M68331EVK/D
REV 1

October 1993

M68331EVK

EVALUATION KIT

USER'S MANUAL

Motorola reserves the right to make changes without further notice to any products herein to improve reliability, function or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part.

TABLE OF CONTENTS

CHAPTER 1 GENERAL INFORMATION

CHAPTER 2 HARDWARE PREPARATION AND INSTALLATION

CHAPTER 3 OPERATING INSTRUCTIONS

CHAPTER 4 FUNCTIONAL DESCRIPTION

LIST OF ILLUSTRATIONS

Figure **Page**

LIST OF TABLES

Table **Page**

CHAPTER 1
GENERAL INFORMATION

1.1 INTRODUCTION

This manual provides general information, hardware preparation, installation instructions, functional description, and support information for the M68331EVK Evaluation Kit (hereafter referred as EVK). The EVK is the M68300EVS Evaluation System in the standalone configuration. Appendix A contains EVK downloading S-record information.

The EVK consists of two printed circuit boards and one software program:

- M68331BCC Business Card Computer (BCC)
- M68300PFB Platform Board (PFB)
- CPU32BUG Debug Monitor (CPU32Bug)

1.2 FEATURES

The BCC consists of:

- MC68331 Microcontroller Unit (MCU)
- 64k x 16 bit, erasable programmable read only memory (EPROM)
- 32k x 16 bit, byte addressable random access memory (RAM)
- RS-232C compatible terminal/host computer input/output (I/O) port
- Background mode interface port
- BCC expansion connectors

The PFB consists of:

- BCC expansion connectors for the BCC and M68300DI Development Interface(1)
- External power supply connector
- Two compatible terminal/host computer I/O ports
- Background mode interface port
- Memory expansion sockets
- Socket for MC68881 or MC68882 Coprocessor
- Logic analyzer interface

CPU32Bug includes:

- Commands for display and modification of memory
- Breakpoint capabilities
- An assembler/disassembler useful for patching programs
- A power-up self test feature which verifies system integrity
- A command-driven user-interactive software debugger (the debugger)
- A user interface which accepts commands from the system console terminal
- A parameter area for user customization

CPU32Bug is described in detail in the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1.

1.3 SPECIFICATIONS

Tables 1-1 and 1-2 list BCC and PFB specifications.

Table 1-1. BCC Specifications

CHARACTERISTICS	SPECIFICATIONS
Internal Clock External Clock	32.768 kHz 25 kHz to 50 kHz(1)
Memory 32k x 16 RAM 64k x 16 EPROM	 85ns (3 clock bus cycle access @ 16.7 MHz) 200ns (5 clock bus cycle access @ 16.7 MHz)
Terminal/Host I/O Port	RS-232C compatible (with internal DC-DC converters for +/-10 volts, 10 mA)
Temperature Operating Storage Relative humidity	 +25° C -40 to +85° C 0 to 90% (non-condensing)
Power Requirements Power Supply Battery Backup	 +5 Vdc @ 200 milliamps (min.) +3 Vdc @ 50 microamps (min.)
Dimensions	2.25 x 3.875 in. (5.7 x 9.84 cm)

1. An optional high frequency clock source (as high as 16.77 MHz) may be used if MODCK (P2, pin 28) is pulled to a logic low level. A hybrid oscillator is recommended as the external clock.

Table 1-2. PFB Specifications

CHARACTERISTICS	SPECIFICATIONS
External Clock	25 kHz to 50 kHz
Expanded Memory Sockets U1 - U4: 32k x 16 RAM (U1, U2, U3, & U4) 32k x 16 EPROM (U2 & U4) 64k x 16 EPROM (U2 & U4)	 25ns (2 clock bus cycle access @ 16.7 MHz) or 85ns (3 clock bus cycle access @ 16.7 MHz) 200ns (5 clock bus cycle access @ 16.7 MHz) 200ns (5 clock bus cycle access @ 16.7 MHz)
Terminal/Host I/O Ports	RS-232C compatible
Temperature Operating Storage Relative humidity	 +25° C -40 to +85° C 0 to 90% (non-condensing)
Power Requirements Power Supply Battery Backup	 +5 Vdc @ 500 milliamps (min.) +3 Vdc @ 50 microamps (min.)
Dimensions	6.25 x 10 in. (15.88 x 25.4 cm)

1.4 GENERAL DESCRIPTION

Using the EVK, the user can design, debug, and evaluate MC68331 MCU-based target systems. The BCC simplifies user evaluation of prototype hardware/software products. The EVK requires a user-supplied power supply and an RS-232C compatible terminal for functional operation. The EVK consists of two printed circuit boards, the BCC and the PFB.

The BCC operates as a single board computer, or as a well-defined core in larger applications. Mounted on the BCC are a microcontroller, on-board memory, and serial level converter circuitry. The BCC also has a 4-pin connector for serial communication. The 64-pin expansion connectors provide access to most of the MC68331 MCU pins.

The PFB is the physical location for installing the BCC and Development Interface (DI). The user may also expand the BCC user accessible memory by installing RAM or EPROM in sockets on the PFB. PFB RAM/EPROM sockets may be configured for autoboot. The PFB has two DB-9 terminal I/O ports for communicating with the BCC and DI from a host computer or terminal. The PFB also has the BCC power connector. See Chapter 2 for I/O connector pin-outs and Chapter 5 for interface connector pin assignments and signal descriptions.

The M68CPU32BUG Debug Monitor (CPU32Bug) is provided in the BCC EPROM. CPU32Bug is a software evaluation and debug tool that may be used to develop systems built around the MCU. Using the debug monitor, the user interacts with the EVK through monitor commands that are entered at the terminal/host computer keyboard. These commands perform functions such as modification of memory, modification of MCU internal registers, program execution under various levels of control, and access to various I/O peripherals in the MCU itself.

User programs may be downloaded into RAM on the BCC or PFB. For a detailed description of the CPU32Bug refer to the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1.

To program the BCC EPROMs you must remove the EPROMs and use an EPROM programmer.

1.5 EQUIPMENT REQUIRED

Table 1-3 lists the external equipment requirements for EVK operation.

Table 1-3. External Equipment Requirements

EXTERNAL EQUIPMENT
A terminal or host computer (RS-232C compatible) with a terminal emulation package (PCKERMIT, PROCOMM, MacTerminal, White Knight, etc.)(1)
Serial communication cable for the terminal or host computer(2).
+5 Vdc at 500 mA power supply(2)

1. Refer to Chapter 3 for details on downloading using a host computer with terminal emulation package.
2. Refer to Chapter 2 for details.

CHAPTER 2
HARDWARE PREPARATION AND INSTALLATION

2.1 INTRODUCTION

This chapter provides unpacking instructions, hardware preparation, and installation instructions for the EVK. This description ensures the EVK is properly configured for target system operation.

2.2 UNPACKING INSTRUCTIONS

Unpack the BCC from shipping carton. Refer to the packing list and verify that all items are present. Save packing material for storing and shipping the BCC.

NOTE

If the product arrives damaged, save all packing material, and contact the carrier's agent.

2.3 HARDWARE PREPARATION

The EVK has been factory tested and is shipped with installed jumpers. The user may reposition these jumpers when the application requires customization of EVK functionality. There are also several connectors on the EVK. These connectors provide power, communication, and access to EVK features. Figure 2-1 shows the BCC installation on the PFB, via the expansion connectors. Figures 2-2 and 2-3 show the locations of switches, jumpers and connectors on the EVK boards.

CAUTION

Use caution when handling the EVK; the signals are not buffered so the EVK is sensitive to static discharge.

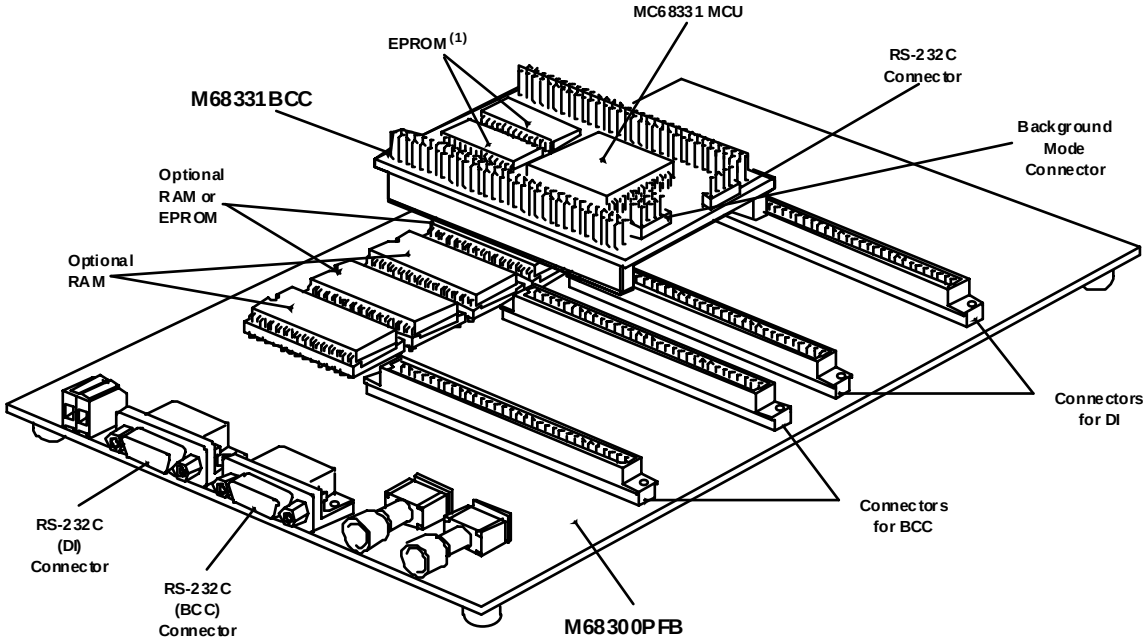


Figure 2-1. M68331EVK Evaluation Kit

1. There is RAM and EPROM on the BCC. The BCC RAM (U3 & U4) is located under the EPROM (U1 & U2). To access RAM, remove the EPROM from the sockets. Use caution when removing or installing the EPROM.

2.3.1 BCC Configuration

The BCC was factory tested and shipped with installed jumpers. The user may reposition these jumpers to customize EVK functionality. Refer to Figure 2-2 for the location of the BCC jumper headers.

CAUTION

Depending on the application, it may be necessary to cut wiring trace shorts (cut-trace shorts) on the PCB. Be careful not to cut adjacent PCB wiring traces.

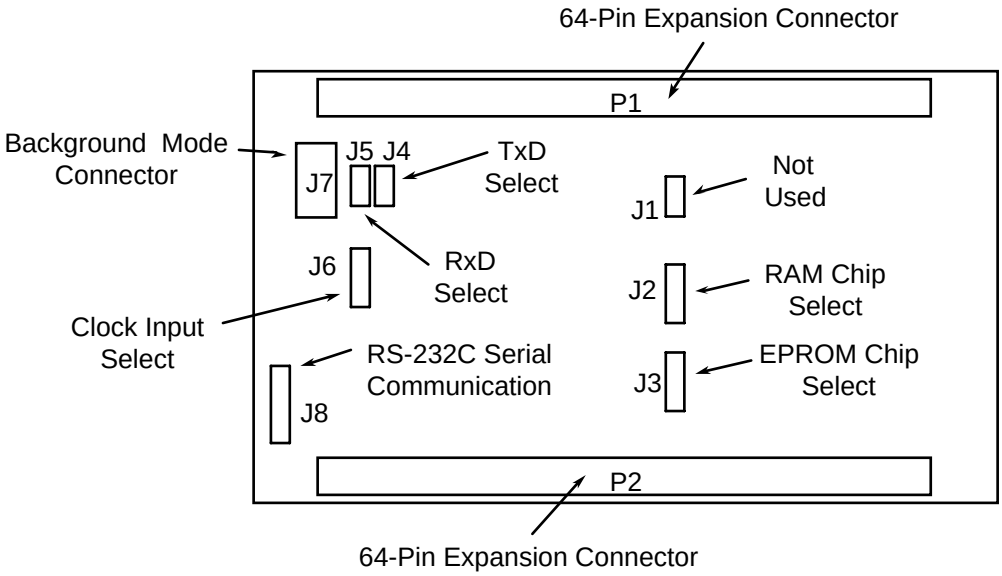
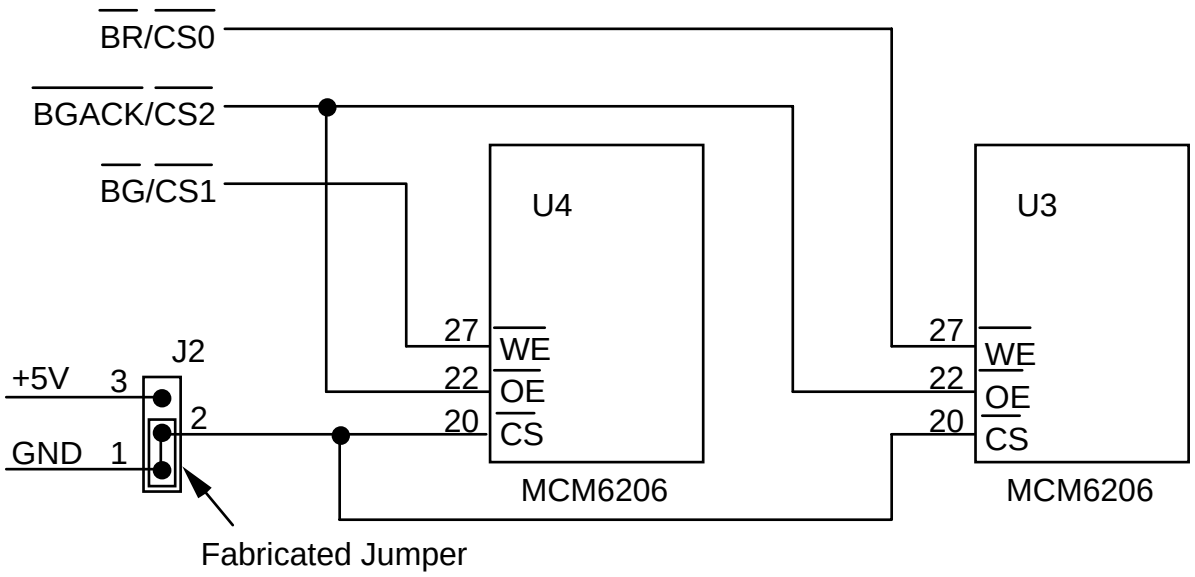


Figure 2-2. BCC Jumper Header and Connector Location Diagram

2.3.1.1 RAM Chip Enable Select Header (J2)

Use the three-pin jumper header J2 (shown below) to enable/disable selection of the on-board RAM. The BCC is shipped from the factory with the RAM chip select connected to GND via a cut-trace short on the bottom of the BCC PCB between pins 1 and 2. A fabricated jumper is also installed between pins 1 and 2. The cut-trace short or a fabricated jumper between pins 1 and 2 enables the BCC on-board RAM.



To disable the RAM from the BCC memory map, cut the trace on the solder side of the board on J2, between pins 1 and 2, and move the fabricated jumper to pins 2 and 3. This jumper disables selection of the on-board RAM by connecting chip enable to +5V. The chip selects are now free for other uses. Refer to the BCC schematic diagram for more detail on RAM chip select signal wiring.

CAUTION

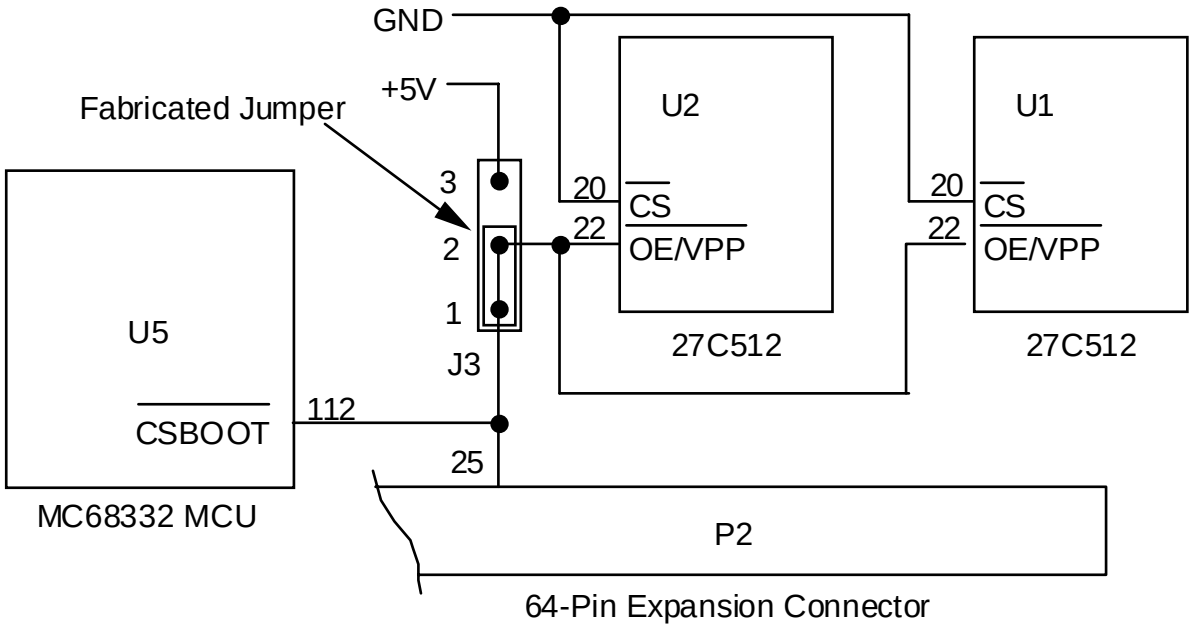
Do not connect the jumper between pins 2 and 3 before removing the cut-trace short between pins 1 and 2. Installing the jumper before the cut-trace short is removed connects +5Vdc to ground. If +5Vdc is shorted to ground the BCC or power supply may be damaged.

NOTE

If the cut-trace short on jumper header J2 is cut, the fabricated jumper must be reinstalled on J2, pins 1 and 2 to return the BCC to its default setting.

2.3.1.2 EPROM Chip Select Header (J3)

When BCC power is applied or reset occurs, the MC68331 MCU device resets itself and downloads the program in EPROM (U1 & U2). The EPROM contains the boot program. Use the three-pin jumper header J3 (shown below) to disable the BCC on-board EPROM. The BCC is shipped from the factory with the EPROM connected to the MCU bootstrap chip select pin ($\overline{\text{CSBOOT}}$) via a cut-trace short on the bottom of the BCC PCB between pins 1 and 2. A fabricated jumper is also installed between pins 1 and 2. To boot from a program stored in memory located in the target system, cut this trace, move the jumper to pins 2 and 3, and connect $\overline{\text{CSBOOT}}$ to the target system via P2, pin 25. Cutting the cut-trace short and moving the fabricated jumper to pins 2 and 3, removes U1 & U2 from the BCC memory map. Refer to the BCC schematic diagram for more detail on $\overline{\text{CSBOOT}}$ signal wiring.



CAUTION

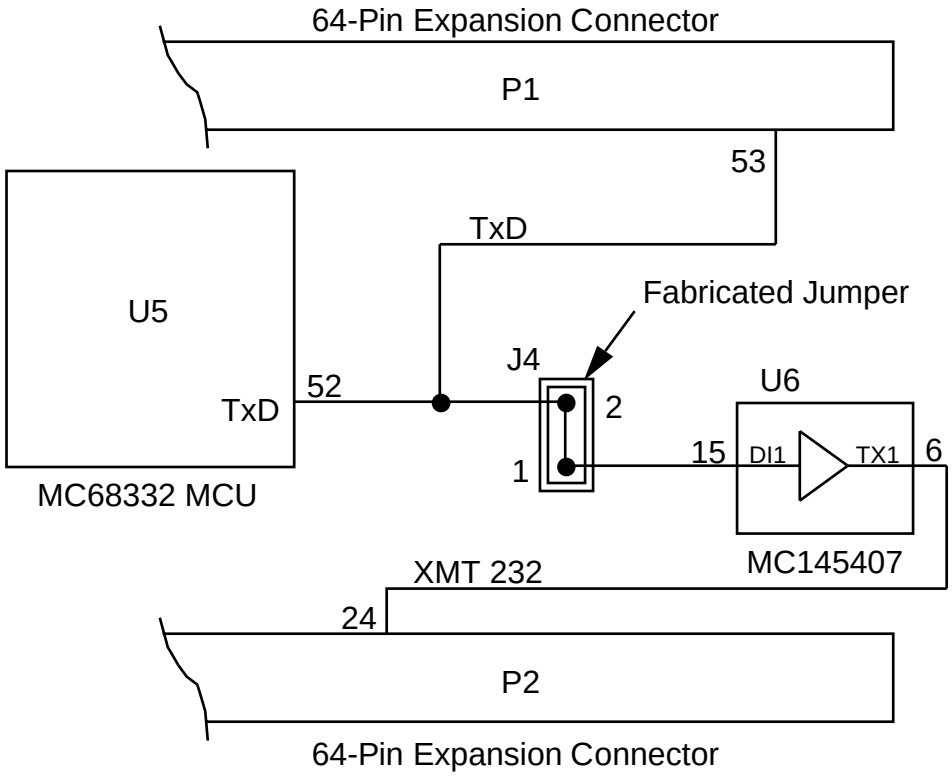
Do not connect the jumper between pins 2 and 3 before removing the cut-trace short between pins 1 and 2. Installing the jumper before the cut-trace short is removed connects +5Vdc to ($\overline{\text{CSBOOT}}$). If +5Vdc is shorted to ($\overline{\text{CSBOOT}}$) the MCU may be damaged.

