If "BCC180" isn't displayed when power is applied, or there is no response to repeated presses of the "Return" key, press the BCC180's reset button (PB1) and try again. If there is still no response, check all connections once again including the power and terminal cables. If you still can't get your BCC180 to respond, you may call our technical support staff at (203) 871-6170.

BCC180 Hardware

The hardware on the BCC180 can be broken into three sections: the HD64180 microprocessor, memory, and I/O. Each of these sections will be discussed in turn.

HD64180 Processor

The power of the BCC180 is made possible by the Hitachi HD64180--a microcoded execution unit based on advanced CMOS manufacturing technology. It provides the benefits of high performance, reduced system cost, and low-power operation while maintaining compatibility with the large base of industry-standard 8-bit software.

Performance is derived from a high clock speed, instruction pipelining, and an integrated Memory Management Unit (MMU). The instruction set is a superset of the Z80 instruction set; twelve new instructions include hardware multiply, bit comparisons, and a SLEEP instruction for low-power mode.

Compared with the Z80 in the same way the 80188 is compared with the 8088, system costs are reduced because many key system functions have been included on-chip. Besides the MMU, the HD64180 boasts a two-channel Direct Memory Access Controller (DMAC), wait-state generator, dynamic-RAM refresh, two-channel Asynchronous Serial Communication Interface (ASCI), Clocked Serial I/O (CSIO), two-channel 16-bit Programmable Reload Timer (PRT), a versatile 12-source interrupt controller, and a "dual" (68xx and 80xx families) bus interface, all on one 68-pin chip.

The HD64180 comprises five functional blocks:

- o Central Processing Unit: The CPU is microcoded to implement an upward-compatible superset of the Z80 instruction set. Besides the twelve new instructions, many instructions require fewer clock cycles for execution than on a standard Z80.

- o Clock Generator: The clock generator produces the system clock from an external crystal or external clock input. The clock is programmably prescaled to generate timing for the on-chip I/O and system support devices.

- Bus State Controller: The bus state controller performs all status/control bus activity. This includes external bus cycle wait-state timing, RESET\, DRAM refresh, and master DMA bus exchange. It generates "dual" bus-control signals for compatibility with both 68xx and 80xx family devices.

- Interrupt Controller: The interrupt controller monitors and prioritizes the four external and eight internal interrupt sources. A variety of interrupt response modes are programmable.

- Memory Management Unit: The MMU maps the CPU's 64K-byte logical address space into a 1-Megabyte physical address space. The MMU organization preserves software object code compatibility while providing extended memory access and uses an efficient "common area/bank area" scheme. I/O accesses (64K-port I/O space) bypass the MMU.

The integrated I/O resources make up the remaining four functional blocks:

- Direct Memory Access Controller: The two-channel DMAC provides high-speed memory-to-memory, memory-to-I/O, and memory-to-memory-mapped I/O transfer. The DMAC features edge- or level-sense request input, address increment/decrement/no-change, and (for memory-to-memory transfer) programmable burst or cycle-steal transfer. In addition, the DMAC can directly access the full 1M-byte physical address space and transfers (up to 64K bytes in length) can cross 64K-byte boundaries.

- Asynchronous Serial Communication Interface: The ASCI provides two separate full-duplex UARTs and includes a programmable baud rate generator, modem control signals, and a multiprocessor communication format. The ASCI can use the DMAC for high-speed serial data transfer, reducing CPU overhead.

- Clocked Serial I/O Port: The CSIO provides a half-duplex clocked serial transmitter and receiver. This can be used for simple, high-speed connection to another microprocessor or microcomputer.

- Programmable Reload Timer: The PRT contains two separate channels, each consisting of 16-bit data and 16-bit timer reload registers. The timebase is divided by 20 (nonprogrammable) from the system clock and one PRT channel has an optional output allowing waveform generation.

Note: The HD64180 Microprocessor Data Book is available from Micromint for $10 plus shipping.

Memory

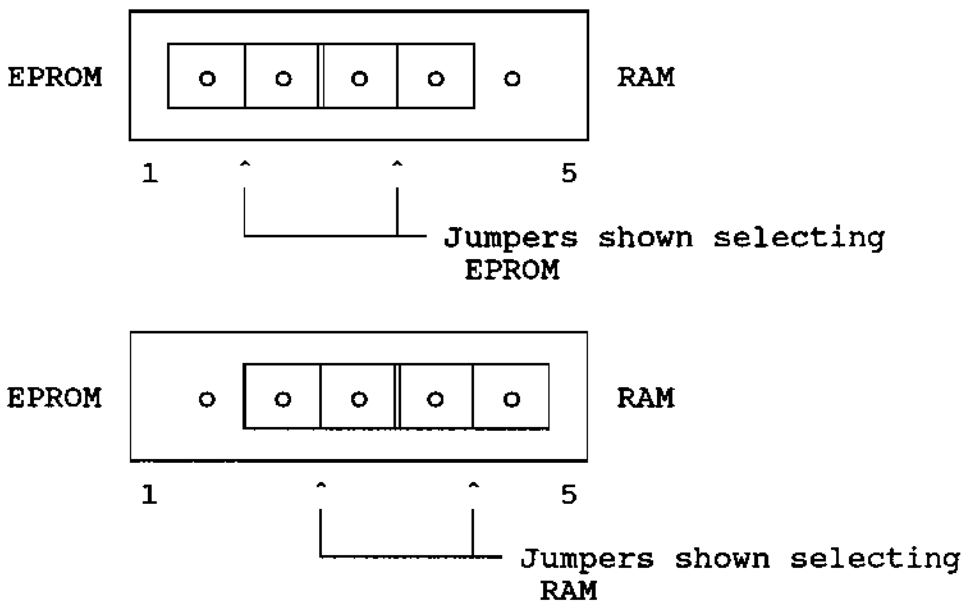
The BCC180 allows the use of up to 384K bytes of memory on the main board. Unfortunately, it isn't easy to expand the memory past this point. The BCC bus is only 16 bits wide and would only allow access to 64K bytes of memory. For this reason, the BCC bus is strictly an I/O bus and all system memory must be on the processor board.

