MMCCMB2114

Controller and Memory Board

User's Manual

metrowerks
A Motorola Company

Important Notice to Users

While every effort has been made to ensure the accuracy of all information in this document, Motorola assumes no liability to any party for any loss or damage caused by errors or omissions or by statements of any kind in this document, its updates, supplements, or special editions, whether such errors are omissions or statements resulting from negligence, accident, or any other cause. Motorola further assumes no liability arising out of the application or use of any information, product, or system described herein: nor any liability for incidental or consequential damages arising from the use of this document. Motorola disclaims all warranties regarding the information contained herein, whether expressed, implied, or statutory, *including implied warranties of merchantability or fitness for a particular purpose*. Motorola makes no representation that the interconnection of products in the manner described herein will not infringe on existing or future patent rights, nor do the descriptions contained herein imply the granting or license to make, use or sell equipment constructed in accordance with this description.

Equal Opportunity

Motorola, Inc., is an Equal Opportunity / Affirmative Action Employer.

Trademarks

This document includes these trademarks:

> Motorola and the Motorola logo are registered trademarks of Motorola, Inc.
>
> Windows and Windows 95 are registered trademarks of Microsoft Corporation in the U.S. and other countries.
>
> Intel is a registered trademark of Intel Corporation.

For an electronic copy of this book, visit Motorola's web site at http://mcu.motsps.com/documentation

Contents

Figures

Figures

Tables

CMB2114 Quick Start Guide

Make sure that power is disconnected from your MMCCMB2114 Controller and Memory Board (CMB2114), and from your development system. Then follow these quick-start steps to make your CMB2114 ready for use as quickly as possible.

ESD CAUTION: *Motorola development systems include open-construction printed circuit boards that contain static-sensitive components. These boards are subject to damage from electrostatic discharge (ESD). To prevent such damage, you must use static-safe work surfaces and grounding straps, as defined in ANSI/EOS/ESD S6.1 and ANSI/EOS/ESD S4.1. All handling of these boards must be in accordance with ANSI/EAI 625.*

Set the User Option Switches to Their Factory Defaults

Your CMB2114 contains two user option switches, S1 and S2. Each switch contains four subswitches. Make sure that subswitches USR0 and USR2 of user option switch S2 are set in the OFF position. The remaining subswitches of both the user option switches should be set in the ON position.

Set the Jumper Headers to Their Factory Defaults

1. Make sure that a jumper is installed in power headers J28, J37, J38, and J48.
2. Make sure that a jumper is installed between pin 2 and 3 of jumper header W3.
3. Make sure that a jumper is installed between pins 1 and 2 of jumper headers W4 and W5.
4. Make sure *there is no jumper installed* between pads 1 and 2 of jumper header J50.

Connect the CMB2114 to Your Computer System

To connect your CMB2114 to a computer system:

- Connect an RS-232 cable between CMB2114 connector J58 labelled PORT A on the silkscreen and the appropriate serial port of your computer.
- Connect your 12-volt power supply to line power and to CMB2114 connector J61 labelled +12V PWR IN on the silkscreen. Use switch S4 to turn power on. The green colored LED DS6 lights to confirm that the CMB2114 is powered.

Note *Should the LED DS6 not light, you may need to replace the fuse at location F1, next to power connector J61. Use a BUS GMA-1.5A fast blow fuse, or comparable.*

Start the Picobug Monitor

The Picobug monitor comes burned into the external FLASH memory devices of your CMB2114.

Notes *Before you start the Picobug monitor, make sure that you have an RS-232 connection between CMB2114 connector J58 and a serial port of your computer. Make sure that power to your CMB2114 is turned off.*

The commands and menu selections in the instructions provided in this subsection are specific to the Hyperterminal terminal-emulation program. If you use a different terminal-emulation program, you must make corresponding changes in the commands and menu selections of the instructions provided in this subsection.

To start the monitor for use as a standalone debugger by using Hyperterminal:

1. Start the Hyperterminal program.
2. Select **File>Properties**.
3. Select the COM port being used by the host computer from the **Connect using** list box in the **Properties** dialog box.
4. Click the **Configure** button in the **Properties** dialog box.
5. Use the **Configuration** dialog box to set the communications properties:
   - 19200 baud
   - 8 data bits

- no parity
- 1 stop bit
- no flow control
- select the appropriate communications port

6. Click **OK**.
7. Apply power to the CMB2114. The Picobug monitor starts automatically, displaying the command prompt: `picobug>`.

Download and Run Sample Application

1. Type `lo` at the picobug command prompt and press Enter.
2. In the HyperTerminal menu bar, select **Transfer>Send Text File**. The **Send Text File** window appears.
3. Navigate to the following directory on the MMC2114 CD-ROM root directory: `Dev_Sys\Mmc2114\Quickstart\Sample_led_app.` This location contains the sample project files.
4. Double-click the `out.elf.s` file. The application code starts to get transferred to the CMB2114. This operation may take a few moments.

   Once the file has been downloaded, the following message appears on the picobug prompt: "`Done downloading. The target PC is set to 8xxx_xxxx`"

CAUTION: *Do not cycle power supply (turn OFF and ON) while the applicaiton is being downloaded to the board. This will cause the download operation to fail.*

5. Type `g` at the picobug prompt and press Enter. This runs the application on the CMB2114.

   The User Status LEDs DS5 through DS2 momentarily display the 5 (0101) flashing pattern. The pattern then changes to A (1010). Next, the program walks a bit from LED DS2 to LED DS5 to light each LED independently. Finally, all LED's are lit up. This pattern then repeats at a faster rate. The fast and slow patterns are repeated continuously till the program is stopped.

6. To stop the application and return to the picobug prompt, press the RESET switch S3 on the CMB2114.

For more information on using picobug monitor and other debugging utilities, refer **Section 3.**

Section 1. General Information

1.1 Introduction

The MMCCMB2114 Controller and Memory Board (CMB2114) is a development tool for Motorola's M•CORE™ processor family that lets you develop code to be embedded in an MMC2114 microcontroller unit (MCU).

As a standalone tool, the CMB2114 uses an RS-232 connection to your computer. This connection lets you use Motorola's M•CORE System Development Software (SysDS), the GNU source-level debugger, or the Metrowerks MetroTRK debug software. The SysDS consists of a loader, the Picobug monitor, and a built-in selftest. The CMB2114 also has a OnCE™ connector, enabling you to use a debugging application that requires one.

Optionally, you may use the CMB2114 with a different emulator product, such as the Motorola Enhanced Background Debug Interface (EBDI), or the Metrowerks CodeWarrior™ integrated development environment (IDE).

Motorola's SysDS loader lets you download your code into the FLASH memory of the MMC2114 MCU and the CMB2114 for execution or for storage in non-volatile memory. You can also use Motorola's SysDS loader to download you code into the static random access memory (SRAM) of CMB2114 for execution.

The CMB2114 combines easily with other, optional development boards from Motorola, such as the MPFB1200 Platform Board. Such an optional board expands CMB2114 capacity, enhances CMB2114 performance, or adds to CMB2114 features.

1.2 CMB2114 Features

The CMB2114 features:

- 144-pin, quad flat pack MMC2114 resident MCU.
- 2 megabytes FLASH memory, configurable for 16- or 32-bit operations.
- 2 megabytes fast static RAM (FSRAM), configurable for 16- or 32-bit operations.
- Xilinx complex programmable logic device (CPLD).
- Connector header for programming Xilinx CPLD.

- On-board 5-volt and 3.3-volt supply.
- ON/OFF power switch and power LED.
- Two RS-232 serial communication ports.
- On Chip Emulation (OnCE) connector.
- External clock input connector.
- Four user-accessible light emitting diodes (LEDs.)
- Two dual inline package (DIP) switches for system configuration and firmware selection.
- User prototyping (breadboard) area.
- A modular, all purpose interface (MAPI 400) connector ring, on the top and bottom of the CMB2114, for easy connection to other, compatible development boards.
- Three 38-pin Mictor logic analyzer connectors.
- Motorola's SysDS.
- Metrowerks MetroTRK debug software
- GNU source-level debugger (from the Free Software Foundation).
- Metrowerks CodeWarrior™ IDE (30-day trial version)
- Four locations for optional, user-installed prototyping connectors
- External clock source connector
- 5 volts or 3.3 volts Analog to Digital Convertor (ADC) operation
- Break out pads for GPIO, interrupts, SCI, timer, and analog-digital converter (ADC) signals.

1.3 System and User Requirements

You need an IBM PC or compatible computer, running the Windows 9x/2000 or Windows NT 4.0 operating system. The computer requires a Pentium or equivalent microprocessor, 64 megabytes of RAM, 150 megabytes of free hard-disk space, a Super Video Graphics Array (SVGA) color monitor, and an RS-232 serial-communications port. To use the Picobug debug monitor, you also need Hyperterminal or a comparable terminal-emulation program.

To get the most from your CMB2114, you should be an experienced C or M•CORE assembly programmer.

The power supply that comes with your CMB2114 converts line power to the input power that the CMB2114 needs: 12 volts @ 1.2 amperes.

1.4 CMB2114 Layout

Figure 1-1 shows the layout of the CMB2114. Connectors P1 through P4, on the top of the board, are the MAPI I/O and interrupt connectors. The corresponding MAPI connectors on the bottom of the CMB2114 are J1 through J4.

Connector J6 is the CPLD programming connector. Connector J7 is the OnCE connector. Connectors J5, J17, and J18 are the logic analyzer connectors. Connector J36 is for external standby power for internal SRAM. Connector J39 is a surface-mount adapter (SMA) connector for external clock input. Connector J57 and J58 are the port B and port A RS-232 serial connectors, respectively. Connector J61 is the connector for 12-volt input power.

Switches S1 and S2 configure several aspects of memory organization and access. Switch S3 is the reset switch. Switch S4 is the power switch.

Several two-pin jumper headers are convenient current measurement points for various power signals:

- J28, 3.3-volt power to the resident MCU at location U10;
- J37, standby power for internal SRAM;
- J38, power for internal FLASH; and
- J48, 5-volt or 3.3-volt power to the queued analog-digital converter (QADC).

To measure the current of any of these signals, temporarily remove the jumper, then connect the leads of your meter to the header pins.

Jumper headers W1 and W2 let you select a 3.3-volt or 5-volt ADC supply. The factory configuration specifies 5-volt ADC supply. These jumper headers are not populated on the board.

Jumper headers W3 through W5 enable you to select either an on-board crystal oscillator or an external clock.

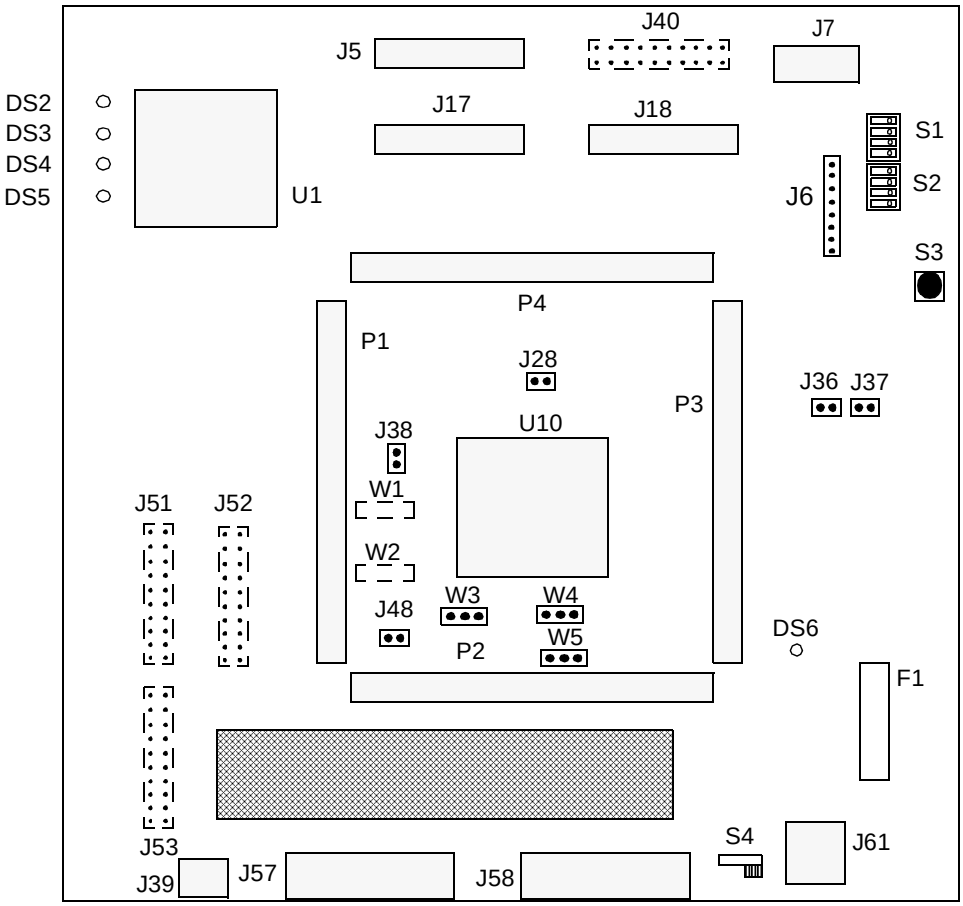


Figure 1-1 MMCCMB2114 Controller and Memory Board Layout

LEDs DS2 through DS5 are general-purpose status indicators. LED DS6 confirms operating power.

The CMB2114 prototyping area is adjacent to MAPI connector P2 and the RS-232 connectors (J57, J58). Ground connections are the left and right columns of this area. The area's top row includes connection points for Analog Power (APWR), Analog Ground (AGND), 3.3 volts, and 5 volts.

Note the four groups of eyelets at the sites J40, J51, J52, and J53 in the upper-right and lower-left areas of the CMB2114. Although the factory does not populate these sites, you may access many signals at these sites. For this, you may install prototyping connectors that must be 2-by-10-pin connectors with pins at 0.1-inch centers, such as the Berg 69192-620 connector.

Location F1 is for the CMB2114 fuse.

The resident MCU, at location U10, is an MMC2114 device, in a 144-pin QFP package. The CPLD is at location U1.

Table 1-1 lists CMB2114 specifications.

Table 1-1 MMCCMB2114 Controller and Memory Board Specifications

Characteristic	Specifications
MCU extension I/O port	High speed complementary metal oxide semiconductor (HCMOS) compatible
Operating temperature	0° to 40° C
Storage temperature	-40° to +85° C
Relative humidity	0 to 90% (non-condensing)
Reference clock crystal frequency	8 megahertz
External clock	32, 24, 16 or 8 megahertz, depending on the Phase Locked Loop (PLL) setting
Power requirements	12 volts dc, at a minimum of 150-milliamperes, provided from a separate power source
Dimensions	6.9 x 8.2 inches (175 x 208 mm)

Section 2. Preparation and Installation

You can follow the instructions in this chapter to configure your CMB2114, and hook it up to your computer system.

ESD CAUTION: *Motorola development systems include open-construction printed circuit boards that contain static-sensitive components. These boards are subject to damage from electrostatic discharge (ESD). To prevent such damage, you must use static-safe work surfaces and grounding straps, as defined in ANSI/EOS/ESD S6.1 and ANSI/EOS/ESD S4.1. All handling of these boards must be in accordance with ANSI/EAI 625.*

2.1 Configuring Board Components

Table 2-1 is a summary of configuration settings.

