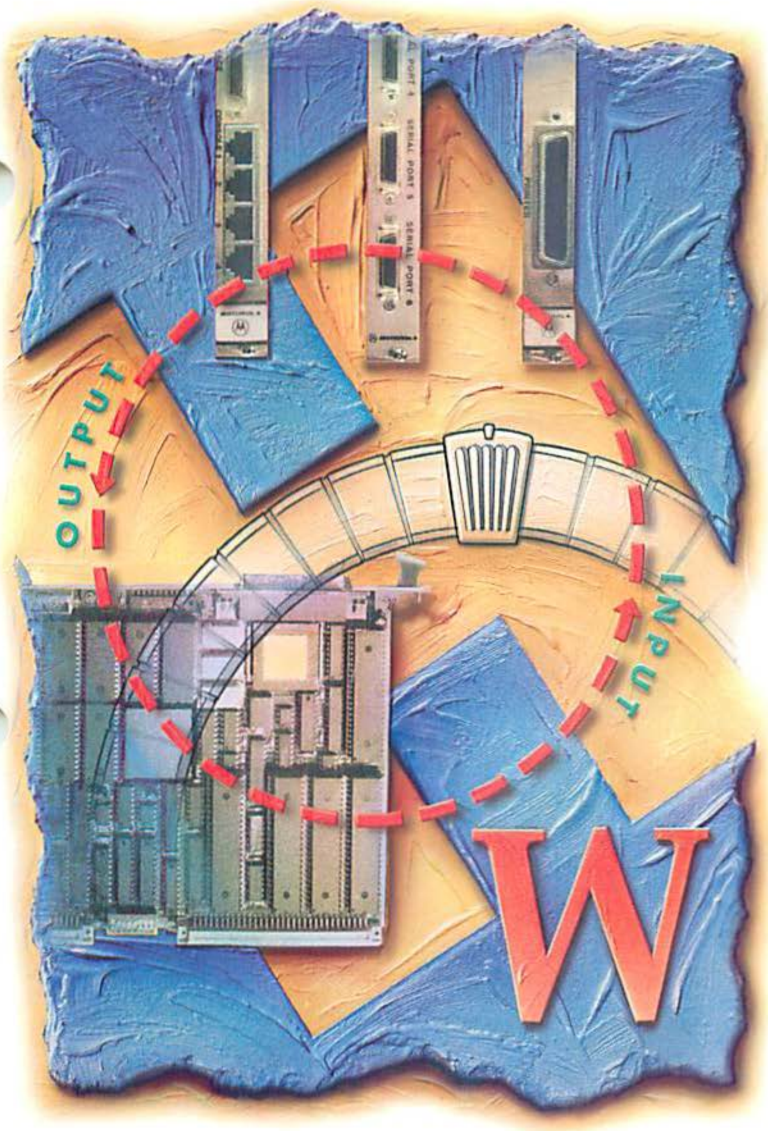


MOTOROLA

MVME133XT VMEmodule 32 Bit Monoboard Microcomputer User's Manual

HARDWARE

READER COMMENTS

Write if we're not right (or even if we are)...

Motorola welcomes your comments on its documentation. We want to know what you think about our manuals and how we can make them better. Please take the time to complete and mail this reply card.

Manual Title ________________________________

Part Number ____________________ Date __________

Your Name ________________________________

Your Title ______________ Telephone (___)__________

Company ________________________________

Address ________________________________

____

____

General Information:

Do you read this manual in order to:

□ Install the product □ Repair the product
□ Use the product □ Reference information
□ Other ________________________

In general, how do you locate information within this manual?

□ Index □ Page Headings
□ Table of Contents □ Other

Completeness:

What topic would you like more information on:

____

____

____

Presentation:

Is the manual easy to use? □ Yes □ No
Please explain what is good or poor: ________________

____

____

____

Is the information easy to understand? □ Yes □ No
If you checked no, please explain: ________________

____

____

Is the information easy to find? □ Yes □ No
If you checked no, please explain: ________________

____

____

Accuracy:

If you have found technical or typographical errors, please list them here.

Page Number	Description of Error

MOTOROLA

MOTOROLA

MVME133XT/D1

APRIL 1988

MVME133XT VMEmodule

32-BIT MONOBOARD MICROCOMPUTER

USER'S MANUAL

The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, Motorola reserves the right to make changes to any products herein to improve reliability, function, or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights or the rights of others.

WARNING

THIS EQUIPMENT GENERATES, USES, AND CAN RADIATE RADIO FREQUENCY ENERGY AND IF NOT INSTALLED AND USED IN ACCORDANCE WITH THE INSTRUCTIONS MANUAL, MAY CAUSE INTERFERENCE TO RADIO COMMUNICATIONS. IT HAS BEEN TESTED AND FOUND TO COMPLY WITH THE LIMITS FOR A CLASS A COMPUTING DEVICE PURSUANT TO SUBPART J OF PART 15 OF FCC RULES, WHICH ARE DESIGNED TO PROVIDE REASONABLE PROTECTION AGAINST SUCH INTERFERENCE WHEN OPERATED IN A COMMERCIAL ENVIRONMENT. OPERATION OF THIS EQUIPMENT IN A RESIDENTIAL AREA IS LIKELY TO CAUSE INTERFERENCE IN WHICH CASE THE USER, AT HIS OWN EXPENSE, WILL BE REQUIRED TO TAKE WHATEVER MEASURES NECESSARY TO CORRECT THE INTERFERENCE.

First Edition

SAFETY SUMMARY
SAFETY DEPENDS ON YOU

The following general safety precautions must be observed during all phases of operation, service, and repair of this equipment. Failure to comply with these precautions or with specific warnings elsewhere in this manual violates safety standards of design, manufacture, and intended use of the equipment. Motorola Inc. assumes no liability for the customer's failure to comply with these requirements. The safety precautions listed below represent warnings of certain dangers of which we are aware. You, as the user of the product, should follow these warnings and all other safety precautions necessary for the safe operation of the equipment in your operating environment.

GROUND THE INSTRUMENT.

To minimize shock hazard, the equipment chassis and enclosure must be connected to an electrical ground. The equipment is supplied with a three-conductor ac power cable. The power cable must either be plugged into an approved three-contact electrical outlet or used with a three-contact to two-contact adapter, with the grounding wire (green) firmly connected to an electrical ground (safety ground) at the power outlet. The power jack and mating plug of the power cable meet International Electrotechnical Commission (IEC) safety standards.

DO NOT OPERATE IN AN EXPLOSIVE ATMOSPHERE.

Do not operate the equipment in the presence of flammable gases or fumes. Operation of any electrical equipment in such an environment constitutes a definite safety hazard.

KEEP AWAY FROM LIVE CIRCUITS.

Operating personnel must not remove equipment covers. Only Factory Authorized Service Personnel or other qualified maintenance personnel may remove equipment covers for internal subassembly or component replacement or any internal adjustment. Do not replace components with power cable connected. Under certain conditions, dangerous voltages may exist even with the power cable removed. To avoid injuries, always disconnect power and discharge circuits before touching them.

DO NOT SERVICE OR ADJUST ALONE.

Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.

USE CAUTION WHEN EXPOSING OR HANDLING THE CRT.

Breakage of the Cathode-Ray Tube (CRT) causes a high-velocity scattering of glass fragments (implosion). To prevent CRT implosion, avoid rough handling or jarring of the equipment. Handling of the CRT should be done only by qualified maintenance personnel using approved safety mask and gloves.

DO NOT SUBSTITUTE PARTS OR MODIFY EQUIPMENT.

Because of the danger of introducing additional hazards, do not install substitute parts or perform any unauthorized modification of the equipment. Contact Motorola Field Service Division for service and repair to ensure that safety features are maintained.

DANGEROUS PROCEDURE WARNINGS.

Warnings, such as the example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed. You should also employ all other safety precautions which you deem necessary for the operation of the equipment in your operating environment.

WARNING

Dangerous voltages, capable of causing death, are present in this equipment. Use extreme caution when handling, testing, and adjusting.

PREFACE

Unless otherwise specified, all address references are in hexadecimal throughout this manual.

An asterisk (*) following the signal name for signals which are level significant denotes that the signal is true or valid when the signal is low.

An asterisk (*) following the signal name for signals which are edge significant denotes that the actions initiated by that signal occur on a high to low transition.

TABLE OF CONTENTS

Page

CHAPTER 1 - GENERAL INFORMATION

CHAPTER 2 - HARDWARE PREPARATION AND INSTALLATION INSTRUCTIONS

TABLE OF CONTENTS (cont'd)

TABLE OF CONTENTS (cont'd)

TABLE OF CONTENTS (cont'd)

Page

LIST OF ILLUSTRATIONS

LIST OF TABLES

TABLE OF CONTENTS (cont'd)

CHAPTER 1 - GENERAL INFORMATION

1.1 INTRODUCTION

This manual provides general information, preparation for use and installation instructions, operating instructions, functional description, and support information for the Motorola MVME133XT VMEmodule 32-Bit Monoboard Microcomputer (referred to as the MVME133XT throughout this manual).

1.2 MODEL DESIGNATIONS

The MVME133XT is available in only one variation, which is listed in Table 1-1.