NOTE

If the cut-trace short on jumper header J3 is cut, the fabricated jumper must be reinstalled on J3 pins 1 and 2 to return the BCC to its default setting.

2.3.1.3 TxD Select Header (J4)

Jumper header J4 allows the user to disconnect the transmit TxD serial data pin of the MC68331 MCU device (U5) from the RS-232C receiver/driver (U6) and use a target system receiver/driver. The BCC is shipped from the factory with the receiver/driver connected to MCU TxD (pin 52) via a cut-trace short on the bottom of the BCC PCB between J4, pins 1 and 2 (shown below). A fabricated jumper is also installed on pins 1 and 2. To disconnect the serial pin of the MCU, cut this trace and remove the jumper.



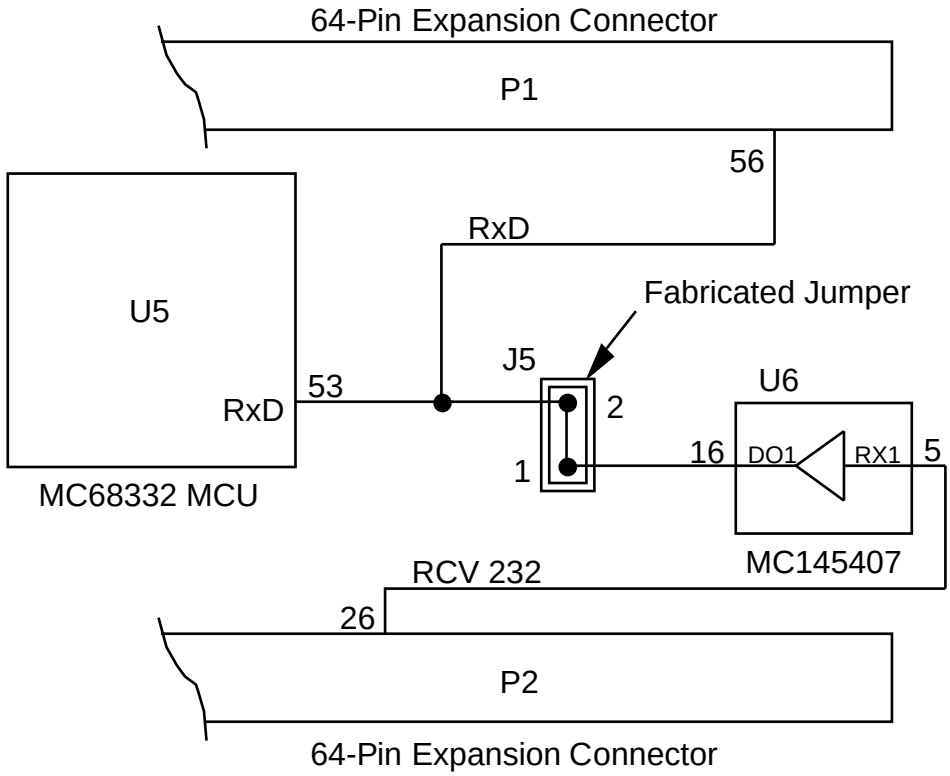
NOTE

If the cut-trace short on jumper header J4 is cut, the fabricated jumper must be reinstalled on J4 to return the BCC to its default setting.

Refer to the BCC schematic diagram for more detail on TxD signal wiring.

2.3.1.4 RxD Select Header (J5)

Jumper header allows the user to disconnect the receive RxD serial data pin of the MC68331 MCU device (U5) from the RS-232C receiver/driver (U6) and use a target system receiver/driver. The BCC is shipped from the factory with receiver/driver connected to MCU RxD via a cut-trace short on the bottom of the BCC PCB between J5, pins 1 and 2 (shown below). A fabricated jumper is also installed on pins 1 and 2. To disconnect the RxD pin of the MCU, cut this trace and remove the jumper.



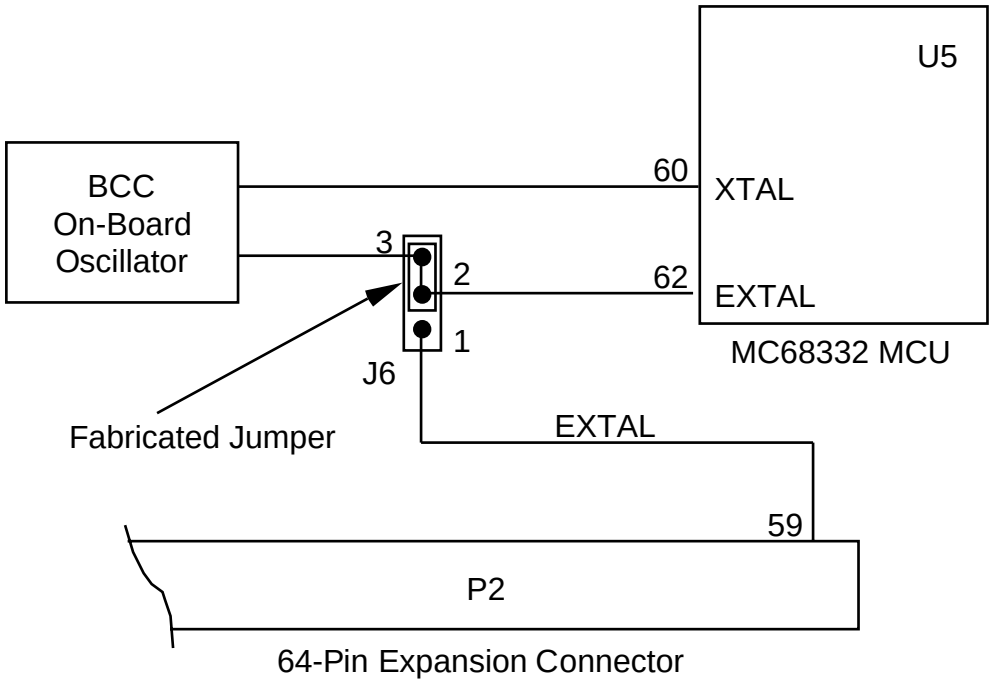
NOTE

If the cut-trace short on jumper header J5 is cut, the fabricated jumper must be reinstalled on J5 to return the BCC to its default setting.

Refer to the BCC schematic diagram for more detail on RxD signal wiring.

2.3.1.5 Clock Input Select Header (J6)

Use the three-pin jumper header J6 (shown below) to select the BCC on-board clock source or an external clock source. The BCC on-board clock source is a 32 kHz crystal, which is frequency multiplied by the MC68331 to a programmable operating frequency. The BCC is shipped with the on-board crystal selected as the clock source. J6 has a cut-trace short on the bottom of the BCC PCB between pins 2 and 3. A fabricated jumper is also supplied, but is not required when the user uses the on-board clock source. Refer to the BCC schematic diagram for more detail on EXTAL signal wiring.



An optional high frequency oscillator (0 to 16.77 MHz) may be used if MODCK (connector P2, pin 28) is pulled to a logic low level. To use the target-system, external-source CMOS clock; follow these steps:

a. Turn off power to the BCC.

b. Cut the printed circuit trace on the bottom of the BCC between pins 2 and 3.

c. Move the fabricated jumper between pins 2 and 3 to pins 1 and 2.

d. Supply an external oscillator to connector P2, pin 59 (EXTAL).

e. Ground connector P2, pin 28 (MODCK).

f. Apply power to the BCC, start the external oscillator, and drive connector P1, pin 57 (RESET), low.

NOTES

If the cut-trace short on jumper header J6 is cut, the fabricated jumper must be reinstalled on J6 pins 2 and 3 to return the BCC to its default setting.

Use a hybrid oscillator when driving the MCU from an external source.

Any change in the MC68331 MCU device clock speed causes a corresponding change in the SCI baud rate. The operational speed of the MCU is determined by the clock and the synthesizer control register value (SYNCR). The SCI baud rate is then set based on this system clock frequency. If changes are made to the MCU speed and the terminal baud rate is not changed appropriately, terminal communication will fail. Refer to Appendix C of the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1.

2.3.2 PFB Configuration

This section of the manual describes the inspection and preparation of the PFB. The PFB was factory tested and shipped with installed jumpers. The user may reposition these jumpers to customize BCC functionality. There are several connectors associated with the PFB. Through these connectors you power, communicate with, and access the available BCC features (connector P10 is not used when the M68331BCC is used with the PFB). Figure 2-3 shows the locations of switches, jumpers and connectors on the PFB.

The PFB is shipped from the factory with the M68331BCC Business Card Computer (BCC) installed in its expansion connectors. Configure the BCC in the desired mode of operation per paragraphs 2.4.1 and 2.4.2. If you remove the BCC, take care to fully reinsert it when installing it back on the PFB.

The PFB contains ABORT and RESET switches, a power connector, a battery backup connector, and 14 jumpers (see Figure 2-1).

The PFB has two pairs of I.C. sockets (U1/U2, and U3/U4) for user-supplied memory devices (RAM or EPROM). See table below for RAM/EPROM capabilities. If EPROM is installed on the PFB, it may be used instead of the BCC on-board EPROM. PFB RAM, expands BCC on-board RAM. The PFB jumper headers must be configured to utilize the user installed RAM or EPROM. To configure these jumpers, follow the instructions in the paragraphs that follow.

Table 2-1. Socket to Memory Device Compatibility

PFB SOCKET	MEMORY DEVICE
U1/U3	32k bytes of RAM per socket
U2/U4	32k bytes of RAM per socket, 32k bytes of EPROM per socket, or 64k bytes of EPROM per socket

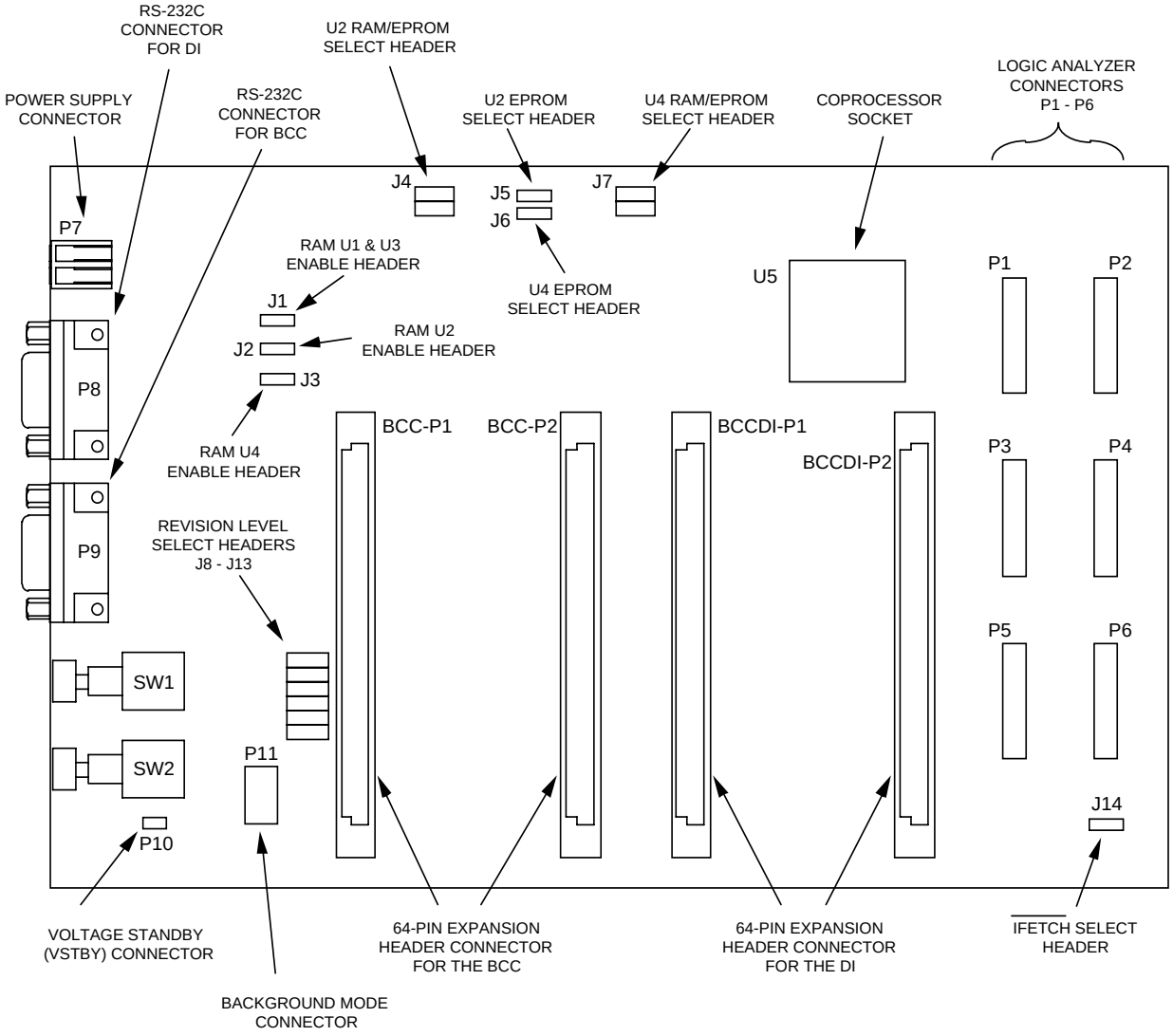
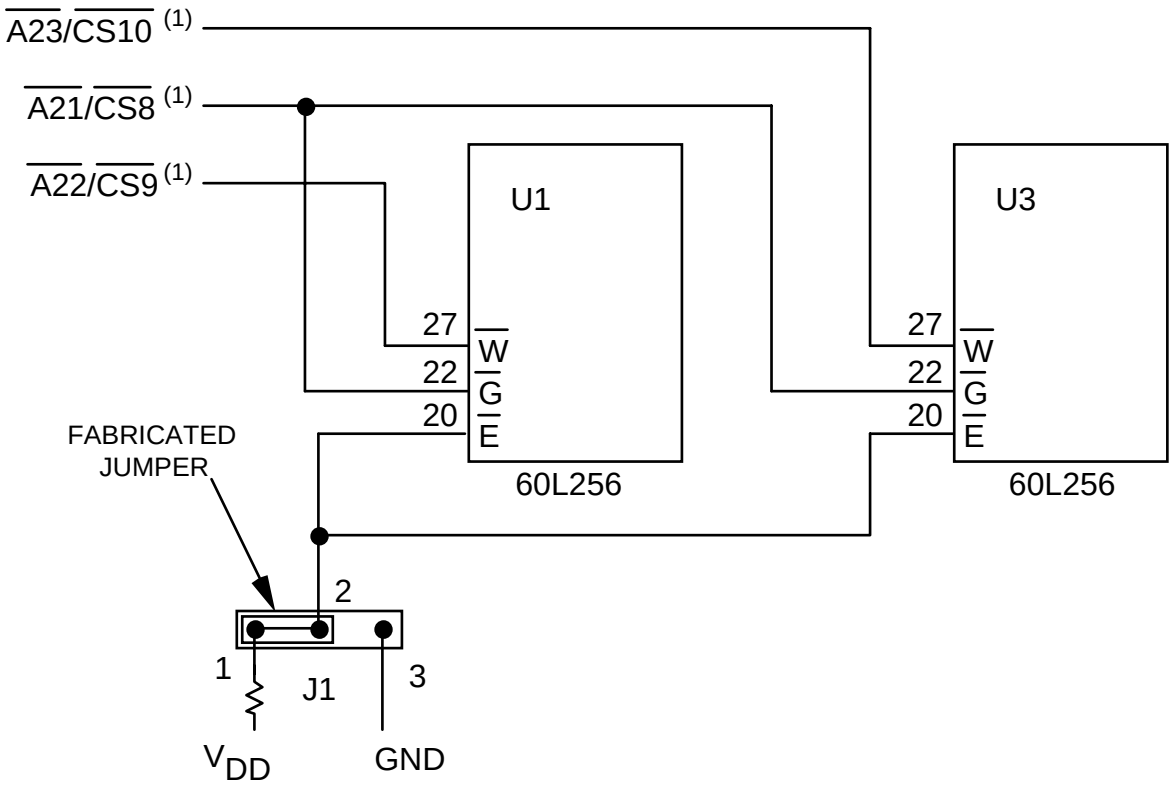


Figure 2-3. PFB Jumper Header and Connector Location Diagram

2.3.2.1 U1 and U3 Enable Header (J1)

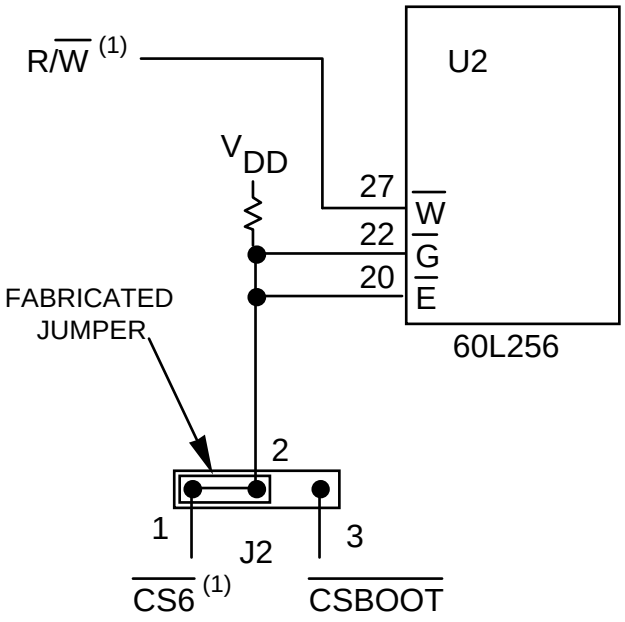
Sockets U1 and U3 are for user-supplied 32k x 16 bit RAM. Use jumper header J1 to enable this additional memory. The PFB is shipped from the factory with a jumper installed on pins 1 and 2; U1 and U3 disabled (shown below). To enable RAM in sockets U1 and U3, move the fabricated jumper to pins 2 and 3. Refer to the PFB schematic diagram for more detail on RAM enable select signal wiring.



1. $\overline{A23}$ /$\overline{CS10}$, $\overline{A21}$ /$\overline{CS8}$, and $\overline{A22}$ /$\overline{CS9}$ are default signals. Jumpers J8, J12, and J13, respectively, can select the alternates.

2.3.2.2 U2 Enable Header (J2)

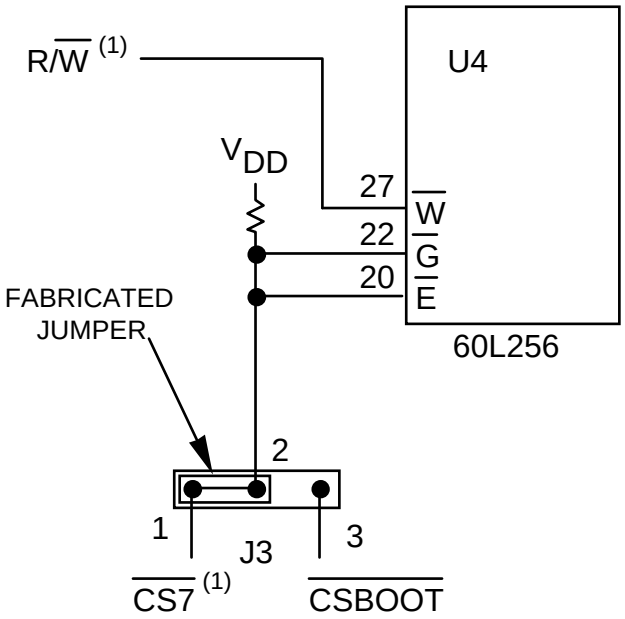
The user may install a user-supplied 32k byte RAM, 32k byte EPROM, or 64k byte EPROM in socket U2. Use jumper header J2 to select between $\overline{CS6}$ and $\overline{CSBOOT}$ chip selects for the memory device installed in U2. The PFB is shipped from the factory with a jumper installed on pins 1 and 2 (shown below). If you install an EPROM in U2 and desire a program to execute during reset or power-up, place the fabricated jumper across pins 2 and 3 and disable the EPROM on the BCC (refer to paragraph 2.3.1.4). Jumper headers J2 and J3 must be configured the same for EPROMs installed in locations U2 and U4. Refer to the PFB schematic diagram for more detail on chip enable signal wiring.



1. R/$\overline{W}$ and $\overline{CS6}$ are default signals. Jumpers J4 and J11, respectively, can select the alternates.