Table 2-1 Component Settings

Component	Position	Effect
User Options Switch S1, Boot Ex/In Subswitch	ON	This setting configures the board to boot from chip-select-0 (CS0) memory, and disables the internal FLASH of the board. Default factory setting.
	OFF	In master mode, this setting configures the board to boot from internal-FLASH memory. In emulation mode, this setting configures internal FLASH emulation on chip-select-1 (CS1) memory. However, the board cannot boot from CS1 memory.
Data 32/16 subswitch of user option switch S1	ON	This setting specifies a 32-bit external data bus. Default factory setting.
	OFF	This setting specifies a 16-bit external data bus. Microcontroller unit (MCU) data lines 15—0 become general purpose input-output (GPIO) lines, and are available at J40.

Table 2-1 Component Settings (Continued)

Component	Position	Effect
Swap 02/20 subswitch of user option switch S1	ON	This setting configures the board for CS0 control of external FLASH and chip-select-2 (CS2) control of external SRAM. Default factory setting.
	OFF	This setting configures the board for CS2 control of external FLASH and CS0 control of external SRAM.
M0 subswitch of user options switch S1 and M1 subswitch of user option switch S2	Subswitch M0 — ON Subswitch M1 — ON	This setting configures the board to run in master mode. Default factory setting.
	Subswitch M0 — ON Subswitch M1 — OFF	This setting configures the board to run in emulation mode and disables internal FLASH emulation on CS1. In addition to this setting, CS1 should be configured for normal operation by turning Boot Ex/In and Swap 02/20 subswitches ON.
	Subswitch M0 — OFF Subswitch M1 — ON	This setting configures the board to run in single-chip mode.
	Subswitch M0 — OFF Subswitch M1 — OFF	This setting configures the board to run in emulation mode and enables internal FLASH emulation on CS1.
User Option Switch S2, USR0 — USR2 Subswitches (1)	Subswitch USR0 — OFF Subswitch USR1 — OFF Subswitch USR2 — OFF	This setting specifies built-in selftest firmware module to be run out of reset.
	Subswitch USR0 — OFF Subswitch USR1 — ON Subswitch USR2 — OFF	This setting specifies Picobug monitor firmware module to be run out of reset. Default factory setting.
	Subswitch USR0 — ON Subswitch USR1 — OFF Subswitch USR2 — OFF	This setting specifies Programmer firmware module to be run out of reset.
	Subswitch USR0 — ON Subswitch USR1 — ON Subswitch USR2 — OFF	This setting specifies MetroTRK firmware module to be run out of reset.
	Subswitch USR0 — ON Subswitch USR1 — ON Subswitch USR2 — ON	This setting specifies user code to be run out of reset.

Table 2-1 Component Settings (Continued)

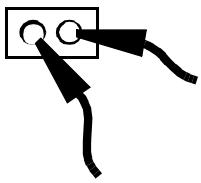
Component	Position	Effect
Reset Switch, S3		Push S3 to reset all board components.
Power Switch, S4	OFF	Setting S4 to the OFF position turns power OFF. Default factory setting.
	ON	Setting S4 to the ON position turns power ON.
Power Headers: • MCU 3V (J28) • Internal RAM standby (J37) • Internal chip FLASH voltage (J38) • ADC voltage (J48)		A jumper installed on a power header connects the specified power signal. A jumper installed on each power header is the default factory setting. You should leave the jumper installed on the power headers during normal use.
		With the jumpers removed, meter leads can be connected to the individual pins of any power header to measure the current of the power signal.

Table 2-1 Component Settings (Continued)

Component	Position	Effect
Jumper headers W3, W4, W5, and J50	W3 — Jumper between pins 2 and 3 W4 — Jumper between pins 2 and 3 W5 — Jumper between pins 2 and 3 J50 — Don't care	Selects on-board crystal (Y1) as clock source with Phase Locked Loop (PLL) enabled.
	W3 — Jumper between pins 2 and 3 W4 — Jumper between pins 1 and 2 W5 — Jumper between pins 1 and 2 J50 — Jumper pads 1 and 2 open	Selects on-board clock oscillator (Y2) as clock source with Phase Locked Loop (PLL) enabled. Default factory setting.
	W3 — Jumper between pins 2 and 3 W4 — Jumper between pins 1 and 2 W5 — Jumper between pins 1 and 2 J50 — Jumper pads 1 and 2 closed	Selects the external clock input at J39 as clock source with Phase Locked Loop (PLL) enabled.
	W3 — Jumper between pins 1 and 2 W4 — Jumper between pins 1 and 2 W5 — Jumper between pins 1 and 2 J50 — Jumper pads 1 and 2 open	Selects on-board clock oscillator (Y2) as clock source with Phase Locked Loop (PLL) disabled.
	W3 — Jumper between pins 1 and 2 W4 — Jumper between pins 1 and 2 W5 — Jumper between pins 1 and 2 J50 — Jumper pads 1 and 2 closed	Selects the external clock input at J39 as clock source with Phase Locked Loop (PLL) disabled.

1. USR0, USR1, and USR2 subswitch settings other than those specified in this table will select the Picobug monitor firmware module to be run out of reset.

2.1.1 Setting the User Option Switches

The subswitches of user option switches S1 and S2 configure several aspects of board initialization and operation, including the mode of operation and the firmware module to be run out of reset. Figure 2-1 shows the factory configuration:

- 32-bit, external data bus;
- CS0 controls the boot memory and the external FLASH;
- Internal FLASH is disabled;
- CS2 controls the SRAM;
- Master mode; and
- Picobug firmware module to be run out of reset.

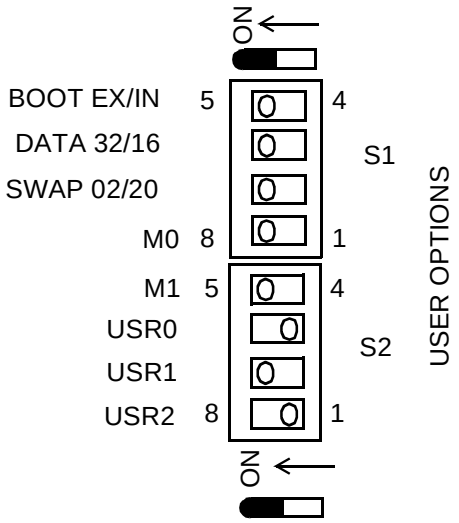


Figure 2-1 User Option Switches (S1, S2) Factory Configuration

For a different configuration, reset the S1 and S2 subswitches per Table 2-1.

Note *Changes you make to settings of the Boot Ex/In, Data 32/16, Swap 02/20, M0, or M1 subswitches do not take effect until the next reset or powerup of your CMB2114.*

2.1.2 Using the Power Headers (J28, J37, J38, J48)

Your CMB2114 has the following power headers:

- J28 — 3.3-volt power to the U10 resident MCU,
- J37 — standby power for internal SRAM of the U10 resident MCU,
- J38 — power for internal FLASH of the U10 resident MCU,
- J48 — 5-volt or 3.3-volt power to the queued analog-digital converter (QADC).

During normal use of your CMB2114, leave the jumpers on all of these headers.

To measure the current of any of these signals:

1. Turn CMB2114 power off by setting switch S4 to the OFF position.
2. Remove the jumper from the power header to be checked.

3. Connect meter leads to the header pins.
4. Turn CMB2114 power ON by using switch S4, and read the current from the meter.
5. Turn CMB2114 power OFF by using switch S4.
6. Disconnect the meter leads from the power header you just checked.
7. Reinstall the jumper in the power header.
8. Restore power by setting switch S4 to the ON position.

2.1.3 Setting Jumper Headers

The jumper headers in CMB2114 allow you to select power supply and clock sources. Detailed explanations of the jumper settings follow.

2.1.3.1 Jumper Headers W1 and W2

Jumper headers W1 and W2 allow you to select power supply for the internal analog-digital converter (ADC). These jumper headers are not populated on the board. The factory ships your CMB2114 with a 0-ohm resistance installed between pads 1 and 2 of components R24 and R25, which are in parallel to jumper headers W1 and W2, respectively. This setting for W1 and W2 specifies 5-volt VDDH and 5-volt VDDA to the ADC, respectively.

Alternatively, you can specify 3.3-volt VDDH and 3.3-volt VDDA to the ADC, respectively. For this, you need to first remove the 0-ohm resistance installed between pads 1 and 2 of components R24 and R25. Next, install the 0-ohm resistance between pads 2 and 3 of components R24 and R25.

2.1.3.2 Jumper Headers W3, W4, W5, and J50

The CMB2114 has an on-board crystal oscillator at locationY1, an on-board clock oscillator at location Y2, and an external clock input at location J39. Additionally, their outputs can be routed through a PLL that allows you to set the desired clock frequency.

Jumper headers W3, W4, W5, and J50 allow you to select either an on-board clock source or an external clock source. The settings of these jumper headers also determine if the PLL is enabled or diabled. Jumper headers W3, W4, and W5 are three-pin headers, while jumper header J50 is a two-pin header in which you can install a 0-Ohm resistance.

The factory setting is:

- A jumper between pins 2 and 3 of jumper header W3
- A jumper between pins 1 and 2 of jumper headers W4 and W5
- A jumper wire installed between pads 1 and 2 of jumper header J50 .

This setting selects the on-board clock oscillator as the clock source to the MMC2114 MCU with the PLL enabled.

For alternative settings of clock sources, refer **Table 2-1**.

2.2 Making Computer System Connections

When you have configured your CMB2114, you are ready to connect it to your computer system:

1. Disconnect power from your CMB2114 and from your development system.
2. For RS-232 communication directly with your host computer, connect an RS-232 cable between CMB2114 connector J58 (port A) and the appropriate serial port of your computer.
3. Alternatively, to you use an Enhanced Background Debug Interface (EBDI) with your CMB2114, connect the 14-lead target cable between CMB2114 connector J7 and the EBDI. Then use an RS-232 cable to connect the EBDI to your host computer. Using an EBDI means that you do not need to use CMB2114 connector J58 at all. However, a target board could communicate through connector J58.

   For using an EBDI, the Boot Ex/In sub switch must be set to the OFF position so that the board boots from the internal memory.
4. Optional: If your code, running in a target board, supports RS-232 communication with the CMB2114, you can connect a second RS-232 cable between the target board and CMB2114 connector J57 (port B).
5. Optional: You may use the CMB2114 with an MPFB1200 platform board.

   To do so, you must:

   a. Connect the boards via their MAPI rings.

   b. Hold the CMB2114 directly above the other board.

c. Turn the CMB2114 so that the right-triangle silk screen markings line up. Then press the CMB2114 down onto the other board. CMB2114 connectors J1 through J4, on the bottom of the board, must connect with the corresponding MAPI connectors P1 through P4, on the top of the other board.

6. Optional: You may use a logic analyzer with the CMB2114. If you do, connect appropriate cables to any of the logic analyzer connectors: J5, J17, or J18. Such a cable must terminate with a compatible Mictor connector. Section 4 includes pin assignments and cable descriptions for the logic analyzer connectors. (Note the Tektronix-pattern pin numbering.)

7. If you are using the CMB2114 as a standalone tool, connect your 12-volt power supply to line power and to CMB2114 connector J61. Use switch S4 to turn on power. The green colored LED DS6 lights to confirm that the CMB2114 is powered.

   Should the LED DS6 *not* light, you may need to replace the fuse at location F1, next to power connector J61. Use a BUS GMA-1.5A fuse, or compatible.

8. If you did connect the CMB2114 to a platform board, apply power to the platform board per its instructions. LED DS6 lights to confirm that the CMB2114 receives power from the platform board. (As in step 6, should the LEDs *not* light, you may need to replace the platform-board fuse.)

This completes system connections. You are ready to perform a selftest, per the instructions of subsection 2.4. You can begin debugging or other development activities, per the instructions of Chapter 3.

2.3 Performing the CMB2114 Selftest

Once you have configured your CMB2114, you can perform a selftest of its components.

1. Turn off CMB2114 power by setting switch S4 to the OFF position. Power LED DS6 should be out.

2. Set switch S2 for the built-in selftest. Subswitches USR0, USR1, and USR2 should all be in the OFF position.

3. Turn on power by using switch S4. LED DS6 lights to confirm power, and the CMB2114 begins its selftest.

4. LEDs DS2 through DS5 light and go out during the test, according to the sequence shown in Table 2-2.

Table 2-2 CMB2114 Selftest LED Sequence

DS2	DS3	DS4	DS5	Test Action
OFF	ON	OFF	ON	8-bit write to memory.
ON	OFF	ON	OFF	8-bit read from memory.(1)
OFF	ON	OFF	ON	16-bit write to memory.
ON	OFF	ON	OFF	16-bit read from memory.(1)
OFF	ON	OFF	ON	32-bit write to memory.
ON	OFF	ON	OFF	32-bit read from memory.(1)

1. Should all four LEDs stay lit at this point, the CMB2114 has failed the SRAM test, aborting the rest of the selftest. Contact Motorola customer support for assistance.

5. Then individual LEDs light several times in the sequence, DS5, DS4, DS3, and DS2.
6. When all four LEDs go out, the CMB2114 has passed the selftest. If any LEDs stay lit, the CMB2114 has failed the selftest. Contact Motorola customer support for assistance.
7. Turn off CMB2114 power by using switch S4.
8. Configure subswitches USR0, USR1, and USR2 for your next development activity before restoring power to the CMB2114.

2.4 Memory Maps

Table 2-3 shows the default memory map where the Swap 02/20 subswitch is ON.

Table 2-3 Default Memory Map

Address Range	Sub Range	Memory Resource	Related Chip Select
0x8000_0000 0x801F_FFFF		CMB FLASH (2 megabytes) System	CS0
	0x8000_0000 0x8001_FFFF	System Software (128 kilobytes Sectors 0—3)	
	0x8002_0000 0x801F_FFFF	User Code (1920 kilobytes Sectors 4 — 18)	
0x8020_0000 0x807F_FFFF		User address space (6 megabytes)	
0x8100_0000 0x811F_FFFF		CMB SRAM (2 megabytes)(1)	CS2
	0x8100_0000 0x8100_BFFF	Reserved for System Software (48 kilobytes)	
	0x8100_C000 0x811F_FFFF	User Code (2000 kilobytes)	
0x8120_0000 0x817F_FFFB		User address space (6 megabytes)(1)	
0x817F_FFFC		MMIO read-only byte (reads in USR0, USR1, USR2 subswitch settings.)	
0x817F_FFFD		MMIO write-only byte (controls LEDs)	

1. If you use the CMB2114 with an MPFB1200 platform board, and if platform-board SRAM is associated with the same chip select as CMB2114 SRAM, the MMIO function uses the last four bytes of CMB SRAM.

Table 2-4 shows the combined memory map for a CMB2114 and an MPFB1200 platform board, each with factory settings. This yields eight megabytes each of FLASH memory and SRAM.

Note *Note that CMB2114 factory settings configure the board to run in master mode. This memory map also is valid for CMB2114 emulation mode, with chip-select 1 internal FLASH emulation disabled.*

The CMB2114 Data 32/16 subswitch must be ON for this MPFB1200 memory map to be valid, but the CMB2114 Boot Ex/In subswitch has no effect on the map.