TABLE 1-1. MVME133XT Model Designations

PRODUCT NUMBER	DESCRIPTION
MVME133XT	25 MHz MC68020 with 25 MHz MC68882

1.3 FEATURES

The MVME133XT is an intelligent single-board processor module containing both the MC68020 microprocessor and the MC68882 Floating Point Coprocessor (FPC). The main features of the MVME133XT are as follows:

- Double-high/single-wide VMEboard
- Address 32/Data 32 (A32/D32) VMEbus master (A32/D16, A24/D32, A24/D16 compatible) interface
- MC68020 Microprocessor with 32-bit address and data, 25 MHz
- MC68882 Floating Point Coprocessor, 25 MHz
- 4Mb of shared local Dynamic RAM, 32-bits wide, accessible from VMEbus
- Four 28-pin JEDEC sockets for ROMs/PROMs/EPROMs/EEPROMs (in two banks, each 16-bits wide) (total 256Kb maximum)
- Three 8-bit programmable timers for tick and watchdog functions

- Battery backup real-time clock (MK48T02)
- 2040 bytes of battery backup SRAM (on the MK48T02)
- Front panel asynchronous DB25 serial debug RS-232C port (on MC68901 MFP)
- Dual multiprotocol (synchronous/asynchronous) serial ports (Z8530)

  one RS-232C (port B)
  one RS-485/RS-422 (port A)
- VMEbus system controller functions with level 3 arbiter
- Single level bus requester (level jumper selectable)
- VMEbus interrupter (selectable level but with status ID $FF only)
- VMEbus interrupt handler for all seven levels
- Front panel FAIL, HALT, RUN, and SCON status LEDs
- Front panel ABORT and RESET switches
- Remote reset through edge connector P2
- Five-position software-readable header; part of Module Status Register (MSR)

1.4 SPECIFICATIONS

General specifications for the MVME133XT are provided in Table 1-2. Paragraphs 1.4.1 and 1.4.2 detail cooling requirements and FCC compliance, respectively.

1.4.1 Cooling Requirements

The Motorola MVME133XT VMEmodule is specified, designed, and tested to operate reliably with an incoming air temperature range from 0 degrees C to 55 degrees C (32 degrees to 131 degrees F) with forced air cooling at a velocity typically achievable by using a 71 CFM axial fan. Temperature qualification is performed in a standard Motorola VMEsystem 1000 chassis. Twenty-five watt load boards are inserted in two card slots, one on each side, adjacent to the board under test, to simulate a high power density system configuration. An assembly of two axial fans, rated at 71 CFM per fan, is placed directly under the VME card cage. The incoming air temperature is measured between the fan assembly and the card cage, where the incoming airstream first encounters the module under test. Test software is executed as the module is subjected to ambient temperature variations. Case temperatures of critical, high power density integrated circuits are monitored to ensure component vendors specifications are not exceeded.

While the exact amount of airflow required for cooling depends on the ambient air temperature and the type, number, and location of boards and other heat sources, adequate cooling can usually be achieved with 5 CFM and 320 LFM flowing over the module. Less airflow is required to cool the module in environments having lower maximum ambients. Under more favorable thermal conditions, it may be possible to operate the module reliably at higher than 55 degrees C with increased airflow. It is important to note that there are several factors, in addition to the rated CFM of the air mover, which determine the actual volume and speed of air flowing over a module.

TABLE 1-2. MVME133XT Module Specifications

CHARACTERISTICS	SPECIFICATIONS
Power requirements (with full set of ROMs/PROMs/ EPROMs/EEPROMs)	+5 Vdc @ 5 A (typical), 7 A (maximum) +12 Vdc @ 100 mA (typical), 250 mA (maximum) -12 Vdc @ 100 mA (typical), 250 mA (maximum)
Microprocessor	MC68020 (MPU)
Coprocessor	MC68882 (FPC)
Clock signal to MPU and FPC	25 MHz (MVME133XT)
Addressing	
Total range (on and offboard)	4 gigabytes
ROM/PROM/EPROM/EEPROM	256Kb maximum: four sockets (two banks of two each, 16 bits wide) for 2K x 8, 8K x 8, 16K x 8, 32K x 8, or 64K x 8 devices
Dynamic RAM	4Mb (32-bits wide)
Serial I/O ports	Port B multiprotocol RS-232C through P2 Port A multiprotocol RS-485/422 through P2 Asynchronous RS-232C debug serial port DCE (to terminal only) through front panel J23
Timers (on MC68901 MFP)	4 total (3 available to user)
Debug port (not available)	8 bit
Watchdog	8 bit
Tick	8 bit
Spare	8 bit

TABLE 1-2. MVME133XT Module Specifications (cont'd)

CHARACTERISTICS	SPECIFICATIONS
Battery backup real-time clock (MK48T02)	1 second resolution Three years storage and operating life
Battery backup SRAM (on MK48T02)	2040 bytes
Bus configuration	Data Transfer Bus (DTB) master or slave, with 32-bit or 24-bit address (A32 or A24) and 32-bit or 16-bit data (D32 or D16)
Interrupt handler	Any or all onboard plus up to seven VMEbus interrupts
Interrupter	Jumper-selectable level with status ID of $FF
Bus arbitration	When MVME133XT is system controller, it arbitrates bus requests/grants on level 3 only
Reset	By SYSRESET*, power-up, RESET switch, watchdog timer time-out, remote reset, or MC68020 RESET instruction.
Temperature	
Operating (Refer to paragraph 1.4.1.)	0 degrees to 55 degrees C at point of entry of forced air (approximately 320 LFM)
Storage	-40 degrees to 85 degrees C
Relative humidity	5% to 90% (non-condensing)
Physical characteristics (not including front panel)	
Height	9.187 inches (233.35 mm)
Depth	6.299 inches (160.00 mm)
Thickness	0.063 inches (1.6 mm)
Connectors	
VMEbus	DIN No. 41612C96 male (P1, P2)
RS-232C	DB-25 female (J23)

1.4.2 FCC Compliance

This VMEmodule (MVME133XT) was tested in an FCC-compliant chassis, and meets the requirements for Class A equipment. FCC compliance was achieved under the following conditions:

a. Shielded cables on all external I/O ports.

b. Cable shields connected to earth ground via metal shell connectors bonded to a conductive module front panel.

c. Conductive chassis rails connected to earth ground. This provides the path for connecting shields to earth ground.

d. Front panel screws properly tightened.

For minimum RF emissions, it is essential that the conditions above be implemented; failure to do so could compromise the FCC compliance of the equipment containing the module.

1.5 GENERAL DESCRIPTION

The MVME133XT 32-Bit Monoboard Microcomputer is a double-high VMEmodule. It takes one slot in a VME system and requires power from both P1 and P2. The module has large onboard DRAM (4Mb), ROM/PROM/EPROM/EEPROM capability (256Kb), serial ports including debug port, Floating Point Coprocessor (FPC), tick timer, watchdog timer, real-time clock with battery backup SRAM, and VMEbus interface with system controller functions.

The MVME133XT is a single-board MPU module intended to be used in a single processor system, but not stand-alone. It is an excellent choice for applications requiring real-time operation such as industrial automation and robotics.

1.6 EQUIPMENT REQUIRED

The following equipment is required to make a complete system using the MVME133XT:

Terminal(s)
Disk drives and controllers
Chassis and power supply
Debug monitor MVME133XTBug
Operating system (such as VERSAdos)

The optionally available MVME133XTBug debug monitor firmware package offers 42 debug, up/downline load, and disk bootstrap load commands, as well as a full set of onboard diagnostics and a one-line assembler/disassembler.

Note that the MVME133XT contains no parallel ports. To use a parallel device, such as a printer, with the MVME133XT, it is necessary to add a module such as the MVME050 System Controller Module to the system.

1.7 RELATED DOCUMENTATION

The following publications are applicable to the MVME133XT and may provide additional helpful information. If not shipped with this product, they may be purchased from Motorola's Literature Distribution Center, 616 West 24th Street, Tempe, Arizona 85282; telephone (602) 994-6561. Non-Motorola documents may be obtained from the sources listed.

DOCUMENT TITLE	MOTOROLA PUBLICATION NUMBER
The VMEbus Specification	HB212
MVME133XT Debug Monitor User's Manual	MVME133XTBUG
VERSAdos to VME Hardware and Software Configuration User's Manual	MVMEVDOS
MC68020 32-Bit Microprocessor User's Manual	MC68020UM
MC68881/MC68882 Floating-Point Coprocessor User's Manual	MC68881UM
MC68901 Multifunction Peripheral Data Sheet	MC68901

MK48T02 2K x 8 Zeropower/Timekeeper RAM Data Sheet, Thompson Components Mostek, 1310 Electronics Drive, Carrollton, TX 75606

Z8530 Serial Communications Controller; data sheet; Zilog, Inc., Corporate Communications, Building A, 1315 Dell Ave, Campbell, CA 95008

CHAPTER 2 - HARDWARE PREPARATION AND INSTALLATION INSTRUCTIONS

2.1 INTRODUCTION

This chapter provides unpacking instructions, hardware preparation, and installation instructions for the MVME133XT.

2.2 UNPACKING INSTRUCTIONS

NOTE

> If the shipping carton is damaged upon receipt, request carrier's agent be present during unpacking and inspection of equipment.

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

CAUTION

> **AVOID TOUCHING AREAS OF INTEGRATED CIRCUITRY; STATIC DISCHARGE CAN DAMAGE CIRCUITS.**

2.3 HARDWARE PREPARATION

To select the desired configuration and ensure proper operation of the MVME133XT, certain modifications may be necessary before installation. These modifications are made through jumper or wire-wrap arrangements on the headers. The location of the headers and connectors on the MVME133XT is illustrated in Figure 2-1. The MVME133XT has been factory tested and is shipped with factory-installed jumper configurations that are described in the following paragraphs. The MVME133XT will operate with its optional add-on Debug Monitor, MVME133XTBug, with the factory-installed jumper configurations. Headers J1 through J22 and test points E1 and E2 are factory-configured as shown in Table 2-1.