EPROM/Static RAM

There are four 28-pin sockets on the BCC180 board (U10-U13), each of which can accommodate either a 27256 (27C256) EPROM or a 62256 static RAM chip. Both types of chips contain 32K bytes of storage and are the only kinds of memory that will work. It isn't possible to use, say, a 27512 EPROM.

The 5-pin jumper header next to each socket determines whether an EPROM or a RAM chip is installed in that socket. When there is one jumper between pins 1 and 2 and a second jumper between pins 3 and 4, that socket is set for EPROM operation. To use the socket with a RAM chip, the jumpers must be moved so they connect pins 2 and 3 and pins 4 and 5. The BCC180 is shipped from the factory with U10 (JP6) set up for EPROM operation and U11-U13 (JP7-JP9) set up for RAM operation. The following shows how these jumpers are set up:

JP6-JP9



The first socket (U10) is mapped to physical address 00000H. Each socket is mapped in 32K-byte increments above that. For example, the second socket (U11) is at physical address 08000H and the third socket (U12) is mapped to 10000H. It is up to whatever program is running to determine how the HD64180's memory management unit (MMU) maps the logical address space into the physical address space.

The default configuration of the BCC180 monitor maps the entire first socket to the first 32K bytes in the logical address space. The other 32K bytes of space is mapped to U13. See the monitor manual for more details about changing active banks while in the monitor and the HD64180 data book for details about configuring the on-board MMU under program control (specifically, the CBAR, CBR, and BBR registers).

NOTE: A 32K-byte 62256 static RAM chip must be installed in socket U13 in order for the BCC180 monitor and the ROM-based version of BASIC-180 to function properly.

Dynamic RAM

In addition to the EPROM and static RAM space provided on the BCC180, there is also provision for using a 256K-byte dynamic RAM SIMM (Single In-line Memory Module). Either an 8-bit or a 9-bit module may be used. While the SIMM socket provided on the BCC180 will allow a 9-bit SIMM to be used, the ninth bit is ignored.

The SIMM is mapped to physical address 40000H and continues through address 7FFFFH. As with the EPROM/static RAM, it is up to the program running on the machine to determine the setup of the MMU.

Wait States

The HD64180 processor has, among other things, an on-board wait-state generator. It's possible to set anywhere from 0 to 3 memory wait states and from 0 to 4 I/O wait states. Memory wait states are dealt with in this section. See the "Input/Output" section for I/O wait state details.

EPROMs and static RAM chips rated for an access time of 150 ns or better can be used with zero memory wait states on a 9.216-MHz BCC180. Memory chips slower than this will need wait states. EPROMs with 200-ns access times need one wait state. <u>The BCC180 monitor defaults to two memory wait states to allow for slow EPROMs.</u> See the BCC180 monitor manual for details on how to change this default.

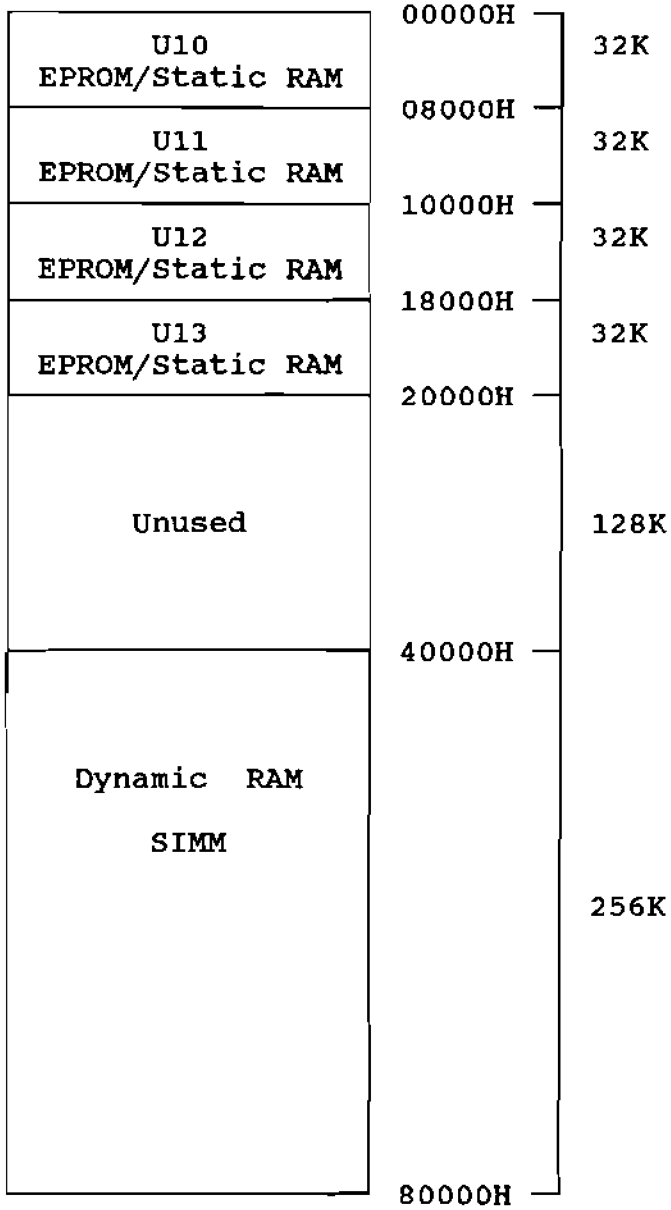
The same applies for the dynamic RAM. SIMMs with RAM chips rated for 120-ns or faster access times can be used on a 9.216-MHz BCC180 with zero wait states. <u>However</u>, an exception to this must always be kept in mind! Whenever the HD64180 fetches an instruction from memory, it accesses memory one-half clock cycle sooner than when it fetches data. This shortens the access time required of the dynamic RAM to values much smaller than is currently available (at an economical price, anyway).