Table 2-4 Joint CMB2114/MPFB1200 Memory Map

Address Range	Use	Size	Chip Select
0x8000_0000 0x801F_FFFF	CMB2114 FLASH	2 megabytes	CS0
0x8020_0000 0x807F_FFFF	MPFB1200 FLASH	6 megabytes	CS0
0x8080_0000 0x809F_FFFF	[Unused]		
0x80A0_0000 0x80FF_FFFF	MPFB1200 SRAM(1)	6 megabytes	CS1
0x8100_0000 0x811F_FFFF	CMB2114 SRAM	2 megabyte	CS2
0x8120_0000 0x817F_FFFF	[Unused]		
0x8180_0000 0x8180_0FFF	MPFB1200 Peripherals	4 kilobytes	CS3
0x8180_1000 0x818F_FFFF	MPFB1200 User Space D	1 megabyte (almost)	CS3

1. For the alternate MPFB SRAM address range 0x8120_0000 through 0x817F_FFFB, use the MPFB main SRAM jumper header (W2) to select CS2.

Table 2-5 shows the alternate CMB2114 memory map where the Swap 02/20 subswitch is OFF.

Table 2-5 Alternate Memory Map

Address Range	Memory Resource	Related Chip Select
0x8000_0000 0x801F_FFFF	CMB SRAM (2 megabytes)	CS0
0x8020_0000 0x807F_FFFB	User address space (6 megabytes)	
0x807F_FFFC	MMIO read-only byte (reads in USR0, USR1, USR2 subswitch settings.)	
0x807F_FFFD	MMIO write-only byte (controls LEDs)	
0x8100_0000 0x811F_FFFF	CMB FLASH (2 megabytes)	CS2
0x8120_0000 0x817F_FFFF	User address space (6 megabytes)	

Note *Using the alternate memory map means that you cannot use the Motorola system software, which is FLASH based. For debugging in the alternate memory map, you must use an EBDI or other product that communicates through the OnCE interface.*

2.5 Chip Select 1 Emulation

Chip select 1 emulation pertains to three cases of M1, M0, and Swap 02/20 subswitch settings.

- Case I—M1 OFF, M0 OFF, Swap 02/20 ON**.**

  These subswitch settings configure emulation mode. MCU memory range 0x0000_0000 — 0x0003_FFFF, under chip-select-1 control, gets mapped to CMB2114 SRAM. Chip select 2 provides access to the same physical memory, but at CMB2114 addresses 0x8104_0000 — 0x8107_FFFF. Motorola system software programs chip select 1 for one wait state.

Note *Chip-select-1 emulation is not an exact simulation of internal FLASH operation, which has no wait states. Actual internal FLASH operation is faster than chip-select-1 emulation.*

- Case II—M1 OFF, M0 OFF, Swap 02/20 OFF.

  These subswitch settings also configure emulation mode. MCU memory range 0x0000_0000 — 0x0003_FFFF, under chip-select-1 control, gets mapped to CMB2114 internal FLASH. Chip select 2 provides access to the same physical memory, but at CMB2114 addresses 0x8004_0000 — 0x8007_FFFF. For debugging in this configuration, you must use an EBDI or other product that communicates through the OnCE interface. (Motorola system software does not support this configuration.)

- Case III—M1 OFF, M0 ON, Swap 02/20 ON.

  These subswitch settings also configure emulation mode. Chip select 1 does not specify any CMB2114 memory. Motorola system software disables chip-select-1 emulation, and programs chip select 1 for three wait states. In this configuration, your code can use chip select 1 to specify platform-board or other user-defined memory.

2.6 Memory Mapped I/O Operation

The MCU operating mode determines the implementation of memory mapped I/O (MMIO) operation.

1. In master or emulation mode, the CPLD MMIO register reads the settings of subswitches USR0 through USR2 and controls the status LEDs DS2 through DS5. The register consists of two bytes, as shown in Figure 2-2. These bytes are mapped to the first two bytes of the last valid SRAM 32-bit address.

817F_FFFC (807F_FFFC)				**817F_FFFD (807F_FFFD)**					
D31	D30	D29	D28 — D24	D23 — D21	D20	D19	D18	D17	D16
USR2	USR1	USR0	Not Used	Not Used	Not Used	LED DS5	LED DS4	LED DS3	LED DS2

Figure 2-2 MMIO Register

The upper byte of the register is read only.

- Bits D31 through D29 show the positions of subswitches USR2 through USR0, respectively. A subswitch OFF setting produces a 1 bit value; a subswitch ON setting produces a 0 bit value.
- This byte is at address 0x817F_FFFC (or 0x807F_FFFC if the Swap 02/20 subswitch is OFF).

The lower byte of the register is write only and is cleared by a reset.

- Bits D19 through D16 control status LEDs DS5 through DS2, respectively: set bits turn ON the corresponding LEDs, clear bits turn OFF the corresponding LEDs.
- This byte is at address 0x817F_FFFD (or 0x807F_FFFD if the swap 02/20 subswitch is OFF).

2. In single chip mode, Port H controls USR subswitch and status LED functionality.
   - Port H bit 7 must be configured as a low output.
   - Port H bits 6 though 4 read the settings of subswitches USR2 through USR0, respectively.
   - Port H bits 3 through 0 control status LEDs DS5 through DS2, respectively: set bits turn ON the corresponding LEDs, clear bits turn OFF the corresponding LEDs.

2.7 Using the Prototyping Area

The CMB2114 prototyping area lets you add your own components to the board. Merely insert the component's feet through holes in the board, then solder the feet in place to hold the component in position. Run appropriate leads from the new component to board power and ground locations.

Note the connection points of the prototyping area:

- Ground — columns on either side,
- Analog power — three points at the upper-left corner,
- Analog ground — three points of the top row,
- 3.3 volt power — three points of the top row, and
- 5-volt power — three points at the upper-right corner.

The prototyping connector site J40 is in the upper-right area and the prototyping connector sites J51, J52, and J53 are in the lower-left area of the board, respectively.

2.8 Reprogramming the CPLD

You can reprogram the CPLD if you want to add test circuits through the CPLD instead of hard wiring to the bread board, and to add circuits to simulate development software, such as an interval interrupt generator.

To reprogram the CPLD you need:

- A parallel cable (model DLC5) for connection between the parallel port of the host PC and the CPLD programming connector J6. This cable can be purchased from Xilinx.
- Xilinx Foundation series software version 3.1i or higher or Webpack ISE software version 3.21 or higher. The Webpack ISE program is freeware and can be downloaded from the Xilinx website:

  `www.xilinx.com/sxpresso/webpack.htm`

You can use the projects provided on the MMC2114 CD to reprogram the CPLD. The MMC2114 CD contains the Verilog version of the project as well as the schematic entry version that uses the Foundation tool.

The schematic entry version and the Verilog version of the project are located at the following path on the MMC2114 CD:

- Schematic entry — `Dev_Sys/MMC2114/CPLD/PD042.zip`

- Verilog — `Dev_Sys/MMC2114/CPLD/PD042_verilog.zip`

Note *The schematic entry version of the project requires Foundation software for synthesis. However, the Verilog version of the project can be used with Webpack as well as Foundation software.*

You can modify the project source files according to your requirements.

CAUTION: *1. Do not remove modules in the project source files as they are critical to system operation.*

2. Do not try to change the pin assignments of the CPLD as these are fixed by the layout.

You can also restore the factory settings of the CPLD by using the same projects. Instructions for restoring the factory settings of the CPLD are provided in a text file on the MMC2114 CD at the following location:

`Dev_Sys/MMC2114/CPLD/readme.txt`

.

Section 3. Support Information

You can follow the instructions in this chapter for using the debugging tools available for your CMB2114 and for using Motorola's SysDS Loader.

3.1 Embedded Code Debugging Options

You have several options available for debugging embedded code. To debug embedded code, you may use:

- The Picobug monitor as standalone software
- The GNU source-level debugger with the Picobug monitor
- Metrowerks Target Resident Kernel (MetroTRK) debugger

Other firms may produce additional software to run, test, and modify the code you develop for embedding in a MMC2114 MCU.

3.1.1 Using the Picobug Monitor

The Picobug monitor comes burned into the external FLASH memory devices of your CMB2114**.** Before you start the Picobug monitor, make sure that you have an RS-232 connection between CMB2114 connector J58 and a serial port of your computer.

You need the Hyperterminal terminal-emulation program to use the Picobug monitor as a standalone debugger. If you use a different terminal-emulation program, you must make corresponding changes in the commands and menu selections of the instructions provided in this subsection.

To start the monitor for use as a standalone debugger by using Hyperterminal:

1. Disconnect power from your CMB2114.
2. Start the Hyperterminal program.
3. Select **File > Properties**. The **Properties** dialog box appears.
4. Select the COM port being used by the host computer from the **Connect using** list box in the **Properties** dialog box.
5. Click the **Configure** button in the **Properties** dialog box. The **Configuration** dialog box appears.

6. Use the configuration dialog box to set the communications properties:
   - 19200 baud
   - 8 data bits
   - no parity
   - 1 stop bit
   - no flow control
   - select the appropriate communications port
7. Click **OK**.
8. Set the subswitches of user option switch S2 to specify the Picobug firmware module. Set the USR0 and USR2 subswitches in the OFF position, and USR1 subswitch in the ON position.
9. Reconnect power to the CMB2114 and press the reset switch (S3.)

The Picobug monitor starts automatically, displaying the command prompt: `picobug>`.

To use the Picobug monitor, merely enter commands at the prompt. Table 3-1 explains these commands. To see a list of these commands on your computer screen, type a question mark or type `he` at the command prompt.

Table 3-1 Picobug Commands

Command	Explanation
br [*address*]	Breakpoint: • With optional *address* value, sets a new breakpoint at that address. • Without any *address* value, lists all current breakpoints.
g [*address*]	Go: • With optional *address* value, starts code execution from that address. • Without any *address* value, starts code execution from the current program-counter value. In either case, execution stops when it arrives at a breakpoint.
gr	Go to Return: Executes code from the current program-counter value to the return address of the calling routine. (Should execution arrive at a breakpoint before encountering the return address, execution stops at the breakpoint.)
gt [*address*]	Go to Address: Executes code from the current program-counter value to the specified *address* value. (Should execution arrive at a breakpoint before encountering the specified address, execution stops at the breakpoint.)
he	Help Displays available commands, identical to the `?` command.

Table 3-1 Picobug Commands (Continued)

Command	Explanation
lo [*address*]	Download: • With optional *address* value, downloads a binary image to that address in SRAM. • Without any *address* value, downloads to SRAM an S-record text file.
md [*address1* [*address2*]] [;*size*]	Memory Display: • With optional *address1* and *address2* values, displays memory contents between the addresses. • With optional *address1* value, displays contents of 16 memory bytes. • With no address value, defaults to the last address viewed. • The optional *size* value specifies the format: **`b`** (bytes, the default), **`h`** (half words), **`w`** (words), or **`i`** (instructions).
mds [*address*]	Memory Display 256: • With optional *address* value, displays contents of 256 memory bytes, starting at that address. • With no address value, displays contents of 256 memory bytes, starting from the last address viewed.
mm [*address* [*value*]] [;*size*]	Modify Memory: • With optional *address* and *value* parameter values, assigns that value to the *address* location. • With optional *address* value but no *value* parameter value, prompts for a value for the *address* location, then prompts for a new value for the next location. To stop modification, enter a period instead of a new value. • With no optional *address* value, prompts for a value for the last address viewed, then prompts for a new value for the next location. To stop modification, enter a period instead of a new value. • The optional *size* value, specifies the format: **`b`** (bytes, the default), **`h`** (half words), **`w`** (words), or **`i`** (instructions).
nobr [*address*]	No Breakpoint: • With optional *address* value, removes the breakpoint from that address. • Without any *address* value, removes all the breakpoints.
reset	Reset: Resets the CPU and peripherals.
rd [*name*]	Register Display: • With optional *name* value, displays the value of that CPU register. • Without any *name* value, displays the values of all CPU registers.
rm [*name* [*value*]]	Register Modify: Assigns the *value* parameter value to the *name* CPU register.
t	Trace (Step): Single steps one instruction; identical to the `s` command.
s	Step (Trace): Single steps one instruction; identical to the `t` command.
?	Help Displays available commands, identical to the `he` command.

3.1.2 Picobug Sample Session

1. This sample session begins with the Picobug prompt:

```
picobug>
```

2. To see the contents of all registers, enter the Register Display (rd) command without any name value:

```
picobug>     rd
```

The system responds with a display such as this:

```
     pc 8101d0c0        epc fffffffe        fpc 50100002
    psr 80000000       epsr 80000000       fpsr 04000200
ss0-ss4 bad0beef   80010040  02200008  00000100  00c90800        vbr 8100dc00
  r0-r7 bad0beef   817ffffd  80010040  00c30000  00002000  00000000  00000009  8100b000
 r8-r15 81000024   80010040  0000000f  00000080  00cc0004  000000c0  8100e7c4  8001125c
```

3. To see the contents of a specific register, such as the epc register, enter the Register Display (rd) command *with* the name value:

```
picobug>  rd epc
```

The system responds with a display such as this:

```
epc:      FFFFFFFE
```

4. To see the contents of a specific memory location, enter the Memory Display (md) command with the location address. An optional size value (in this case w, for word) may be part of the command:

```
picobug>  md    0x8101d000   ; w
```

The system responds with a display such as this:

```
8101D000:  710B1210
```

5. To see the contents of a memory range, enter the Memory Display (md) command with the beginning and ending addresses. An optional size value (in this case b, for byte) may be part of the command:

```
picobug>  md    0x8101d000   0x8101d016   ; b
```

The system responds with a display such as this:

```
8101D000: 71  0B  12  10  7F  0B  00  00  24  70  9F  00  8F  00  20  70 q.......$p....p
8101D010: 00  CF  00  00  24  70  9F
```

6. To download into SRAM a program executable, in S-record format, enter the Download (lo) command without any address value:

```
picobug>  lo
```

The system waits for you to send the program executable file. To do so, open the Transfer menu and select Send Text File. This opens a file-select dialog box. Use this dialog box to specify the appropriate S-record file, then click the Open button. As soon as the download is complete (this may take several minutes), a confirmation message appears, followed by the Picobug prompt:

```
Done downloading. The target PC is set to 8101d000.
picobug>
```

7. To see the new contents of registers, enter the Register Display (rd) command again, without any name value:

```
picobug>  rd
```

The system responds with an updated display, which shows that the pc register value reflects the start of the program just downloaded:

```
     pc 8101d000        epc fffffffe        fpc 50100002
    psr 80000000       epsr 80000000       fpsr 04000200
ss0-ss4 bad0beef   80010040  02200008  00000100  00c90800        vbr 8100dc00
  r0-r7 bad0beef   817ffffd  80010040  00c30000  00002000  00000000  00000009  8100b000
 r8-r15 81000024   80010040  0000000f  00000080  00cc0004  000000c0  8100e7c4  8001125c
```

8. To set a breakpoint at address 0x8101d11e, enter this address as part of the Breakpoint (br) command:

```
picobug>  br    0x8101d11e
```

The Picobug prompt reappears, confirming that the system set the breakpoint:

```
picobug>
```

9. To see the list of breakpoints, enter the Breakpoint (br) command *without* any address value:

```
picobug>  br
```

The system responds with the addresses of breakpoints, in this case only the breakpoint set in step 8:

```
8101D11E
```

10. To start program execution, enter the Go (g) command:

```
picobug>  g
```

In this instance, the breakpoint set during step 8 stops code execution. The system responds with this new display of register values:

```
At breakpoint!!
```

```
       pc 8101d11e         epc 8101d11e         fpc 50100002
      psr 80000100        epsr 80000100        fpsr 04000200
  ss0-ss4 bad0beef   80010040  02200008   00000100  00c90800         vbr 8100dc00
    r0-r7 8101efd8   8101f000  00000000   00000001  00002000   00000000   00000001  817ffffd
   r8-r15 8101efd8   80010040  0000000f   00000080  00cc0004   000000c0   8100e7c4  8101d056
8101D11E:    B607       stb    r6, (r7)
```

11. To remove all breakpoints, enter the No Breakpoint (nobr) command, without any address value:

```
picobug>  nobr
```

The Picobug prompt reappears, confirming that the system has removed the breakpoints:

```
picobug>
```

12. To see the list of breakpoints again, once more enter the Breakpoint (br) command without any address value:

```
picobug>  br
```

As there are no longer any breakpoints, the system responds with the Picobug prompt:

```
picobug>
```

13. To continue with this example session, enter another appropriate command. For example, to resume program execution, enter the Go (g) command.
14. To end your Picobug session, remove power from the CMB and close the terminal-emulation program.