2.3.2.3 U4 Enable Header (J3)

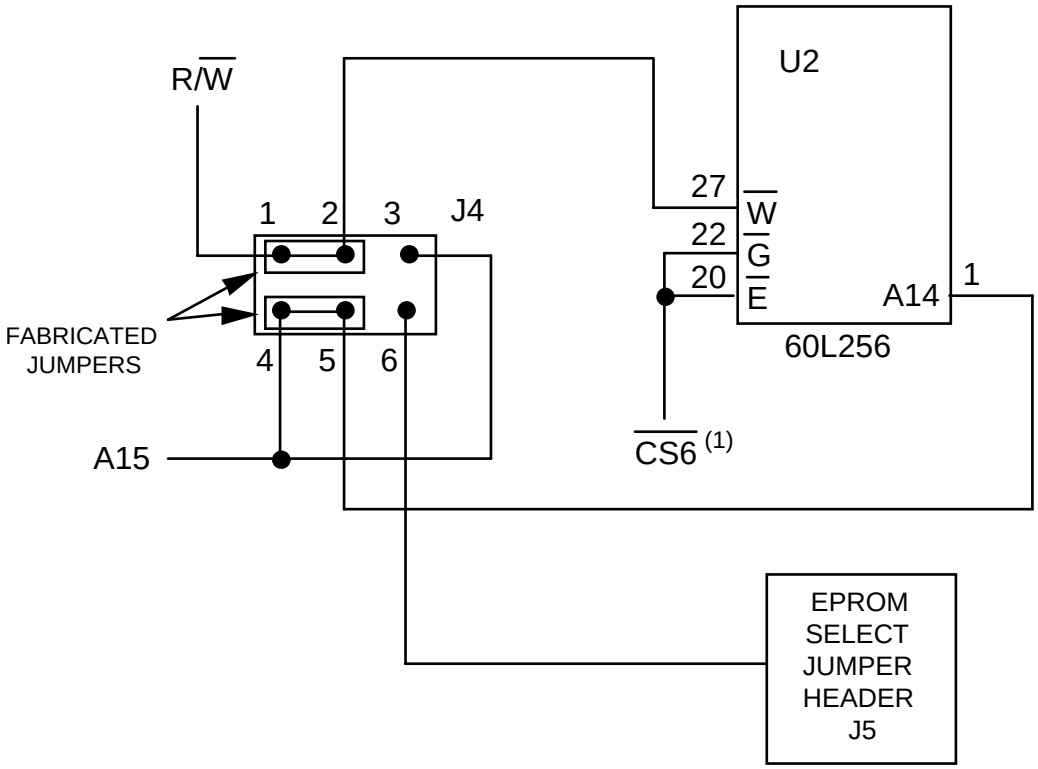
The user may install a user-supplied 32k byte RAM, 32k byte EPROM, or 64k byte EPROM in socket U4. Use jumper header J3 to select between $\overline{CS7}$ and $\overline{CSBOOT}$ chip selects for the memory device installed in U4. The PFB is shipped from the factory with a jumper installed on pins 1 and 2 (shown below). If you install an EPROM in U4 and desire a program to execute during reset or power-up, place the fabricated jumper across pins 2 and 3 and disable the EPROM on the BCC (refer to paragraph 2.3.1.4). Jumper headers J2 and J3 must be configured the same for EPROMs installed in locations U2 and U4. Refer to the PFB schematic diagram for more detail on chip enable signal wiring.



1. R/$\overline{W}$ and $\overline{CS7}$ are default signals. Jumpers J7 and J10, respectively, can select the alternates.

2.3.2.4 U2 RAM/EPROM Select Header (J4)

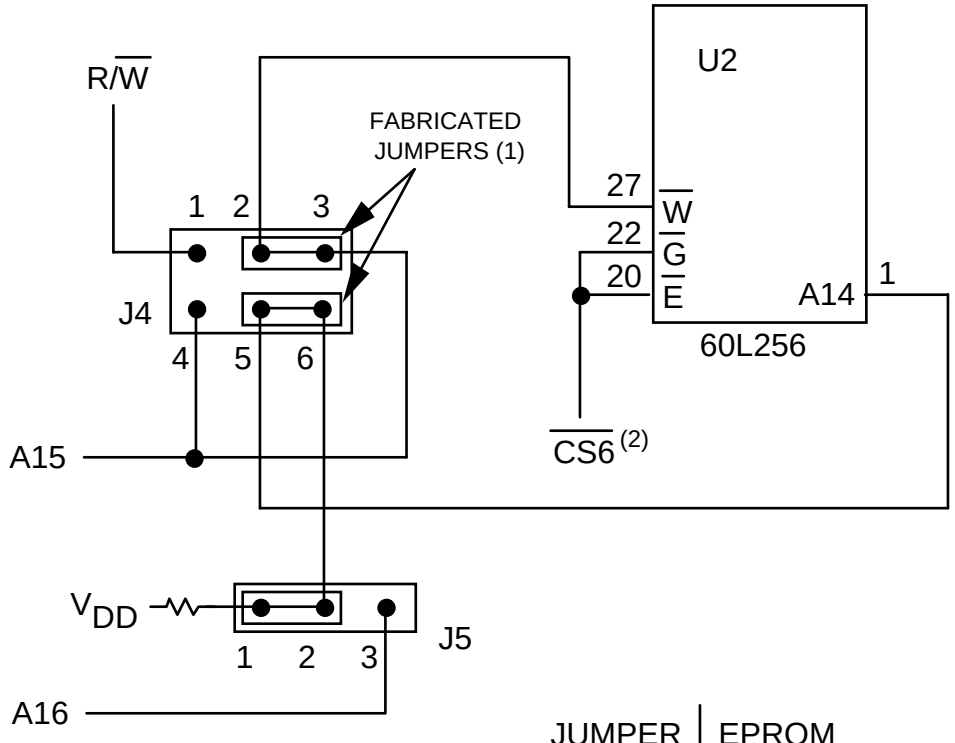
Use jumper header J4 to select RAM or EPROM as the memory device type. The PFB is shipped from the factory with fabricated jumpers installed on pins 1 and 2 and pins 4 and 5 (shown below). This is correct if RAM is installed in U2. When an EPROM is installed in U2, place the jumpers on pins 2 and 3 and pins 5 and 6 (refer to paragraph 2.3.2.5). Refer to the PFB schematic diagram for more detail on RAM/EPROM type select signal wiring.



1. $\overline{CS6}$ is the default signal. Jumpers J2 and J11 can select the alternate.

2.3.2.5 U2 EPROM Select Header (J5)

Use jumper header J5 (shown below) when an EPROM is to be installed in U2. Two types of EPROM can be used: 27C256 and 27C512. For a 27C256, install a fabricated jumper between pins 1 and 2; for a 27C512, install a jumper between pins 2 and 3. Refer to Paragraph 2.3.2.4 explains jumper header J4 configuration to match memory device in location U2. Locations U2 and U4 must have the same type of memory devices; either both RAM or both EPROM. Refer to the PFB schematic diagram for more detail on EPROM type selection signal wiring.

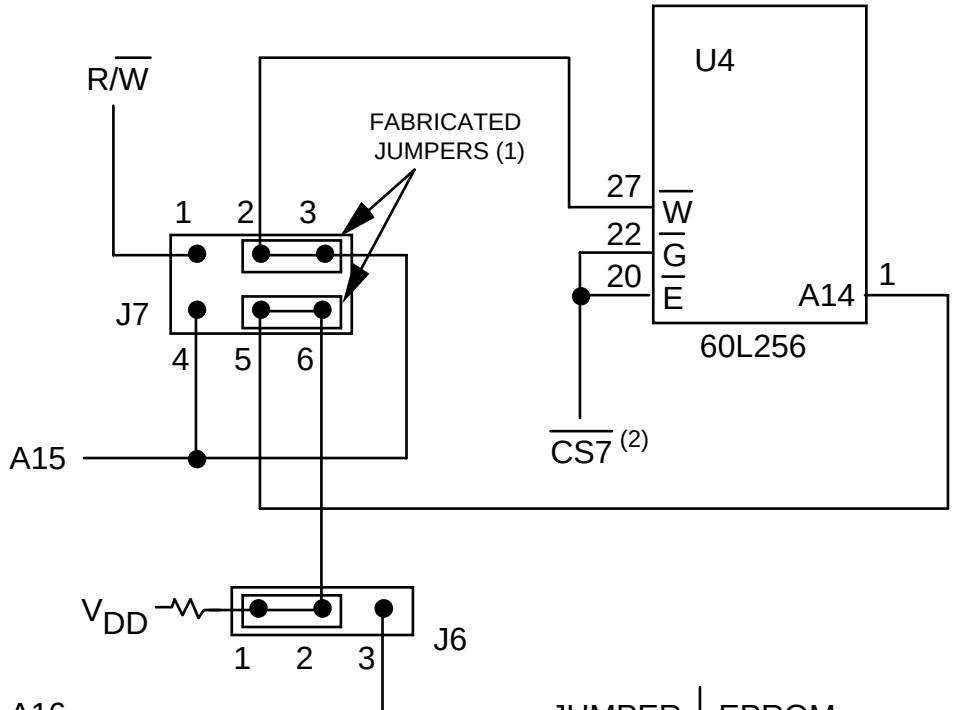


JUMPER	EPROM
J5, 1 & 2	27C256
J5, 2 & 3	27C512

1. Above diagram shows J4 in the EPROM select mode.
2. $\overline{CS6}$ is the default signal. Jumpers J2 and J11 can select the alternate.

2.3.2.6 U4 EPROM Select Header (J6)

Use jumper header J6 (shown below) when an EPROM is to be installed in U4. Two types of EPROM can be used: 27C256 and 27C512. For a 27C256, install a fabricated jumper between pins 1 and 2; for a 27C512, install a fabricated jumper between pins 2 and 3. Paragraph 2.3.2.7 explains jumper header J7 configuration to match memory device in location U4. Locations U4 and U2 must have the same type of memory devices; either both RAM or both EPROM. Refer to the PFB schematic diagram for more detail on EPROM type selection signal wiring.

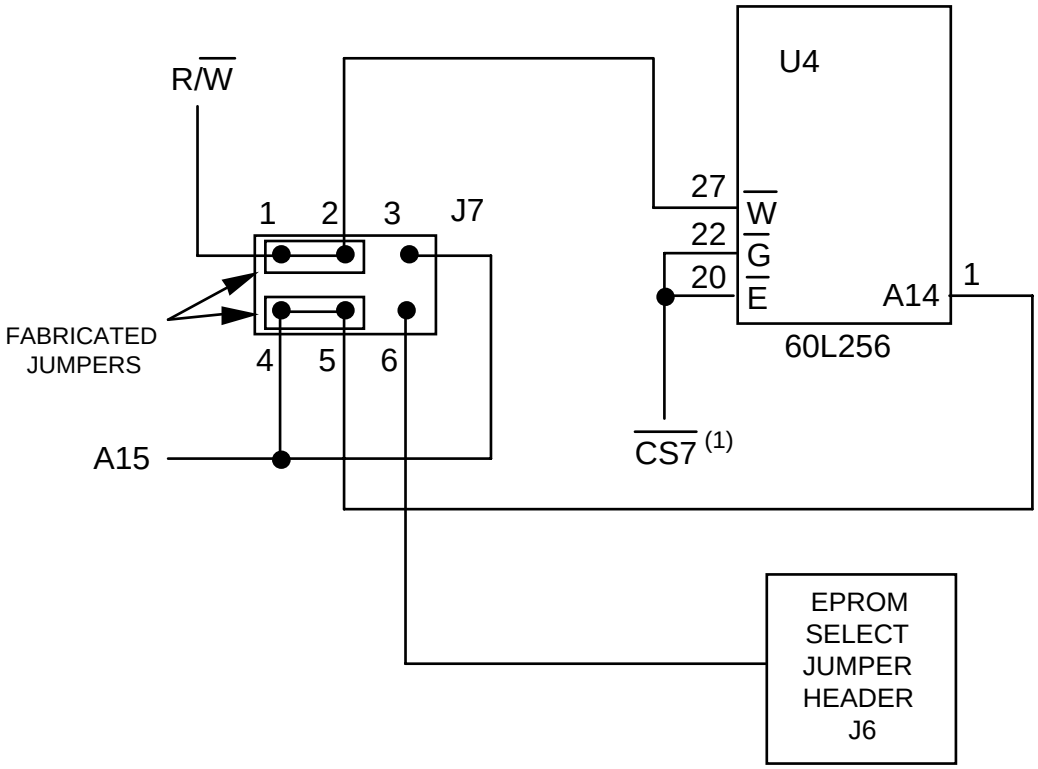


JUMPER	EPROM
J6, 1 & 2	27C256
J6, 2 & 3	27C512

1. Above diagram shows J7 in the EPROM select mode.
2. $\overline{CS7}$ is the default signal. Jumpers J3 and J10 can select the alternate.

2.3.2.7 U4 RAM/EPROM Select Header (J7)

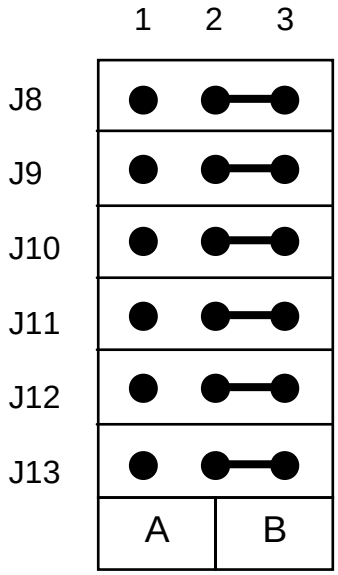
Use jumper header J7 to select RAM or EPROM as the memory device type. The PFB is shipped from the factory with fabricated jumpers installed on pins 1 and 2 and pins 4 and 5 (shown below). This is correct if RAM is installed in U4. When an EPROM is installed in U4, place the jumpers on pins 2 and 3 and pins 5 and 6 (refer to paragraph 2.3.2.6). Refer to the PFB schematic diagram for more detail on RAM/EPROM type select signal wiring.



1. $\overline{CS7}$ is the default signal. Jumpers J3 and J10 can select the alternate.

2.3.2.8 Revision Level Select Headers (J8 - J13)

Jumpers J8 through J13 (shown below) configure the PFB for either a revision A, B, or C M68331BCC. The PFB is shipped from the factory configured for operation with a B or C revision of the BCC via cut-trace shorts between pins 2 and 3 of J8 - J13. The M68331BCC does not have an A revision so do not change the factory settings.

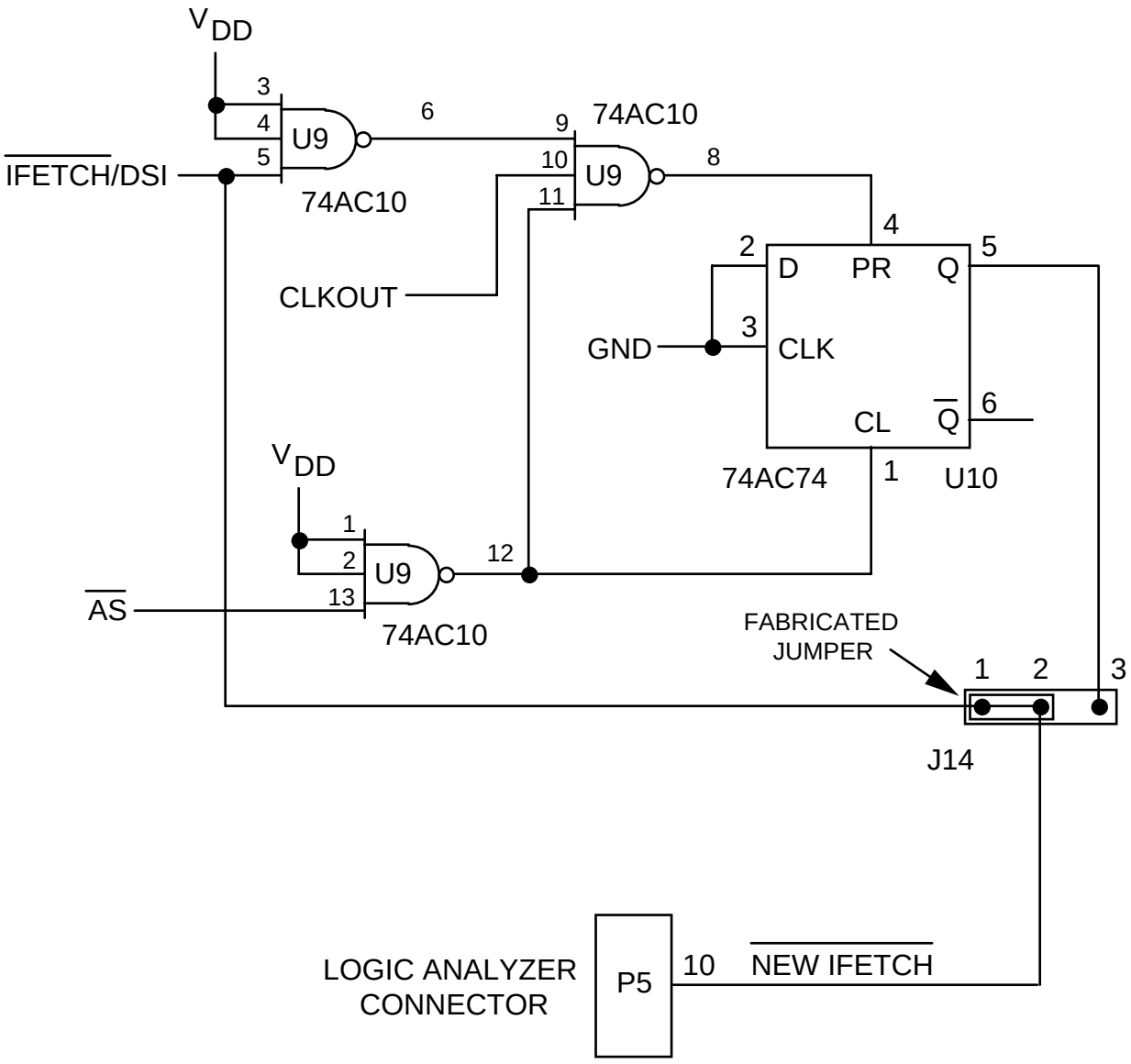


NOTE

If the cut-trace shorts on jumper headers J8 through J13 are cut, user-supplied fabricated jumpers must be installed on J8 through J13 (pins 2 and 3) to return the BCC to its factory setting.

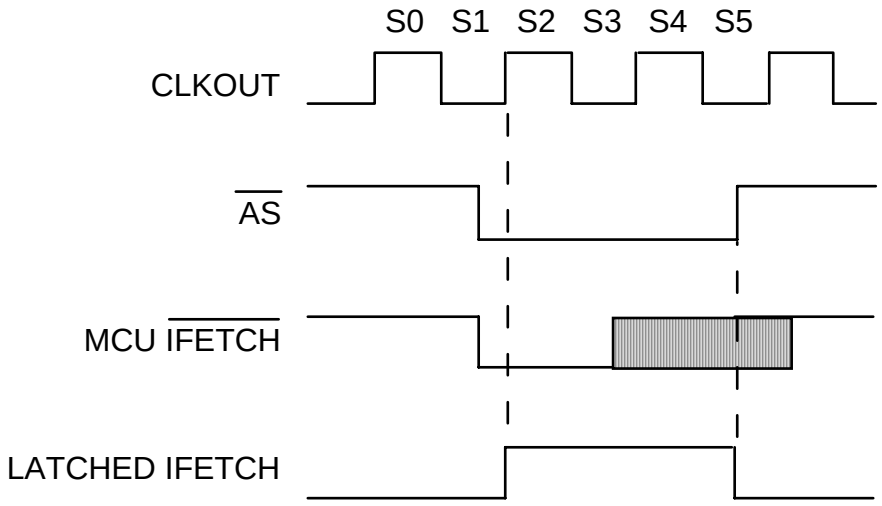
2.3.2.9 $\overline{\text{IFETCH}}$ Select Header (J14)

Use jumper header J14 (shown below) to select the latched IFETCH or MCU $\overline{\text{IFETCH}}$ signal. The PFB is shipped from the factory with fabricated jumpers installed between pins 1 and 2 of J14; this routes $\overline{\text{IFETCH}}$ directly from the MC68331 MCU to the logic analyzer connector, P5, pin 10. When the jumper is installed on pins 2 and 3, a latched IFETCH signal is routed to the logic analyzer connector, P5, pin 10. Refer to the PFB schematic diagram for more detail on $\overline{\text{IFETCH}}$ signal wiring.



Latched IFETCH is an active high signal (opposite of $\overline{\text{IFETCH}}$) that signals the logic analyzer to decode an instruction from the data bus. The signal characteristics are:

- When MCU $\overline{\text{IFETCH}}$ goes low, latched IFETCH goes high on the rising edge of CLKOUT (beginning of S2).
- Latched IFETCH goes low (inactive) 2 - 10 nanoseconds after address strobe ($\overline{\text{AS}}$) goes high.
- The timing diagram for these signals is:



2.3.2.10 Coprocessor Socket (U5)

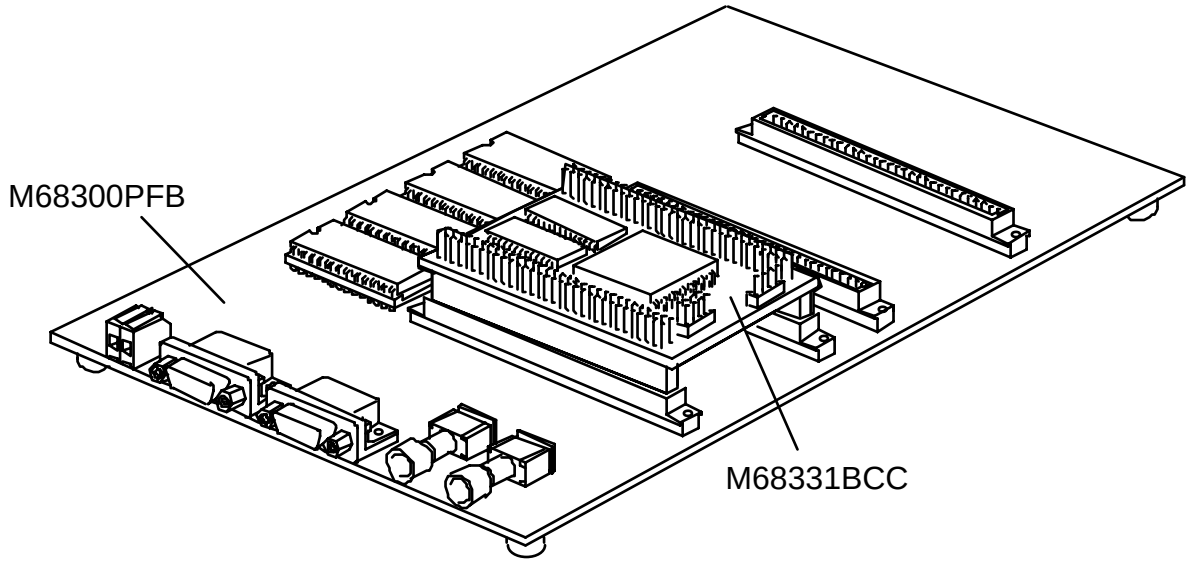
Use the 68-pin gate array (PGA) socket (U5) for installing a user-supplied MC68881 or MC68882 floating point coprocessor, so floating point instructions execute concurrently with MPU integer instructions. Execution of floating point instructions requires user supplied interface routines. Addition of the floating-point coprocessor substantially decreases processor instruction execution time for floating-point arithmetic.