What does this mean in terms of system operation? When using dynamic RAM strictly for data storage (all programs are stored in either EPROM or static RAM), zero memory wait states are required on a 9.216-MHz BCC180 using a 120-ns SIMM. However, when a program is to be stored and run in dynamic memory, one memory wait state must always be used. If you are unsure of these requirements, select one wait state just to be safe.

See the HD64180 data book for more details about configuring the on-board wait-state generator (specifically, the DCNTL register).

Recommended Memory Wait States
(for a 9.216-MHz BCC180)

Board Position	Part	Access Time	Suggested Memory Wait States
U10-U13	27x256 EPROM 62256 SRAM	≤120 ns	0
U10-U13	27x256 EPROM 62256 SRAM	150 ns	1
U10-U13	27x256 EPROM 62256 SRAM	≥200 ns	2 or more
SIMM1	Samsung KMM59256	≤120 ns	1 (0 possible, see above)
SIMM1	Samsung KMM59256	150 ns	1
SIMM1	Samsung KMM59256	≥200 ns	2 or more



BCC180 System Memory Map

Input/Output

The BCC180 has nearly unlimited expansion potential when it comes to I/O. It features an I/O space of 65535 (64K) I/O ports accessible through the BCC bus interface. Expansion is only limited by the number of boards that can be physically placed on the bus. (Use of more than eight expansion boards on the same bus requires additional bus drivers).

The HD64180 has 64 on-board I/O ports or registers that are used to control the chip's serial ports, memory management unit, DMA controller, and so on. These are mapped to ports 0000H to 003FH by default. They can also be set to start at 0040H, 0080H, or 00C0H. The use of these registers is well-documented in the HD64180 data book, so won't be covered in this manual.

The BCC180 also has on-board two 8255 programmable peripheral interface chips that provide six 8-bit parallel I/O ports. The port addresses for these are jumper selectable so they can be set to avoid conflicts with other peripherals. See the following section on parallel I/O for more details.

In order to access any peripheral device, the complete 16-bit port address must be specified by any program. The normal HD64180 "IN A,port" and "OUT port,A" instructions cannot be used since they only specify an 8-bit port address. (They actually do specify 16-bit addresses, but in a manner that is totally useless.) Instead, the "IN reg,(C)" and "OUT (C),reg" instructions should be used. These use the 16-bit value found in register pair BC as the port address and use the specified register for data. The BCC180 monitor's I and O commands support 16-bit port addresses as do BASIC-180's INP function and OUT statement.

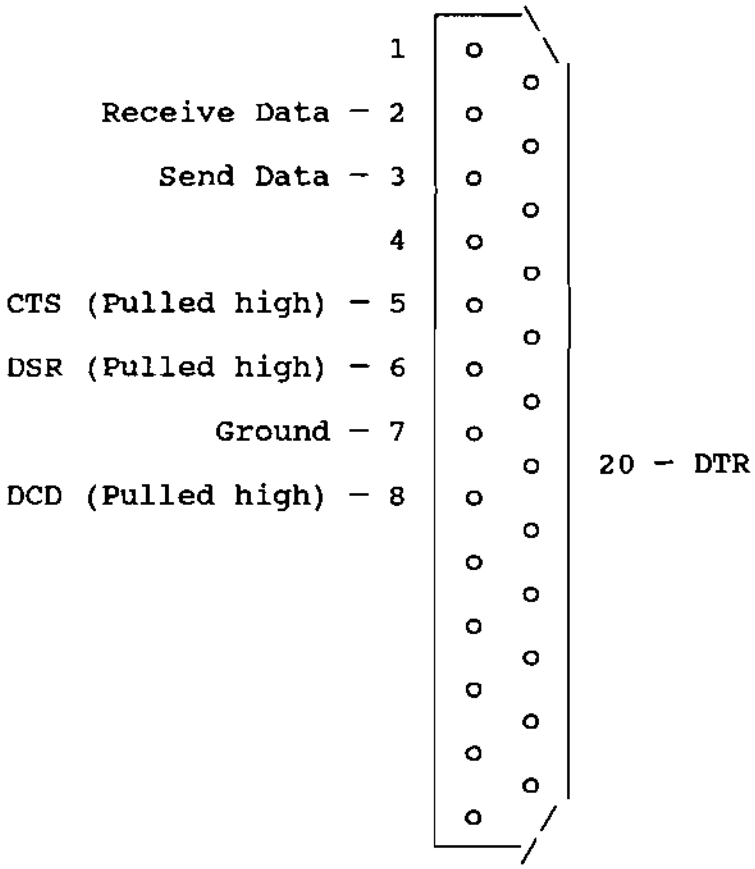
The HD64180 has an on-board wait-state generator that allows the system to generate anywhere from 0 to 4 I/O wait states. Wait states are used to compensate for a fast processor trying to access a slow peripheral device. It's difficult to predict exactly how many wait states a particular I/O device is going to need. It is safest to set the maximum number of I/O wait states (4), get the device working, then gradually reduce the number of wait states until the device stops working. You may actually want to leave the wait-state generator set to a value one greater than the minimum found by experimenting. This will prevent flakey operation caused by a device being pushed to its limits. Refer to the HD64180 data book for more details on how to set the on-board wait-state generator (specifically, the DCNTL register).

Serial I/O

The HD64180 processor used on the BCC180 contains three serial ports on the chip; two asynchronous and one synchronous. These have been brought out to connectors/headers on the BCC180 so they can be used in applications.