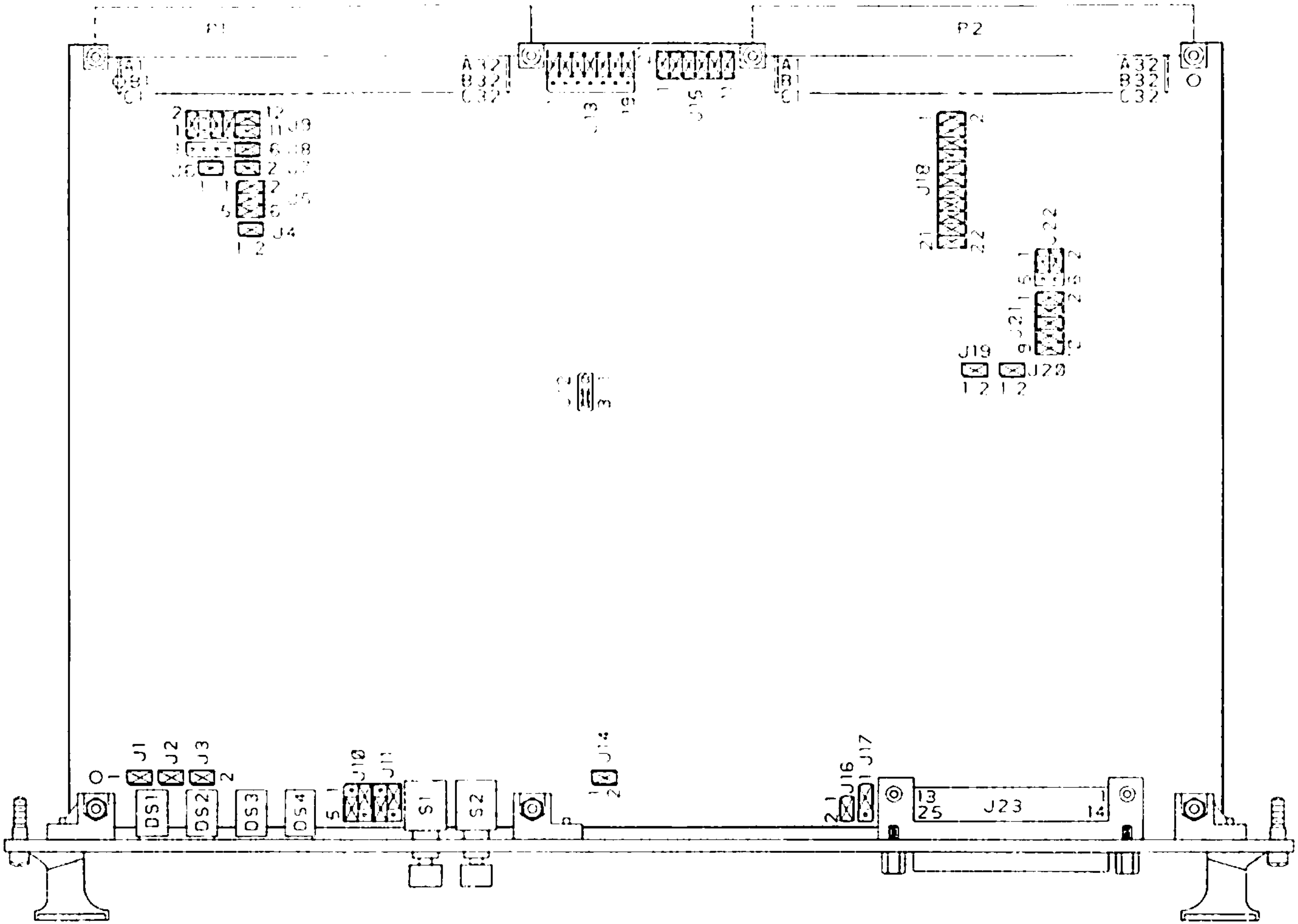


FIGURE 2-1. MVME133XT Headers and Connectors

TABLE 2-1. MVME133XT Header and Test Point Factory Configuration

FUNCTION	CONFIGURATION	CONDITION
ABORT switch	J1, 1-2	Enabled
Watchdog reset	J2, 1-2	Enabled
RMW cycle type select	J3, 1-2	MVME133XT requests VMEbus mastership on all multiple-address RMW cycles.
System controller enable	J4, 1-2	MVME133XT module is system controller.
VMEbus interrupter	J5, 1-2, 3-4, 5-6	Disabled. No interrupt. Must match J13.
VMEbus address size select	J6, 1-2	VMEbus contains both 24-bit and 32-bit address devices.
VMEbus slave interface addressing	J7, 1-2	Onboard DRAM responds to both 24-bit and 32-bit addressing.
VMEbus requester level select	J8, 5-6 and J9, 1-2, 5-6, 7-8, 9-11, 10-12	Level 3 requested.
ROM/PROM/EPROM EEPROM size	J10, 2-4, 3-5 and J11, 2-4, 3-5	Banks 1 & 2 each set for two 64K x 8 ROMs/PROMs/EPROMs
Floating point coprocessor speed	J12, 2-3	Factory-wired for 25 MHz. **DO NOT CHANGE THIS FACTORY CONFIGURATION.**
VMEbus interrupter and interrupt handler	J13, 2-3, 5-6, 8-9, 11-12, 14-15, 17-18, 20-21	Interrupter disabled to match J5. Interrupts IRQ1* through IRQ7* all enabled.
RESET switch	J14, 1-2	Enabled
Shared DRAM offset address select	J15, 1-2, 3-4, 5-6, 7-8, 9-10, 11-12	Onboard DRAM offset address is $00000000 on the VMEbus.
Bus error interrupt	J16, 1-2	Enabled.
VMEbus data width select	J17, 1-2	VMEbus is 32-bit data for physical addresses $00400000 through $EFFFFFFF; 16-bit data for physical addresses $F0000000 through $FFEFFFFF and $FFFF0000 through $FFFFFFFF.

TABLE 2-1. MVME133XT Header and Test Point Factory Configuration (cont'd)

FUNCTION	CONFIGURATION	CONDITION
Serial port B configuration	J18, 1-2, 3-4, 5-6, 7-8, 9-10, 11-12, 13-14, 15-16, 17-18, 19-20, 21-22	Port B as DCE (to terminal)
Local time-out	J19, 1-2	Enabled.
Global time-out	J20, 1-2	Enabled if MVME133XT is the system controller.
Software-readable header	J21, 1-2, 3-4, 5-6, 7-8, 9-10	Module Status Register (MSR) bits 0 through 4 all = 0.
Serial ports RTXCx source select	J22, 1-3, 2-4	RTXCA and RTXCB are driven by onboard 1.230769 MHz signal.
Cache disable	E1 and E2 not connected	MC68020 onchip cache memory not disabled.

NOTE

J23 is the front-panel RS-232C connector.

2.3.1 ABORT Switch Enable Header (J1)

```
    J1
   1   2
 +-------+
 | o---o |
 +-------+
```

ABORT SWITCH ENABLED
(FACTORY CONFIGURATION)

```
    J1
   1   2
 +-------+
 | o   o |
 +-------+
```

ABORT SWITCH DISABLED

2.3.2 Watchdog Reset Enable Header (J2)

```
    J2
   1   2
 +-------+
 | o---o |
 +-------+
```

WATCHDOG RESET IS ENABLED. WHEN THE WATCHDOG COUNTER OUTPUT FROM THE MULTIFUNCTION PERIPHERAL (MFP) TIMER B IS HIGH, A MODULE (BOARD) RESET HAPPENS. SYSRESET* IS ALSO ACTIVATED IF THE MVME133XT IS THE SYSTEM CONTROLLER.
(FACTORY CONFIGURATION)

```
    J2
   1   2
 +-------+
 | o   o |
 +-------+
```

WATCHDOG RESET IS DISABLED. THE MFP TIMER B MAY BE USED AS THE TIMER A OVERFLOW FOR LONGER TICK TIMING.

2.3.3 RMW Cycle Type Select Header (J3)

```
    J3
   1   2
 +-------+
 | o---o |
 +-------+
```

MVME133XT REQUIRES ITS VMEbus REQUESTER TO OBTAIN VMEbus MASTERSHIP FOR ALL MULTIPLE-ADDRESS RMW CYCLES IN ORDER TO MAINTAIN THE INTEGRITY OF THESE CYCLES.
(FACTORY CONFIGURATION)

```
    J3
   1   2
 +-------+
 | o   o |
 +-------+
```

MVME133XT DOES NOT REQUIRE ITS VMEbus REQUESTER TO OBTAIN VMEbus MASTERSHIP FOR MULTIPLE-ADDRESS RMW CYCLES TO ITS ONBOARD RAM. SOFTWARE MUST NEVER GENERATE MULTIPLE-ADDRESS RMW CYCLES TO VMEbus, AND OTHER VMEbus MASTERS MUST NEVER PERFORM MULTIPLE-ADDRESS RMW CYCLES TO THE MVME133XT ONBOARD SHARED DRAM.