2.4 INSTALLATION INSTRUCTIONS

The following paragraphs describe EVK interconnections, expansion connector pin assignments, target system dimensions, power supply interconnections, the serial communication connector and cable, and logic analyzer connectors.

2.4.1 BCC – PFB Interconnection

For BCC to PFB interconnection the BCC mounts on the PFB (shown below). This configuration is the EVS standalone configuration. The EVK may be used to evaluate the MCU and verify functionality of user developed code. For this configuration match BCC P1 to PFB BCC-P1 and BCC P2 to PFB BCC-P2. Figure 2-4 illustrates the expansion header pin assignments for the BCC and PFB. Table 2-2 shows revision level compatibility between the BCC and PFB.



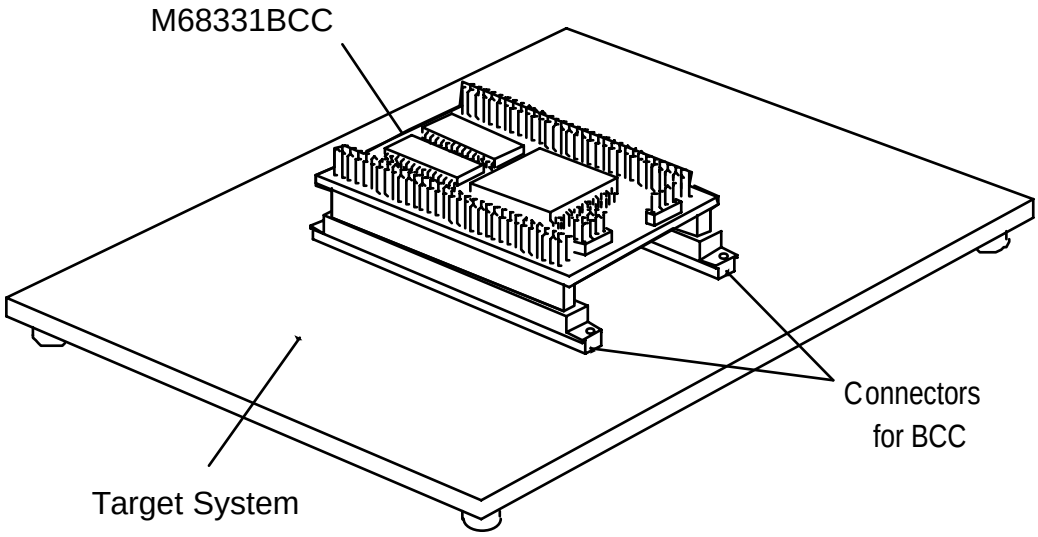
BCC to PFB Interconnection

Table 2-2. BCC/PFB Compatibility

BCC REV. Level	PFB REV. Level
A	A
B	B
C	B

2.4.2 Target System – BCC Interconnection

For target-system-to-BCC interconnection the BCC mounts on the target system as shown below. Use this configuration to evaluate your hardware design. The 64-pin expansion connectors provide access to most of the MC68331 MCU device pins. Figure 2-4 illustrates the expansion header pin assignments for the BCC. Figure 2-5 shows physical dimension requirements for installing the BCC on a target system.



Target System to BCC Interconnection

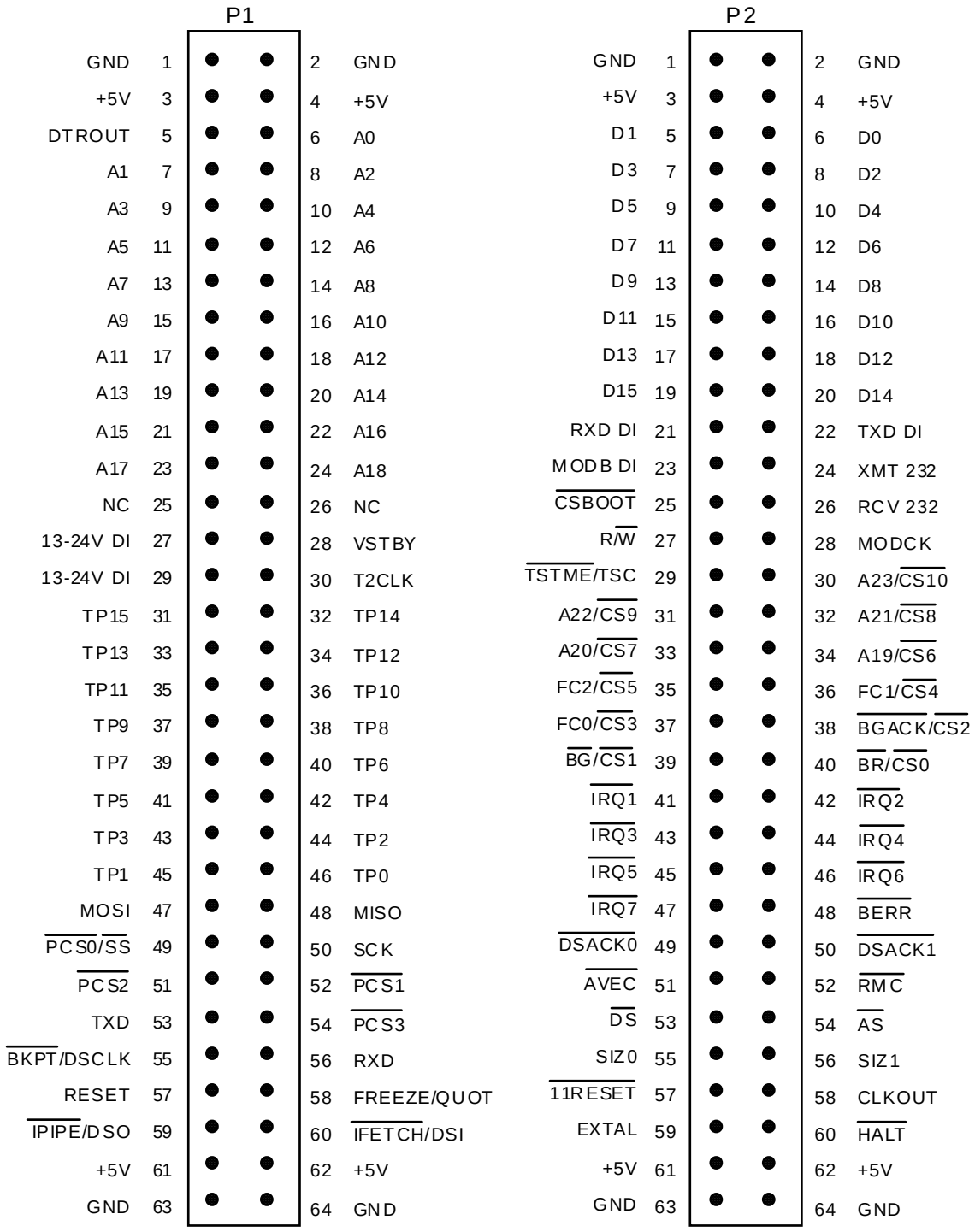


Figure 2-4. Expansion Connector Pin Assignments

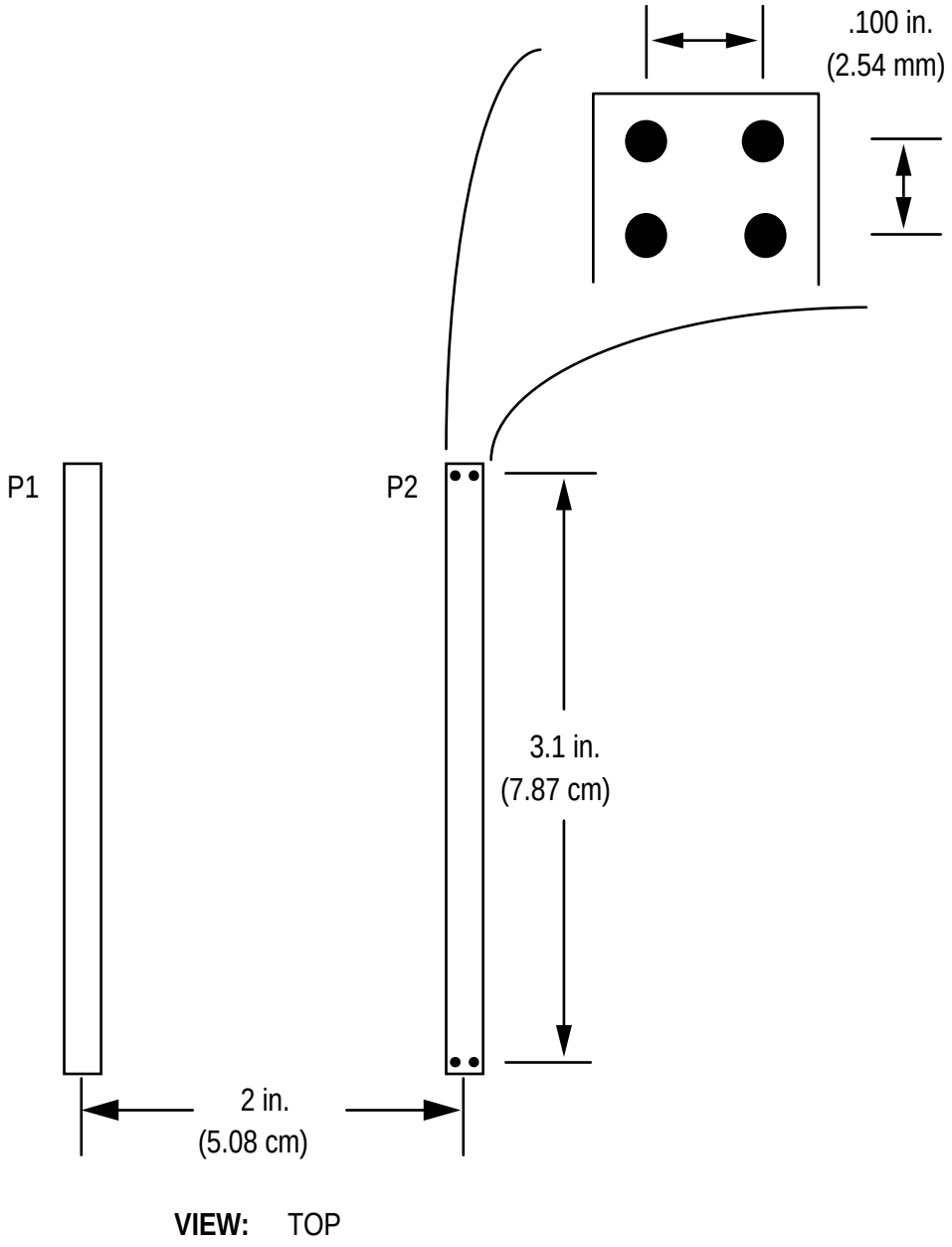
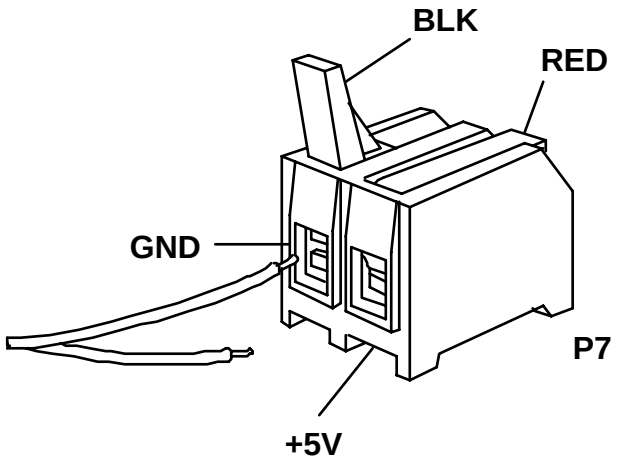


Figure 2-5. Target System Connector Dimension Requirements

2.4.3 Power Supply – EVK Connection

The EVK requires a +5 Vdc @ 200 milliamps power supply for operation.

Use PFB connector P7 to connect power to the EVK. Contact 1 is ground; black lever. Contact 2 is VDD (+5 volts); red lever. Use 20 or 22 AWG wire for power connections. For each wire, trim back the insulation 1/4 in. (.635 cm), lift the appropriate lever of P7 to release tension on the contacts, then insert the bare wire into P7 and close the lever.



CAUTIONS

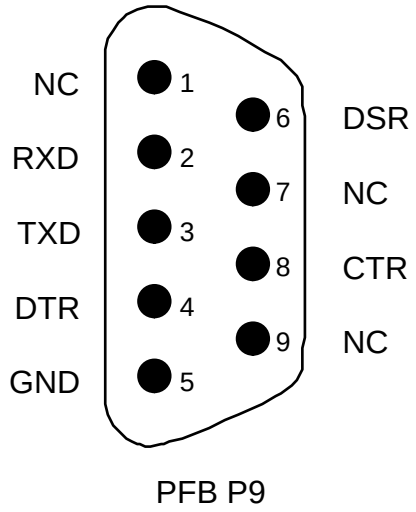
Do not use wire larger than 20 AWG in connector P7. Such wire could damage the connector.

Turn off PFB power when installing the BCC or removing the BCC from the PFB. Sudden power surges could damage EVK integrated circuits.

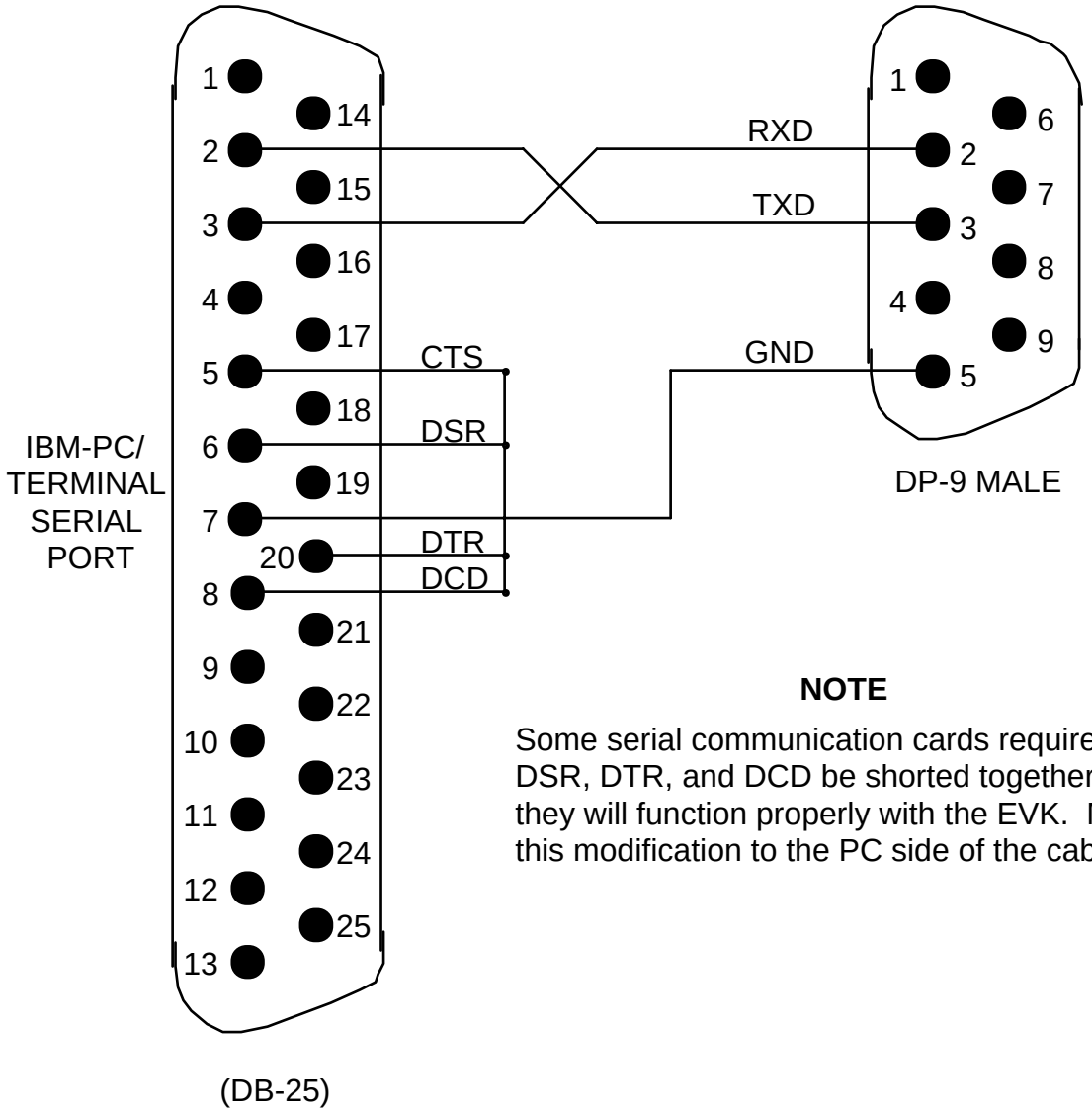
2.4.4 Terminal – EVK Connection

The PFB has two 9-pin serial ports (P8 and P9) for connection to a terminal or host computer (with terminal emulation). Use P9 to communicate with the BCC from a terminal or host computer. P8 is not used with the EVK.

To connect an RS-232C compatible terminal or host computer, use a user-supplied cable assembly, as shown in Figure 2-6. Connect one end of the cable assembly to PFB connector P9 (shown below). Connect the other end of the cable assembly to a user-supplied terminal or host computer. Refer to Chapter 5 for connector pin assignments and signal descriptions of PFB connector P9.



Terminal/PC I/O Port



NOTE

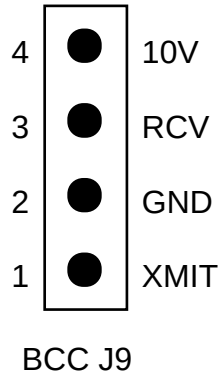
Some serial communication cards require CTS, DSR, DTR, and DCD be shorted together before they will function properly with the EVK. Make this modification to the PC side of the cable.

Figure 2-6. Terminal/PC Cable Diagram for PFB P9

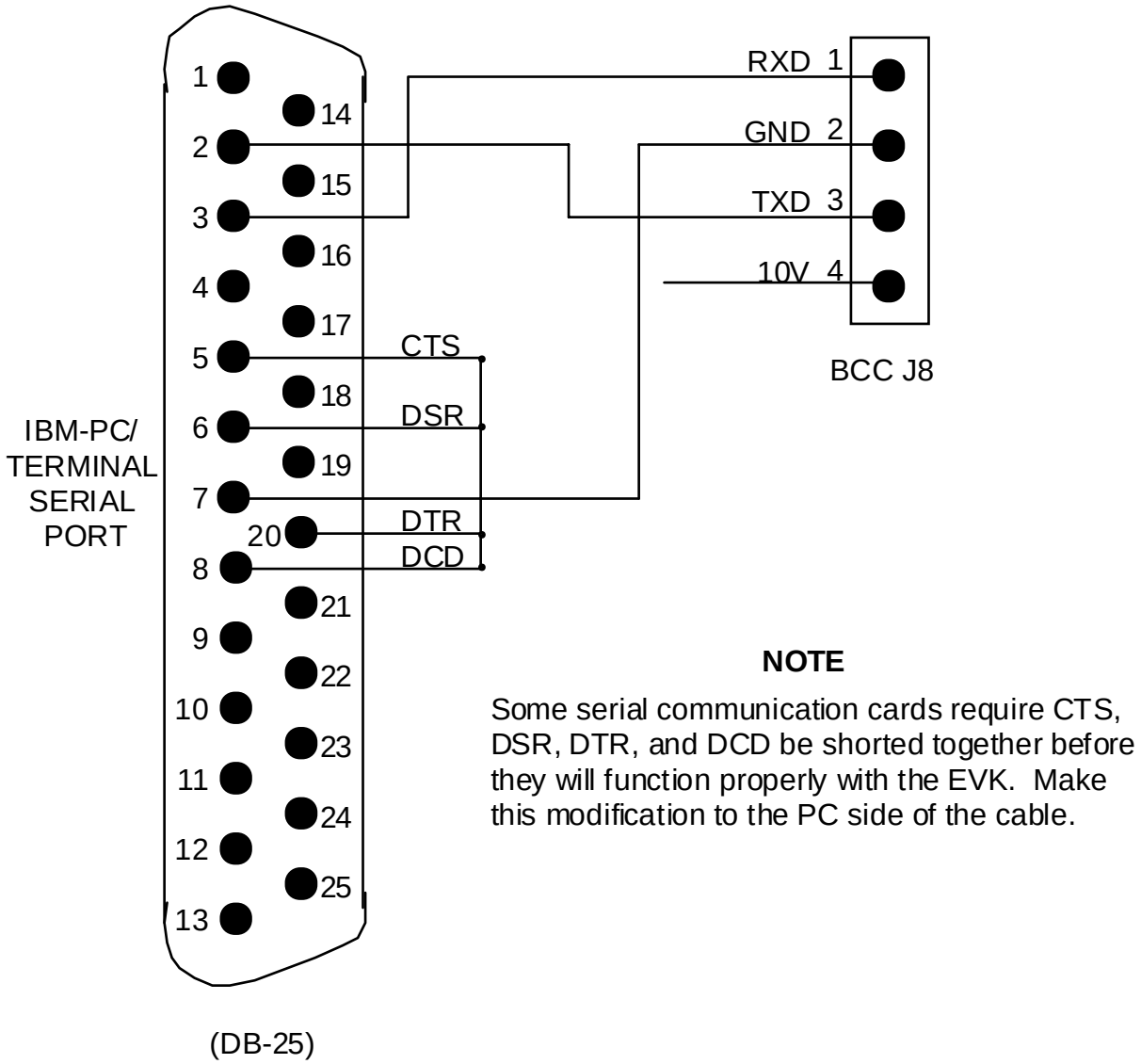
2.4.5 Terminal – BCC Connection

The BCC has a 4-pin serial port (P4) for connection to a terminal or host computer (with terminal emulation). Use P4 to communicate with the BCC from a terminal or host computer.