Async Port 1

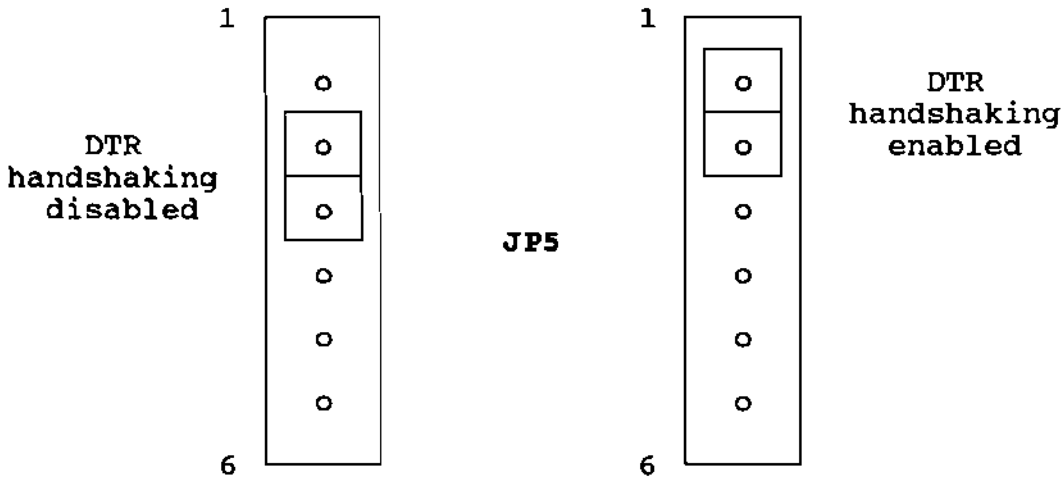
Asynchronous port 1 is normally used as the console port. The serial transmit, receive, and DTR handshake lines are sent through a level shifter to interface with a standard RS-232 terminal and are brought out to a standard DB-25S connector (J2) on the BCC180 board. Pins 5 (CTS), 6 (DSR), and 8 (DCD) are pulled high by the BCC180 to satisfy the requirements of some terminals. The following shows the pinout of J2:



J3

A jumper on JP5 determines whether DTR handshaking will be in effect for the console. If there is a jumper between pins 2 and 3 on JP5, handshaking is disabled. If there is a jumper between pins 1 and 2, handshaking is enabled. In this case, whenever the terminal is ready to receive characters, it should produce a high level on pin 20 of the DB-25 connection. When the terminal starts to fall behind and it wants to signal the computer to pause for a moment, it should drop pin 20 to a low level.

The following shows how to set up JP5. The default factory setting disables DTR handshaking and is shown on the left.



There is also a pair of console serial lines defined on the BCC bus (pins M and N on the edge connector, J1). Using an expansion board with serial capability (e.g., Micromint's BCC22 TermMite terminal board) it's possible to communicate through the BCC bus without the need for an extra cable between the boards. It also eliminates the need for the ±12V power supplies since RS-232 isn't being used. In order to use the BCC bus serial lines, the jumper on JP4 must be removed. Also be sure that console handshaking is disabled by installing a jumper between pins 2 and 3 on JP5.

Async Port 0 (RS-232 Operation)

Asynchronous port 0 is an auxiliary serial port that can be used in several different configurations. The most likely configuration will be as an RS-232 port. When there is a jumper between pins 2 and 3 on JP2, the port's transmit and receive lines are passed through an RS-232 level shifter and sent to J3. A cable made up of a 2x10 Berg-type connector at one end and a DB-25 connector at the other must be plugged into J3 on the BCC180. For the benefit of whatever device might be connected to J3, pins 4 and 20 and pulled high. The following shows the pin-

Signal	Pin			Pin	Signal
	1	o	o	14	
Send Data	2	o	o	15	
Receive Data	3	o	o	16	
RTS (Pulled high)	4	o	o	17	
	5	o	o	18	
	6	o	o	19	
Ground	7	o	o	20	DTR (Pulled high)
	8	o	o	21	
	9	o	o	22	
	10	o	o	23	

J3

A serial printer may be connected to J3, however hardware handshaking (Printer Busy) is not supported. The ROM monitor (and most likely whatever ROM language is installed; check its manual) supports software XON/XOFF handshaking with the printer. Make certain that the send and receive data lines are properly matched for the printer you're using (most require that pins 2 and 3 be swapped by use of a null modem cable) and any unused handshaking inputs to the printer are pulled high. An RS-232 breakout box is an invaluable tool in making a stubborn printer work.

A modem may also be connected to J3. In this case, it isn't necessary to swap pins 2 and 3. Just be sure that the modem is set to ignore its DTR input line (check the modem's manual). DCD from the modem is ignored by the BCC180; there is no way for the computer to check the modem's DCD output signal.

Async Port 0 (RS-422/RS-485 Operation)

The other mode in which the auxiliary serial port can run is RS-422/RS-485. RS-232 signalling is called "single ended." That is, there is a common ground between the devices connected together, and all signals are referenced to that ground. For example, a high signal is +12V referenced to ground and a low signal is -12V referenced to ground.

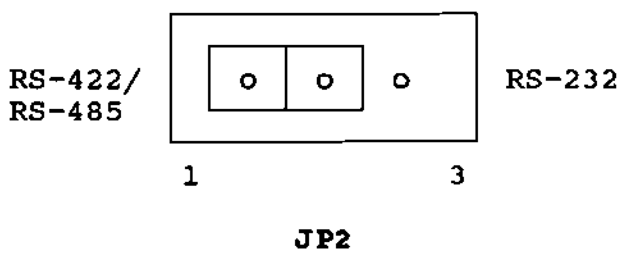
RS-422 and RS-485 use what is called "double ended" or "balanced" signalling. In other words, there is no common ground between the devices connected together. Instead, the voltage differential between a pair of wires is used. When that differential voltage is positive, a high signal is denoted. A low signal is represented by a negative differential voltage between the wires.