3.1.3 Using the GNU Source-Level Debugger

The GNU source-level debugger is on the CD-ROM that comes with your CMB2114. The GNU software works with the Picobug monitor to provide source-level debugging for your code.

Install the GNU software by performing the following steps:

1. Insert the MMC2114 CD-ROM into your CD-ROM drive. The install shield should begin automatically. If the install shield does not start automatically, select **Start>Run**. The **Run** dialog box appears. Use the **Run** dialog box to run the `Autorun.exe` file of the CD-ROM.

2. An M•CORE install shield screen appears. Click the **M•CORE Tools** button. This brings up a second install shield screen.
3. To install GNU software, click the second screen's **Install GNU Tools** button, then follow the instructions of successive screens. An installation-complete message appears, indicating successful installation.

Note *Install GNU software into the default directory. Do not browse. Changing the installation directory may give undesirable results.*

4. Click the **OK** in the installation-complete message box to return to the second install shield screen.

After the installation is complete, configure your CMB2114 for using GNU software by performing the following steps:

1. Disconnect power from your CMB2114.
2. Connect the RS-232 cable between CMB2114 connector J58 and the serial port of your computer. If appropriate for your serial port, use the DB9/DB25 adapter.
3. Make sure switches S1 and S2 have the factory settings. Subswitches USR0 and USR2 should be in the OFF position and subswitch USR1 should be in the ON position.
4. Make sure that your +12-volt power supply is turned off or disconnected from line power. Connect the power supply cable to CMB2114 connector J61.
5. Apply power to your CMB2114. Green LED DS6 lights to confirm power.

Next, build a sample application by performing the following steps:

1. Copy the CD-ROM subdirectory `dev_sys\MMC2114\gnusample` to the root directory of your hard disk. For example, the root directory `C:\`.
2. Open an MS-DOS window, and change to the `gnusample` subdirectory of your hard disk.
3. When the MS-DOS prompt returns, type `testgnu` and press the Enter key. This action loads, compiles, and links the sample application. Ignore any "could not find" messages. Leave the MS-DOS window open.

Note: *If the GNU tools do not work after you have installed them, your initial environment space may be lesser than 4096 bytes. The way to make this setting may be different for each system. For assistance, refer to your Windows documentation, to Microsoft technical support, or to Motorola M2CORE technical support.*

To debug the sample application, perform the following steps:

1. At the MS-DOS prompt, type `gdb-mcore testgnu.elf` and press the Enter key.
2. At the GNU Debug (GDB) prompt, type `target picobug com2`. If you are using a different communications port, change this command appropriately. Next, press the Enter key.
3. Ignore any "0x0 in ?? ()" line. At the GDB prompt, type `load testgnu.elf` and press the Enter key.
4. At the GDB prompt, type `list`. Press the Enter key several times to view the application.
5. At the GDB prompt, set several breakpoints. Type:

   `BR 30` (for application line 30) and press Enter;

   `BR 33` and press Enter;

   `BR 36` and press Enter;

   `BR 39` and press Enter.
6. At the GDB prompt, type `run` and press the Enter key. The application executes until it reaches line 30. LED DS5 lights.
7. At the GDB prompt, type the continue command `c` and press Enter. The application executes to line 33 (LED DS4 lights). Type additional continue commands for execution to break at line 36 (LED DS3 lights), line 39 (LED DS2 lights), and line 30 (again).
8. At the GDB prompt, type `BR 41` and press Enter. This sets another breakpoint at the end of the program.
9. At the next four GDB prompts, type `del 1`, `del 2`, `del 3`, and `del 4`, pressing Enter after each command. These four commands delete the breakpoints set during step 5, above.

10. At the GDB prompt, type `c` and presse Enter. The application runs through its normal loop, flashing each LED 10 times before stopping at line 41.
11. To quit the debugger application, type `q` at the GDB prompt and press Enter.

You can now create and debug your own application.

3.2 Using Metrowerks MetroTRK Debugger

Metrowerks Target Resident Kernel (MetroTRK) is a debug monitor for use with applications built with the CodeWarrior™ Integrated Development Environment (IDE). To use MetroTRK Debugger, you need to install the CodeWarrior tools on your computer.

To install the CodeWarrior IDE on your computer:

1. Insert the CodeWarrior CD-ROM into your CD-ROM drive. The CodeWarrior installation starts automatically.
2. Click the **Launch CodeWarrior Setup** button and follow the instructions to install the IDE. Instructions for activating the CodeWarrior license in file License_Readme.txt. (The default path for this file is C:\Program Files\Metrowerks\CodeWarrior.)

After you have installed the CodeWarrior IDE on your computer, follow these steps to create a serial connection and configure the CMB2114 components:

1. Connect the CMB2114 to your computer.
    a. Connect the RS-232 cable between CMB2114 connector J58 and the serial port of your computer. (If appropriate for your serial port, use the DB9/DB25 adapter.)
    b. Make sure that the +12-volt power supply is turned off (or disconnected from line power), and that CMB2114 power switch S4 is set to the OFF position. Connect the power supply cable to CMB/EVB2114 connector J61.
2. Start CMB2114 communication.
    a. Start HyperTerminal program. (If a message asks about installing a modem, click the **No** button.) The **Connection Description** dialog box appears.

b. Enter the connection name and click OK. The **Connect To** dialog box appears.

c. Select the COM port to which you connected your serial cable, then click **OK**. The **Properties** dialog box appears.

d. Set the port properties to 115200 bits per second, 8 data bits, no parity, 1 stop bit, and no flow control.

e. Click **OK**.

3. Configure board components.

a. Set the subswitches of switch S2 to run MetroTRK. Set subswitches USR0 and USR1 to the ON position. Set USR2 subswitch to the OFF position.

b. Set all the remaining switches to their factory settings.

c. Apply power to the board by setting switch S4 to the ON position. Green LED DS6 lights to confirm power.

Once you apply power to the board, The MetroTRK welcome message appears in the Hyper Terminal window. If desired, you can save the HyperTerminal session for later use, and close the HyperTerminal window.

Now you can build and debug the example application by performing the following steps:

1. Copy the sample application to your computer.

a. Insert the M•CORE MMC2114 CD-ROM in your CD-ROM drive. The install screen appears.

b. Click the **Explore the CD** button.

c. Copy the `Dev_Sys\Mmc2114\gnusample` directory to a working directory on your host computer.

2. Start the CodeWarrior IDE by selecting **Start>Program**. Alternatively, double click the `IDE.exe` file in the `CodeWarrior\bin` directory.

3. Create a new project.

a. Select **File>New**. The **New** window appears.

b. Select M•Core EABI 2114 Stationery.

c. Click the **Set** button to select the location of your project file and directory; and specify an appropriate name for the project, such as M2114Led.

d. Click the **Save** button to return to the **New** window.

e. Click **OK**. The **New Project** window appears.

f. Click the **CMB/EVB** control tree to expand it.

g. Click the **TRK** control tree to expand it.

h. Finally, click the **Debugger Channel** control tree to expand it.

i. Select the **C** item in **Debugger Channel** control tree.

j. Click **OK**. The project file window appears. This window displays the following files and folders: `Source`, `MSL`, and `Runtime`.

4. Add files from the gnusample project to the new project.

a. Select **Project>Add Files**. The Select files to add window appears.

b. Navigate to the `gnusample` directory and select `m2114_led.c` file.

c. Click the **Add** button. The `m2114_led.c` file appears in the project window.

d. Click the **Source folder** control tree to expand it.

e. In the project window, drag the `m2114_led.c` file to the `Source` folder.

f. Next, select the `main.c` file in the `Source` folder.

g. Press the **Delete** button. The `main.c` file is deleted.

h. Select **Project>Add Files**. The **Select files to add** window appears.

i. Navigate to the `gnusample` directory and select `main.c` file.

j. Click the **Add** button. The `main.c` file appears in the project window.

k. In the project window, drag the `main.c` file to the `Source` folder.

5. Configure remote debugging and target settings for the new project.

a. Press Alt-F7 to bring up the **Project Settings** window.

b. Select **Remote Debugging** from the **Target Settings Panels** list. The **Remote Debugging** settings panel appears.

c. In the **Remote Debugging** settings panel, make sure that the **Use Memory Configuration File** checkbox is checked.

The text box below should contain filename `MemCfg_CMB2114.txt or MemCfg_EVB2114.txt`, according to your board. These files are in the CodeWarrior layout at the following location: `{CodeWarrior Directory}\MCore_EABI_Support\Debugger Files`.

d. Next, select the **MCore Target Settings** from the **Target Settings Panels** list. The **MCore Target Settings** panel appears.

e. In the **MCore Target Settings** panel, if the **Protocol** list box does not already display MetroTRK, select that value.

6. Check connection settings.

a. Select the **Connection Settings** panel and make sure that the Primary Serial Port Options are set to the port connected to your serial cable, 115200 bits per second, 8 data bits, no parity, 1 stop bit, and no flow control.

b. Click **OK**. The settings are saved and **Target Settings** window disappears.

7. Debug the project.

a. Select **Project>Enable Debugger**.

b. Press function key F5 to build the project and start the debugger. The debugger window appears.

c. In the debugger window, click **Run** to run the program. LEDs DS5—DS2 on the board flash in sequence 10 times, stopping with DS2.

8. End the program.

Click the **Kill** button to end the program and close the debugger window.

You have just set up your CMB2114 and run a simple application. For more details on how to use the CodeWarrior IDE, refer to the relevant documentation supplied with the CodeWarrior tools.

3.3 Using the SysDS Loader

Motorola's SysDS Loader is on the CD-ROM that comes with your CMB2114. There are two versions of SysDS Loader present on the CD-ROM: one for on-chip FLASH programming and the other for on-board FLASH programming. The install shield installs both the versions of SysDS Loader on your computer in different directories. The subsections that follow explain you how to install and use both the versions.

3.3.1 Installing SysDS Loader

Follow these steps to install the both versions of SysDS Loader on your computer:

1. Start install shield.

   Insert the MMC2114 CD-ROM into your CD-ROM drive. The install shield should begin automatically. If the install shield does not start automatically, select **Start>Run** on your desktop. Use the **Run** dialog box to run the `Autorun.exe` file of the CD-ROM.

Note *Be sure to use the install shield to install the SysDS Loader. Merely copying the SysDS Loader files from the MMC2114 CD-ROM does not install the program correctly.*

2. An M•CORE install shield screen appears. Click the MMC2114 button on this screen. This brings up a new screen that includes **Install Device Drivers**, **Install SysDS Loader** and **Read Me** buttons.
3. To install the SysDS loader, click the **Install SysDS Loader** button, then follow the instructions on successive screens. After installation is complete, a copy-successful message appears.

Note *The install shield installs both versions of SysDS loader on the following location on your computer:* `C:\Motorola\loader\2114`. *Within the* `2114` *directory there are two other directories:* `On Chip` *and* `On Board`. *The* `On Chip` *and* `On Board` *directories contain the on-chip and on-board FLASH programming versions of SysDS Loader, respectively.*

4. Click the OK button of the copy-successful message box to return to the first install shield screen.
5. To see the CMB2114 readme file, click again on the MMC2114 button, then click the Read Me button.

When you are done with installation activities, click the Exit button of any screen.

3.3.2 Using On-Chip FLASH Programming Version of SysDS Loader

The Motorola SysDS Loader for on-chip FLASH programming lets you:

- Program code into on-chip FLASH memory
- Upload on-chip FLASH contents to a PC file
- Verify that on-chip FLASH contents match those of a download file
- Display memory contents
- Erase on-chip FLASH memory
- Erase a valid single bank of on-chip FLASH memory
- Erase a valid page of on-chip FLASH memory
- Blank check on-chip FLASH memory
- Write the security word of user entry
- Write the back door keyword of user entry
- Select an on-chip FLASH type (SGFM_128K or SGFM_256K)

Notes *For the first action of an SysDS Loader session (downloading, verifying, displaying, erasing, or blank checking), the software will download an algorithm file before carrying out the action.*

If the software cannot find the algorithm file, an appropriate error message identifies the file. Click the message's OK button to bring up a file-select dialog box, then use this dialog box to specify the location of the algorithm file. If necessary, recopy the file from the transmittal CD-ROM. Click the OK button to resume your SysDS Loader action.

Follow these steps to use the on-chip FLASH programming version of SysDS Loader:

1. In case the Hyperterminal program is using the same COM port being used by EBDI, stop Hyperterminal. The SysDS Loader needs the same computer serial port that Hyperterminal uses.
2. Set the subswitches of user option switch S1 to the following positions:
   – Boot Ex/In Subswitch is OFF — board boots from internal-FLASH memory

- Data 32/16 subswitch is ON — specifies 32-bit external data bus
- Swap 02/20 subswitch is ON — configures the board for CS0 control of external FLASH and CS2 control of external SRAM
- M0 subswitch is ON — configures the board to run in master mode

3. Set the M1 subswitch of user option switch S2 to the ON position. This configures the board to run in master mode.
4. Specify clock reference frequency (in hex) in the `loader.ini` file.
    a. Navigate to the following directory on your computer's hard disk where you installed SysDS Loader: `Motorola/loader/On Chip`. This location contains the `loader.ini` file.
    b. Double-click the `loader.ini` file. The file opens.
    c. For the `ClockRef` parameter in this file, specify the clock frequency of the board in hex. The default `ClockRef` parameter is 3D0900 (4-megahertz).
    d. Close the `loader.ini` file.
5. Press the reset switch, S3, to reset the CMB2114.
6. Start the SysDS Loader.
    a. Navigate to the following directory on your computer's hard disk where you installed SysDS Loader: `Motorola/loader/On Chip`. This location contains an executable file `loader.exe`.
    b. Double-click the `loader.exe` file. SysDS Loader starts and the main screen appears as shown in Figure 3-1.