To connect an RS-232C compatible terminal or host computer, use a user-supplied cable assembly, as shown in Figure 2-7. Connect one end of the cable assembly to BCC connector P4 (shown below). Connect the other end of the cable assembly to a user-supplied terminal or host computer. Refer to Chapter 5 for connector pin assignments and signal descriptions of BCC connector P4.



Terminal/PC I/O Port



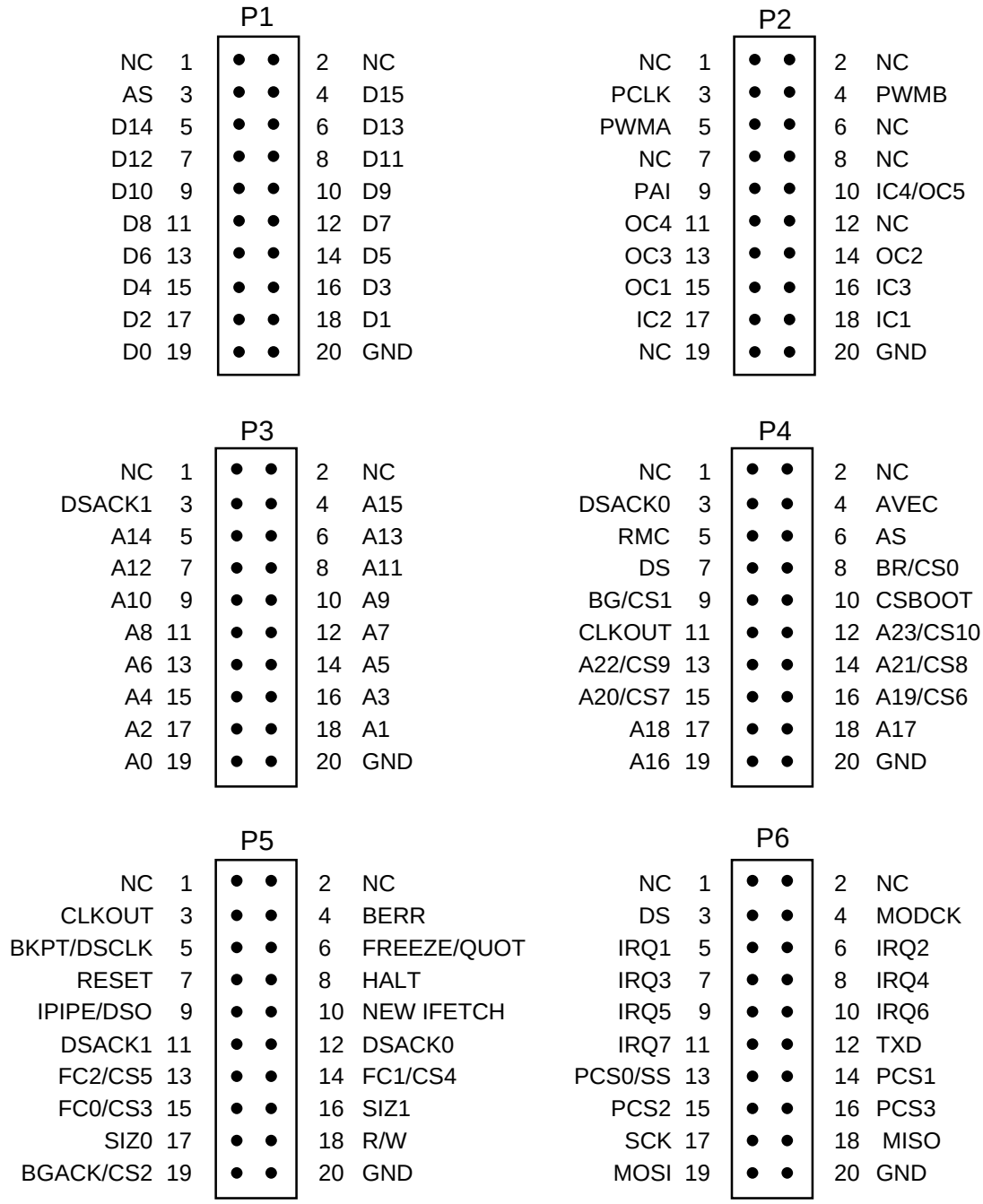
NOTE

Some serial communication cards require CTS, DSR, DTR, and DCD be shorted together before they will function properly with the EVK. Make this modification to the PC side of the cable.

Figure 2-7. Terminal/PC Cable Diagram for BCC P4

2.4.6 Logic Analyzer – EVK Connection

A logic analyzer can be connected to the PFB, as an aid to debugging target system hardware and software. Logic analyzer connector pin assignments are shown below. The MC68331 MCU consists of a general purpose timer (GPT) instead of the time processor unit (TPU) found in the MC68332 MCU. Therefore, the pinout names for connector P2 on the PFB silk-screen are wrong. Use the figure below for P2 pinouts when connecting a logic analyzer to the EVK. For EVK connectors P1 - P6 signal descriptions, refer to Chapter 5.



CHAPTER 3
OPERATING INSTRUCTIONS

3.1 INTRODUCTION

This chapter provides general information, control switch descriptions, limitations, operating procedure, and monitor description for the EVK debug monitor (CPU32Bug). MCU assembling/disassembling and EVK downloading procedures are also provided. This information is in this order:

- General information
- Control switch descriptions
- Limitations
- Operating procedure
- Monitor description
- Assembling/disassembling procedures
- Downloading procedures

The EVK is factory tested for target system evaluation.

The monitor is the resident firmware (CPU32Bug) for the EVK, which provides a self-contained operating environment. The monitor interacts with the user through commands entered at a terminal. These commands perform functions such as memory display or modification, MCU internal register display or modification, program execution under various levels of control, and control access to various I/O peripherals connected to the EVK.

3.2 GENERAL INFORMATION

The EPROMs on the BCC contain the M68CPU32BUG debug monitor program (hereafter referred to as CPU32Bug). CPU32Bug is a software tool for evaluating and debugging systems built around the MC68331 MCU. CPU32Bug allows loading, debugging, and executing of user programs. Various CPU32Bug routines that handle I/O, data conversion, timer, and string functions are available to user programs through system calls. For a detailed description of the CPU32Bug functions, refer to M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1.

CPU32Bug consists of:

- Memory and register display and modification commands
- Breakpoint capabilities
- System calls
- Diagnostic commands
- A single-line assembler/disassembler
- A parameter area for user customization

There are two modes of operation in the CPU32Bug monitor; the debugger mode and the diagnostic mode. When the user is in the debugger directory the prompt **CPU32Bug>** appears, and the user has access to the debugger commands (refer to paragraph 3.5). When the user is in the diagnostic mode the prompt **CPU32Diag>** appears, and the user has access to the diagnostic commands (refer to paragraph 3.5.4). These modes are also called directories.

CPU32Bug is command-driven and performs various operations in response to user commands entered at the keyboard. CPU32Bug executes entered commands; upon completion the prompt reappears. However, if a command is entered which causes execution of user target code (i.e., GO), control may or may not return to CPU32Bug. This depends upon the user program function. Entering the help (HE) command provides a list of all possible commands and their structure.

3.3 CONTROL SWITCHES

The two EVK switches (PFB switches SW1 and SW2) control the reset and abort functions. Table 3-1 lists these switches by name, description, and function.

Table 3-1. EVK Control Switches

Name	Description And Function
BCC RESET Switch (PFB SW1)	The PFB reset switch returns the BCC to a known state. Reset causes a total BCC initialization equivalent to the power-up sequence. All static variables are restored to their default states. The serial ports are reset to their default state. The breakpoint table is cleared. The offset registers are cleared. The target registers are invalidated. Input and output character queues are cleared. On-board devices (timer, serial ports, etc.) are reset. Use reset if the MC68331 MCU halts, for example, after a halt monitor fault, or if the user program environment is lost (vector table is destroyed, etc.).
ABORT Switch (PFB SW2)	The PFB abort switch terminates all in-process instructions. When the abort switch is pressed while running target code, a snapshot of the processor state is captured and stored in the target registers. For this reason abort is appropriate when terminating a user program that is being debugged. Use abort to regain control if the program gets caught in a loop, etc. The target PC, stack pointers, etc. help pinpoint malfunctions. Abort generates a non-maskable, level-seven interrupt. The target registers reflect the MC68331 MCU device state at the time of an abort and are displayed on the terminal screen. Any breakpoints installed in the user code are removed and the breakpoint table remains intact. Control is then returned to the user.

3.4 LIMITATIONS

CPU32Bug requires these system resources to operate properly: several chip selects, level 7 interrupt, periodic interrupt timer (for timer system calls; SYSCALL), and system exception vectors.

3.4.1 Chip Select Usage

The MC68331 MCU has chip select signals that enable peripheral devices. The BCC requires some of these chip selects for BCC operation making them unavailable to the user. Do not remove the chip selects used by the BCC, or CPU32Bug will not operate. In addition to the chip selects employed on the BCC, other chip selects are used elsewhere in the M68331EVK (refer to Tables 3-2 and 3-3).

Depending on the user's environment, any or all of the chip selects may be re-configured for an alternate function (i.e., as I/O or address lines). Chip select pins used by the BCC and EVK cannot be used in their alternate capacities.

Although a chip select (CS4 on Rev. B and Rev. C BCCs) is dedicated for the ABORT switch, the pin is not used. Instead the chip select decodes the interrupt acknowledge (IACK) cycle in response to a level 7 interrupt generated by the ABORT switch. Refer the System Integration Module User's Manual (SIM32UM/AD) for more information.

When the BCC is not mounted on the M68300PFB Platform Board (PFB), these chip selects are available to the user. Also if the PFB RAM/EPROM socket pairs (U1/U3, U2/U4) and coprocessor socket U5 are not populated, the chip selects are available to the user.

Table 3-2. BCC Rev. B(1) Chip Selection Summary

Signal	Board/Chip	Description	Memory Type
CSBOOT	BCC U2	CPU32Bug EPROM	
CS0	BCC U1	write enable for MSB=UPPER=EVEN	RAM
CS1	BCC U3	write enable for LSB=LOWER=ODD	RAM
CS2	BCC U3/U1	read enable for MSB/LSB=BOTH	RAM
CS3	<unused>		
CS4	PFB	ABORT push-button autovector	
CS5	PFB U5	chip enable for MC68881/882. cut/-jump U5-J3 from CS2 to CS5 required.	
CS6	PFB U2	read enable for LSB=LOWER=ODD	RAM/EPROM
CS7	PFB U4	read enable for MSB=UPPER=EVEN	RAM/EPROM
CS8	PFB U1/U3	read enable for MSB/LSB=BOTH	RAM
CS9	PFB U1	write enable for LSB=LOWER=ODD	RAM
CS10	PFB U3	write enable for MSB=UPPER=EVEN	RAM

(1) There is no Rev. A for the M68331BCC board.

Table 3-3. BCC Rev. C Chip Selection Summary

Signal	Board/Chip	Description	Memory Type
CSBOOT	BCC U3	CPU32Bug EPROM for MSB=UPPER=EVEN	
CSBOOT	BCC U4	CPU32Bug EPROM for LSB=LOWER=ODD	
CS0	BCC U1	write enable for MSB=UPPER=EVEN	RAM
CS1	BCC U2	write enable for LSB=LOWER=ODD	RAM
CS2	BCC U3/U1	read enable for MSB/LSB=BOTH	RAM
CS3	<unused>		
CS4	PFB	ABORT push-button autovector	
CS5	PFB U5	chip enable for MC68881/882. cut/-jump U5-J3 from CS2 to CS5 required.	
CS6	PFB U2	read enable for LSB=LOWER=ODD	RAM/EPROM
CS7	PFB U4	read enable for MSB=UPPER=EVEN	RAM/EPROM
CS8	PFB U1/U3	read enable for MSB/LSB=BOTH	RAM
CS9	PFB U1	write enable for LSB=LOWER=ODD	RAM
CS10	PFB U3	write enable for MSB=UPPER=EVEN	RAM

3.4.2 Other MCU Resources Used by CPU32Bug

Avoid writing the value zero to bit 7 of the port F pin assignment register (PFPAR); such a value disables the ABORT switch.

The software watchdog timer is disabled via a write-once register (SYPCR) during power-up or reset, so the software watchdog timer cannot be used or re-enabled by the user unless the user modifies the SYPCR_OR and SYPCR_AND parameters. Modification of the SYPCR_OR and SYPCR_AND parameters is detailed in Appendix C of the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1.

The monitor uses the system exception vectors, so they are unavailable to the BCC user. The monitor debug exception vectors are listed in Table 3-4. The associated debugger facilities (breakpoints, trace mode, etc.) will not operate if the vector offsets in the target program vector table are changed.

Table 3-4. CPU32Bug Exception Vectors

Vector Number	Offset	Exception	CPU32Bug Commands
4	$10	Illegal	Instruction breakpoints (Used by GO, GN, GT)
9	$24	Trace	T, TC, TT
31	$7C	Level 7 interrupt	ABORT switch
47	$BC	TRAP #15	System calls (see Chapter 5 of the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1)
66	$108	User Defined	Timer Trap #15 Calls ($4X)

Any change in the MC68331 MCU device clock speed causes a corresponding change in the SCI baud rate. The operational speed of the MCU is determined by the clock and the synthesizer control register value (SYNCR) or by an external clock signal applied to the EXTAL pin of the MCU. The SCI baud rate is then set based on this system clock frequency. If changes are made to the MCU system clock frequency, changes must be made to the customization parameter area (FCRYSTAL or FEXTAL) so the correct baud rate can be calculated for SCI communications by CPU32Bug. See M68CPU32BUG Debug Monitor User's Manual (M68CPU32BUG/AD1), Appendix C for details.

Additionally, CPU32Bug writes a one (1) to the module mapping (MM) bit of the module control register (MCR). This configures the register block to start at address $FFF000. As the MM bit is a write-once bit, the user cannot clear it to move the register block to low memory ($7FF000). The user can move the register block by modifying the MCR_AND parameter detailed in Appendix C of the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1.

3.5 OPERATING PROCEDURE

A Power On Reset (POR) occurs when the user applies power to the EVK. This POR resets the MCU and user I/O port circuitry, and passes processing control to the monitor program. All MC68331 MCU registers are set to their reset state during monitor power-up.

The input serial format for the BCC terminal I/O port must be configured for 8 data bits, 1 stop bit, no parity, and 9600 baud.

The terminal then displays this message:

```
CPU32Bug Debugger/Diagnostics - Version X.XX
    (C) Copyright 1991 by Motorola Inc.

CPU32Bug>
```

where:

X.XX is the software revision level

After initialization or return of control to the monitor, the terminal displays the prompt "CPU32Bug>" and waits for a response. If an incorrect response is entered, the terminal displays "Invalid command" followed by the prompt "CPU32Bug>".

CPU32Bug waits for a command line input from the user terminal. When a proper command is entered, the operation continues in one of two basic modes. If the command causes execution of a user program, the monitor may or may not be reentered, depending upon the desire of the user. For the alternate case, the command is executed under the control of the monitor, and the system returns to a waiting condition after the command is completed. During command execution, additional user input may be required, depending on the command function.

The user can use any of the commands supported by the monitor. A standard input routine controls the BCC operation while the user types a command line. Command processing begins only after the command line has been terminated by pressing the keyboard carriage return <CR> key.

3.6 MONITOR DESCRIPTION

CPU32Bug performs various operations in response to user commands entered at the keyboard. When the debugger prompt **CPU32Bug**> appears on the terminal screen, the debugger is ready to accept commands.

As the command line is entered it is stored in an internal buffer. Execution begins only after the carriage return is entered. This lets the user correct entry errors using the control characters described in the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1.

After the debugger executes a command it returns with the **CPU32Bug**> prompt. However, if the entered command causes execution of user target code, (i.e., **GO**), then control may or may not return to the debugger. This depends on the user program function. For example, if a breakpoint is specified, control returns to the debugger when the breakpoint is encountered during user program execution. The user program also returns control to the debugger by means of the TRAP #15 system call function, **.RETURN**.

Included as part of the CPU32Bug firmware is a single-line assembler/disassembler function. The assembler is an interactive assembler/editor in which source programs are not saved. Each source line is translated into MC68331 MCU machine language code and stored line-by-line into memory as it is entered. In order to display an instruction, the machine code is disassembled and the instruction mnemonic and operands are displayed. All valid MC68331 MCU instructions are supported.

The CPU32Bug assembler is effectively a subset of the M68000 Family Structured Assembler (M68MASM). It has some limitations as compared with the M68MASM assembler, such as not allowing line numbers and labels; however, it is a useful tool for creating, modifying, and debugging MC68331 MCU code.

3.6.1 Memory and Register Display and Modification Commands

Various commands are available to the user for displaying and modifying memory. For more information, refer to Chapter 3 of the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1. The memory display and modification commands are:

- **BF** (block of memory fill) fills the specified range of memory with a data pattern.
- **BM** (block of memory move) copies the contents of the memory addresses defined by <RANGE> to another place in memory, beginning at <ADDR>.
- **BS** (block of memory search) searches the specified range of memory for a match with a user-entered data pattern.
- **BV** (block of memory verify) compares the specified range of memory against a data pattern.
- **MD** (memory display) displays the contents of multiple memory locations.
- **MM** (memory modify) examines and changes memory locations.
- **MS** (memory set) writes data to memory starting at a specified address.
- **RD** (register display) displays the contents of the MCU registers.
- **RM** (register modify) examines and changes register contents.
- **RS** (register set) writes data to specified register.

3.6.2 Breakpoint Capabilities

A breakpoint lets the user set a target code instruction address stopping point for debugging purposes. Target code execution halts when a breakpoint is encountered. For more information on breakpoints, refer to Chapter 3 of the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1. The breakpoint commands are:

- **BR**/**NOBR** (breakpoint insert/delete) lets the user set a target code instruction address as a breakpoint address for debugging purposes.
- **GD** (go direct) starts target code execution and ignores breakpoints.
- **GN** (go to temporary breakpoint) sets a temporary breakpoint at the address of the next instruction, that is, the one following the current instruction.
- **GT** (go to temporary breakpoint) sets a temporary breakpoint at the current instruction and starts target code execution.
- **TT** (trace to temporary breakpoint) sets a temporary breakpoint at a specified address and traces until encountering a breakpoint.

3.6.3 System Calls

The CPU32Bug TRAP #15 handler allows system calls from user programs. A system call accesses selected functional routines contained in CPU32Bug, including input and output routines. TRAP #15 also transfers control to CPU32Bug at the end of a user program. For more information on system calls, refer to Chapter 5 of the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1. System calls include:

- **.INCHR** (input character) reads a character from the default input port.
- **.INSTAT** (input serial port status) checks for characters in the default input port buffer.
- **.INLN** (input line (pointer/pointer format)) reads a line from the default input port.
- **.READSTR** (input string (pointer/count format)) reads a string of characters from the default input port into a buffer.
- **.READLN** (input line (pointer/count format)) reads a string of characters from the default input port.
- **.CHKBRK** (check for break) returns zero (0) status in condition code register if break status is detected at the default input port.
- **.OUTCHR** (output character) outputs a character to the default output port.
- **.OUTSTR** (output string in pointer/pointer format) outputs a string of characters to the default output port.
- **.OUTLN** (output line in pointer/pointer format) outputs a character strings followed by a carriage return (<**CR**>) and a line feed (<**LF**>) sequence.
- **.WRITE** (output string in pointer/count format) formats character strings with a count byte and outputs the string to the default output port. After formatting, the count byte is the first byte in the string.
- **.WRITELN** (output line in pointer/count format) formats character strings with a count byte and outputs the string to the default output port. After formatting, the count byte is the first byte in the string.
- **.PCRLF** (output carriage return and line feed) sends a <**CR**> and a <**LF**> sequence to the default output port.
- **.ERASLN** (erase line) erases the line at the present cursor position.
- **.WRITD** (output string with data in pointer/count format) uses the monitor I/O routine which outputs a user string containing embedded variable fields. The user passes the starting address of the string and the data stack address containing the data which is inserted into the string. The output goes to the default output port.

- **.WRITDLN** (output line with data in pointer/count format) uses the monitor I/O routine which outputs a user string containing embedded variable fields. The user passes the starting address of the string and the data stack address containing the data which is inserted into the string. The output goes to the default output port.
- **.SNDBRK** (send break) sends a break to the default output port.
- **.TM_INI** (timer initialization) initializes the MC68331 MCU device periodic interrupt timer.
- **.TM_STR0** (start timer at T=0) resets the timer to zero and starts it.
- **.TM_RD** (read timer) reads the timer count.
- **.DELAY** (timer delay) generates timing delays.
- **.RETURN** (return to CPU32Bug) returns control to CPU32Bug from the target program.
- **.BINDEC** (convert binary to Binary Coded Decimal) calculates the BCD equivalent of a specified binary number.
- **.CHANGEV** (parse value) parses value in user specified buffer.
- **.STRCMP** (compare two strings in pointer/count format) compares equality and returns a Boolean flag to the caller.
- **.MULU32** (multiply two 32-bit unsigned integers) multiplies two 32-bit unsigned integers and returns the product on the stack as a 32-bit unsigned integer.
- **.DIVU32** (divide two 32-bit unsigned integers) divides two 32-bit unsigned integers and returns the quotient on the stack as a 32-bit unsigned integer.