2.3.4 System Controller Enable Header (J4)

```
                    J4                                       J4
                  1   2                                    1   2
                +-------+                                +-------+
                | o---o |                                | o   o |
                +-------+                                +-------+
MVME133XT IS THE SYSTEM CONTROLLER.   MVME133XT IS NOT THE SYSTEM CONTROLLER.
ALL BUS MASTERS IN THE SYSTEM
MUST REQUEST BUS MASTERSHIP ON
LEVEL 3 ONLY.
(FACTORY CONFIGURATION)
```

2.3.5 VMEbus Interrupter Header (J5)

J5 indicates to the MVME133XT the level that it is generating interrupts at. The configuration of J5 must match that of J13 on the interrupter side for proper operation. (Refer to paragraph 2.3.11.) The factory configuration is that the MVME133XT interrupter is disabled, while its interrupt handler handles all seven VMEbus interrupts.

```
        J5                 J5                 J5                 J5
     +-------+          +-------+          +-------+          +-------+
   1 | o---o | 2      1 | o   o | 2      1 | o---o | 2      1 | o   o | 2
     |       |          |       |          |       |          |       |
     | o---o |          | o---o |          | o   o |          | o   o |
     |       |          |       |          |       |          |       |
   5 | o---o | 6      5 | o---o | 6      5 | o---o | 6      5 | o---o | 6
     +-------+          +-------+          +-------+          +-------+
    NO INTERRUPT         LEVEL 1            LEVEL 2            LEVEL 3
(FACTORY CONFIGURATION)

        J5                 J5                 J5                 J5
     +-------+          +-------+          +-------+          +-------+
   1 | o---o | 2      1 | o   o | 2      1 | o---o | 2      1 | o   o | 2
     |       |          |       |          |       |          |       |
     | o---o |          | o---o |          | o   o |          | o   o |
     |       |          |       |          |       |          |       |
   5 | o   o | 6      5 | o   o | 6      5 | o   o | 6      5 | o   o | 6
     +-------+          +-------+          +-------+          +-------+
      LEVEL 4            LEVEL 5            LEVEL 6            LEVEL 7
```

2.3.6 VMEbus Address Size Select Header (J6)

```
     J6
   1   2
 +-------+
 | o---o |
 +-------+
```

VMEbus IS MIXED A24 AND A32. MVME133XT GENERATES A24 ACCESSES FOR THE PHYSICAL ADDRESS RANGE OF $00400000 THROUGH $00EFFFFF, AND A32 ACCESSES FOR THE PHYSICAL ADDRESS RANGE OF $00F00000 THROUGH $FFEFFFFF.
(FACTORY CONFIGURATION)

```
     J6
   1   2
 +-------+
 | o   o |
 +-------+
```

VMEbus IS A32. MVME133XT GENERATES A32 ACCESSES TO THE VMEbus FOR THE PHYSICAL ADDRESS RANGE OF $00400000 THROUGH $FFEFFFFF.

NOTE

Refer to paragraph 4.3.7.1 for an explanation of the MVME133XT address bus and VMEbus memory map.

2.3.7 VMEbus Slave Interface Addressing Header (J7)

```
     J7
   1   2
 +-------+
 | o---o |
 +-------+
```

MVME133XT ONBOARD DRAM RESPONDS TO BOTH STANDARD (A24) ADDRESSING AND TO EXTENDED (A32) ADDRESSING.
(FACTORY CONFIGURATION)

```
     J7
   1   2
 +-------+
 | o   o |
 +-------+
```

MVME133XT ONBOARD DRAM RESPONDS TO ONLY EXTENDED (A32) ADDRESSING.

2.3.8 VMEbus Requester Level Select Headers (J8, J9)

```
      BG0     BG1 BG2     BG3
      OUT*   OUT* OUT*    OUT*
       2   4   6   8  10  12
     +-----------------------+
     | o   o   o   o   o---o |
  J9 | |       |   |         |
     | o   o   o   o   o---o |
     +-----------------------+
       1   3   5   7   9  11
      BG0     BG1 BG2     BG3
      IN*     IN* IN*     IN*

     BR0*     BR1* BR2*    BR3*
     +-----------------------+
  J8 | o   o   o   o   o---o |
     +-----------------------+
       1   2   3   4   5   6
  LEVEL 3 (FACTORY CONFIGURATION)
```

```
      BG0     BG1 BG2     BG3
      OUT*   OUT* OUT*    OUT*
       2   4   6   8  10  12
     +-----------------------+
     | o   o   o   o---o   o |
  J9 | |       |           | |
     | o   o   o   o---o   o |
     +-----------------------+
       1   3   5   7   9  11
      BG0     BG1 BG2     BG3
      IN*     IN* IN*     IN*

     BR0*     BR1* BR2*    BR3*
     +-----------------------+
  J8 | o   o   o   o---o   o |
     +-----------------------+
       1   2   3   4   5   6
              LEVEL 2
```

```
      BG0     BG1 BG2     BG3
      OUT*   OUT* OUT*    OUT*
       2   4   6   8  10  12
     +-----------------------+
     | o   o---o   o   o   o |
  J9 | |           |       | |
     | o   o---o   o   o   o |
     +-----------------------+
       1   3   5   7   9  11
      BG0     BG1 BG2     BG3
      IN*     IN* IN*     IN*

     BR0*     BR1* BR2*    BR3*
     +-----------------------+
  J8 | o   o---o   o   o   o |
     +-----------------------+
       1   2   3   4   5   6
              LEVEL 1
```

```
      BG0     BG1 BG2     BG3
      OUT*   OUT* OUT*    OUT*
       2   4   6   8  10  12
     +-----------------------+
     | o---o   o   o   o   o |
  J9 |         |   |       | |
     | o---o   o   o   o   o |
     +-----------------------+
       1   3   5   7   9  11
      BG0     BG1 BG2     BG3
      IN*     IN* IN*     IN*

     BR0*     BR1* BR2*    BR3*
     +-----------------------+
  J8 | o---o   o   o   o   o |
     +-----------------------+
       1   2   3   4   5   6
              LEVEL 0
```

2.3.9 ROM/PROM/EPROM/EEPROM Size Headers (J10, J11)

```
 J10(BANK 1),J11(BANK 2)   J10(BANK 1),J11(BANK 2)   J10(BANK 1),J11(BANK 2)

            1   2                     1   2                     1   2
          +-------+                 +-------+                 +-------+
      +5V | o   o | [LA15]      +5V | o   o | [LA15]      +5V | o   o | [LA15]
          | |     |                 | |   | |                 |     | |
    PIN 1 | o   o | PIN 27    PIN 1 | o   o | PIN 27    PIN 1 | o   o | PIN 27
          |       |                 |       |                 | |     |
   [LA16] | o   o | [ROMWR*] [LA16] | o   o | [ROMWR*] [LA16] | o   o | [ROMWR*]
          +-------+                 +-------+                 +-------+
            5   6                     5   6                     5   6
     8K x 8 OR 16K x 8         32K x 8 ROM/PROM/EPROM     64K x 8 ROM/PROM/EPROM
       ROM/PROM/EPROM                                     (FACTORY CONFIGURATION)
                                                         (WORKS WITH MVME133XTBUG)
```

```
 J10(BANK 1),J11(BANK 2)            NOTES              J10(BANK 1),J11(BANK 2)

            1   2           1. Customer provides all            1   2
          +-------+            ROM, PROM, EPROM, or           +-------+
      +5V | o   o | [LA15]     EEPROM chips to put        +5V | o  /o | [LA15]
          |       |            in the four sockets.           |   /   |
    PIN 1 | o   o | PIN 27                              PIN 1 | o/  o | PIN 27
          |     | |         2. Bank 1:  XU31 (even)           |     | |
   [LA16] | o   o | [ROMWR*]              XU12 (odd).   [LA16] | o   o | [ROMWR*]
          +-------+            Bank 2:  XU20 (even)           +-------+
            5   6                       XU3 (odd).              5   6 (WIRE-WRAP
     2K x 8 OR 8K x 8          Each bank may be              32K x 8  AT PINS
          EEPROM               jumpered differently.         EEPROM   2 AND 3)

                            3. Refer to paragraph
                               4.3.13.1.
```

2.3.10 Floating Point Coprocessor Speed Header (J12)

```
              +---+                              +---+
              | o | 1                            | o | 1
              |   |                              | | |
          J12 | o |                          J12 | o |
              | | |                              |   |
              | o | 3                            | o | 3
              +---+                              +---+
        25 MHz CLOCK INPUT                 16 MHz CLOCK INPUT
     (FACTORY CONFIGURATION)
```

J12 is factory-wired to provide a 25 MHz clock signal to the MC68882 FPC.

CAUTION

DO NOT CHANGE THIS FACTORY CONFIGURATION. THE COPROCESSOR MAY NOT OPERATE PROPERLY OR AT ALL IF THIS CONNECTION IS REMOVED OR CHANGED.