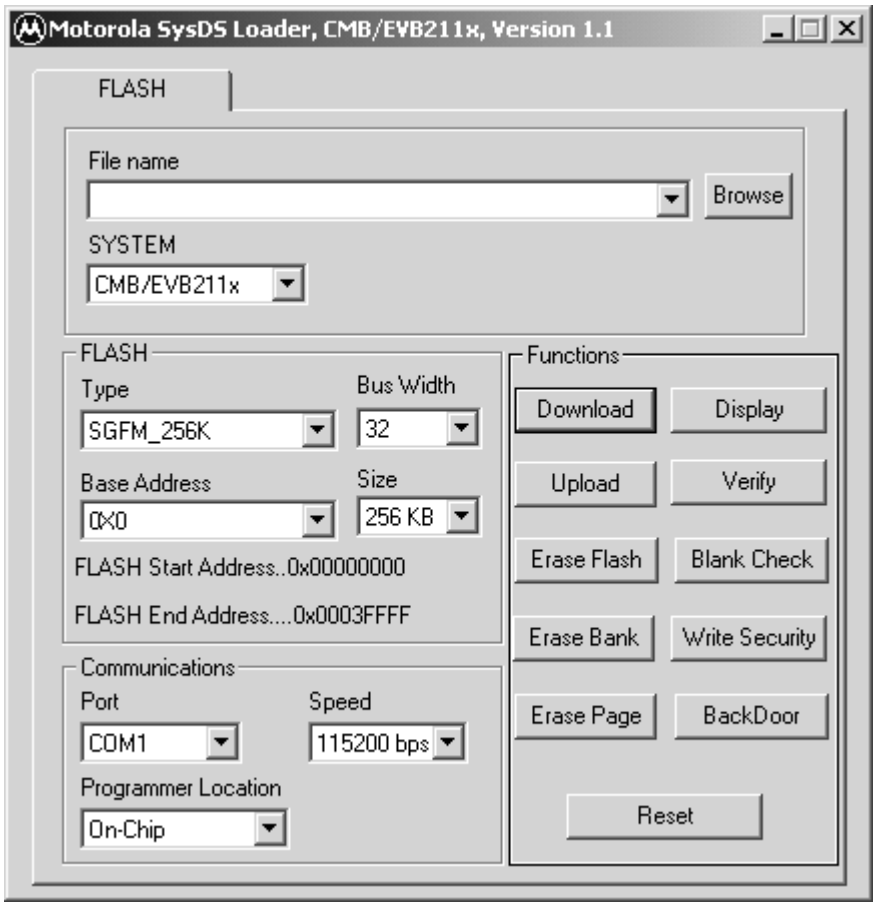


Figure 3-1 On Chip FLASH Version SysDS Loader Main Screen

7. Specify file name.

Note *If your only action for this Loader session will be uploading FLASH contents, you may leave the File name field blank.*

- If you know the full path name of the file to be programmed, enter the path name in the **File name** field.
- If you do not know the full pathname of the file to be programmed:
  a. Click the **Browse** button. This brings up a standard file-select dialog box.

b. In the file-select dialog box, select the file and click **OK**. This returns you to the main screen. The pathname appears in the **File name** field.

8. Use the **FLASH** area of the main screen to select the FLASH type. To program FLASH, make sure to specify that value in the **Type** list box.

   The bus width, bus size, and base address values are automatic and depend on the FLASH type selected. However, for base address, you may select the optional value <CUSTOM>, which brings up the **Custom Address** dialog box. Enter an appropriate address, then click the dialog box OK button to return to the main screen.

9. In the **Communications** area of the main screen, use the **Port** field to specify the PC serial port, and use the **Speed** field to specify the communications rate. The default rate is 115200 baud.

10. To program FLASH memory, click the **Download** button on the main screen. As the software downloads the file you specified, a progress message appears in a Status dialog box. A message informing you that the download has been completed successfully appears at the end of downloading.

11. To upload FLASH memory contents to a file in your PC, click the **Upload** button. The **Upload To File** dialog box appears, as shown in Figure 3-2.

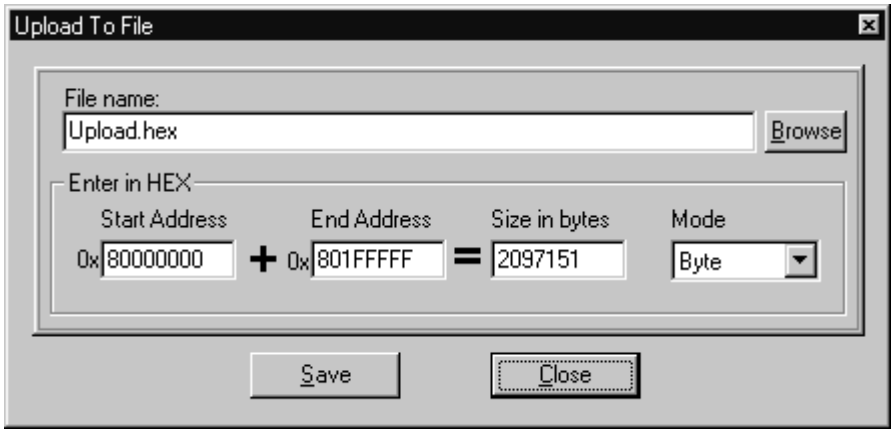


Figure 3-2 Upload To File Dialog Box

- Type the name of the destination file in the **Upload to File** dialog box. Optionally, you can click the **Browse** button, to select a file via a standard file-select dialog box.
- The **Start Address** field of the **Upload to File** dialog box indicates the start of CMB2114 FLASH memory or RAM. The default address value corresponds to the value of the SYSTEM field of the main screen, but you may enter a different address, if appropriate.
- Enter the appropriate value in the **End Address** field of the **Upload to File** dialog box. The system automatically determines the value and displays it in the **Size in Bytes** field of the **Upload to File** dialog box.
- The value displayed in the **Size in Bytes** field of the **Upload to File** dialog box corresponds to the value of the **Size** field of the main screen. If appropriate, you may enter a different value.
- The default value in the **Mode** field of the **Upload to File** dialog box is Byte.
- When the **Upload To File** dialog box shows appropriate values, click the **Save** button in this dialog box. A progress message appears during uploading.

12. To verify that the contents of FLASH memory match the selected download file, click the **Verify** button on the main screen. A progress message appears as verification begins. A message informing you that the verification is successful appears at the end of verification.
    - If verification fails, an error message specifies the location that did not have the expected contents.
    - To recover from a verification failure, try downloading FLASH again, to replace the selected download file.
13. To view the contents of FLASH memory, click the **Display** button on the main screen. The Display Flash/Ram screen appears, as shown in Figure 3-3.

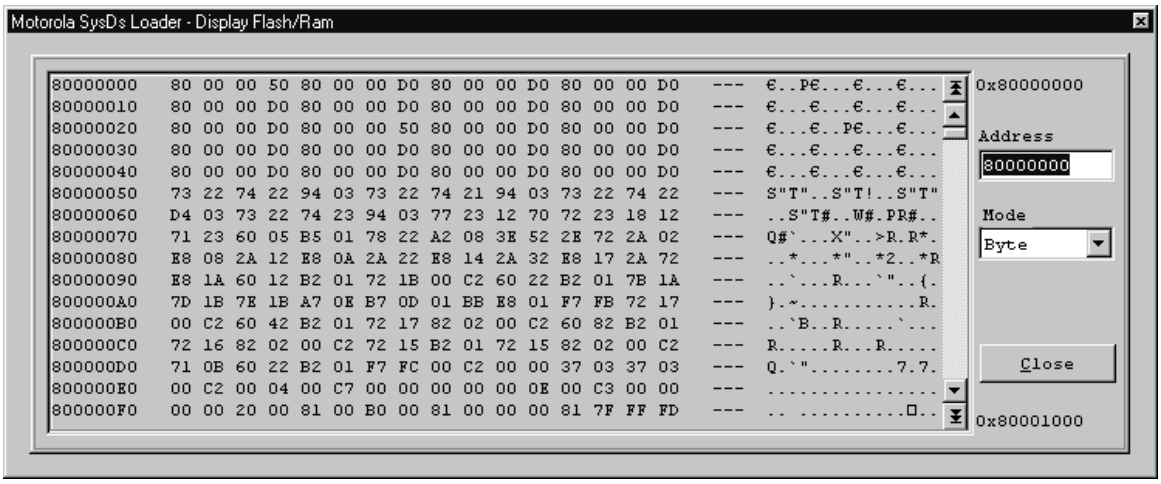


Figure 3-3 Display Flash/Ram Screen

- The Address field of the Display Flash/RAM screen shows the first address of the value display. One way to change the display is to enter a different address in this field.
- Another way to change the value display is to use the scroll bars.
- Use the Mode field of the Display Flash/RAM screen to specify byte, half-word, or word values in the display.
- When you are done viewing the display, click the **Close** button to return to the main screen.

14. To erase FLASH memory, click the **Erase Flash** button on the main screen. The SysDS Loader erases all contents of the FLASH memory depending on the FLASH type you selected. The FLASH configuration field area is also erased.
15. To erase a single bank of FLASH memory, click the **Erase Bank** button on the main screen. This brings up a dialog box where you can enter the number of the bank to be erased, then click **OK**.
16. To erase a particular page of FLASH memory, click the **Erase Page** button. This brings up a dialog box where you can type the address (in HEX) of the page to be erased, and click **OK**.
17. To verify if a page or bank in FLASH memory is blank:

a. Click the **Blank Check** button on the Main Screen. This brings up the **Blank Check Selection** dialog box.

b. If you want to check for a blank bank, type 0 and click **OK**. Alternatively, if you want to check for a blank page, type 1 and click **OK**. A dialog box will appear where you can specify the bank number or page address.

c. Type the bank number or page address and click **OK**. A message tells you the results of the blank check.

18. To write a security word, click the **Write Security** button on the main screen. A pre-determined security word is written at the appropriate SGFM configuration address.

19. To write a back door entry key, click the **Back Door** button. A back door entry key is written at the appropriate SGFM configuration address.

20. To end your SysDS Loader session, close the main screen.

3.3.3 Using On-Board FLASH Programming version of SysDS Loader

The on-board FLASH programming version of SysDS Loader lets you:

- Program code into FLASH memory
- Upload FLASH contents to a PC file
- Verify that FLASH contents match those of a download file
- Display memory contents
- Erase FLASH memory or erase a sector of FLASH memory
- Blank check a sector of FLASH memory

To use the on-board FLASH programming version of SysDS Loader, perform the following steps:

1. In case the Hyperterminal program is running on your computer, stop the program. The SysDS Loader needs the same computer serial port that Hyperterminal uses.

2. Set the subswitches of user option switch S2 to select the Programmer firmware module. Set the USR0 subswitch to the ON position and set the USR1 and USR2 subswitches to the OFF position.

3. Press the reset switch, S3, to reset the CMB2114.

4. Navigate to the following directory on your computer's hard disk where you installed SysDS Loader: `Motorola/loader/On Board.` This location contains an executable file `loader.exe`.
5. Double-click the `loader.exe` file. SysDS Loader starts and the main screen appears as shown in Figure 3-4.

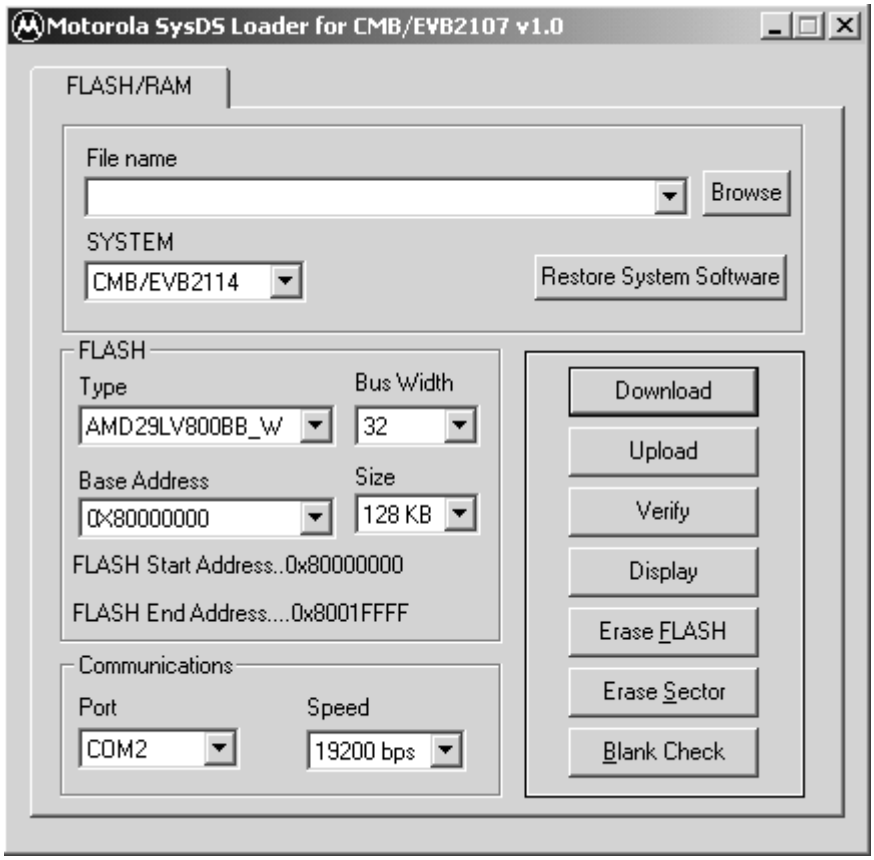


Figure 3-4 On Board SysDS Loader Version Main Screen

6. Specify file name.

Note *If your only action for this Loader session will be uploading FLASH contents, you may leave the File name field blank.*

- If you know the full pathname of the file to be programmed, enter the path name in the **File name** field.
- If you do not know the full path name of the file to be programmed:

a. Click the **Browse** button. This brings up a standard file-select dialog box.

b. In the file-select dialog box, select the file and click **OK**. This returns you to the main screen. The pathname appears in the **File name** field.

7. Use the **FLASH** area of the main screen to configure the FLASH type, bus width, and size. To program FLASH, make sure to specify that value in the **Type** field.

   The value in the Base Address field is automatic. However, you may select the optional value <CUSTOM>, which brings up the **Custom Address** dialog box. Enter an appropriate address, then click the dialog box OK button to return to the main screen.

8. In the **Communications** area of the main screen, use the **Port** field to specify the PC serial port, and use the **Speed** field to specify the communications rate. The default rate is 19200 baud.

9. To program FLASH memory, click the **Download** button on the main screen. As the software downloads the file you specified, a progress message appears in a Status dialog box. A message informing you that the download has been completed successfully appears at the end of downloading.

   The error message "Unable to Validate Flash configuration" indicates some problem with the programming. One such problem could be that the chip select base address does not correspond to the configured chip select. Correct the problem, then click the **Download** button again.

10. To upload FLASH memory contents to a file in your PC, click the **Upload** button. The **Upload To File** dialog box appears, as shown in Figure 3-5.