3.6.4 Diagnostic Monitor

The diagnostic monitor is a series of self-tests for the MC68331 MCU device. The diagnostic monitor is programmed into the BCC EPROM. For more information on the diagnostic monitor, refer to Chapter 6 of the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1. The diagnostic monitor commands are:

- **HE** (help) displays a menu of the top level directory.
- **ST** (self test) executes self-test diagnostics.
- **SD** (switch directories) toggles between the CPU32Bug directory and the CPU32Diag directory.
- **LE** (loop-on-error) loops a test at the point where an error is detected.
- **SE** (Stop-On-Error) halts a test at the point where an error is detected.
- **LC** (loop-continue) loops a test or series of tests.
- **NV** (non-verbose)suppresses all error messages except PASSED or FAILED.
- **DE** (display error counters) displays the results of a particular test.
- **ZE** (clear error counters) resets all error counters to zero.
- **DP** (display pass count) displays the number of passes in loop-continue mode. At the conclusion of each pass, this command also displays other information.
- **ZP** (zero pass count) resets the pass counter to zero.

Available utilities are:

  – Write Loop (**WL**)
  – Read Loop (**RL**)
  – Read/Write Loop (**WR**)

CPU tests are diagnostics tests on the CPU portion of the MC68331 MCU:

  – **CPU A** - Register Test
  – **CPU B** - Instruction Test
  – **CPU C** - Address Mode Test
  – **CPU D** - Exception Processing Test

Memory tests verify random access memory that may reside on the BCC or the PFB:

- **MT A** - Set Function Code
- **MT B** - Set Start Address
- **MT C** - Set Stop Address
- **MT D** - Set Bus Data Width
- **MT E** - March Address Test
- **MT F** - Walk a Bit Test
- **MT G** - Refresh Test
- **MT H** - Random Byte Test
- **MT I** - Program Test
- **MT J** - TAS Test

Bus error test (**BERR**) tests for internal bus access time-out and internal to external bus access time-out error conditions, including:

- Bus errors when accessing the BCC RAM.
- Bus errors when reading the BCC EPROM.
- Bus errors when accessing the PFB optional memory.
- Internal bus access time-outs when reading and writing from an undefined function code/memory location internal to the MCU.
- Internal to external bus access time-outs reading or writing to an undefined function code/memory location external to the MCU.

3.7 ASSEMBLING/DISASSEMBLING PROCEDURES

The assembler/disassembler is an interactive, one-line assembler/editor in which the source program is not saved. Each source line is converted into machine language code and is stored in memory on a line-by-line basis at the time of entry. In order to display an instruction, the machine code is disassembled and the instruction mnemonic and operands are displayed. All valid opcodes are converted to assembly language mnemonic. All invalid opcodes return a Declare Constant Word (DC.W) conversion.

The memory modify (**mm ;di**) command lets the user create, modify, and debug MC68331 MCU code. Assembler input must have exactly one space between the mnemonic and the operand. There must be no space inside the operand field. Assembler input must be terminated by a carriage return. No comments are allowed after the instruction input and no line labels are permitted.

After each new assembler input line, the new line is disassembled for the user before stepping to the new instruction. The new line may assemble to a different number of bytes than the previous one.

For Branch if Higher or Same (BHS)/Branch if Carry Clear (BCC) mnemonics, disassembly displays the BCC mnemonic. For Branch if Lower (BLO)/Branch if Carry Set (BCS) mnemonics, disassembly displays the BCS mnemonic.

Branch address offsets are automatically calculated by the assembler, so the user should input the destination address rather than an offset value.

The assembler is terminated by entering a period (.) followed by a carriage return as the only entry on the command input line. Entering a carriage return alone on an input line steps to the next instruction.

The following pages describe how to operate the assembler/disassembler by creating a typical program loop, and debugging the program using CPU32Bug monitor commands. A typical program loop is first assembled. Routine examples then illustrate how to set a breakpoint, proceed from a breakpoint, display and modify registers, and initiate user program execution.

Enter the periodic interrupt timer (PIT) time-out program starting at address $5000:

EXAMPLE PROGRAM		PROGRAM DESCRIPTION
CPU32Bug>**MM 5000;DI**<**CR**>		*Memory modify at location $5000 with disassembly option.*
5000	**MOVE.L #$501C,$78**<**CR**>	*Set-up level six vector table.*
5008	**MOVE.L #$061E0120,$FFFA22**<**CR**>	*Initialize PIT.*
5012	**LPSTOP #$2500**<**CR**>	*Execute LPSTOP Instruction.*
5018	**BRA.W $5012**<**CR**>	*Loop*
501C	**MOVEA.W #$5100,A0**<**CR**>	*Beginning of message.*
5020	**MOVEA.W #$510E,A1**<**CR**>	*End of message.*
5024	**BTST.B #$0,$FFFC0C**<**CR**>	*Check for SCI not busy.*
502C	**BEQ.B $5024**<**CR**>	*Branch until free.*
502E	**MOVE.B (A0)+,$FFFC0F**<**CR**>	*Send message byte.*
5034	**CMPA.W A0,A1**<**CR**>	*Check for end of message.*
5036	**BNE.W $5024**<**CR**>	*Branch until done.*
503A	**RTE**<**CR**>	*Return from print routine.*

Enter the ASCII code for the output message, `PIT TIME-OUT`. Each time this message appears when running the program, indicates the program has completed a loop. Enter the following ASCII code at memory location $5100:

CPU32Bug>**MS 5100 'PIT TIME-OUT'ODOA**<**CR**> *Memory modify at location $5100.*

After entering the PIT time-out program display the instructions at location $5000

EXAMPLE PROGRAM			PROGRAM DESCRIPTION
CPU32Bug>**MD 5000;DI**<**CR**>			*Display memory at location $5000 with disassembly option.*
5000	21FC0000 501C0078	MOVE.L	#$501C,($78).W
5008	23FC061E 012000FF FA22	MOVE.L	#$061E0120,($FFFA22).L
5012	F80001C0 2500	LPSTOP.W	#$2500
5018	6000FFF8	BRA.W	$5012
501C	307C5100	MOVEA.W	#$5100,A0
5020	327C510E	MOVEA.W	#$510E,A1
5024	08390000 00FFFC0C	BTST.B	#$0,($FFFC0C).L
502C	67F6	BEQ.B	$5024
CPU32Bug><**CR**>			*Display next eight instructions.*
502E	13D800FF FC0F	MOVE.B	(A0)+,(FFFC0F).L.
5034	B2C8	CMPA.W	A0,A1
5036	6600FFEC	BNE.W	$5024
503A	4E73	RTE	
503C	0000FFFF	ORI.B	#$FF,D0
5040	0000FFFF	ORI.B	#$FF,D0
5044	0000FFFF	ORI.B	#$FF,D0
5048	0000FFFF	ORI.B	#$FF,D0

The message displayed on the terminal

```
CPU32Bug>MD 5100<CR>
00005100 5049 5420 5449 4D45  2D4F 5554 0D0A FFFF    PIT TIME-OUT....
```

The following routines are performed on the preceding program loop:

TERMINAL | **ROUTINE DESCRIPTION**

```
CPU32Bug>MD 5000;DI<CR>                          Display memory at address 5000
00005000   21FC0000 501C0078   MOVE.L      #$501C,($78).W
00005008   23FC061E 012000FF
           FA22                MOVE.L      #$061E0120, (FFFA22)
00005012   F80001C0 2500       LPSTOP.W    #$2500
00005018   6000FFF8            BRA.W       $5012
0000501C   307C5100            MOVEA.W     #$5100,A0
00005020   327C510E            MOVEA.W     #$510E,A1
00005024   08390000 00FFFC0C   BTST.B      #$0,($FFFC0C).L
0000502C   67F6                BEQ.B       $5024
CPU32Bug>MM 500C<CR>                             Modify memory at location 500C.
0000500C   0120? 00FF.                           Change PIT time-out speed.
CPU32Bug>g 5000<CR>                              Go to address 5000 and begin execution.
Effective address: 00005000
PIT TIME-OUT                                     Periodic interrupt timer time-out  message.
PIT TIME-OUT
PIT TIME-OUT
      :
      :
```

Press the ABORT switch on the PFB to terminate the loop program.

```
Exception: ABORT
PC    =00005018      SR    =2500=TR:OFF_S_5_.....              VBR   =00000000
SFC   =5=SD          DFC   =5=SD          USP   =0000FC00      SSP*  =00010000
D0    =00000000      D1    =00000000      D2    =00000000      D3    =00000000
D4    =00000000      D5    =00000000      D6    =00000000      D7    =00000000
A0    =0000510E      A1    =0000510E      A2    =00000000      A3    =00000000
A4    =00000000      A5    =00000000      A6    =00000000      A7    =00010000
00005018   6000FFF8                BRA.W.W          $5012
CPU32Bug>T 1<CR>                                 Trace one instruction.
PC    =00005018      SR    =2500=TR:OFF_S_5_.....              VBR   =00000000
SFC   =5=SD          DFC   =5=SD          USP   =0000FC00      SSP*  =00010000
D0    =00000000      D1    =00000000      D2    =00000000      D3    =00000000
D4    =00000000      D5    =00000000      D6    =00000000      D7    =00000000
A0    =0000510E      A1    =0000510E      A2    =00000000      A3    =00000000
A4    =00000000      A5    =00000000      A6    =00000000      A7    =00010000
00005012   F80001C0   2500           LPSTOP.W         #$2500
CPU32Bug>BR 5034<CR>                             Set breakpoint at 5034.
BREAKPOINTS
00005034
CPU32Bug>g 5000<CR>                              Go to address 5000 and begin execution.
Effective address: 00005018
At Breakpoint
PC    =00005034      SR    =2600=TR:OFF_S_6_.....              VBR   =00000000
SFC   =5=SD          DFC   =5=SD          USP   =0000FC00      SSP*  =0000FFF8
D0    =00000000      D1    =00000000      D2    =00000000      D3    =00000000
D4    =00000000      D5    =00000000      D6    =00000000      D7    =00000000
A0    =00005101      A1    =0000510E      A2    =00000000      A3    =00000000
A4    =00000000      A5    =00000000      A6    =00000000      A7    =0000FFF8
00005034   B2C8                     CMPA.W           A0,A1
```

3.8 DOWNLOADING PROCEDURES

Downloading transfers information from a host computer to the EVK, via the load (**LO**) command. The procedure described below lets the user download with an IBM personal computer (PC) or Apple Macintosh host computer.

The **LO** command moves data in S-record format (see Appendix A) from an external host computer to the EVK user pseudo ROM.
Subsections 3.7.1 through 3.7.4 list instructions for downloading to the EVK from an Apple Macintosh with MacTerminal or Red Ryder, or from an IBM-PC with Kermit or PROCOMM.

3.8.1 Apple Macintosh (with MacTerminal) to EVK

The MacTerminal downloading program serves as a terminal emulator for the Apple Macintosh computer. To download a Motorola S-record file from the Apple Macintosh computer to the EVK, follow these steps:

a. Select these menu terminal settings:

Terminal:	VT100
Mode:	ANSI
Cursor Shape:	Underline
Line Width:	80 Columns
Select:	On Line Auto Repeat
Click on:	OK

b. Select these menu compatibility settings:

Baud rate:	9600
Bits per Character:	8 Bits
Parity:	None
Handshake:	XOn/XOff
Connection:	Another Computer
Connection Port:	Modem
Click on:	OK

c. Select these menu file transfer settings:

Settings for Pasting or Sending Text:	Word Wrap Outgoing Text
File Transfer Protocol:	Text
Settings for Saving Lines Off Top:	Retain Line Breaks
Click on:	OK

d. Apply power to the EVK.

e. Press the carriage return (<**CR**>) key to display the applicable EVK monitor prompt.

f The computer displays the CPU32Bug> prompt.

g. Enter the EVK monitor download command:

CPU32Bug>**LO** (Press <**CR**> after entering **LO**.)

h. Operate the pull-down File menu, and select (choose): Send File ...

i. Use the dialog box to select applicable S-record object file.

Click on: Send

Motorola S-record file is now transferred to the EVK.

NOTE

The S-record file is not displayed during the file transfer to the EVK.

The underline cursor flashes and the beeper sounds when the S-record finishes downloading. Press the carriage return twice to return to the CPU32Bug prompt:

<**CR**>
<**CR**>
CPU32Bug>

3.8.2 Apple Macintosh (with White Knight) to EVK

The White Knight downloading program serves as a terminal emulator for the Apple Macintosh computer. To download a Motorola S-record file from the Apple Macintosh computer to the EVK, follow these steps:

a. Execute the White Knight program.

b. Set up the computer program to match the EVK baud rate (typically):

   9600 baud, no parity, 8 bits, 1 stop bit, full duplex

c. Apply power to the EVK.

d. Press the computer keyboard carriage return (<**CR**>) key to display the applicable EVK monitor prompt.

e Enter the EVK monitor download command:

   CPU32Bug>**LO** (Press <**CR**> after entering **LO**.)

f. Operate the pull-down File menu, and select (choose):

   Send File - ASCII...

g. Use the dialog box to select the applicable S-record object file.

   Click on: Send

   Motorola S-record file is now transferred to the EVK.

NOTE

The S-record file is not displayed during the file transfer to the EVK.

The underline cursor flashes and the beeper sounds when the S-record finishes downloading. Press the carriage return twice to return to the CPU32Bug prompt:

<**CR**>
<**CR**>
CPU32Bug>

3.8.3 IBM-PC (with KERMIT) to EVK

Before performing any IBM-PC operation, ensure that both IBM-PC and EVK baud rates are 9600, and that the IBM-PC asynchronous port is configured for terminal mode of operation. If the asynchronous port is hard wired for host mode of operation and cannot be re-configured for a terminal mode of operation, the use a null modem (cross-coupled transmit (TxD), and receive (RxD), and associated handshake lines) is required.

NOTE

IBM-PC to EVK connection requires one serial communication cable assembly. This cable is connected to the EVK terminal I/O port connector P9 for downloading operations.

To perform IBM-PC to EVK downloading procedure, follow this example:

EXAMPLE	**DESCRIPTION**
C>**KERMIT**<**CR**> IBM-PC Kermit-MS VX.XX Type ? for help	*IBM-PC prompt. Enter Kermit program.*
Kermit-MS>**SET BAUD 9600**<**CR**>	*Set IBM-PC baud rate.*
Kermit-MS>**CONNECT**<**CR**>	*Connect IBM-PC to EVK.*
[Connecting to host, type Control-] C to return to PC]	
<CR> CPU32Bug>**LO**<**CR**>	*EVK download command (via terminal port) entered.*
(CTRL)]C Kermit-MS>**PUSH**<**CR**>	
The IBM Personal Computer DOS Version X.XX (C)Copyright IBM Corp. 1981, 1982, 1983	
C>**TYPE (File Name) > COM1**<**CR**>	*Motorola S-record file name.*
C>**EXIT**<**CR**>	*S-record downloading completed.*
Kermit-MS>**CONNECT**<**CR**>	*Return to EVK monitor program.*

The underline cursor flashes and the beeper sounds when the S-record finishes downloading. Press the carriage return twice to return to the CPU32Bug prompt.

<**CR**>
<**CR**>
CPU32Bug>

CPU32Bug>**(CTRL)]C**
KERMIT-MS>**EXIT**<**CR**> *Exit Kermit program.*

3.8.4 IBM-PC (with PROCOMM) to EVK

To perform the IBM-PC to EVK downloading with PROCOMM, follow these steps:

a. Execute the PROCOMM.EXE program.

b. Set up PROCOMM to match the EVK baud rate and protocol (type Alt-P, then the number 11) as follows:

    9600 baud, no parity, 8 bits, 1 stop bit, full duplex

c. Setup ASCII transfer parameters (type Alt-S, then the number 6) as follows:

    Echo Local - No

    Expand Blank Lines - Yes

    Pace Character - 0

    Character pacing - 15 (milliseconds)

    Line Pacing - 10

    CR Translation - None

    LF Translation - None

    Save above settings to disk for future use.

d. Apply power to the EVK.

e. Press the keyboard carriage return (<**CR**>) key to display the applicable EVK monitor prompt.

f Enter the EVK monitor download command:

    CPU32Bug>**LO** (Press <**CR**> after entering **LO**.)

g. Press the PC Pg Up key, to instruct PROCOMM to send the S-record file. Then follow PROCOMM instructions on the screen to select the S-record file using an ASCII protocol.

The file transfer is done when the beeper sounds and the underline cursor flashes. Press the carriage return twice to return to the CPU32Bug prompt.

<**CR**>
<**CR**>
CPU32Bug>

CHAPTER 4
FUNCTIONAL DESCRIPTION

4.1 INTRODUCTION

This chapter is a functional description of the EVK and its components.

4.2 EVK DESCRIPTION

The EVK may be configured in either of two ways; the BCC mounted on the PFB or the BCC mounted on the target system. Figure 4-1 is the EVK block diagram.

When the BCC is mounted on the PFB, you may evaluate the MCU and debug user developed code. To do is, connect a terminal or host computer to PFB connector P9 and run the CPU32Bug debug monitor program. Logic analyzer connection may be made to connectors P1 through P6 of the PFB.

Mount the BCC on the target system to verify hardware design. With the BCC mounted on the target system, MC68331 MCU device emulation with hardware breakpoints is possible by connecting a PC to BCC connector P4 and running CPU32Bug debug monitor. Logic analyzer connection may be made to connectors P1 and P2 of the BCC.

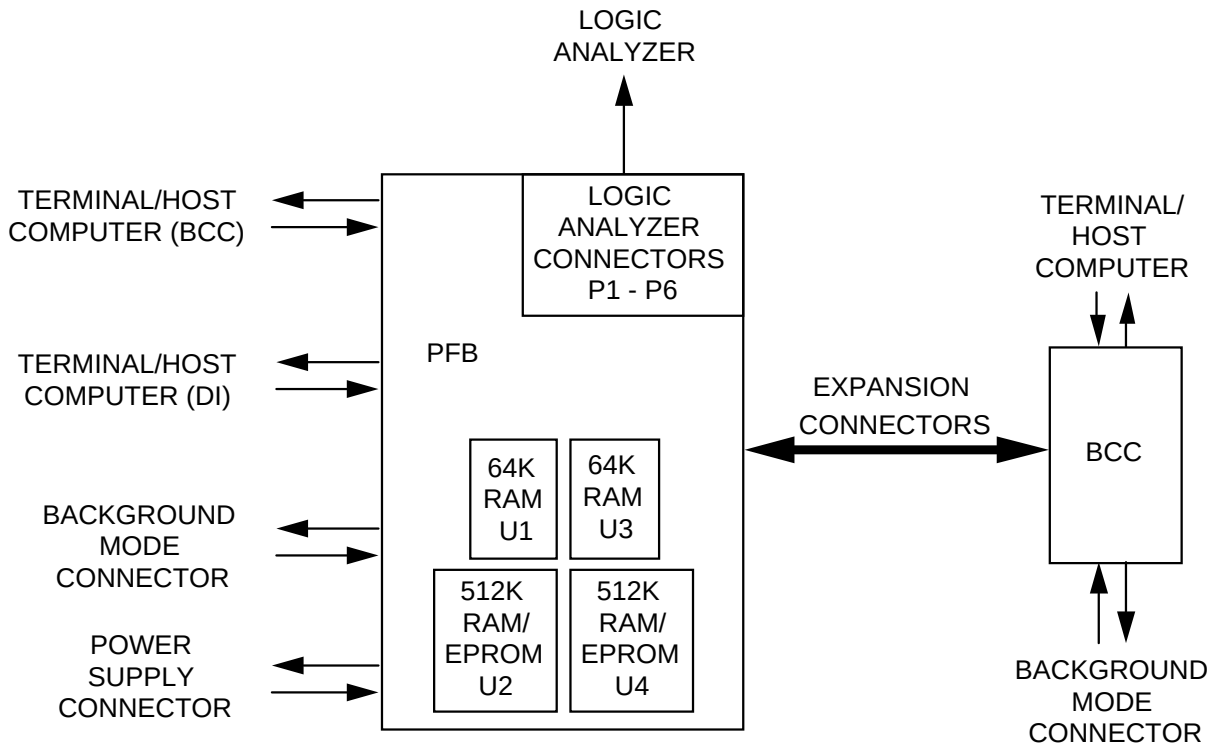


Figure 4-1. EVK Block Diagram

4.3 BCC DESCRIPTION

The BCC is a tool for designing, debugging, and evaluating MC68331 MCU based target systems. The BCC simplifies user evaluation of prototype hardware/software products. The BCC may be used by itself, as part of the EVK, or as part of the M68300EVS Evaluation System (EVS).