2.3.11 VMEbus Interrupter and Interrupt Handler Header (J13)

J5 indicates to the MVME133XT the level that it is generating interrupts at. The configuration of J5 (refer to paragraph 2.3.5) must match that of J13 on the interrupter side for proper operation. J13 is a dual function header: one side is used to configure the interrupter and the other side is used to enable or disable each individual interrupt level for the interrupt handler. The factory configuration is that the MVME133XT interrupter is disabled, while its interrupt handler handles all seven VMEbus interrupts.

```
          I   I   I   I   I   I   I
          R   R   R   R   R   R   R    INTERRUPT HANDLER
          Q   Q   Q   Q   Q   Q   Q    (CLOSED = ENABLED, OPEN = DISABLED)
          1   2   3   4   5   6   7
          *   *   *   *   *   *   *

          3   6   9  12  15  18  21
        +---------------------------+
        | o   o   o   o   o   o   o |  MVME133XT HANDLES
        | |   |   |   |   |   |   | |  IRQ1* - IRQ7*.
    J13 | o   o   o   o   o   o   o |                          (FACTORY
        |                           |  MVME133XT INTERRUPTER     CONFIGURATION)
        | o   o   o   o   o   o   o |  IS DISABLED.
        +---------------------------+
          1   4   7  10  13  16  19

          L   L   L   L   L   L   L    INTERRUPTER LEVELS
          1   2   3   4   5   6   7    (CLOSED = ENABLED, OPEN = DISABLED)
```

2.3.12 RESET Switch Header (J14)

```
             J14                                   J14
            1   2                                 1   2
          +-------+                             +-------+
          | o---o |                             | o   o |
          +-------+                             +-------+
     RESET SWITCH ENABLED                 RESET SWITCH DISABLED
    (FACTORY CONFIGURATION)
```

2.3.13 Shared DRAM Offset Address Select Header (J15)

The MVME133XT shared DRAM occupies a total of 4Mb in the VMEbus address range. Its base address is controlled by U65 and J15. U65 selects one 256Mb block within the 4Gb range for the MVME133XT. The default factory program for U65 places the base address of this 256Mb block at $00000000. (Refer to Appendix A for U65 program details.) J15 then selects one of the 64 possible positions within this 256Mb block for the 4Mb of shared DRAM on the MVME133XT. As shipped, the MVME133XT is jumpered for a base address of $00000000 as follows:

```
  2        J15        12    2        J15        12    2        J15        12
+-----------------------+ +-----------------------+ +-----------------------+
| o   o   o   o   o   o | | o   o   o   o   o   o | | o   o   o   o   o   o |
| |   |   |   |   |   | | |     |   |   |   |   | | | |       |   |   |   | |
| o   o   o   o   o   o | | o   o   o   o   o   o | | o   o   o   o   o   o |
+-----------------------+ +-----------------------+ +-----------------------+
 A22 A23 A24 A25 A26 A27   A22 A23 A24 A25 A26 A27   A22 A23 A24 A25 A26 A27
  1                   11    1                   11    1                   11
    OFFSET $00000000          OFFSET $00400000          OFFSET $00800000
ONBOARD DRAM IS MAPPED
FROM $00000000 THROUGH        NOTE:  OPEN = 1, CLOSED = 0.
$003FFFFF ON THE VMEbus.
(FACTORY CONFIGURATION)

  2        J15        12    2        J15        12    2        J15        12
+-----------------------+ +-----------------------+ +-----------------------+
| o   o   o   o   o   o | | o   o   o   o   o   o | | o   o   o   o   o   o |
|         |   |   |   | | | |   |       |   |   | | |     |       |   |   | |
| o   o   o   o   o   o | | o   o   o   o   o   o | | o   o   o   o   o   o |
+-----------------------+ +-----------------------+ +-----------------------+
 A22 A23 A24 A25 A26 A27   A22 A23 A24 A25 A26 A27   A22 A23 A24 A25 A26 A27
  1                   11    1                   11    1                   11
    OFFSET $00C00000          OFFSET $01000000          OFFSET $01400000
```

AND SO ON THROUGH THE HIGHEST POSSIBLE ADDRESS OFFSETS:

```
  2        J15        12    2        J15        12    2        J15        12
+-----------------------+ +-----------------------+ +-----------------------+
| o   o   o   o   o   o | | o   o   o   o   o   o | | o   o   o   o   o   o |
|     |                 | | |                     | |                       |
| o   o   o   o   o   o | | o   o   o   o   o   o | | o   o   o   o   o   o |
+-----------------------+ +-----------------------+ +-----------------------+
 A22 A23 A24 A25 A26 A27   A22 A23 A24 A25 A26 A27   A22 A23 A24 A25 A26 A27
  1                   11    1                   11    1                   11
    OFFSET $0F400000          OFFSET $0F800000          OFFSET $0FC00000
```

2.3.14 Bus Error Interrupt Header (J16)

```
          +---+                                      +---+
          | o | 1                                    | o | 1
      J16 | | |                                  J16 |   |
          | o | 2                                    | o | 2
          +---+                                      +---+
```

MPU IS INTERRUPTED ON LEVEL SEVEN IF AN MPU CYCLE IS TERMINATED WITH [BERR*]. REFER TO BUS ERROR PROCESSING, APPENDIX E, FOR DETAILS. (FACTORY CONFIGURATION)

BUS ERROR INTERRUPT (BEIRQ) IS DISABLED.REFER TO BUS ERROR PROCESSING, APPENDIX E, FOR DETAILS.

2.3.15 VMEbus Data Width Select Header (J17)

```
          +---+                                      +---+
          | o | 1                                    | o | 1
          | | |                                      |   |
      J17 | o |                                  J17 | o |
          |   |                                      |   |
          | o | 3                                    | o | 3
          +---+                                      +---+
```

VMEbus IS TREATED AS D32 FOR THE PHYSICAL ADDRESS RANGE OF $00400000 THROUGH $EFFFFFFF, AND AS D16 FOR PHYSICAL ADDRESS RANGE $F0000000 THROUGH $FFEFFFFF AND $FFFF0000 THROUGH $FFFFFFFF. (FACTORY CONFIGURATION)

VMEbus IS ALWAYS TREATED AS D16; MVME133XT NEVER ACTIVATES LWORD*. A 32-BIT LONGWORD ALIGNED TRANSFER IS PERFORMED WITH TWO SEPARATE 16-BIT OPERATIONS.

```
          +---+
          | o | 1
          |   |
      J17 | o |
          | | |
          | o | 3
          +---+
```

VMEbus IS TREATED AS D32 WHEN LA24 = 0 (I.E., $00XXXXXX, $02XXXXXX, $04XXXXXX, ..., $FAXXXXXX, $FCXXXXXX, OR $FEXXXXXX) AND AS D16 WHEN LA24 = 1 (I.E., $01XXXXXX, $03XXXXXX, $05XXXXXX, ..., $FBXXXXXX, $FDXXXXXX, OR $FFXXXXXX).

NOTE

Refer to paragraphs 4.2.1 and 4.3.7.2 for an explanation of the MVME133XT data bus.

2.3.16 Serial Port B Configuration Header (J18)

```
      J18
   +-------+
 1 | o---o | 2
   |       |
 3 | o---o | 4
   |       |
 5 | o---o | 6
   |       |
 7 | o---o | 8
   |       |
 9 | o---o | 10
   |       |
11 | o---o | 12
   |       |
13 | o---o | 14
   |       |
15 | o---o | 16
   |       |
17 | o---o | 18
   |       |
19 | o---o | 20
   |       |
21 | o---o | 22
   +-------+
```

PORT B AS DCE (TO TERMINAL). (FACTORY CONFIGURATION)

```
      J18
   +-------+
 1 | o   o | 2
   |       |
 3 | o   o | 4
   |       |
 5 | o   o | 6
   |       |
 7 | o   o | 8
   | |   | |
 9 | o   o | 10
   |       |
11 | o   o | 12
   | |   | |
13 | o   o | 14
   |       |
15 | o   o | 16
   | |   | |
17 | o   o | 18
   |       |
19 | o   o | 20
   | |   | |
21 | o   o | 22
   +-------+
```

PORT B AS DTE (TO MODEM). TTXC IS OUTPUT FROM TRXCB PIN OF Z8530 (SCC). RTXC IS NOT USED.

2.3.17 Local Time-out Header (J19)

```
    J19
   1   2
 +-------+
 | o---o |
 +-------+
```

LOCAL TIME-OUT IS ENABLED. TIME-OUT PERIOD IS 21 MSEC FOR 25 MHz OPERATION. (FACTORY CONFIGURATION)

```
    J19
   1   2
 +-------+
 | o   o |
 +-------+
```

LOCAL TIME-OUT IS DISABLED. MVME133XT HANGS UP IF SOFTWARE ACCESSES NON-EXISTENT LOCATIONS (SUCH AS WRITING TO $FFFEXXXX)

2.3.18 Global Time-out Header (J20)

```
          J20                                      J20
         1   2                                    1   2
       +-------+                                +-------+
       | o---o |                                | o   o |
       +-------+                                +-------+
```

GLOBAL TIME-OUT ENABLED: IF CONFIGURED AS SYSTEM CONTROLLER (REFER TO PARAGRAPH 2.3.4) MVME133XT ACTIVATES BERR* IF DS0* AND/OR DS1* ARE LOW FOR MORE THAN 72 TO 82 MICROSECONDS. (FACTORY CONFIGURATION)

GLOBAL TIME-OUT DISABLED: THIS MAY CAUSE A SYSTEM PROBLEM. REFER TO PARAGRAPH 2.4.3. IN THIS CONFIGURATION, THE SYSTEM HANGS UP IF MVME133XT IS THE SYSTEM CONTROLLER AND THE SOFTWARE ATTEMPTS TO ACCESS A NON-EXISTENT VMEbus DEVICE.

2.3.19 Software-Readable Header for Module Status Register (MSR) (J21)

```
          J21                                      J21
       +-------+                                +-------+
     1 | o---o | 2  MSRBIT0 = 0               1 | o   o | 2  MSRBIT0 = 1
       |       |                                |       |
     3 | o---o | 4  MSRBIT1 = 0               3 | o   o | 4  MSRBIT1 = 1
       |       |                                |       |
     5 | o---o | 6  MSRBIT2 = 0               5 | o   o | 6  MSRBIT2 = 1
       |       |                                |       |
     7 | o---o | 8  MSRBIT3 = 0               7 | o   o | 8  MSRBIT3 = 1
       |       |                                |       |
     9 | o---o | 10 MSRBIT4 = 0               9 | o   o | 10 MSRBIT4 = 1
       +-------+                                +-------+
  (FACTORY CONFIGURATION)
```

J21 is used to set five bits of the MSR. For details, refer to the MSR description in Chapter 4.