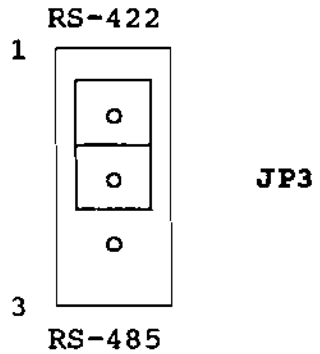
The biggest advantage to using balanced signalling is its increased noise immunity and, hence, increased communication distance. The reason for the increased noise immunity is as follows: when the signal reaches the receiver, the voltage on one wire is subtracted from the voltage on the other. The result is either a positive or negative voltage indicating either a high or a low signal. If some noise is introduced onto the signal on its way from the driver to the receiver, that noise appears on both wires with equal magnitude. When the voltages on the wires are subtracted at the receiver, the noise is cancelled out and a clean signal results.

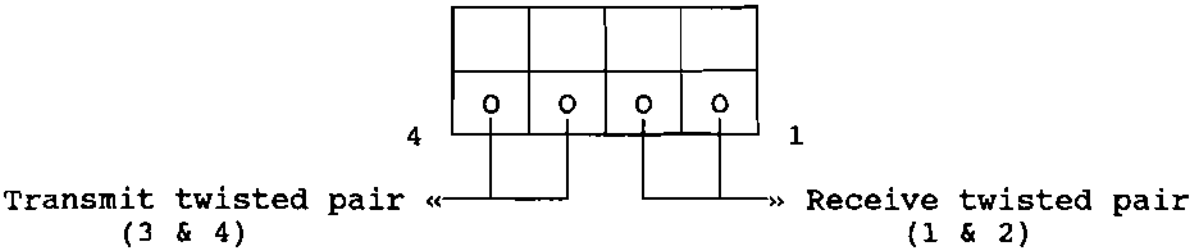
In RS-422 operation, one twisted pair is used for communicating in one direction between two devices and a second twisted pair is used to communicate in the other direction. This is similar to RS-232 in that only two devices can be connected together and full-duplex operation is possible.

RS-485 is simply an extension of RS-422. It uses a single twisted pair to communicate in both directions. As a result, half-duplex must be used (i.e., data can only flow in one direction at a time). The advantage gained by using RS-485 is that more than two devices may be connected to the same twisted pair, party line fashion. The RS-485 spec states that up to 32 devices may be connected to the same twisted pair. Using this setup, it's possible to implement a low-cost (albeit low-performance) LAN. Protocols used by the devices to arbitrate who talks when are strictly up to the system designer.

To use the auxiliary port in an RS-422 application, there must first be a jumper installed between pins 1 and 2 of JP2. This disables the RS-232 portion of the auxiliary port. Next, a jumper should be installed between pins 2 and 3 of JP3. This permanently enables the receiver circuit. Finally, a jumper should be installed in JP1 to enable receiver termination. To make the connection with a remote device, the receive data twisted pair should be connected to pins 1 and 2 of J8 and the transmit data twisted pair should be connected to pins 3 and 4 of J8. The following shows how to set up JP2, JP3, and J8 for RS-422 operation:





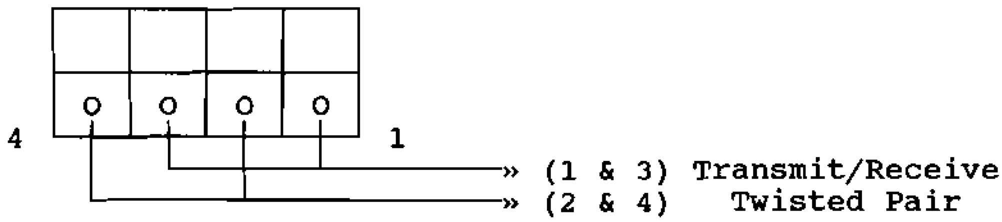


J8

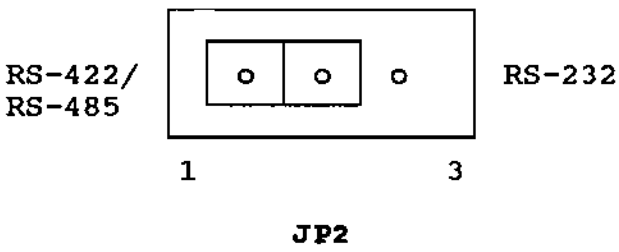
From a software standpoint, the only important item to remember is to enable the driver circuit. That is done by dropping the HD64180's RTS0\ output low. To do this, suppose a data format of 8 data bits, 1 stop bit, and no parity is desired (the data format is controlled by the same port). The assembly instruction "OUT0 (00H),64H" would drop RTS0\ low and the command "OUT0 (00H),74H" would raise it again. If BASIC-180 is installed, the statements "OUT $0,$64" and "OUT $0,$74" would do the same thing. For details on the other bits in this control register, refer to the HD64180 data book and the CNTLA0 port. Once the driver is enabled, data is sent and received in the same manner as an RS-232 connection.

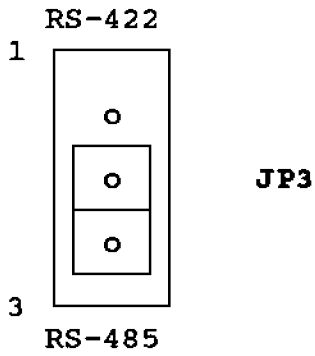
RS-485 operation is a little more involved. Set up JP2 as above (jumper pins 1 and 2). On JP3, connect a jumper between pins 2 and 3. The purpose of this jumper will be explained in a moment. Termination (JP1) is only needed on the two ends of the main twisted pair. That is, the proper way to set up a multidrop RS-485 connection is to have a main twisted pair that goes from a device on one end to a device and the other. To connect devices in the middle, short "stubs" are used to connect the device to the main twisted pair. One of these middle devices should not be terminated.