Figure 4-2 is the BCC block diagram. The BCC consists of these functional circuits:

- MCU
- User memory
- I/O connectors

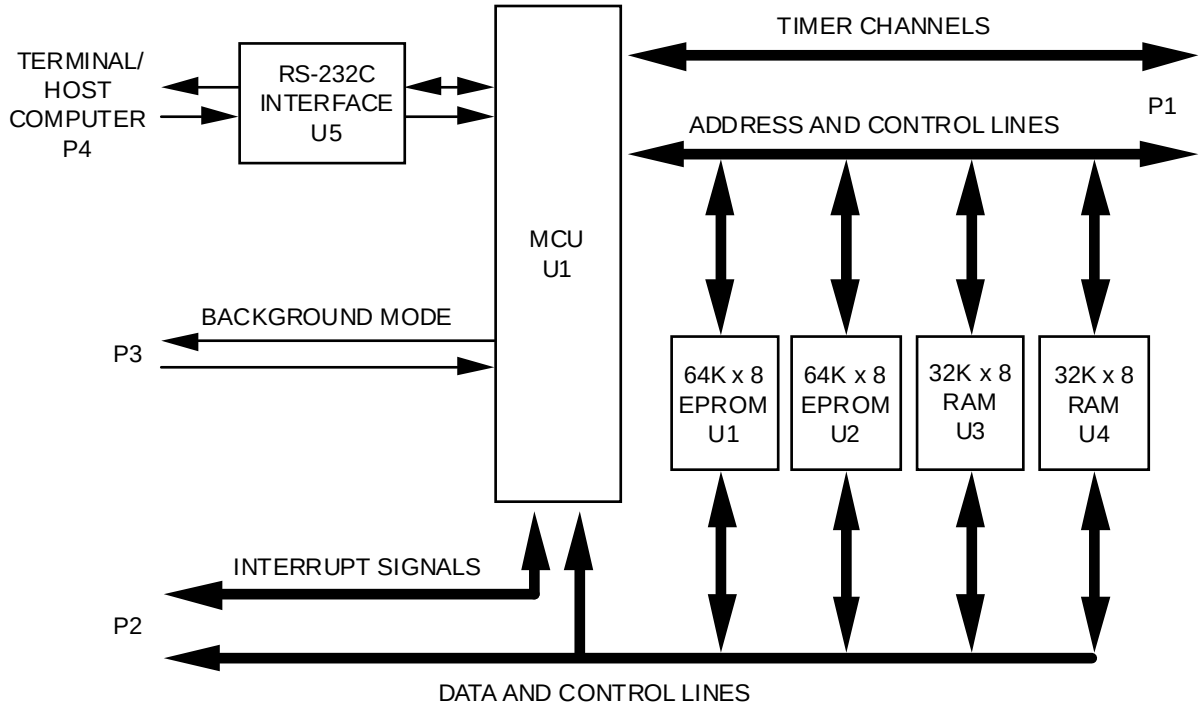


Figure 4-2. BCC Block Diagram

4.3.1 MCU Summary

The resident MC68331 Microcontroller Unit (MCU) of the BCC provides resources for designing, debugging, and evaluating MC68331 MCU based target systems and simplifies user evaluation of prototype hardware/software products.

The MCU device is a 32-bit integrated microcontroller, combining high-performance data manipulation capabilities with powerful peripheral subsystems. The MCU includes:

- 32-bit central processor unit (CPU32)
- General Purpose Timer (GPT)
- Queued serial module (QSM)
- External bus interface
- Chip selects
- System clock
- Test module

4.3.1.1 32-Bit Central Processor Unit

The CPU32 is the MC68331 MCU central processor. The CPU32 is source- and object-code compatible with the MC68000 and MC68010. All user-state programs can be executed unchanged. The CPU32 features are:

- 32-bit internal data path and arithmetic hardware - 16-bit external data bus
- 32-bit internal address bus - 24-bit external address bus
- Powerful instruction set
- Eight 32-bit general purpose data registers
- Seven 32-bit general purpose address registers
- Separate user and supervisor stack pointers and address spaces
- Separate program and data address spaces
- Flexible addressing modes
- Full interrupt processing

4.3.1.2 General-Purpose Timer

The GPT is a simple, flexible 11-channel timer for use in systems that require a moderate degree of external visibility and control. The GPT consists of two nearly independent submodules: the compare/capture unit and the pulse-width modulation (PWM) unit. The former features three input capture channels, four output compare channels, and one input capture/output compare channel (function selected via control register). These channels share a 16-bit free-running counter (TCNT) that derives its clock signal from a nine-stage prescaler or from the external clock input pin, PCLK. The timer counter also contains one pulse accumulator channel. The pulse accumulator logic includes its own 8-bit counter and can operate in either event counting mode or gated time accumulation mode.

The PWM submodule is associated with two output pins. The outputs are periodic waveforms whose duty cycles may be independently selected and modified by user software. The PWM unit has its own 16-bit free-running counter that is clocked by an output of the nine-stage prescaler (the same prescaler used by the compare/capture unit) or by the clock input pin, PCLK.

If not needed for timing functions, any of the pins associated with the GPT may be used for general-purpose input/output. The input capture and output compare pins are bi-directional and may be used to form an 8-bit parallel port. The PWM pins are outputs only. PAI and PCLK are inputs only.

4.3.1.3 Queued Serial Module

The QSM contains two serial ports. The queued serial peripheral interface (QSPI) port provides easy peripheral expansion or inter-processor communications via a full-duplex, synchronous, three-line bus: data-in, data-out, and a serial clock. Four programmable peripheral select pins provide address-ability for as many as 16 peripheral devices. A QSPI enhancement is an added queue in a small RAM. This lets the QSPI handle as many as 16 serial transfers of 8- to 16-bits each, or to transmit a stream of data as long as 256 bits without CPU intervention. A special wrap-around mode lets the user continuously sample a serial peripheral, automatically updating the QSPI RAM for efficient interfacing to serial peripheral devices (such as analog-to-digital converters).

The serial communications interface (SCI) port provides a standard non-return to zero (NRZ) mark/space format. Advanced error detection circuitry catches noise glitches to 1/16 of a bit time in duration. Word length is software selectable between 8- or 9-bits, and the SCI modulus-type, baud rate generator provides baud rates from 64 to 524k baud, based on a 16.77 MHz system clock. The SCI features full- or half-duplex operation, with separate transmitter and receiver enable bits and double buffering of data. Optional parity generation and detection provide either even or odd parity check capability. Wake-up functions let the CPU run uninterrupted until either a true idle line is detected or a new address byte is received.

4.3.1.4 External Bus Interface

The external bus consists of 24 address lines and a 16-bit data bus. The data bus allows dynamic sizing between 8- and 16-bit data accesses. A read-modify-write cycle (RMC) signal prevents bus cycle interruption. External bus arbitration is accomplished by a three-line handshaking interface.

4.3.1.5 Chip Selects

Twelve independently programmable chip selects provide fast, two-cycle external memory, or peripheral access. Block size is programmable from 2 kilobytes through 1 megabyte. Accesses can be selected for either 8- or 16-bit transfers. As many as 13 wait states can be programmed for insertion during the access. All bus interface signals are automatically handled by the chip select logic.

4.3.1.6 System Clock

An on-chip phase locked loop circuit generates the system clock signal to run the device up to 16.78 MHz from a 32.768 kHz watch crystal. The system speed can be changed dynamically, providing either high performance or low power consumption under software control. The system clock is a fully-static CMOS design, so it is possible to completely stop the system clock via a low power stop instruction, while still retaining the contents of the registers and on-board RAM.

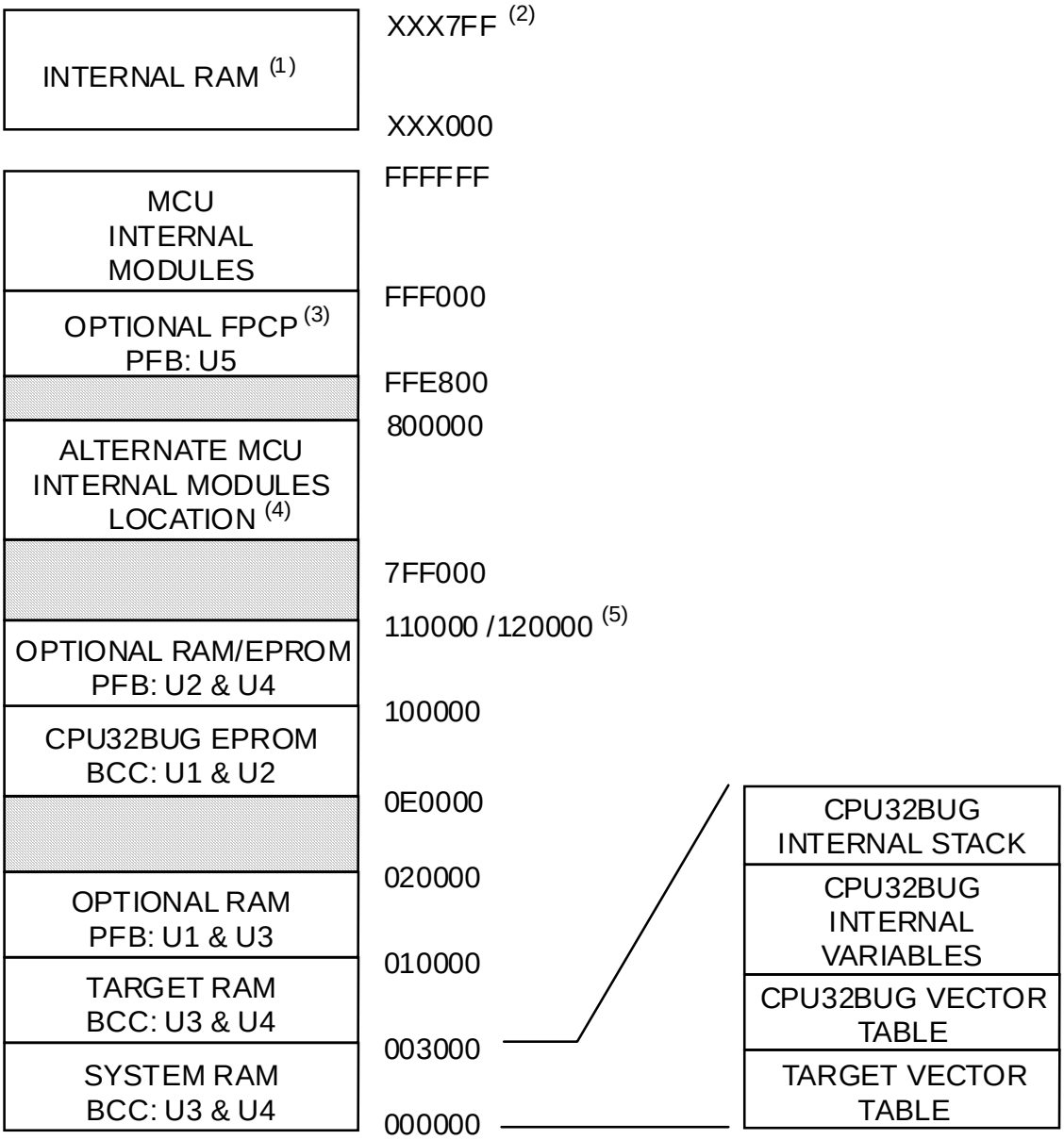
4.3.1.7 Test Module

The test module consolidates the microcontroller test logic into a single block to facilitate production testing, user self-test, and system diagnostics. Scan paths throughout the MC68331 provide signature analysis checks on internal logic. User self-test is initiated by asserting the test pin to enter test mode. This test provides a pass/fail response to various externally supplied test vectors.

4.3.2 User Memory

On board the BCC is 32k x 16 bits of RAM and 64k x 16 bits of EPROM. The RAM is the debug monitor storage area and user accessible memory space; the M68CPU32BUG Debug Monitor is stored in the BCC EPROM. For debug monitor functionality see the M68CPU32BUG Debug Monitor User’s Manual, M68CPU32BUG/AD1. Figure 4-3 is the EVK memory map.

The PFB has sockets for 32k x 16 or 64k x 16 bit RAM or 64k x 16 bit EPROM. The RAM and/or EPROM, supplied by the user, is user-accessible memory space.



1. Consult the MCU device user's manual.
2. XXbase address is user programmable. Internal modules, such as internal RAM, can be configured on power-up/reset by using the initilization table (INITTBL) covered in Appendix C of the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1.
3. Floating point coprocessor - MC68881/MC68882.
4. See Appendix C of the M68CPU32BUG Debug Monitor User's Manual, M68CPU32BUG/AD1.
5. Depends on the memory device type used.

Figure 4-3. EVK Memory Map

4.3.3 I/O Connectors

There are two 64-pin expansion connectors on the BCC (P1 and P2). Through these connectors the BCC communicates with the PFB or target system. Background mode operation is available through P3 and serial communication through P4. Chapter 5 contains a description of the interface connectors pin assignments.

4.3.3.1 64-Pin Expansion Connectors

The expansion connectors interconnect the BCC to the PFB or target system. The pin-outs of the MC68331 MCU device, serial communication, and background mode interface are available on the expansion connectors.

4.3.3.2 Serial Communication Connectors

A terminal or host computer with terminal emulation (PCKERMIT.EXE, PROCOMM, etc.), can be connected to the BCC or to the PFB. Terminal connections are provided through the serial communication connectors, BCC P4 or to PFB P9.

4.3.3.3 Background Mode Interface Connector

The background debug mode is implemented in MCU microcode. In background mode, registers can be viewed or altered, memory can be read or written, and test features can be executed. Background mode is initiated by one of several sources: externally generated breakpoints, internal peripherally generated breakpoints, software, and catastrophic exception conditions. Instruction execution is suspended for the duration of background mode. Background mode communications between the BCC and the development system are via a serial link (P3).

4.4 PFB DESCRIPTION

The PFB is the physical location for installing the BCC. The user may expand the user accessible memory. I/O connectors are available for communication, power, and a logic analyzer.

4.4.1 Floating-Point Coprocessor Socket (U5)

Socket U5 on the PFB accommodates an optional coprocessor for the EVK. Either an MC68881 or an MC68882 coprocessor can be used in socket U5. The coprocessor software interface is not part of the EVK, so must be provided by the user. For developing the coprocessor software interface, see the application note MC68881 Floating-Point Coprocessor as a Peripheral in an M68000 System, AN947.

The coprocessor interface is a transparent, logical extension of the MC68331 MCU device registers and instructions. To the external environment the CPU and coprocessor execution model appear to be on the same chip.

A coprocessor interface is an execution model based on sequential instruction execution by the CPU and coprocessor. For optimum performance, the coprocessor interface lets floating-point instructions execute concurrently with CPU integer instructions. Concurrent instruction execution is further extended by the coprocessor, which executes multiple floating-point instructions simultaneously.

4.4.2 Logic Analyzer Connectors

To debug hardware and software developed for the MC68331 MCU device, connect a logic analyzer to the desired pins of PFB connectors P1 - P6.

CHAPTER 5
SUPPORT INFORMATION

5.1 INTRODUCTION

This chapter provides the connector signal descriptions, parts lists and associated parts location diagrams, and schematic diagrams of the M68331EVK Evaluation Kit (EVK) components (BCC and PFB).

5.2 CONNECTOR SIGNAL DESCRIPTIONS

EVK connector pin assignments are defined in Tables 5-1 through 5-12. Connector signals are identified by pin number, signal mnemonic, and signal name and description. Table 5-1 is the EVK connector descriptions for the BCC and PFB.

Table 5-1. EVK Connector Descriptions

Connector	Description	Table
BCC P1	Interconnect the BCC and PFB.	5-1
BCC P2	Interconnect the BCC and PFB.	5-2
BCC J8	Background mode connector.	5-3
BCC J9	Serial communication between a PC and the BCC.	5-4
PFB BCC-P1	Interconnect the PFB and BCC.	5-1
PFB BCC P2	Interconnect the PFB and BCC.	5-2
PFB BCCDI-P1	Interconnect the PFB and DI.	5-1
PFB BCCDI-P2	Interconnect the PFB and DI.	5-2
PFB P1 – P6	Logic analyzer connections.	5-5 – 5-10
PFB P7	Supplies power to the EVK.	None
PFB P8	Serial communication between a PC and the DI.	5-11
PFB P9	Serial communication between a PC and the BCC.	5-12
PFB P10	Battery backup for MCUs with internal RAM.	None
PFB P11	Background mode connector.	5-3

Table 5-2. P1 Expansion Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1, 2	GND	GROUND
3, 4	VDD	SYSTEM POWER SUPPLY – +5 volt power source.
5	DTROUT	DATA TERMINAL READY OUT – RS-232C handshake from the DI.
6 – 24	A0 – A18	ADDRESS BUS – Three-state output address bus.
25	EPROM–W	EPROM WRITE – Active-low input write strobe that enables the on-board EPROM.
26	EPROM–VPP	EPROM PROGRAMMING VOLTAGE – 12.5 Vdc input programming voltage.
27	+13–24V DI	+13–24 VOLTS DEVELOPMENT INTER-FACE – DI switching voltage.
28	—	Not connected
29	+13–24V DI	$^{+}$13–24 VOLTS DEVELOPMENT INTER-FACE – DI switching voltage.
30	PCLK	AUXILIARY TIMER CLOCK INPUT – External input clock source for the GPT.
31, 32	PWMB, PWMA	PULSE WIDTH MODULATION A and B – Repetitive output signals whose high time to low time ratio can be controlled by the CPU.
33, 34, 35	—	Not connected
36	PAI	PULSE ACCUMULATOR INPUT – Input signal that increments an 8-bit counter.
37	IC4 / OC5	INPUT CAPTURE 4 – Input signal that latches the contents of the GPT timer counter (TCNT) into the input capture register TI4O5 when a selected edge occurs at the pin. OUTPUT COMPARE 5 – Output signal that is generated when the GPT timer counter (TCNT) and TI4O5 comparator register contain the same value.

Table 5-2. P1 Expansion Connector Pin Assignments (continued)

Pin Number	Signal Mnemonic	Signal Name and Description
38	OC4	OUTPUT COMPARE 4 – Output signal that is generated when the GPT timer counter (TCNT) and TOC4 comparator register contain the same value.
39	—	Not connected
40 – 42	OC3 – OC1	OUTPUT COMPARE 3 - 1 – Output signals that are generated when the GPT timer counter (TCNT) and TOC3 - TOC1 comparator registers contain the same value.
43 – 45	IC3 – IC1	INPUT CAPTURE 3 - 1 – Input signals that latch the contents of the GPT timer counter (TCNT) into the input capture registers TIC3 - TIC1 when a selected edge occurs at the pin.
46	—	Not connected
47	MOSI	MASTER-OUT SLAVE-IN – Serial output from the OSPI in master mode, and serial input to the QSPI in slave mode.
48	MISO	MASTER-IN SLAVE-OUT – Serial input to the QSPI in master mode, and serial output from the QSPI in slave mode.
49	$\overline{\text{PCS0}}$ / $\overline{\text{SS}}$	PERIPHERAL CHIP SELECT 0 – Active-low output QSPI peripheral chip select signal. SLAVE SELECT – Bi-directional active-low signal that places the QSPI in slave mode.
50	SCK	QSPI SERIAL CLOCK – Input/output QSPI clock source.
51	$\overline{\text{PCS2}}$	PERIPHERAL CHIP SELECT 2 – Active-low output QSPI peripheral chip select.
52	$\overline{\text{PCS1}}$	PERIPHERAL CHIP SELECT 1 – Active-low output QSPI peripheral chip select.
53	TXD	TRANSMIT DATA – Serial data output line.
54	$\overline{\text{PCS3}}$	PERIPHERAL CHIP SELECT 3 – Active-low output QSPI peripheral chip select.

Table 5-2. P1 Expansion Connector Pin Assignments (continued)

Pin Number	Signal Mnemonic	Signal Name and Description
55	$\overline{\text{BKPT}}$ /	BREAKPOINT – An active-low input signal that places the CPU32 in background debug mode.
	DSCLK	DEVELOPMENT SYSTEM CLOCK – Serial input clock for background debug mode.
56	RXD	RECEIVE DATA – Serial data input line.
57	$\overline{\text{RESET}}$	RESET – Active-low input/output signal for initiating a system reset.
58	FREEZE /	FREEZE – Output signal that indicates that the CPU32 has entered background debug mode.
	QUOT	QUOTIENT OUT – Output signal that furnishes the quotient bit of the polynomial divider for test purposes.
59	$\overline{\text{IPIPE}}$ /	INSTRUCTION PIPE – Active-low output signal that tracks movement of words through the instruction pipeline.
	DSO	DEVELOPMENT SERIAL OUT – Serial output for background debug mode.
60	$\overline{\text{IFETCH}}$ /	INSTRUCTION FETCH – Active-low output signal that indicates when the CPU is performing an instruction word pre-fetch and when the instruction pipeline has been flushed.
	DSI	DEVELOPMENT SERIAL IN – Serial data input for background debug mode.
61, 62	VDD	+5 VOLT INPUT POWER
63, 64	GND	GROUND

Table 5-3. P2 Expansion Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1, 2	GND	GROUND
3, 4	VDD	+5 VOLT INPUT POWER
5 – 20	D0 – D15	DATA BUS – 16 bi-directional data pins.
21	RXD DI	RECEIVE DATA DEVELOPMENT INTER-FACE – DI receive data.
22	TXD DI	TRANSMIT DATA DEVELOPMENT INTER-FACE – DI transmit data.
23	MODB DI	MODE B DEVELOPMENT INTERFACE – MC68HC11 mode B pin on the DI.
24	XMT 232	TRANSMIT DATA – Serial data output signal.
25	$\overline{\text{CSBOOT}}$	BOOT CHIP SELECT – An active-low output chip select.
26	RCV 232	RECEIVE DATA – Serial data input signal.
27	R/$\overline{\text{W}}$	READ/WRITE – Active-high output signal that indicates the direction of data transfer on the bus.
28	MODCK	CLOCK MODE SELECT – Input signal that selects the source of the internal system clock.
29	$\overline{\text{TSTME}}$ /	TEST MODE ENABLE – Active-low input signal that enables hardware for test mode.
	TSC	THREE STATE CONTROL – When TSC is 10 volts (2 X VDD), this input signal forces all output drivers to a high-impedance state.
30 – 34	A23/$\overline{\text{CS10}}$, A22/$\overline{\text{CS9}}$, A21/$\overline{\text{CS8}}$, A20/$\overline{\text{CS7}}$, A19/$\overline{\text{CS6}}$	ADDRESS BUS 19-23 – Three-state output address bus.
		CHIP SELECTS 6-10 – Output signals that select peripheral/memory devices at programmed addresses.