2.3.20 Serial Ports RTXCx Source Select Header (J22)

```
                  J22
               +-------+
[1.23 MHZ] 1   | o   o | 2 [1.23 MHZ]
               | |   | |
    RTXCA 3    | o   o | 4 RTXCB
               |       |
 TT OR RT 5    | o   o | 6 TTXC OR RXC
               +-------+
```

RTXCA PIN OF THE SCC IS DRIVEN BY ONBOARD 1.230769 MHz. RTXCB IS DRIVEN BY ONBOARD 1.230769 MHz. (FACTORY CONFIGURATION)

```
                  J22
               +-------+
[1.23 MHZ] 1   | o   o | 2 [1.23 MHZ]
               | |     |
    RTXCA 3    | o   o | 4 RTXCB
               |     | |
 TT OR RT 5    | o   o | 6 TTXC OR RXC
               +-------+
```

RTXCA IS DRIVEN BY ONBOARD 1.230769 MHz. RTXCB IS DRIVEN BY TTXC IF PORT B IS CONFIGURED DCE (TO TERMINAL), OR FROM RXC IF PORT B IS CONFIGURED DTE (TO MODEM).

```
                  J22
               +-------+
[1.23 MHZ] 1   | o   o | 2 [1.23 MHZ]
               |     | |
    RTXCA 3    | o   o | 4 RTXCB
               | |     |
 TT OR RT 5    | o   o | 6 TTXC OR RXC
               +-------+
```

RTXCA IS DRIVEN BY TT+/- IF PORT A IS CONFIGURED AS A SLAVE, OR FROM RT+/- IF PORT A IS CONFIGURED AS A MASTER. RTXCB IS DRIVEN BY ONBOARD 1.230769 MHz.

```
                  J22
               +-------+
[1.23 MHZ] 1   | o   o | 2 [1.23 MHZ]
               |       |
    RTXCA 3    | o   o | 4 RTXCB
               | |   | |
 TT OR RT 5    | o   o | 6 TTXC OR RXC
               +-------+
```

RTXCA IS DRIVEN BY TT+/- IF PORT A IS CONFIGURED AS A SLAVE, OR FROM RT+/- IF PORT A IS CONFIGURED AS A MASTER. RTXCB IS DRIVEN BY TTXC IF PORT B IS CONFIGURED DCE (TO TERMINAL), OR FROM RXC IF PORT B IS CONFIGURED DTE (TO MODEM).

NOTE

Refer to paragraph 2.4.3, System Considerations, for possible installation of terminators for port A signals.

2.3.21 Cache Disable Test Points (E1, E2)

```
o E1                      o E2
```

The user may hardware disable the MC68020 onchip cache memory by wire-wrapping test point pins E1 and E2 together. (E1 and E2 are next to J14 and J16, respectively.) This connects a ground to the CDIS* pin of the MC68020, preventing cache hit. The factory configuration is with E1 and E2 not connected, leaving the cache function under software control.

2.4 INSTALLATION INSTRUCTIONS

The following paragraphs discuss installation of the MVME133XT module into a VME chassis, connection of an RS-232C terminal and cable, and system considerations. Ensure that desired ROM/PROM/EPROM/EEPROM devices, such as those for the MVME133XTBug debug monitor in sockets XU31 (even bytes) and XU12 (odd bytes), are installed and configured, and that all other headers are configured for desired operation.

2.4.1 MVME133XT Module Installation

Now that the MVME133XT module is ready for installation, proceed as follows:

a. Turn all equipment power OFF.

CAUTION

INSERTING OR REMOVING MODULES WHILE POWER IS APPLIED COULD RESULT IN DAMAGE TO MODULE COMPONENTS.

b. The MVME133XT module requires power from both P1 and P2. It may be installed in any double-height unused card slot, if it is not configured as system controller. If the MVME133XT is configured as system controller, it must be installed in the left-most card slot (slot 1) to correctly initiate the bus-grant daisy-chain and to have proper operation of the IACK-daisy-chain driver.

c. Carefully slide the MVME133XT module into the card slot. Be sure module is seated properly into connectors on the backplane. Fasten module in chassis with screws provided, making good contact with the transverse mounting rails to minimize RFI emissions.

d. Connect any desired cables to the MVME133XT module at the P2 backplane connector, to mate with (optional) peripherals at the RS-232C and/or RS-485 serial ports, and optionally with a remote reset switch. These cables are not provided with the MVME133XT module, and therefore are made or provided by the user. (Motorola recommends using shielded cables for all connections to peripherals to minimize radiation.) Connect the peripherals to the cables.

Two suggested cabling arrangements are shown. Suggestion 1 is in Figure 2-2. Suggestion 2 is in Figure 2-3. A 64-pin flat-ribbon cable female connector may be used to connect to P2. This flat-ribbon cable may, then, be separated and crimped to a flat-cable female DB-25 connector for the RS-232C port connected as given in Table 2-2. The flat-ribbon cable may also be crimped to a flat-cable DB-25 connector for the RS-485/RS-422 port per Table 2-3. Alternately, the cable may be crimped to a DB-9 connector for the RS-485/RS-422 port per Table 2-4. The RS-232C port is defined to interface directly with a standard RS-232C connector, but the RS-485/RS-422 port may require cross-over connections for the user's specific interface. Note that the optional remote reset switch must be connected to P2 pin A20 and/or pin A32 and to ground.

Install any other required VMEmodules in the system.

e. Turn equipment power ON.

TABLE 2-2. Mating Cable Connections for RS-232C Port

P2	64-PIN MATING	DB-25	RS-232C
PIN NO.	CONNECTOR PIN NO.	PIN NO.	SIGNAL NAME
C1	1	1	Not used
A1	2	14	Not used
C2	3	2	TXD
A2	4	15	RTXC
C3	5	3	RXD
A3	6	16	Not used
C4	7	4	RTS
A4	8	17	RXC
C5	9	5	CTS
A5	10	18	Not used
C6	11	6	DSR
A6	12	19	Not used
C7	13	7	Signal Return GND
A7	14	20	DTR
C8	15	8	DCD
A8	16	21	Not used
C9	17	9	Not used
A9	18	22	Not used
C10	19	10	Not used
A10	20	23	Not used
C11	21	11	Not used
A11	22	24	TTXC
C12	23	12	Not used
A12	24	25	Not used
C13	25	13	Not used

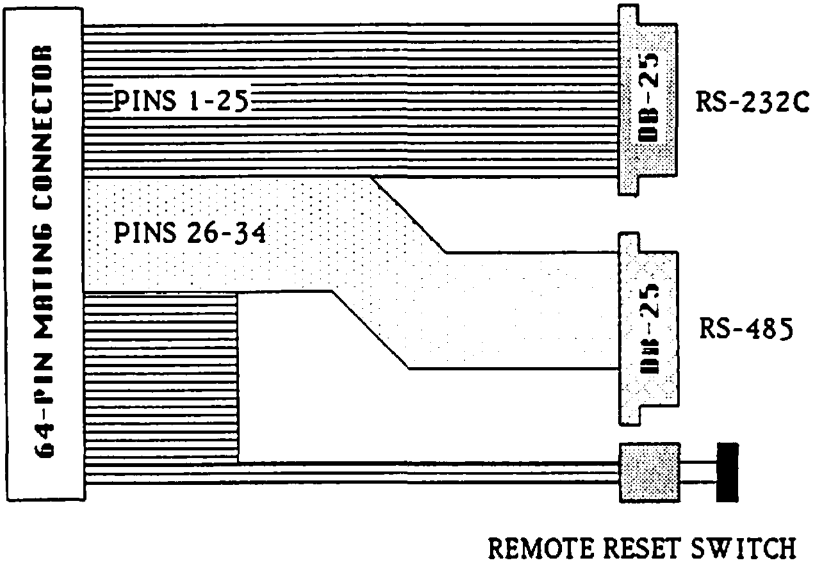


FIGURE 2-2. Cabling Connections to P2, Suggestion 1

TABLE 2-3. DB-25 Mating Cable Connections for RS-485/RS-422 Port

P2 PIN NO.	64-PIN MATING CONNECTOR PIN NO.	DB-25 PIN NO.	RS-485/RS-422 SIGNAL NAME
A13	26	1	SD+
C14	27	14	SD-
A14	28	2	TT+
C15	29	15	TT-
A15	30	3	RD+
C16	31	16	RD-
A16	32	4	RT+
C17	33	17	RT-
A17	34	5	Signal Return GND