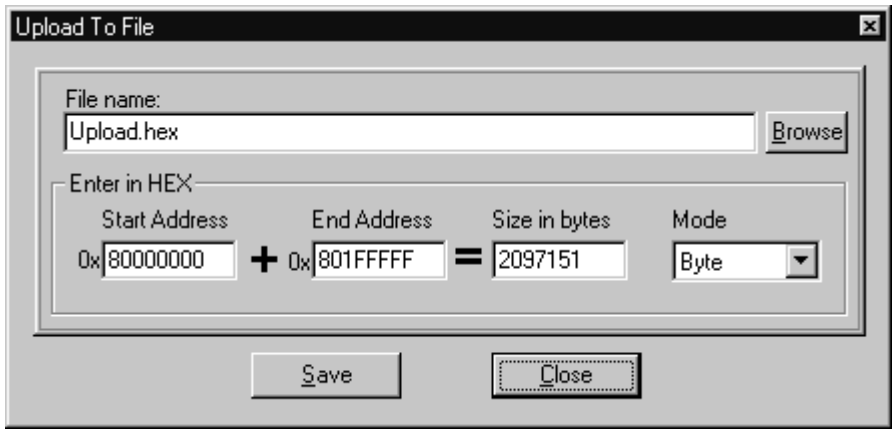


Figure 3-5 Upload To File Dialog Box

- Type the name of the destination file in the **Upload to File** dialog box. Optionally, you can click the **Browse** button, to select a file via a standard file-select dialog box.
- The **Start Address** field of the **Upload to File** dialog box indicates the start of CMB2114 FLASH memory or RAM. The default address value corresponds to the value of the SYSTEM field of the main screen, but you may enter a different address, if appropriate.
- Enter the appropriate value in the **End Address** field of the **Upload to File** dialog box. The system automatically determines the value and displays it in the **Size in Bytes** field of the **Upload to File** dialog box.
- The value displayed in the **Size in Bytes** field of the **Upload to File** dialog box corresponds to the value of the **Size** field of the main screen. If appropriate, you may enter a different value.
- The default value in the **Mode** field of the **Upload to File** dialog box is Byte.
- When the **Upload To File** dialog box shows appropriate values, click the **Save** button in this dialog box. A progress message appears during uploading.

11. To verify that the contents of FLASH memory match the selected download file, click the **Verify** button on the main screen. A progress message appears as verification begins. A message informing you that the verification is successful appears at the end of verification.
    - If verification fails, an error message specifies the location that did not have the expected contents.
    - To recover from a verification failure, try downloading FLASH again, to replace the selected download file.
12. To view the contents of FLASH memory, click the **Display** button on the main screen. The Display Flash/Ram screen appears, as shown in Figure 3-6.

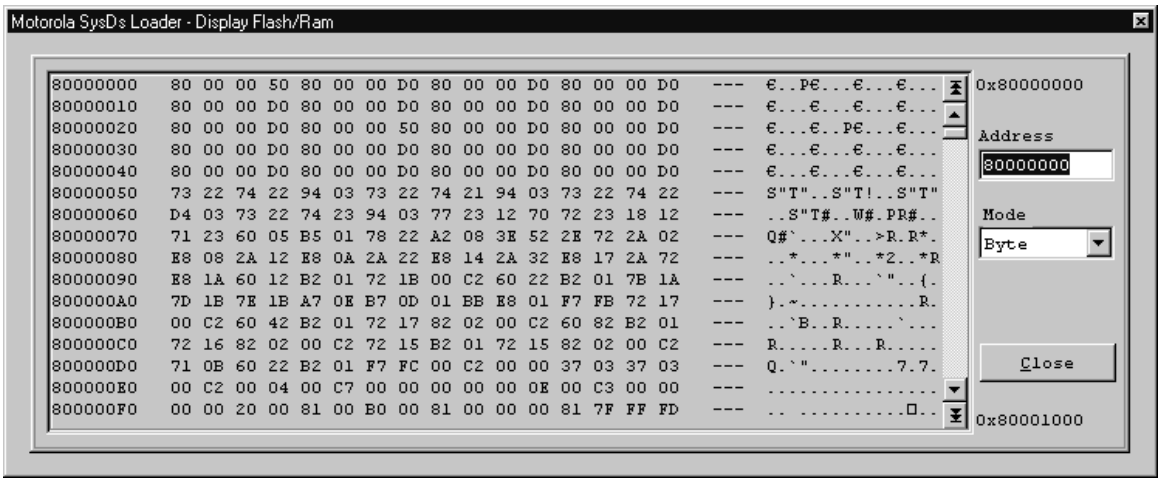


Figure 3-6 Display Flash/Ram Screen

- The Address field of the Display Flash/RAM screen shows the first address of the value display. One way to change the display is to enter a different address in this field.
- Another way to change the value display is to use the scroll bars.
- Use the Mode field of the Display Flash/RAM screen to specify byte, half-word, or word values in the display.
- When you are done viewing the display, click the **Close** button to return to the main screen.

13. To erase FLASH memory, click the **Erase Flash** button on the main screen. The SysDS Loader erases all contents of the FLASH memory except for the sectors that contain system software.
14. To erase a particular sector of FLASH memory, click the **Erase Sector** button in the main screen. This brings up the **Flash Sector Number** dialog box. In this dialog box, specify the number of the sector to be erased and click **OK**.

Note *To avoid erasing sectors 0 through 3, which contain system software, make sure that the sector number you specify for erasing is 4 or greater.*

15. To verify if a page or bank in FLASH memory is blank:
    a. click the **Blank Check** button on the main Screen. This brings up a dialog box that asks for a sector number.
    b. Type the number of the sector to be blank checked.
    c. Click **OK**. A message tells you the results of the blank check. If the sector is not blank, you can erase the sector or try a different sector.
16. To end your SysDS Loader session, close the main screen.

Section 4. Connector Information

4.1 MAPI Connectors

Connectors P1 through P4, all 2-by-50-pin connectors, are the CMB2114 MAPI connectors. Connectors J1 through J4, on the bottom of the CMB2114, have the same pin assignments. Figure 4-1 shows the orientation of the CMB2114 MAPI connectors. Figure 4-2 through Figure 4-5, and Table 4-1 through Table 4-4, give the pin assignments and signal descriptions for these connectors.

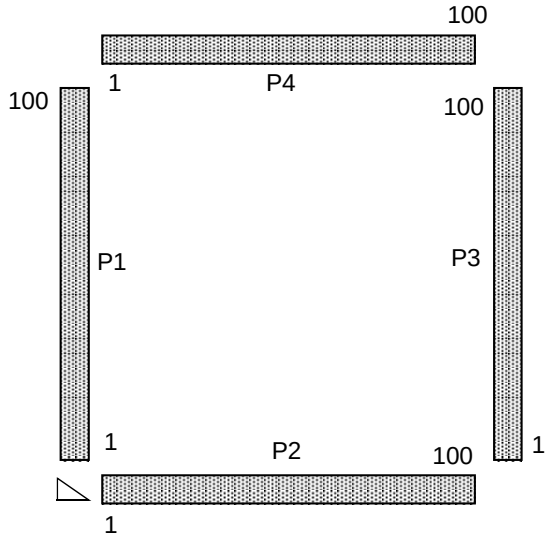


Figure 4-1 MAPI Connectors Orientation

P1/J1

PTJ1[100]	100	• •	99	VDD3V
PTJ1[98]	98	• •	97	PTJ1[97]
PTJ1[96]	96	• •	95	PTJ1[95]
PTJ1[94]	94	• •	93	PTJ1[93]
GND	92	• •	91	PTJ1[91]
VDD5V	90	• •	89	GND
PTJ1[88]	88	• •	87	PTJ1[87]
PTJ1[86]	86	• •	85	PTJ1[85]
PTJ1[84]	84	• •	83	PTJ1[83]
PTJ1[82]	82	• •	81	PTJ1[81]
PTJ1[80]	80	• •	79	PTJ1[79]
PTJ1[78]	78	• •	77	PTJ1[77]
PTJ1[76]	76	• •	75	PTJ1[75]
GND	74	• •	73	DEVSP_B[0]
PTJ1[72]	72	• •	71	GND
PTJ1[70]	70	• •	69	PTJ1[69]
INT_B[6]	68	• •	67	INT_B[7]
INT_B[4]	66	• •	65	INT_B[5]
INT_B[2]	64	• •	63	INT_B[3]
INT_B[0]	62	• •	61	INT_B[1]
VDD3V	60	• •	59	VDD3V
IDVDD (MID0)	58	• •	57	ICOC2[3]
ICOC2[2]	56	• •	55	ICOC2[1]
ICOC1[3]	54	• •	53	ICOC2[0]
IDVDD (MID1)	52	• •	51	ICOC1[2]
ICOC1[1]	50	• •	49	ICOC1[0]
SCK	48	• •	47	PTJ1[47]
GND	46	• •	45	PTJ1[45]
MOSI	44	• •	43	PTJ1[43]
MISO	42	• •	41	GND
GND (MID2)	40	• •	39	PTJ1[39]
SS_B	38	• •	37	PTJ1[37]
PTJ1[36]	36	• •	35	PTJ1[35]
IDVDD (MID3)	34	• •	33	PTJ1[33]
PTJ1[32]	32	• •	31	PTJ1[31]
PTJ1[30]	30	• •	29	PTJ1[29]
GND	28	• •	27	PTJ1[27]
GND1	26	• •	25	GND1
PTJ1[24]	24	• •	23	PTJ1[23]
PTJ1[22]	22	• •	21	PTJ1[21]
PTJ1[20]	20	• •	19	PTJ1[19]
PTJ1[18]	18	• •	17	PTJ1[17]
PTJ1[16]	16	• •	15	PTJ1[15]
PTJ1[14]	14	• •	13	PTJ1[13]
GND1	12	• •	11	PTJ1[11]
AGND	10	• •	9	PTJ1[9]
AGND	8	• •	7	PTJ1[7]
AGND	6	• •	5	MAPIVRH
AGND	4	• •	3	MAPIVRL
AGND	2	• •	1	PQA[0]

Figure 4-2 MAPI Connector P1/J1 Pin Assignments

Table 4-1 MAPI Connector P1/J1 Signal Descriptions

Pin	Mnemonic	Signal
100, 98 — 93, 91, 88 — 75, 72, 70, 69, 47, 45, 43, 39, 37 — 35, 33 — 29, 27, 24 — 13, 11, 9, 7	PTJ1[x]	Pass through.
99, 60, 59,	VDD3V	+3.3-volt power
92, 89, 74, 71, 46, 41, 28	GND	GROUND
90	VDD5V	+5-volt power.
73	DEVSP_B[0]	DEVELOPMENT SPACE (line 0) — Active-low signal indicating that the current memory cycle is addressing on-board devices.
68 — 61	INT_B[7] — INT_B[0] (not in exact order)	EXTERNAL INTERRUPT (lines 7—0) — Active-low lines for external interrupts or general-purpose I/O. In addition, certain lines can show processor core signal states: • INT_B[7:6]: reflects the states of TSIZ[1:0] signals, provided that the chip configuration register (CCR) SZEN bit is set. • INT_B[5:2]: reflects the states of PSTAT[3:0] signals, provided that the CCR PSTEN bit is set.
34, 52, 58	IDVDD (MID0, MID1 MID3)	IDENTIFICATION POWER — Special 3-volt MAPI identification code (MID) bits 0, 1, and 3 for the identification code signals.
57 — 55, 53	ICOC2[3] — ICOC2[0]	TIMER 2 INPUT CAPTURE OUTPUT CAPTURE (lines 3—0) — Signals for internal timer channel 2.
54, 51 — 49	ICOC1[3] — ICOC1[0]	TIMER 1 INPUT CAPTURE OUTPUT CAPTURE (lines 3—0) — Signals for internal timer channel 1.
40	GND (MID2)	GROUND. MID bit 2 — signal that identifies the host processor board.
48	SCK	SERIAL CLOCK — Synchronization signal for master-slave communication: an output if SPI is configured as master, an input if SPI is configured as slave.
44	MOSI	MASTER OUT/SLAVE IN — If SPI is enabled, the data master-out/slave-in signal. If SPI is disabled, a general-purpose port E I/O signal.
42	MISO	MASTER IN/SLAVE OUT — If SPI is enabled, the data master-in/slave-out signal. If SPI is disabled, a general-purpose port E I/O signal.
38	SS_B	SLAVE SELECT — Active-low slave select signal, in slave mode. In master mode, a peripheral chip-select signal.
26, 25, 12	GND1	GROUND — Connection to the GROUND 1 plane.
10, 8, 6, 4, 2	AGND	ANALOG GROUND — Analog ground connection for the analog-digital converter.
5	MAPIVRH	MAPI VOLTAGE REFERENCE HIGH — High reference for voltage supplied via the MAPI ring.
3	MAPIVRL	MAPI VOLTAGE REFERENCE LOW — Low reference for voltage supplied via the MAPI ring.
1	PQA[0]	A ANALOG INPUT (line 0) — A analog input to the QADC, also usable for general-purpose digital I/O.

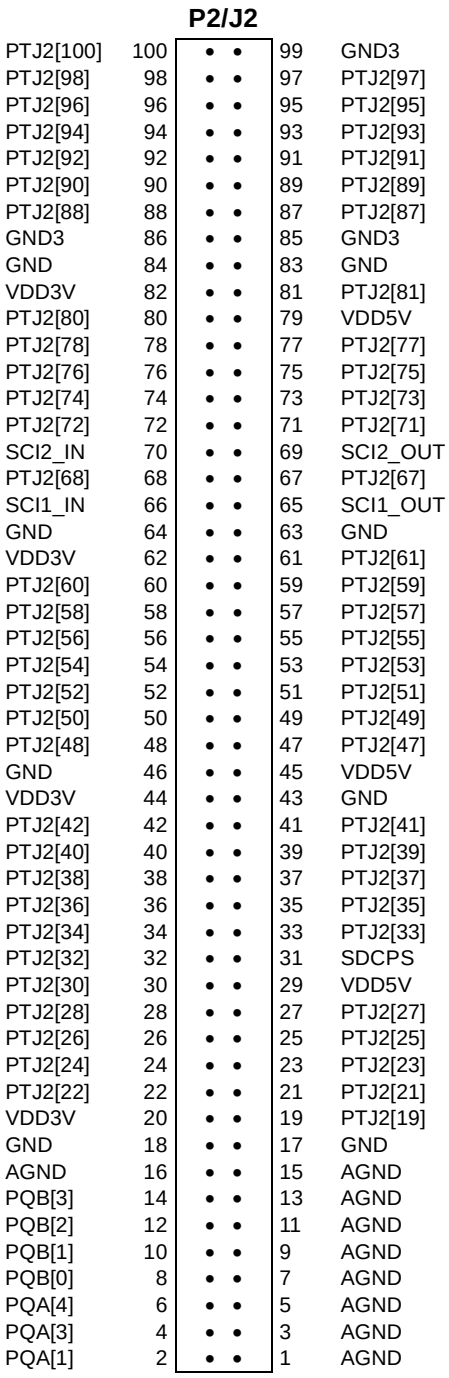


Figure 4-3 MAPI Connector P2/J2 Pin Assignments

Table 4-2 MAPI Connector P2/J2 Signal Descriptions

Pin	Mnemonic	Signal
100, 98 — 87, 81, 80, 78—71, 68, 67, 61—47, 42—32, 30, 28—21, 19	PTJ2[x]	Pass through.
99, 86, 85,	GND3	GROUND — Connection to the GROUND 3 plane.
84, 83, 64, 63, 46, 43, 18, 17	GND	GROUND
82, 62, 44, 20	VDD3V	+3.3-volt power
79, 45, 29	VDD5V	+5-volt power.
70, 65	SCI2_IN, SCI1_IN	SCI INPUT — Serial communications interface (SCI) input lines 2 and 1, otherwise available for general-purpose I/O use. (These lines also are known as RXDB and RXDA.)
69, 65	SCI2_OUT, SCI1_OUT	SCI OUTPUT — SCI output lines 2 and 1, otherwise available for general-purpose I/O use. (These lines also are known as TXDB and TXDA.)
31	SDCPS	SHUT DOWN CMB POWER SUPPLY — Input signal. If low, disables the 5-volts power supply. The 3.3-volt power supply remains operational.
16, 15, 13, 11, 9, 7, 5, 3, 1	AGND	ANALOG GROUND — Analog ground connection for the analog-digital converter.
14, 12, 10, 8	PQB[3] — PQB[0]	B ANALOG INPUTS (lines 3—0) — B analog inputs to the queued analog to digital converter (QADC), also usable as general-purpose digital inputs.
6, 4, 2	PQA[4], PQA[3], PQA[1]	A ANALOG INPUTS (lines 4, 3, 1) — A analog inputs to the QADC, also usable for general-purpose digital I/O.