Table 5-3. P2 Expansion Connector Pin Assignments (continued)

Pin Number	Signal Mnemonic	Signal Name and Description
35 – 37	FC2/$\overline{CS5}$, FC1/$\overline{CS4}$, FC0/$\overline{CS3}$	FUNCTION CODES – Three-state output signals that identify the processor state and address space of the current bus cycle. CHIP SELECTS 3-5 – Output signals that select peripheral/memory devices at programmed addresses.
38	$\overline{BGACK}$ / $\overline{CS2}$	BUS GRANT ACKNOWLEDGE – Active-low input signal that indicates that an external device has assumed control of the bus. CHIP SELECT 2 – Output signal that selects peripheral/memory devices at programmed addresses.
39	$\overline{BG}$ / $\overline{CS1}$	BUS GRANT – Active-low output signal that indicates that the current bus cycle is complete and the MC68331 has relinquished the bus. CHIP SELECT 1 – Output signal that selects peripheral/memory devices at programmed addresses.
40	$\overline{BR}$ / $\overline{CS0}$	BUS REQUEST – Active-low input signal that indicates that an external device requests bus master-ship. CHIP SELECT 0 – Output signal that selects peripheral/memory devices at programmed addresses.
41 – 47	$\overline{IRQ1}$ – $\overline{IRQ7}$	INTERRUPT REQUEST (1-7) – Seven prioritized active-low input lines that request MCU synchronous interrupts. IRQ7 has the highest priority.
48	$\overline{BERR}$	BUS ERROR – Active-low input signal that indicates an erroneous bus operation attempt.

Table 5-3. P2 Expansion Connector Pin Assignments (continued)

Pin Number	Signal Mnemonic	Signal Name and Description
49	$\overline{\text{DSACK0}}$	DATA AND SIZE ACKNOWLEDGE 0 – Active-low input signals that allow asynchronous data transfers and dynamic bus sizing between the MC68331 and external devices.
50	$\overline{\text{DSACK1}}$	DATA AND SIZE ACKNOWLEDGE 1 – Active-low input signals that allow asynchronous data transfers and dynamic bus sizing between the MC68331 and external devices.
51	$\overline{\text{AVEC}}$	AUTOVECTOR – Active-low input signal that requests an automatic vector during an interrupt acknowledge cycle.
52	$\overline{\text{RMC}}$	READ-MODIFY-WRITE CYCLE – Active-low output signal that identifies the bus cycle as part of an indivisible read-modify-write operation.
53	$\overline{\text{DS}}$	DATA STROBE – Active-low output signal, that during a read cycle, indicates that an external device should place valid data on the data bus. During the write cycle, DS indicates that valid data is on the data bus.
54	$\overline{\text{AS}}$	ADDRESS STROBE – Active-low output signal that indicates that a valid address is on the address bus.
55, 56	SIZ0, SIZ1	TRANSFER SIZE – Active-high output signals that indicate the number of bytes remaining to be transferred during this cycle.
57	$\overline{\text{11RESET}}$	MC68HC11 RESET –.Active-low input signal that resets the DI HC11.
58	CLKOUT	SYSTEM CLOCK OUT – Output signal that is the MC68331 internal system clock.
59	EXTAL	EXTERNAL CLOCK INPUT – External clock input for the MC68331 MCU device.
60	$\overline{\text{HALT}}$	HALT – Active-low input/output signal that suspends external bus activity, to request a retry when used with $\overline{\text{BERR}}$, or for single-step operation.
61, 62	VDD	+5 VOLT INPUT POWER
63, 64	GND	GROUND

Table 5-4. BCC J8 and PFB P11 Background Mode Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	$\overline{DS}$	DATA STROBE – Active-low output signal, that during a read cycle, indicates that an external device should place valid data on the data bus. During the write cycle, DS indicates that valid data is on the data bus.
2	$\overline{BERR}$	BUS ERROR – Active-low input signal that indicates that an erroneous bus operation is being attempted.
3	GND	GROUND
4	$\overline{BKPT}$ /	BREAKPOINT – An active-low input signal that places the CPU32 in background debug mode.
	DSCLK	DEVELOPMENT SYSTEM CLOCK – Serial input clock for background debug mode.
5	GND	GROUND
6	FREEZE /	FREEZE – Output signal that indicates that the CPU32 has entered background debug mode.
	QUOT	QUOTIENT OUT – Output signal that furnishes the quotient bit of the polynomial divider for test purposes
7	$\overline{RESET}$	RESET – Active-low input/output signal for initiating a system reset.
8	$\overline{IFETCH}$ /	INSTRUCTION FETCH – Active-low output signal that indicates when the CPU is performing an instruction word pre-fetch and when the instruction pipeline has been flushed.
	DSI	DEVELOPMENT SERIAL IN – Serial data input for background debug mode.
9	+5V	+5 VOLT INPUT POWER
10	$\overline{IPIPE}$ /	INSTRUCTION PIPE – Active-low output signal used to track movement of words through the instruction pipeline.
	DSO	DEVELOPMENT SERIAL OUT – Serial output for background debug mode.

Table 5-5. BCC J9 Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	XMIT	TRANSMIT DATA – RS-232C serial output data.
2	GND	GROUND
3	RCV	RECEIVE DATA – RS-232C serial input data.
4	+10V	+10 VOLTS DC – Output voltage that may be used to drive RS-232C handshake lines.

Table 5-6. PFB P1 Logic Analyzer Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1, 2	NC	Not connected.
3	$\overline{AS}$	ADDRESS STROBE – Active-low output signal that indicates a valid address is on the address bus.
4 – 19	D0 – D15	DATA BUS – 16 bi-directional data pins.
20	GND	GROUND

Table 5-7. PFB P2 Logic Analyzer Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1, 2	NC	Not connected.
3	PCLK	AUXILIARY TIMER CLOCK INPUT – External input clock source to the GPT.
4, 5	PWMB, PWMA	PULSE WIDTH MODULATION A and B – Repetitive output signals whose high time to low time ratio can be controlled by the CPU.
6, 7, 8	NC	Not connected.
9	PAI	PULSE ACCUMULATOR INPUT – Input signal that increments an 8-bit counter.
10	IC4 / OC5	INPUT CAPTURE 4 – Input signal that latches the contents of the GPT timer counter (TCNT) into the input capture register TI4O5 when a selected edge occurs at the pin. OUTPUT COMPARE 5 – Output signal that is generated when the GPT timer counter (TCNT) and TI4O5 comparator register contain the same value.
11	OC4	OUTPUT COMPARE 4 – Output signal that is generated when the GPT timer counter (TCNT) and TOC4 comparator register contain the same value.
12	NC	Not connected.
13, 14, 15	OC3 – OC1	OUTPUT COMPARE 3 - 1 – Output signals that are generated when the GPT timer counter (TCNT) and TOC3 - TOC1 comparator registers contain the same value.
16, 17, 18	IC3 – IC1	INPUT CAPTURE 3 - 1 – Input signals that latch the contents of the GPT timer counter (TCNT) into the input capture registers TIC3 - TIC1 when a selected edge occurs at the pin.
19	NC	Not connected.
20	GND	GROUND

Table 5-8. PFB P3 Logic Analyzer Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1, 2	NC	Not connected.
3	$\overline{\text{DSACK1}}$	DATA AND SIZE ACKNOWLEDGE – Active-low input signals that allow asynchronous data transfers and dynamic bus sizing between the MC68331 and external devices.
4 – 19	A0 – A15	ADDRESS BUS – Three-state output address bus.
20	GND	GROUND

Table 5-9. PFB P4 Logic Analyzer Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1, 2	NC	Not connected.
3	$\overline{\text{DSACK0}}$	DATA AND SIZE ACKNOWLEDGE – Active-low input signals that allow asynchronous data transfers and dynamic bus sizing between the MC68331 and external devices.
4	$\overline{\text{AVEC}}$	AUTOVECTOR – Active-low input signal that requests an automatic vector during an interrupt acknowledge cycle.
5	$\overline{\text{RMC}}$	READ-MODIFY-WRITE CYCLE – Active-low output signal that identifies the bus cycle as part of an indivisible read-modify-write operation.
6	$\overline{\text{AS}}$	ADDRESS STROBE – Active-low output signal that indicates that a valid address is on the address bus.

Table 5-9. PFB P4 Logic Analyzer Connector Pin Assignments (continued)

Pin Number	Signal Mnemonic	Signal Name and Description
7	$\overline{DS}$	DATA STROBE – Active-low output signal, that during a read cycle, indicates that an external device should place valid data on the data bus. During the write cycle, DS indicates that valid data is on the data bus.
8	$\overline{BR}$ / $\overline{CS0}$	BUS REQUEST – Active-low input signal that indicates that an external device requires bus master-ship. CHIP SELECT 0 – Output signal that selects peripheral/memory devices at programmed addresses.
9	$\overline{BG}$ / $\overline{CS1}$	BUS GRANT – Active-low output signal that indicates that the current bus cycle is complete and the MC68331 has relinquished the bus. CHIP SELECT 1 – Output signal that selects peripheral/memory devices at programmed addresses.
10	$\overline{CSBOOT}$	BOOT CHIP SELECT – An active-low output chip select.
11	CLKOUT	SYSTEM CLOCK OUTPUT – MCU internal clock output signal.
12 – 16	A23/$\overline{CS10}$, A22/$\overline{CS9}$, A21/$\overline{CS8}$, A20/$\overline{CS7}$, A19/$\overline{CS6}$	ADDRESS BUS 19-23 – 5-bits of the 24-bit address bus. CHIP SELECTS 6-10 – Enables peripherals at programmed addresses.
17 – 19	A16 – A18	ADDRESS BUS 16-18 – Three bits of the 24-bit address bus.
20	GND	GROUND

Table 5-10. PFB P5 Logic Analyzer Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1, 2	NC	Not connected.
3	CLKOUT	SYSTEM CLOCK OUTPUT – MCU internal clock output signal.
4	$\overline{\text{BERR}}$	BUS ERROR – Active-low input signal that indicates that an erroneous bus operation is being attempted.
5	$\overline{\text{BKPT}}$ /	BREAKPOINT – Active-low input signal that places the CPU32 in background debug mode.
	DSCLK	DEVELOPMENT SYSTEM CLOCK – Serial input clock for background debug mode.
6	FREEZE /	FREEZE – Indicates that the CPU has acknowledged a breakpoint or has entered background mode.
	QUOT	QUOTIENT OUT – Furnishes the quotient bit of the polynomial divider for test purposes.
7	$\overline{\text{RESET}}$	RESET – System reset.
8	$\overline{\text{HALT}}$	HALT – Suspend external bus activity.
9	$\overline{\text{IPIPE}}$ /	INSTRUCTION PIPE – Tracks movement of words through the instruction pipeline.
	DSO	DEVELOPMENT SERIAL OUT – Serial output for background debug mode.
10	$\overline{\text{NEW IFETCH}}$	NEW INSTRUCTION FETCH – $\overline{\text{NEW IFETCH}}$ is either normal $\overline{\text{IFETCH}}$ or latched IFETCH per the configuration of jumper header J14.

Table 5-10. PFB P5 Logic Analyzer Connector Pin Assignments (continued)

Pin Number	Signal Mnemonic	Signal Name and Description
11, 12	$\overline{\text{DSACK1}}$, $\overline{\text{DSACK0}}$	DATA AND SIZE ACKNOWLEDGE – Terminates asynchronous data transfers and dynamic bus sizing.
13 – 15	FC2/$\overline{\text{CS5}}$, FC1/$\overline{\text{CS4}}$, FC0/$\overline{\text{CS3}}$	FUNCTION CODES – Identify the processor state and address space of the current bus cycle. CHIP SELECTS 3-5 – Enable peripherals at programmed addresses.
16, 17	SIZ1, SIZ0	SIZE – Indicates the number of bytes remaining to be transferred during this cycle.
18	R/$\overline{\text{W}}$	READ/WRITE – Indicates the direction of data transfer on the bus.
19	$\overline{\text{BGACK}}$ / $\overline{\text{CS2}}$	BUS GRANT ACKNOWLEDGE – Indicates that an external device has assumed control of the bus. CHIP SELECT 2 – Enables peripherals at programmed addresses.
20	GND	GROUND

Table 5-11. PFB P6 Logic Analyzer Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1, 2	NC	Not connected.
3	$\overline{DS}$	DATA STROBE – During a read cycle, DS indicates that an external device should place valid data on the data bus. During the write cycle, DS indicates that valid data is on the data bus.
4	MODCK	CLOCK MODE SELECT – Active-high input signal that selects the source of the internal system clock.
5 – 11	$\overline{IRQ1}$ – $\overline{IRQ7}$	INTERRUPT REQUEST (1-7) – Seven prioritized active low input lines that request MCU synchronous interrupts. $\overline{IRQ7}$ has the highest priority.
12	TXD	TRANSMIT DATA – Serial data output line.
13	$\overline{PCS0}$ / $\overline{SS}$	PERIPHERAL CHIP SELECT 0 – Active-low output QSPI peripheral chip select signal. SLAVE SELECT – Bi-directional active-low signal that places the QSPI in slave mode.
14	$\overline{PCS1}$	PERIPHERAL CHIP SELECT 1 – Active-low output QSPI peripheral chip select.
15	$\overline{PCS2}$	PERIPHERAL CHIP SELECT 2 – Active-low output QSPI peripheral chip select.
16	$\overline{PCS3}$	PERIPHERAL CHIP SELECT 3 – Active-low output QSPI peripheral chip select.
17	SCK	QSPI SERIAL CLOCK – Furnishes the clock from the QSPI in master mode or to the QSPI in slave mode.

Table 5-11. PFB P6 Logic Analyzer Connector Pin Assignments (continued)

Pin Number	Signal Mnemonic	Signal Name and Description
18	MISO	MASTER-IN SLAVE-OUT – Serial input to the QSPI in master mode, and serial output from the QSPI in slave mode.
19	MOSI	MASTER-OUT SLAVE-IN – Serial output from the OSPI in master mode, and serial input to the QSPI in slave mode.
20	GND	GROUND

Table 5-12. PFB P8 Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	DCD	DATA CARRIER DETECT – Not connected.
2	RXD	RECEIVE DATA – RS-232C serial input signal.
3	TXD	TRANSMIT – RS-232C serial output signal.
4	DTR	DATA TERMINAL READY – Not connected.
5	GND	GROUND
6	DSR	DATA SET READY – An output signal (held high) that indicates an on-line/in-service/active status.
7	RTS	REQUEST TO SEND – Not connected.
8	CTS	CLEAR TO SEND – An output signal that indicates a ready-to-transfer data status.
9	NC	Not connected.

Table 5-13. PFB P9 Connector Pin Assignments

Pin Number	Signal Mnemonic	Signal Name and Description
1	DCD	DATA CARRIER DETECT – Not connected.
2	RXD	RECEIVE DATA – RS-232C serial input signal.
3	TXD	TRANSMIT – RS-232C serial output signal.
4	DTR	DATA TERMINAL READY – An output line that indicates an on-line/in-service/active status.
5	GND	GROUND
6	DSR	DATA SET READY – An output signal (held high) that indicates an on-line/in-service/active status.
7	RTS	REQUEST TO SEND – Not connected.
8	CTS	CLEAR TO SEND – An output signal that indicates a ready-to-transfer data status.
9	NC	Not connected.

APPENDIX A
S-RECORD INFORMATION

The S-record format for output modules was devised for the purpose of encoding programs or data files in a printable format for transportation between computer systems. The transportation process can thus be visually monitored and the S-records can be more easily edited.

S-RECORD CONTENT

When viewed by the user, S-records are essentially character strings made of several fields which identify the record type, record length, memory address, code/data and checksum. Each byte of binary data is encoded as a 2-character hexadecimal number; the first character representing the high-order 4 bits, and the second the low-order 4 bits of the byte.

The five fields which comprise an S-record are shown below:

Where the fields are composed as follows:

Field	Printable Characters	Contents
type	2	S-records type -- S0, S1, etc.
record length	2	The count of the character pairs in the record, excluding type and record length.
address	4,6,or 8	The 2-, 3-, or 4-byte address at which the data field is to be loaded into memory.
code/data	0-n	From 0 to n bytes of executable code, memory-loadable data, or descriptive information. For compatibility with teletypewriters, some programs may limit the number of bytes to as few as 28 (56 printable characters in the S-record).
checksum	2	The least significant byte of the one's complement of the sum of the values represented by the pairs of characters making up the records length, address, and the code/data fields.

Each record may be terminated with a CR/LF/NULL. Additionally, an S-record may have an initial field to accommodate other data such as line numbers generated by some time-sharing systems. An S-record file is a normal ASCII text file in the operating system of origin.

Accuracy of transmission is ensured by the record length (byte count) and checksum fields.

S-RECORD TYPES

Eight types of S-records have been defined to accommodate the several needs of the encoding, transportation and decoding functions. The various Motorola upload, download and other records transportation control programs, as well as cross assemblers, linkers and other file-creating or debugging programs, utilize only those S-records which serve the purpose of the program. For specific information on which S-records are supported by a particular program, the user's manual for the program must be consulted. 332Bug supports S0, S1, S2, S3, S7, S8, and S9 records.

An S-record format module may contain S-records of the following types:

Type	Description
S0	The header record for each block of S-records, The code/data.field may contain any descriptive information identifying the following block of S-records. The address field is normally zeros.
S1	A record containing code/data and the 2-byte address at which the code/data is to reside.
S2	A record containing code/data and the 3-byte address at which the code/data is to reside.
S3	A record containing code/data and the 4-byte address at which the code/data is to reside.
S5	A record containing the number of S1, S2, and S3 records transmitted in a particular block. This count appears in the address field. There is no code/data field.
S7	A termination record for a block of S3 records, The address field may optionally contain the 4-byte address of the instruction to which control is passed. There is no code/data field.
S8	A termination record for a block of S2 records. The address field may optionally contain the 3-byte address of the instruction to which control is passed. There is no code/data field.
S9	A termination record for a block of S1 records. The address field may optionally contain the 2-byte address of the instruction to which control is passed. If not specified, the first entry point specification encountered in the object module input will be used. There is no code/data field.

Only one termination record is used for each block of S-records. S7 and S8 records are usually used only when control is to be passed to a 3 or 4 byte address. Normally, only one header record is used, although it is possible for multiple header records to occur.

S-RECORDS CREATION

S-record format files may be produced by dump utilities, debuggers, linkage editors, cross assemblers or cross linkers. Several pro- grams are available for downloading a file in S-record format from a host system to a microprocessor-based system.

EXAMPLE

Shown below is a typical S-record format module, as printed or displayed:

```
S00600004844521B
S1130000285F245F2212226A000424290008237C2A
S11300100002000800082629001853812341001813
S113002041E900084E42234300182342000824A952
S113003000144ED492
S9030000FC
```

The module consists of one S0 record, four S1 records, and an S9 record.

The S0 record is comprised of the following character pairs:

S0	S-record type S0, indicating that it is a header record.
06	Hexadecimal 06 (decimal 6), indicating that six character pairs (or ASCII bytes) follow.
00 00	Four-character, 2-byte, address field; zeros in this example.
48 44 52	ASCII H, D and R - "HDR".
1B	The checksum.

The first S1 record is explained as follows:

S1	S-record type S1, indicating that it is a code/data record to be loaded/verified at a 2-byte address.
13	Hexadecimal 13 (decimal 19), indicating that 19 character pairs, representing 19 bytes of binary data, follow.
00 00	Four-character, 2-byte, address field; hexadecimal address 0000, where the data which follows is to be loaded.

The next 16 character pairs of the first S1 record are the ASCII bytes of the actual program code/data. In this assembly language example, the hexadecimal opcodes of the program are written in sequence in the code/data fields of the S1 records:

OPCODE	INSTRUCTION	
285F	MOVE.L	(A7)+,A4
245F	MOVE.L	(A7)+,A2
2212	MOVE.L	(A2),D1
226A0004	MOVE.L	4(A2),A1
24290008	MOVE.L	FUNCTION(A1),D2
237C	MOVE.L	#FORCEFUNC,FUNCTION(A1)

(The balance of this code is continued in the code/data fields of the remaining S1 records and stored in memory.)

2A The checksum of the first S1 record.

The second and third S1 records also each contain $13 (19) character pairs and are ended with checksums 13 and 52 respectively. The fourth S1 record contains 07 character pairs and has a checksum of 92.

The S9 record is explained as follows:

S9 S-record type S9, indicating that it is a termination record.

03 Hexadecimal 03, indicating that three character pairs (3 bytes) follow.

00
00 The address field, zeros.

FC The checksum of the S9 record.

Each printable character in an S-record is encoded in a hexadecimal (ASCII in this example) representation of the binary bits which are actually transmitted. For example, the first S1 record above is sent as:

TYPE				LENGTH				ADDRESS								CODE/DATA										CHECKSUM			
S		1		1		3		0		0		0		0		2		8		5		F		•••		2		A	
5	3	3	1	3	1	3	3	3	0	3	0	3	0	3	0	3	2	3	8	3	5	4	6	•••		3	2	4	1
0101	0011	0011	0001	0011	0001	0011	0011	0011	0000	0011	0000	0011	0000	0011	0000	0011	0010	0011	1000	0011	0101	0100	0110	•••		0011	0010	0100	0001