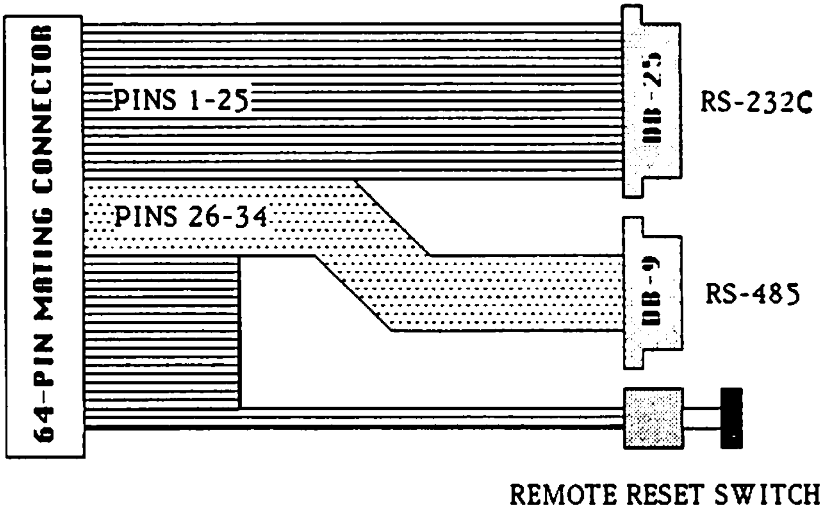


FIGURE 2-3. Cabling Connections to P2, Suggestion 2

TABLE 2-4. DB-9 Mating Cable Connections for RS-485/RS-422 Port

P2 PIN NO.	64-PIN MATING CONNECTOR PIN NO.	DB-9 PIN NO.	RS-485/RS-422 SIGNAL NAME
A13	26	1	SD+
C14	27	6	SD-
A14	28	2	TT+
C15	29	7	TT-
A15	30	3	RD+
C16	31	8	RD-
A16	32	4	RT+
C17	33	9	RT-
A17	34	5	Signal Return GND

2.4.2 Terminal Connection

The RS-232C port on the front panel is configured for DCE (to terminal) operation only. A 25-pin RS-232C cable may be connected to the front panel female connector J23, with the other end connected to a compatible terminal. This cable is not provided with the MVME133XT module, and must be made or provided by the user. (Motorola recommends using shielded cables for all connections to peripherals to minimize RFI radiation.) Note that J23 has a metal shell and jack posts that are electrically connected to the MVME133XT front panel. If the MVME133XT front panel is electrically connected to the chassis ground, then the shell and jack posts on J23 are connected to chassis ground. This allows for shielded cabling to be used for effective reduction of EMI and EMC problems. Detailed information on the signals supported is found in Appendix B and Table 5-3.

NOTE

The user may change J23 to a "to modem" configuration by providing a "null-modem" cable that switches certain signals.

2.4.3 System Considerations

The MVME133XT needs to draw power from both P1 and P2 of the VMEbus backplane. P2 is also used for the upper 16 bits of data for 32-bit transfers, and for the upper 8 address lines for extended addressing mode. The MVME133XT may not operate properly without both P1 and P2 of the VMEbus backplane.

The MVME133XT may be used by itself or with other VMEbus controllers. The MVME133XT is not intended to be used as an Intelligent Peripheral Controller (IPC). It is intended to be used as a VMEbus master. As the system controller, the MVME133XT contains only a single-level arbiter which arbitrates VMEbus mastership on level three. If it is to be used as the system controller, then all bus masters in the system must request bus mastership on level three only.

Whether the MVME133XT operates as a VMEbus master or as a VMEbus slave, it may be configured for 32- or 24-bits of address and for 32- or 16-bits of data (A32 or A24/ D32 or D16). Note that other D16 devices in the system must be located in the MVME133XT module D16 address range. Otherwise, they must only be accessed with 16-bit and 8-bit data transfers only. Refer to VMEbus data width and address size theory details in Chapter 4, and to the memory maps in Chapter 3. Refer to Chapter 4 for details in handling bus error (BERR*) and the use of Read-Modify-Write (RMW) cycles.

The MVME133XT uses the address modifier lines in such a way that it performs short, standard, or extended addressing (AM = $2D, $29; $3E, $3D, $3A, $39; $0E, $0D, $0A, or $09) when it is VMEbus master, but it responds to standard or extended addressing (AM = $3E, $3D, $3A, $39; $0E, $0D, $0A, or $09) when it is a VMEbus slave. Refer to the VMEbus specification for a complete description of all the address modifier codes.

The MVME133XT contains 4Mb of shared DRAM whose offboard address is jumper-selectable with J15. The onboard MPU always sees this local DRAM at physical address $00000000 through $003FFFFF. This address may, however, be changed by reprogramming PAL U39. Refer to Appendix A for details.

Note that the MVME133XT contains no parallel ports. To use a parallel device, such as a parallel printer, with the MVME133XT, it is necessary to add a module such as the MVME050 System Controller Module to the system.

A single SIP resistor package, R24, with four 120-ohm resistors, is used for proper and reliable system operations with the RS-485 serial port (port A). In systems where RS-485 multi-drop cable is used to connect many RS-485 ports (e.g., many MVME133XTs), noise on the cable may be read as spurious data and/or cause undesired interrupts unless the cable is terminated properly. The recommended method is to terminate each of the two ends of the cable with a 120-ohm resistor. For systems using MVME133XTs, proper termination is accomplished by the existing R24 (eight-pin resistor pack with four 120-ohm resistors) on the two MVME133XT modules, one at each end of the RS-485 cable. When more than two MVME133XT modules are on the same cable, remove R24 from its socket from all modules except those at the cable ends. This termination is also useful in case devices connecting to the MVME133XT RS-485 port may be OFF or not online when the RS-485 port is enabled.

If the MVME133XT tries to access offboard resources in a non-existent location, and if the system does not have a global bus time-out, the MVME133XT waits forever for the VMEbus cycle to complete. (Local bus time-out on the MVME133XT does not terminate a VMEbus access.) This would cause the system to hang up. There are two situations in which the system might lack this global bus time-out: (1) the MVME133XT is the system controller but its onboard global bus time-out is disabled (J20 has no jumper), and (2) the MVME133XT is not the system controller, and there is no global bus time-out elsewhere in the system.

THIS PAGE INTENTIONALLY LEFT BLANK.

CHAPTER 3 - OPERATING INSTRUCTIONS

3.1 INTRODUCTION

This chapter provides necessary information to use the MVME133XT module in a system configuration. This includes controls and indicators, memory map details, and software initialization of the module.

3.2 CONTROLS AND INDICATORS

The MVME133XT module has ABORT and RESET switches, and FAIL, HALT, RUN, and SCON indicators, all of which are located on the front panel of the module.

3.2.1 ABORT Switch S1

The ABORT switch is debounced and generates a level 7 interrupt to the interrupt handler. Refer to the interrupt handler details in Chapter 4.

3.2.2 RESET Switch S2

The front panel RESET switch resets all onboard devices (including the MPU) and drives SYSRESET* low if the MVME133XT is the system controller. (The MVME133XT also drives SYSRESET* low at power up if it is configured as the system controller. Refer to the reset details in Chapter 4.)

3.2.3 FAIL, HALT, RUN, and SCON Indicators DS1, DS2, DS3, and DS4

MVME133XT has four LEDs: FAIL, HALT, RUN, and SCON. FAIL is on (red) when [BRDFAIL] is high. HALT is on (red) when reset (any reset except the RESET instruction from the MPU) is true or when [HLED] is high. RUN is on (green) when MPU address strobe [PAS] is high. SCON is on (green) when the MVME133XT is configured as the system controller, which is when [SYSCON*] is jumpered low by J4. The module status for all possible combinations of these LEDs is described in Table 3-1.

3.3 MVME133XT MEMORY MAPS AND MAP DECODER

At the beginning of each MPU cycle, the map decoder determines what kind of cycle takes place and which device or function is selected within that cycle type. Cycle types are determined by the function code lines FC2-FC0, which are driven by the MC68020 MPU. The cycle types and the devices that respond are shown in Table 3-2.

TABLE 3-1. Front Panel LEDs and MVME133XT Status

FAIL	HALT	RUN	
DS1	DS2	DS3	
RED	RED	GREEN	MVME133XT STATUS
off	off	off	No power is applied to the module, or the MPU is not the current local bus master.
off	off	ON	Normal operation.
off	ON	off	MPU is halted.
off	ON	ON	MPU is running and encountering VMEbus deadlocks. Frequency of VMEbus deadlocks determines intensity of HALT LED.
ON	off	off	MPU is not current local bus master. Also, [BRDFAIL] has not been cleared since reset or has been set by software. FAIL indicator is also on if MVME133XT is system controller and SYSFAIL* is detected low on the VMEbus.
ON	off	ON	[BRDFAIL] has not been cleared since reset or has been set by software. FAIL indicator is also on if MVME133XT is system controller and SYSFAIL* is detected low on the VMEbus.
ON	ON	off	MPU is halted and [BRDFAIL] has not been cleared since reset or has been set by software. FAIL indicator is also on if MVME133XT is system controller and SYSFAIL* is detected low on the VMEbus.
ON	ON	ON	MPU is running and encountering VMEbus deadlocks. Frequency of VMEbus deadlocks determines intensity of HALT LED. Also [BRDFAIL] has not been cleared since reset or has been set by software. FAIL indicator is also on if MVME133XT is system controller and SYSFAIL* is detected low on the VMEbus.

TABLE 3-2. Cycle Types and Responding Devices

FC2	FC1	FC0	CYCLE TYPE	MVME133XT DEVICES THAT RESPOND
0	0	0	reserved	None (causes local time-out)
0	0	1	User Data	All except interrupt handler and MC68882
0	1	0	User Program	All except interrupt handler and MC68882
0	1	1	reserved	None (causes local time-out)
1	0	0	reserved	None (causes local time-out)
1	0	1	Supervisory Data	All except interrupt handler and MC68882
1	1	0	Supervisory Program	All except interrupt handler and MC68882
1	1	1	CPU (IACK)	VMEbus, Z8530, MK68901, interrupt handler
1	1	1	CPU (coprocessor)	MC68882 FPC

3.3.1 Main Memory Map

The memory map of devices that respond in User Data, User Program, Supervisory Data, and Supervisory Program spaces is shown in Table 3-3.