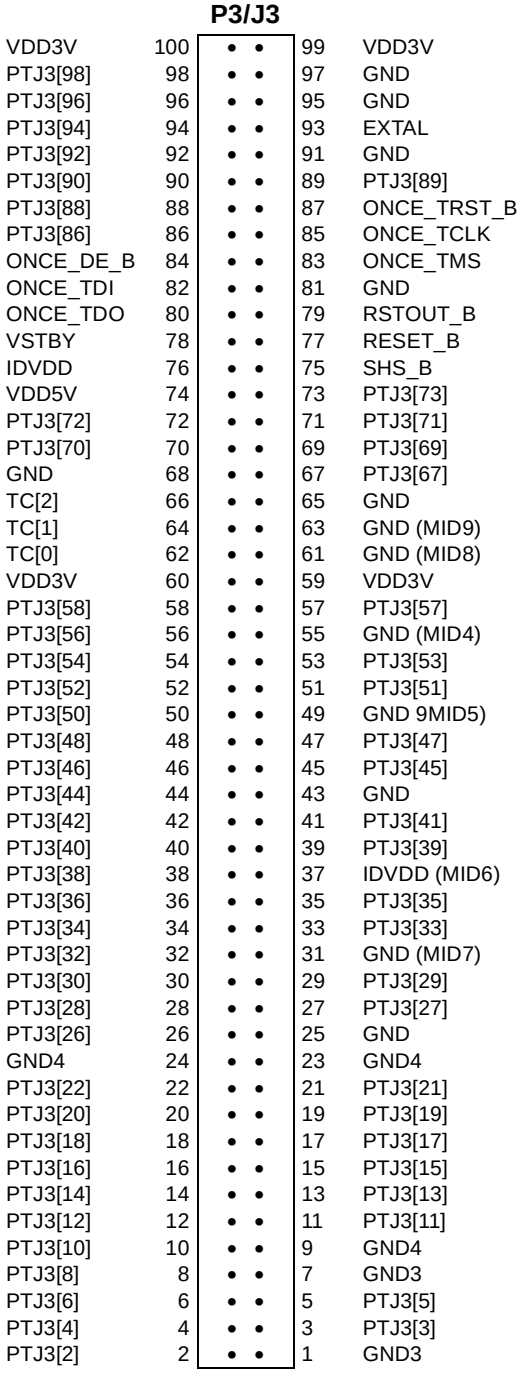


Figure 4-4 MAPI Connector P3/J3 Pin Assignments

Table 4-3 MAPI Connector P3/J3 Signal Descriptions

Pin	Mnemonic	Signal
100, 99, 60, 59,	VDD3V	+3.3-volt power
98, 96, 94, 92, 90 — 88, 86, 73 — 69, 67, 58 — 56, 54 — 50, 48 — 44, 42 — 38, 36 — 32, 30 — 26, 22 — 10, 8, 6 — 2	PTJ3[x]	Pass Through.
97, 95, 91, 81, 68, 65, 43, 25	GND	GROUND
93	EXTAL	EXTERNAL CLOCK — Off-board clock signal.
87	ONCE_TRST_B	OnCE TEST RESET – Active-low input that asynchronously initializes JTAG and On Chip Emulation (OnCE) logic.
85	ONCE_TCLK	OnCE TEST CLOCK – Input signal that synchronizes JTAG and OnCE logic.
84	ONCE_DE_B	OnCE DEBUG EVENT – Open-drain, active-low debug signal, via the OnCE connector. If an input signal from an external command controller, causes the processor to enter debug mode. If an output signal, acknowledges that the MCU is in debug mode.
83	ONCE_TMS	OnCE TEST MODE SELECT – Input signal that sequences the JTAG test controller's state machine, sampled on the rising edge of the ONCE_TCLK signal.
82	ONCE_TDI	OnCE TEST DATA INPUT – Serial input for JTAG test instructions and data, sampled on the rising edge of the ONCE_TCLK signal.
80	ONCE_TDO	OnCE TEST DATA OUTPUT – Serial output for JTAG test instructions and data. Tri-stateable and actively driven in the Shift-IR and Shift-DR controller states, this signal changes on the falling edge of the ONCE_TCLK signal.
79	RSTOUT_B	RESET OUT – Active-low output signal, controlled by the processor, that resets external components. Activation of any internal reset sources asserts this line.
78	VSTBY	STANDBY POWER — Standby power source for the RAM array, should main power (VDD) be lost.
77	RESET_B	RESET IN – Active-low input signal that starts a system reset: a reset of the MMC2114 device and most peripherals.
76, 37	IDVDD	IDENTIFICATION POWER — Special 3-volt power signals (pin 37 also is MID6) for the identification code signals.
75	SHS_B	SHOW CYCLE STROBE — Active-low, output strobe signal for capturing addresses, controls, and data during show cycles. Emulation mode forces this signal active. In master mode, software must enable this signal.
74	VDD5V	+5-volt power.
63, 61, 55, 49, 31	GND (MID9, MID8, MID4, MID5, MID7)	GROUND. MID bits 9, 8, 4, 5, and 7 — signals that identify the host processor board.
66, 64, 62	TC[2] — TC[0]	TRANSFER CODE (lines 2—0) — Outputs indicating the data transfer code for the current bus cycle.
24, 23, 9	GND4	GROUND — Connection to the GROUND 4 plane.
7, 1	GND3	GROUND — Connection to the GROUND 3 plane.

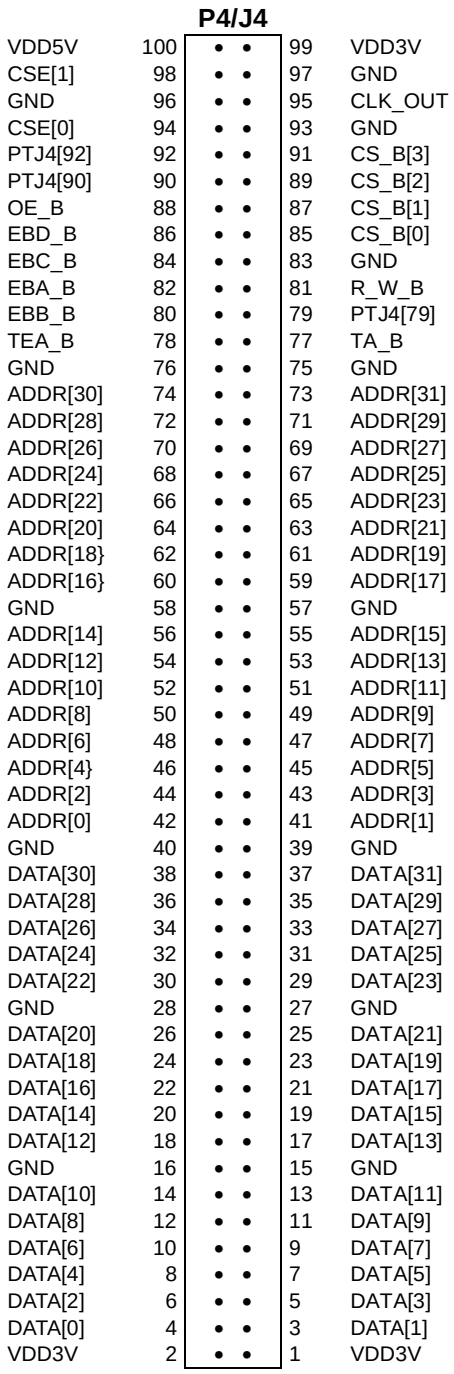


Figure 4-5 MAPI Connector P4/J4 Pin Assignments

Table 4-4 MAPI Connector P4/J4 Signal Descriptions

Pin	Mnemonic	Signal
100	VDD5V	+5-volt power.
99, 2, 1	VDD3V	+3.3-volt power.
98, 94	CSE1, CSE0	EMULATION CHIP SELECTS (lines 1, 0) — Emulation-mode output chip-select signals.
97, 96, 93, 83, 76, 75, 58, 57, 40, 39, 28, 27, 16, 15	GND	GROUND
95	CLK_OUT	CLOCK OUTPUT — System clock output.
92, 90, 79	PTJ4[x]	Pass Through
91, 89, 87, 85	CS_B[3] — CS_B[0]	CHIP SELECTS (lines 3—0) — Active-low output lines that provide chip selects to external devices.
88	OE_b	OUTPUT ENABLE — Active-low output that indicates that a bus access is a read access; enables slave devices to drive the data bus.
86, 84, 82, 80	EBD_B, EBC_B, EBA_B, EBB_B	ENABLE BYTES D, C, B, A — Active-low outputs active during an operation to corresponding data bits (D31-D24 for enable byte D, D23-D16 for enable byte C, D15-D8 for enable byte B, D7-D0 for enable byte A.)
81	R_W_B	READ/WRITE ENABLE — Active-low signal indicating that the current bus access is a write access. Otherwise, the current bus access is a read access.
78	TEA_B	TRANSFER ERROR ACKNOWLEDGE — Active-low input that indicating that a bus transfer error has occurred.
77	TA_B	TRANSFER ACKNOWLEDGE — Active-low input indicating completion of a data transfer, for either a read or a write cycle.
74 — 59, 56—41	ADDR[31] — ADDR[0] (not in exact order)	ADDRESS BUS (lines 31—0) — Output lines for addressing external devices. These lines change state only during external-memory accesses.
38—29, 26—17, 14—3	DATA[31] — DATA[0] (not in exact order)	DATA BUS (lines 31–0) — Bi-directional data lines for accessing external memory. A hardware reset or no external-bus activity hods these lines in their previous logic state.

4.2 CPLD Programming Connector

Connector J6 is an eight pin connector used for in-system programming (ISP) of the Complex Programmable Logic Device (CPLD). You connect the Xilinx provided, JTAG compliant ISP cable between connector J6 and the serial or parallel port of the host PC. Figure 4-6 and Table 4-5 show the pin assignments and signal descriptions of connector J6, respectively.

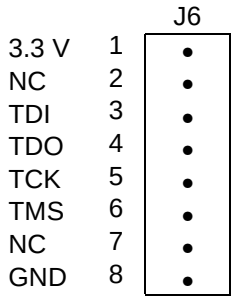


Figure 4-6 CPLD Programming Connector J6 Pin Assignments

Table 4-5 CPLD Programming Connector J6 Signal Descriptions

Pin	Mnemonic	Signal
1	3.3 V	3.3-volt power
2, 7	NC	No connection
3	TDI	TEST DATA INPUT – Serial data input to CPLD.
4	TDO	TEST DATA OUTPUT – Serial output from CPLD.
5	TCK	CPLD TEST CLOCK — Input signal for synchronization.
6	TMS	MODE SELECT – Test mode select input to Test Access Port (TAP) controller on CPLD.
8	GND	Ground

4.3 OnCE Connector

Connector J7, a 2x7-pin connector, conveys data and control signals to and from the OnCE control block. Figure 4-7 and Table 4-6 give the pin assignments and signal descriptions for this connector.

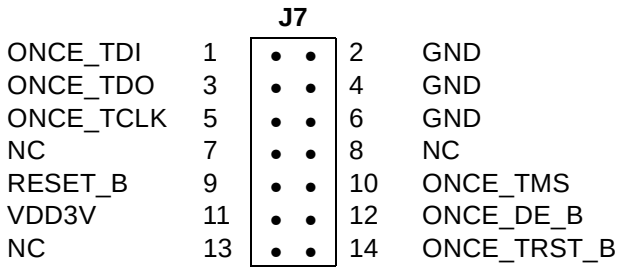


Figure 4-7 OnCE Connector J7 Pin Assignments

Table 4-6 OnCE Connector J7 Signal Descriptions

Pin	Mnemonic	Signal
1	ONCE_TDI	OnCE TEST DATA INPUT – Serial input for JTAG test instructions and data, sampled on the rising edge of the ONCE_TCLK signal.
2, 4, 6	GND	GROUND
3	ONCE_TDO	OnCE TEST DATA OUTPUT – Serial output for JTAG test instructions and data. Tri-stateable and actively driven in the Shift-IR and Shift-DR controller states, this signal changes on the falling edge of the ONCE_TCLK signal.
5	ONCE_TCLK	OnCE TEST CLOCK – Input signal that synchronizes JTAG and OnCE logic.
7, 8, 13	NC	No connection
9	RESET_b	RESET IN – Active-low input signal that starts a system reset: a reset of the MMC2114 device and most peripherals.
10	ONCE_TMS	OnCE TEST MODE SELECT – Input signal that sequences the JTAG test controller's state machine, sampled on the rising edge of the ONCE_TCLK signal.
11	VDD3V	+3.3-volt power.
12	ONCE_DE_B	DEBUG EVENT – Active-low debug-mode control line for the OnCE controller. An input signal from an external command controller makes the OnCE controller immediately enter debug mode. An output signal acknowledges debug-mode-entry to the external command controller.
14	ONCE_TRST_B	OnCE TEST RESET – Active-low input that asynchronously initializes JTAG and OnCE logic.

4.4 Logic Analyzer Connectors

Connectors J5, J17, and J18, all 2-by-19-pin Mictor connectors, are the logic analyzer connectors. Figure 4-8 through Figure 4-10 give the pin assignments for these connectors. Table 4-7 through Table 4-9 give the signal descriptions for these connectors. Note that these figures and tables follow the Tektronix pin-numbering pattern.

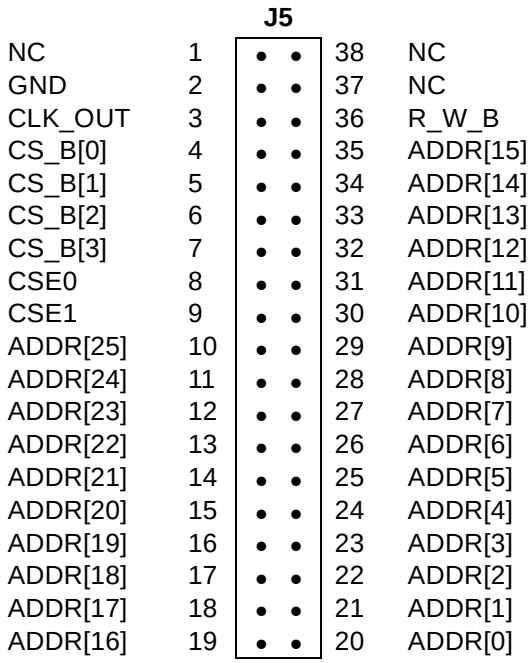


Figure 4-8 Logic Analyzer Connector J5 (A) Pin Assignments

Table 4-7 Logic Analyzer Connector J5 (A) Signal Descriptions

Pin	Mnemonic	Signal
1, 37, 38	NC	No connection
2	GND	Ground
3	CLK_OUT	CLOCK OUTPUT — System clock output.
4 — 7	CS_B[0] — CS_B[3]	CHIP SELECTS (lines 0—3) — Active-low output lines that provide chip selects to external devices.
8, 9	CSE0, CSE1	EMULATION CHIP SELECTS (lines 0, 1) — Emulation-mode output chip-select signals.
10 — 35	ADDR[25] — ADDR[0] (not in exact order)	ADDRESS BUS (lines 25—0) – Output lines for addressing external devices. These lines change state only during external-memory accesses. **Exception:** Pins 10 — 12 (ADDR[25] — ADDR[23]) always have the value 0.
36	R_W_B	READ/WRITE ENABLE – Active-low signal indicating that the current bus access is a write access. Otherwise, the current bus access is a read access.