Once the jumpers are correct, connect pins 1 and 3 on J8 to one side of the twisted pair and connect pins 2 and 4 to the other side. Normal operation would have the driver circuit (U9) turned off and the receiver circuit (U8) turned on. This is done by setting the HD64180's RTS0\ output high (see above). When it's time to send a byte, the driver circuit must be turned on. However, we don't necessarily want to receive the byte we just sent, so the receiver should be turned off. Dropping the RTS0\ output low will accomplish both tasks. The following shows how to set up JP2, JP3, and J8 for RS-485 operation:



J8



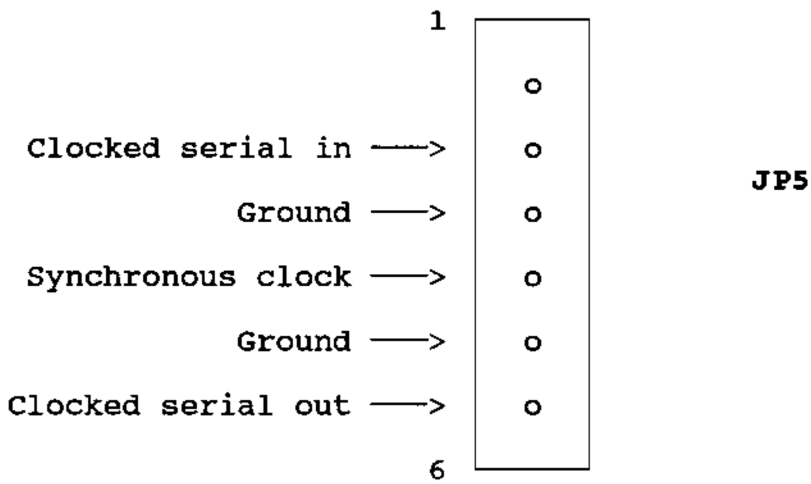


In summary, for RS-422 operation, RTS0\ should be low all the time. For RS-485 operation, RTS0\ should be high to receive data and low to send data.

Synchronous Port

In addition to the two asynchronous ports, the HD64180 has a clocked serial I/O (CSIO) port for doing synchronous communication. The operation of this port won't be discussed here since it is well-covered in the HD64180 data book.

JP5 is used to bring the CSIO lines to the outside world. Since the serial in line is multiplexed with the console handshaking line, use of CSIO disables console handshaking. Any jumpers installed on JP5 should be removed. The following shows the pinout for the CSIO section of JP5:



Parallel I/O

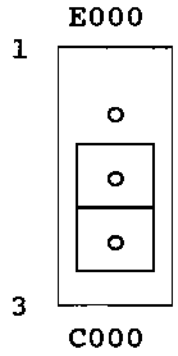
Six parallel input/output ports are provided on the BCC180 board with two 8255 peripheral interface adapter chips. The 48 I/O bits and ground are available on two 26-pin Berg connectors located between the 8255s and the RS-232 connector.

The 8255 is a programmable interface device. The 24 I/O lines on each chip are divided into three ports, A, B, and C, each separately configurable either as input, output, or handshaking lines under software control. A control register on each chip defines the characteristics of each of the ports on that chip.

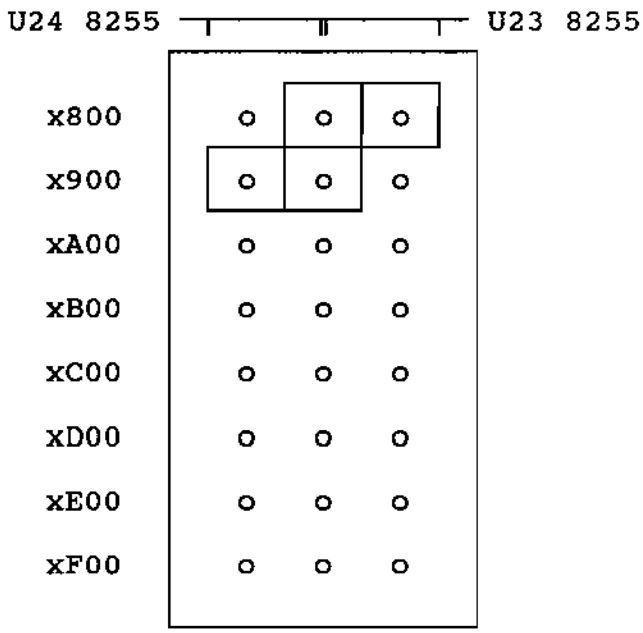
Each 8255 occupies 256 ports in the BCC180's I/O space (although only four of these ports are actually used). The actual port addresses can be selected by jumpers on JP10 and JP11. The following table shows the possible jumper configurations and the resulting port addresses:

		I/O Port Address			
JP10 BASE +	JP11 OFFSET =	PORT A	PORT B	PORT C	CONTROL REGISTER
C000H	800H	C800H	C801H	C802H	C803H
"	900H	C900H	C901H	C902H	C903H
"	A00H	CA00H	CA01H	CA02H	CA03H
"	B00H	CB00H	CB01H	CB02H	CB03H
"	C00H	CC00H	CC01H	CC02H	CC03H
"	D00H	CD00H	CD01H	CD02H	CD03H
"	E00H	CE00H	CE01H	CE02H	CE03H
"	F00H	CF00H	CF01H	CF02H	CF03H
E000H	800H	E800H	E801H	E802H	E803H
"	900H	E900H	E901H	E902H	E903H
"	A00H	EA00H	EA01H	EA02H	EA03H
"	B00H	EB00H	EB01H	EB02H	EB03H
"	C00H	EC00H	EC01H	EC02H	EC03H
"	D00H	ED00H	ED01H	ED02H	ED03H
"	E00H	EE00H	EE01H	EE02H	EE03H
"	F00H	EF00H	EF01H	EF02H	EF03H