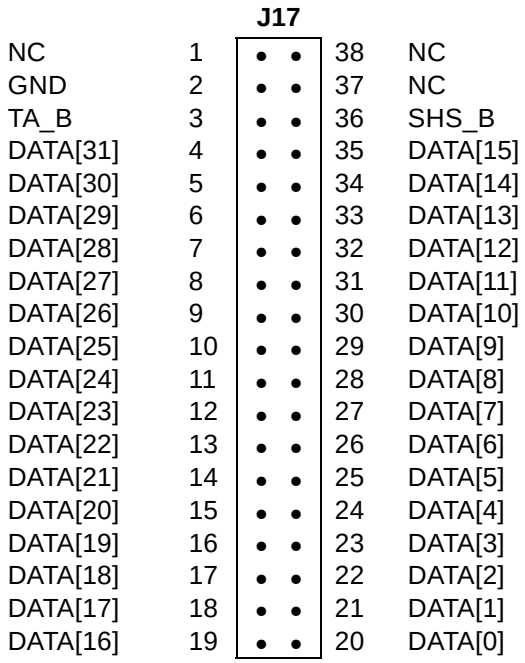


Figure 4-9 Logic Analyzer Connector J17 (D) Pin Assignments

Table 4-8 Logic Analyzer Connector J17 (D) Signal Descriptions

Pin	Mnemonic	Signal
1, 37, 38	NC	No connection
2	GND	Ground
3	TA_B	TRANSFER ACKNOWLEDGE — Active-low input indicating completion of a data transfer, for either a read or a write cycle.
4—35	DATA[31] — DATA[0] (not in exact order)	DATA BUS — Bi-directional data lines 31—0, for accessing external memory.
36	SHS_B	SHOW CYCLE STROBE — Active-low, output strobe signal for capturing addresses, controls, and data during show cycles. Emulation mode forces this signal active. In master mode, software must enable this signal.

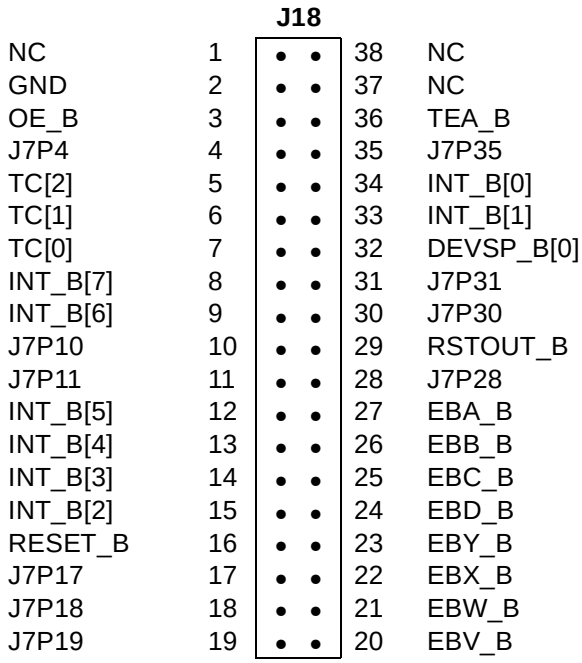


Figure 4-10 Logic Analyzer Connector J18 (C) Pin Assignments

Table 4-9 Logic Analyzer Connector J18 (C) Signal Descriptions

Pin	Mnemonic	Signal
1, 37, 38	NC	No connection
2	GND	Ground
3	OE_B	OUTPUT ENABLE — Active-low output that indicates that a bus access is a read access; enables slave devices to drive the data bus.
4, 10, 11, 17 — 19, 28, 30, 31, 35	J7P[x]	MICTOR PINS — Open pins of this Mictor connector, pins that may be used to connect other system signals to a logic analyzer.
5 — 7	TC[2] — TC[0]	TRANSFER CODE (lines 2—0) — Outputs indicating the data transfer code for the current bus cycle.
8, 9, 12 — 15, 33, 34	INT_B[7] — INT_B[0]	EXTERNAL INTERRUPT (lines 7—0) — Active-low lines for external interrupts or general-purpose I/O. In addition, certain lines can show processor core signal states: • INT_B[7:6]: reflects the states of TSIZ[1:0] signals, provided that the chip configuration register (CCR) SZEN bit is set. • INT_B[5:2]: reflects the states of PSTAT[3:0] signals, provided that the CCR PSTEN bit is set.
16	RESET_B	RESET IN – Active-low input signal that starts a system reset: a reset of the MMC2114 device and most peripherals.
20 — 23	EBV_B — EBY_B	ENABLE BYTES V—Y — CPLD general enable bytes for control of on-board SRAM. (Enable byte V: bits MD7—MD0, enable byte W: bits MD15—8, enable byte X: MD23—MD16, enable byte Y: bits MD31—MD24.)

Table 4-9 Logic Analyzer Connector J18 (C) Signal Descriptions (Continued)

Pin	Mnemonic	Signal
24 — 27	EBD_B — EBA_B	ENABLE BYTES D—A — Active-low outputs active during an operation to corresponding data bits (D31-D24 for enable byte D, D23-D16 for enable byte C, D15-D8 for enable byte B, D7-D0 for enable byte A).
29	RSTOUT_B	RESET OUT – Active-low output signal, controlled by the processor, that resets external components. Activation of any internal reset sources asserts this line.
32	DEVSP_B[0]	DEVELOPMENT SPACE 0 — Active-low signal indicating that the current memory cycle is addressing on-board devices.
36	TEA_B	TRANSFER ERROR ACKNOWLEDGE — Active-low input that indicating that a bus transfer error has occurred.

4.5 RS-232 Connectors

Connectors J57 and J58, the RS-232 connectors, have DCE format. Figure 4-11shows the pin numbering of these connectors. Table 4-10 lists the pin assignments and signal directions for these connectors.

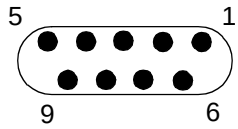


Figure 4-11 RS-232 Connector

Table 4-10 RS-232 Connector J57, J58 Pin Assignments

Pin	Signal	Signal Direction
1	CD Carrier Detect	Out — hard wired active (positive)
2	TXD (SCI_OUT) Transmitted Data	Out
3	RXD (SCI_IN) Received Data	In
4, 7	No connection	—
5	GROUND	—
6	DSR Data Set Ready	Out — hard wired active (positive)
8	RTS Request to Send	Out — hard wired active (positive)
9	RI Ring Indicator	In — hard wired inactive (negative)

Notes *Connector J57 is for channel B, and connector J58 is for channel A. Accordingly, the respective pin 1 assignments can be thought of as CDB and CDA. Similarly, the respective pin 2 assignments can be thought of as TXDB and TXDA, and so forth.*

TXD signals are designated SCI_OUT for other connectors: TXDB is SCI2_OUT; TXDA is SCI1_OUT. RXD signals are designated SCI_IN for other connectors: RXDB is SCI2_IN; RXDA is SCI1_IN.

4.6 SRAM External Standby Power Connector

Connector J36 is for internal SRAM standby external power. The positive SRAM standby voltage should never exceed 3.3-volt. If you do not connect such external power, internal SRAM does not retain data when you turn off board power.

The pin numbering for connector J36 is shown in Figure 4-12.

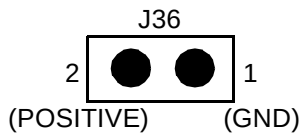


Figure 4-12 Internal SRAM Standby External Power Connector

Standby external power must be provided by a user-supplied power supply. The MMC2114 chip specifications available with the CMB2114 kit explain the correct voltage (VSTBY) level.

4.7 Prototyping Connector Sites

Board locations J40 and J51 through J53 are available for optional, user installation of Berg 69192-620 2-by-10-pin headers for wire wrapping, probing, or cabling to external prototype circuits. Figure 4-13 through Figure 4-16 display the pin assignments for these connectors. Table 4-11 through Table 4-14 provides the signal descriptions for the these connectors.

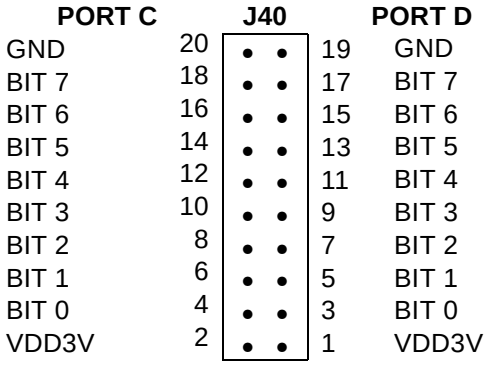


Figure 4-13 Connector Location J40 Pin Assignments

Table 4-11 Connector Location J40 Signal Description

Pin	Mnemonic	Signal
1,2	VDD3V	3.3-volt power
3, 5, 7, 9, 11, 13, 15, 17	BIT0, BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7	Bits 0 to 7 for port D
4, 6, 8, 10, 12, 14, 16, 18	BIT0, BIT1, BIT2, BIT3, BIT4, BIT5, BIT6, BIT7	Bits 0 to 7 for port C

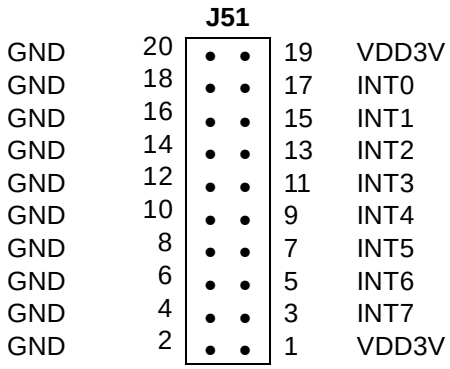


Figure 4-14 Connector Location J51 Pin Assignments

Table 4-12 Connector Location J51 Signal Descriptions

Pin	Mnemonic	Signal
20, 18, 16, 14, 12, 10, 8, 6, 4, 2	GND	GROUND
19, 1, 19	VDD3V	OPERATING VOLTAGE – Transmission line for +3.3-volt MCU operating power.
17, 15, 13, 11, 9, 7, 5, 3	INT0 — INT7	EXTERNAL INTERRUPT (lines 0—7) — Active-low lines for external interrupts or general-purpose I/O. In addition, certain lines can show processor core signal states: • INT_B[7:6]: reflects the states of TSIZ[1:0] signals, provided that the chip configuration register (CCR) SZEN bit is set. • INT_B[5:2]: reflects the states of PSTAT[3:0] signals, provided that the CCR PSTEN bit is set.

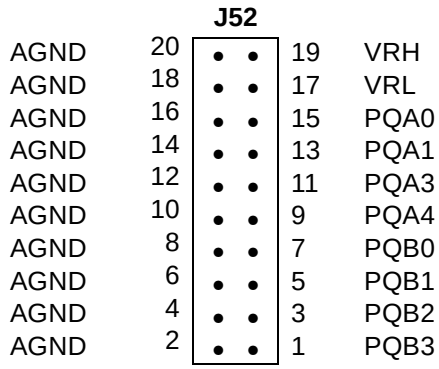


Figure 4-15 Connector Location J52 Pin Assignments

Table 4-13 Connector Location J52 Signal Descriptions

Pin	Mnemonic	Signal
20, 18, 16, 14, 12, 10, 8, 6, 4, 2	AGND	ANALOG GROUND — Analog ground connection for the analog-digital converter.
19	VRH	VOLTAGE REFERENCE HIGH — High reference for the QADC.
17	VRL	VOLTAGE REFERENCE LOW — Low reference for the QADC.
15, 13, 11, 9	PQA0, PQA1, PQA3, PQA4	A ANALOG INPUTS (lines 0, 1, 3, 4) — A analog inputs to the QADC, also usable for general-purpose digital I/O.
7, 5, 3, 1	PQB0 — PQB3	B ANALOG INPUTS (lines 0—3) — B analog inputs to the QADC, also usable as general-purpose digital inputs.

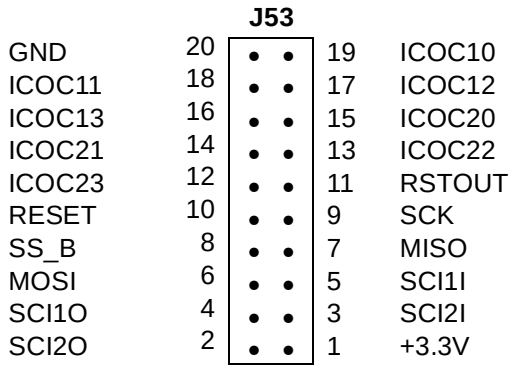


Figure 4-16 Connector Location J53 Pin Assignments

Table 4-14 Connector Location J53 Signal Descriptions

Pin	Mnemonic	Signal
20	GND	GROUND
19 — 16	ICOC10 — ICOC13	TIMER 1 INPUT CAPTURE OUTPUT CAPTURE (lines 0—3) — Signals for internal timer channel 1.
15 — 12	ICOC20 — ICOC23	TIMER 2 INPUT CAPTURE OUTPUT CAPTURE (lines 0—3) — Signals for internal timer channel 2.
11	RSTOUT	RESET OUT – Active-low output signal, controlled by the processor, that resets external components. Activation of any internal reset sources asserts this line.
10	RESET	RESET IN – Active-low input signal that starts a system reset: a reset of the MMC2114 device and most peripherals.
9	SCK	SERIAL CLOCK — If SPI is enabled, the serial clock signal. If SPI is disabled, a general-purpose port E I/O signal.
8	SS_B	SLAVE SELECT — Active-low slave select signal, in slave mode. In master mode, a peripheral chip-select signal.
7	MISO	MASTER IN/SLAVE OUT — If SPI is enabled, the data master-in/slave-out signal. If SPI is disabled, a general-purpose port E I/O signal.
6	MOSI	MASTER OUT/SLAVE IN — If SPI is enabled, the data master-out/slave-in signal. If SPI is disabled, a general-purpose port E I/O signal.
5, 3	SCI1I, SCI2I	SCI INPUT — Serial communications interface (SCI) input lines 1 and 2, otherwise available for general-purpose I/O use. (These lines also are known as RXDA and RXDB.)
4, 2	SCI1O, SCI2O	SCI OUTPUT — Serial communications interface (SCI) output lines 1 and 2, otherwise available for general-purpose I/O use. (These lines also are known as TXDA and TXDB.)
1	+3.3V	OPERATING VOLTAGE – Transmission line for +3.3-volt MCU operating power.

Index

C

R

S

U

How to reach us:

USA: 1-800-377-5416

International: +1-512-997-4700, Operator 4

World Wide Web Addresses

Metrowerks: http://metrowerks.com/

Motorola: http://www.motorola.com/General/index.html

MMCCMB2114UM/D