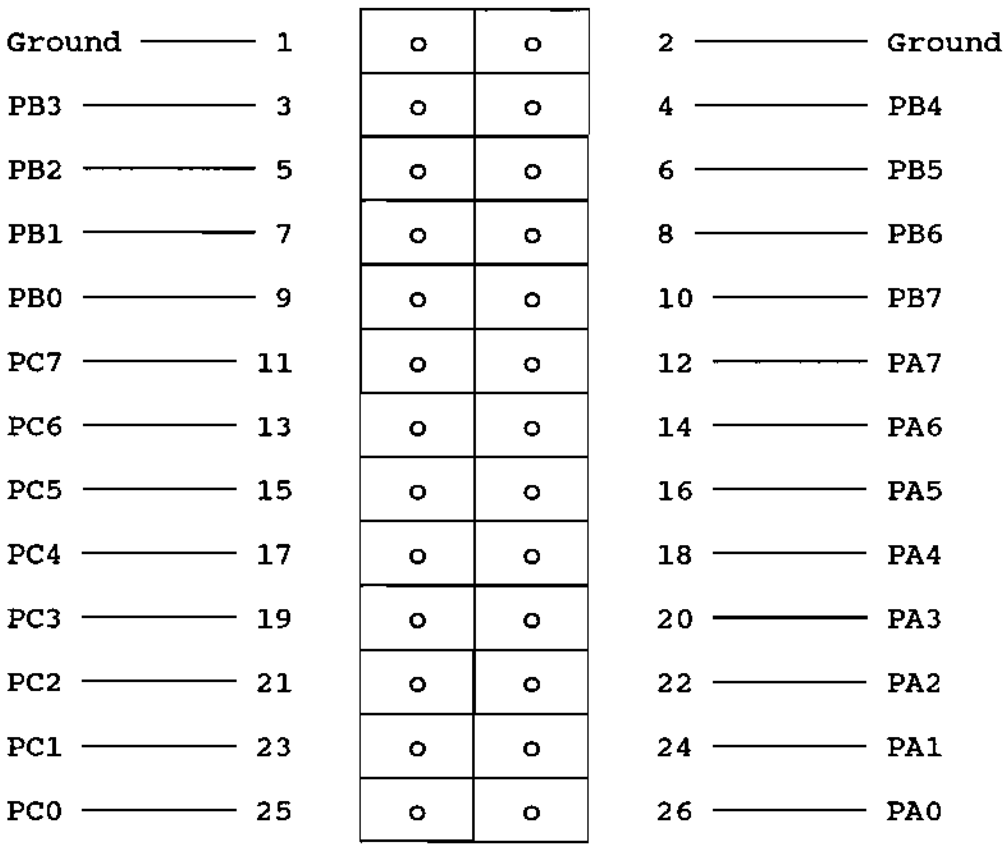
For example, the jumper settings on JP10 and JP11 on the following page configure U23/J5 for a base port address of C800H while U24/J6 is set up for C900H (this is how the BCC180 is set up at the factory). In addition, the pinout of each 26-pin parallel I/O header is shown.



JP10



JP11



J5/J6

The 24 lines of each 8255 are divided into two groups of 12 lines each: group A (port A and the upper half of port C) and group B (port B and the lower half of port C). The functional configuration of each port is set by the system software. The BCC180 outputs a control word to the 8255 which contains information such as "mode," "bit set," "bit reset," and so on, that initializes the 8255. The control register is a write-only port (i.e., the control word cannot be read by examining the contents of the control register port).

When the power to the computer is turned on, the 8255s are in an unknown configuration. Before the ports can be used they must be initialized by loading a control word into the control register port. For example, if BASIC-180 is installed, the statement "OUT $C803,$80" or "SETPIO 0,$80" will output the value 80H to the control register of the 8255 with a base port address of C800H. (Note that "SETPIO 0,xx" only works if the 8255 is jumpered for a base port address of C800H. See the BASIC-180 manual for more details about the SETPIO statement.) The value 80H sets all three ports for output and mode 0 operation (bit 7 of the control register must always be set to a 1). At this point, 8-bit values can be directed to the specific ports with either the OUT statement or the PIOOUT statement. Setting port B to 56H can be done using either "OUT $C801,$56" or "PIOOUT 1,$56".

In a similar manner, all three ports of one of the 8255s can be set for mode 0 input operation by using a control word of 9BH. The BASIC statement "OUT $C803,$9B" or "SETPIO 0,$80" will do this. To display the current value of port A, either "PRINT INP($C800)" or "PRINT PIOIN(0)" can be used. The following is a list of control word values for some typical 8255 port configurations:

Control Word Value	Port A	Port B	Port C
80H	output	output	output
89H	output	output	input
82H	output	input	output
8BH	output	input	input
90H	input	output	output
99H	input	output	input
92H	input	input	output
9BH	input	input	input

A complete specification of the 8255 is not included in this manual. Should you need to configure one of the 8255s for a more complicated I/O configuration or require the use of handshaking, refer to an 8255 data sheet published by one of the following manufacturers:

Intel Corporation
3065 Bowers Avenue
Santa Clara, CA 95051

NEC Electronics USA, Inc.
One Natick Executive Park
Natick, MA 01760

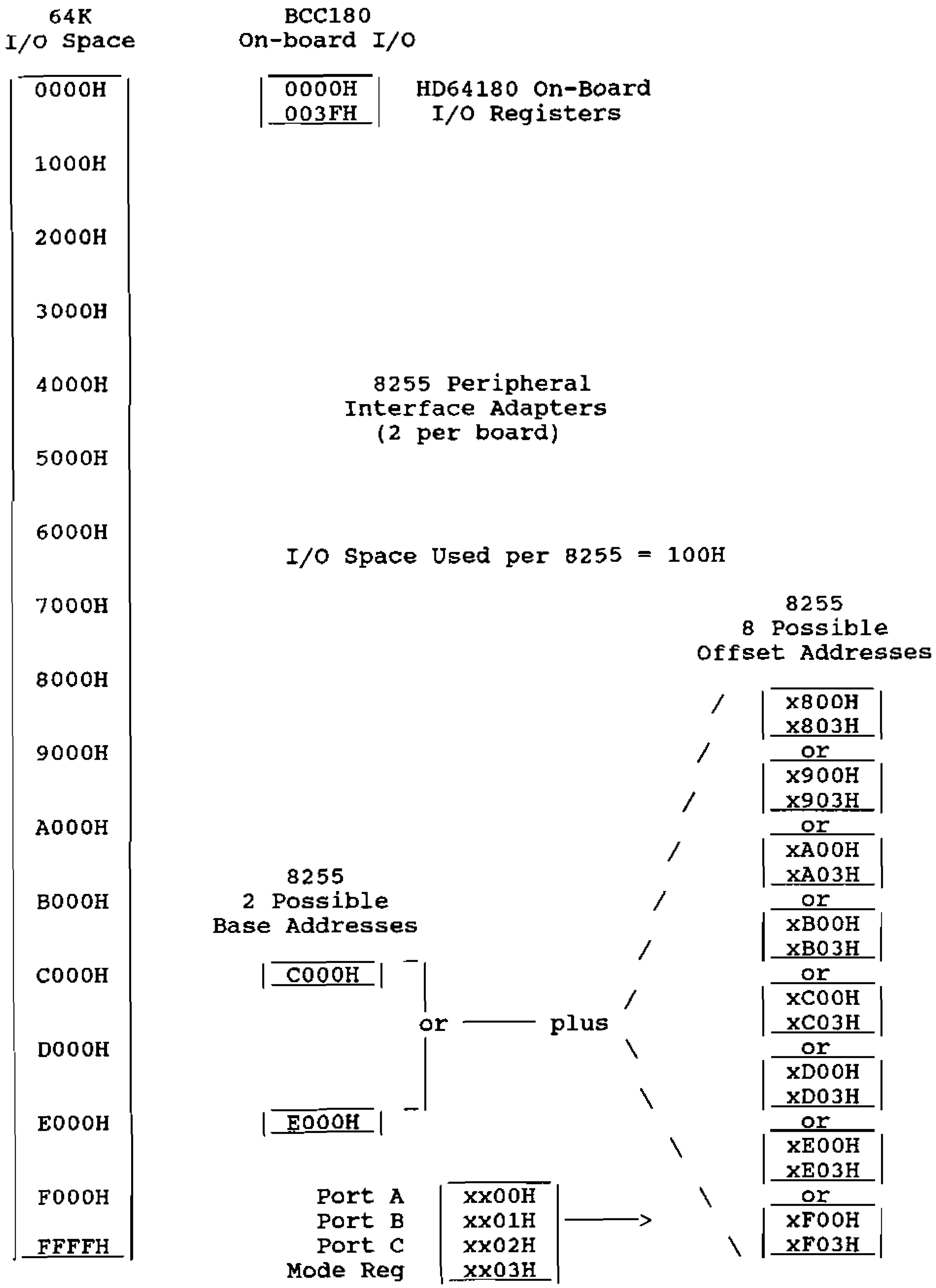
National Semiconductor Corporation
2900 Semiconductor Drive
Santa Clara, CA 95051

Interfacing with other BCC-series boards

The BCC180 is completely bus compatible with Micromint's BCC expansion boards originally designed for the BCC11 (Z8) and BCC52 (8052) computers. Some of the boards, however, wouldn't be especially useful to the BCC180. For example, any board that boasts memory expansion could be used by the BCC180, but each memory location on the expansion board would use a separate I/O port address and would have to be accessed by a separate IN or OUT instruction.

The boards shown on the following pages are those that have been determined to be useful on a BCC180 system at the time of this manual revision. Again, any BCC-series board will work fine with the BCC180. For each board, the potential port locations that the board can occupy are shown. Each board's actual port address can be selected with jumpers on the board. The manual that comes with each board should be consulted for details on changing those jumpers.

BCC180 Computer/Controller



BCC08 Serial Expansion

I/O Space Used per Board = 1000H

The BCC08 is a single-port, UART-based serial expansion board. It provides both RS-232 and 20-mA connections.

```
   64K
I/O Space

| 0000H |
|       |
| 1000H |
|       |
| 2000H |
|       |
| 3000H |
|       |
| 4000H |
|       |
| 5000H |
|       |
| 6000H |
|       |
| 7000H |
|       |
| 8000H |
|       |
| 9000H |
|       |
| A000H |
|       |
| B000H |
|       |          BCC08
| C000H |        I/O Space
|       |
| D000H |        | D000H |  \
|       |        |       |    BCC08 #1
|       |        | DFFFH |  /
| E000H |        | E000H |  \
|       |        |       |    BCC08 #2
|       |        | EFFFH |  /
| F000H |        | F000H |  \
|       |        |       |    BCC08 #3
| FFFFH |        | FFFFH |  /
```

BCC13 8-bit Analog-to-Digital Converter

64K
I/O Space

64K I/O Space
0000H
1000H
2000H
3000H
4000H
5000H
6000H
7000H
8000H
9000H
A000H
B000H
C000H
D000H
E000H
F000H
FFFFH

I/O Space Used per Board = 100H

The BCC13 is an 8-channel, 8-bit A/D converter board. It can run at up to 10,000 samples per second and its inputs can either range from 0V to 10V or from -5V to +5V.

Port Address = Base + Offset + Channel #

BCC13 16 Possible Base Addresses	BCC13 8 Possible Offset Addresses
8000H / 8800H	x000H / x0FFH
9000H / 9800H	x100H / x1FFH
A000H / A800H	x200H / x2FFH
B000H / B800H	x300H / x3FFH
C000H / C800H	x400H / x4FFH
D000H / D800H	x500H / x5FFH
E000H / E800H	x600H / x6FFH
F000H / F800H	x700H / x7FFH