TABLE 3-3. MVME133XT Main Memory Map

PHYSICAL ADDRESS RANGE (HEXADECIMAL)	DEVICES ACCESSED	PORT SIZE	SIZE (BYTES)	NOTES
00000000 - 003FFFFF	Onboard DRAM	D32	4Mb	1
00400000 - 00EFFFFF	VMEbus A32/A24	D32/D16	11Mb	2,3
00F00000 - FFEFFFFF	VMEbus A32	D32/D16	4Gb	3
FFF00000 - FFF1FFFF	ROM/EEPROM bank 1	D16	128Kb	4
FFF20000 - FFF3FFFF	ROM/EEPROM bank 2	D16	128Kb	4
FFF40000 - FFF5FFFF	ROM/EEPROM bank 1 repeats in this space.	D16	128Kb	4
FFF60000 - FFF7FFFF	ROM/EEPROM bank 2 repeats in this space.	D16	128Kb	4
FFF80000 - FFF9FFFF	MSR & MC68901 MFP	D16	128Kb	5
FFFA0000 - FFFBFFFF	Z8530 Serial Communications Controller (SCC)	D08	128Kb	8
FFFC0000 - FFFCFFFF	MK48T02 real-time clock (RTC) with 2Kb SRAM	D08	64Kb	8
FFFD0000 - FFFDFFFF	PWRUP* flag	--	64Kb	7
FFFE0000 - FFFEFFFF	VMEbus interrupter	D08	64Kb	6
FFFF0000 - FFFFFFFF	VMEbus short I/O space	D16	64Kb	3

TABLE 3-3. MVME133XT Main Memory Map (cont'd)

PHYSICAL ADDRESS RANGE (HEXADECIMAL)	DEVICES ACCESSED	PORT SIZE	SIZE (BYTES)	NOTES

NOTES: 1. Onboard ROM/PROM/EPROM/EEPROM bank 1 for first four cycles after a reset, onboard DRAM thereafter.

2. VMEbus address size is selectable with J6. Refer to Chapter 2.

3. VMEbus data width option is selectable with J17. Refer to Chapter 2.

4. Writes to EEPROMs must always be 16-bit wide.

5. The Module Status Register (MSR) appears on the upper byte, while the MC68901 Multifunction Peripheral (MFP) appears on the lower byte. The MSR is read-only; write accesses are ignored by the MSR but affect the MFP. For MFP registers, refer to paragraph 3.3.4 and Table 3-5.

6. A VMEbus interrupt is generated on the selected level when a read access is performed in this area. Locations $FFFE0000 through $FFFEFFFF are read-only; write accesses are not allowed and will cause local bus time-outs.

7. Any access to locations $FFFD0000 through $FFFDFFFF resets the Power Up (PWRUP*) flag in the MSR to a logical 1.

8. A write access to the MK48T02 or to the SCC also resets the PWRUP* status flag in the MSR to a logical 1.

3.3.2 Local Processor CPU Space Memory Map

Only two types of CPU space cycles (FC2-FC0 = %111) are supported by the MVME133XT: Coprocessor and Interrupt Acknowledge (IACK) . (Refer to Table 3-2.) All other types of CPU space cycles generated by the MPU are ignored and cause local bus time-out on the MVME133XT. Among the other types of CPU space cycles which the MC68020 is capable of generating but which the MVME133XT does not support are those using Breakpoint Acknowledge, Access level control, or MOVES instructions.

3.3.2.1 Coprocessor Interface Register Map

The only coprocessor on the MVME133XT is the MC68882 Floating Point Coprocessor (FPC). The map decoder selects the MC68882 FPC any time the MPU executes a coprocessor cycle (FC2-FC0 = %111 and [A19]-[A16] = %0010). The Coprocessor ID (Cp-ID) (bits 9-11 of the coprocessor instruction word) for the MC68882 is binary %001. The MC68882 FPC coprocessor interface register locations in the CPU space that are used for communications between the MPU and the FPC are identified in Table 3-4.

TABLE 3-4. MC68882 Floating Point Coprocessor (FPC) Interface Register Map

REGISTER	A04 - A00 (BINARY)	OFFSET (HEX)	DATA WIDTH	R/W
Response	%0000X	$00	16-bit	R
Control	%0001X	$02	16-bit	W
Save	%0010X	$04	16-bit	R
Restore	%0011X	$06	16-bit	R/W
(Reserved)	%0100X	$08	16-bit	--
Command	%0101X	$0A	16-bit	W
(Reserved)	%0110X	$0C	16-bit	--
Condition	%0111X	$0E	16-bit	W
Operand	%100XX	$10	32-bit	R/W
Register Select	%1010X	$14	16-bit	R
(Reserved)	%1011X	$16	16-bit	--
Instruction Address	%110XX	$18	32-bit	W
Operand Address	%111XX	$1C	32-bit	R/W

NOTES: 1. Read accesses to write-only locations return with all ones; write accesses to read-only locations are ignored. In all cases, the MC68882 terminates the cycle properly with DSACK0*/DSACK1* in response to all CPU space cycles accessing coprocessor one (FC2-FC0 = $7, CPU space type = $2, and Cp-ID = 1).

2. X means don't care.

3.3.2.2 Interrupt Acknowledge Map

The MC68020 distinguishes Interrupt Acknowledge (IACK) cycles from other CPU space cycles by placing the binary value %1111 on [A19] - [A16]. The interrupt handler is thus selected when [FC2] - [FC0] = $7 and [A19] - [A16] = %1111. The MPU also indicates the level that is being acknowledged with address lines [A03] - [A01]. The interrupt handler selects which device within that level is being acknowledged. Refer to the interrupt handler description in Chapter 4 for further details.

3.3.3 Shared DRAM Address Map on the VMEbus

The onboard shared DRAM address is controlled by PAL U65 and by header J15. U65 selects one 256Mb block within the 4 Gigabytes range for the MVME133XT. The default factory program for U65 puts the base address of this 256Mb block at $00000000. J15 then selects one of the 64 possible positions within this 256Mb block for the 4Mb of onboard shared DRAM. When U65 contains the default factory program, J15 selects the offset addresses as given in paragraph 2.3.13. Refer to Appendix A for U65 program details (as well as those of PALs U33 and U39).

Moreover, the user selects the onboard DRAM to respond to either 32-bit address accesses only or to both 24-bit and 32-bit address accesses by the VMEbus. J7 defines the address size for the VMEbus slave interface. (Refer to paragraph 2.3.7.) The onboard DRAM may be jumpered to respond to extended address accesses only, or to respond to both extended and standard address accesses. Furthermore, the MVME133XT onboard DRAM responds to the VMEbus accesses only when the addresses match and the address modifiers (AM0-AM5) indicate privileged or non-privileged, data or program space. Also, an MVME133XT may not access its own onboard memory via the VMEbus.

3.3.4 Multifunction Peripheral (MFP) Registers Map

The MFP and the Module Status Register (MSR) are combined together to form a 16-bit port to the MPU and are located at a physical base address of $FFF80000. The MFP is accessed with 8-bit accesses at the odd-byte locations or with 16-bit accesses at the even-byte locations. When accessed with 16-bit transfers, the MFP appears at the least significant byte of the 16-bit word. The register map of the MFP is shown in Table 3-5.

TABLE 3-5. Multifunction Peripheral (MFP) Registers Map

OFFSET	PHYSICAL ADDRESS (HEX)		REGISTER	REGISTER
(HEX)	16-BIT ACCESS	8-BIT ACCESS	NAME	DESCRIPTION
1	FFF80000	FFF80001	GPIP	General Purpose I/O
3	FFF80002	FFF80003	AER	Active Edge Register
5	FFF80004	FFF80005	DDR	Data Direction Register
7	FFF80006	FFF80007	IERA	Interrupt Enable Register A
9	FFF80008	FFF80009	IERB	Interrupt Enable Register B
B	FFF8000A	FFF8000B	IPRA	Interrupt Pending Register A
D	FFF8000C	FFF8000D	IPRB	Interrupt Pending Register B
F	FFF8000E	FFF8000F	ISRA	Interrupt In-service Register A
11	FFF80010	FFF80011	ISRB	Interrupt In-service Register B
13	FFF80012	FFF80013	IMRA	Interrupt Mask Register A
15	FFF80014	FFF80015	IMRB	Interrupt Mask Register B
17	FFF80016	FFF80017	VR	Vector Register
19	FFF80018	FFF80019	TACR	Timer A Control Register
1B	FFF8001A	FFF8001B	TBCR	Timer B Control Register
1D	FFF8001C	FFF8001D	TCDCR	Timer C and D Control Register
1F	FFF8001E	FFF8001F	TADR	Timer A Data Register
21	FFF80020	FFF80021	TBDR	Timer B Data Register
23	FFF80022	FFF80023	TCDR	Timer C Data Register
25	FFF80024	FFF80025	TDDR	Timer D Data Register
27	FFF80026	FFF80027	SCR	Sync Character Register
29	FFF80028	FFF80029	UCR	USART Control Register
2B	FFF8002A	FFF8002B	RSR	Receive Status Register
2D	FFF8002C	FFF8002D	TSR	Transmit Status Register
2F	FFF8002E	FFF8002F	UDR	USART Data Register
	FFF80030 FFF9FFFF			The MSR and MFP registers appear repeatedly in this space.

3.3.5 Serial Communications Controller (SCC) Registers Map

The MVME133XT uses the Z8530 SCC to implement its two multiprotocol serial ports, port A as an RS-485 port, and port B as an RS-232C port. The SCC occupies 128Kb in the MVME133XT memory map and is located at a physical base address of $FFFA0000. The address map for the SCC is shown in Table 3-6.

TABLE 3-6. Serial Communications Controller (SCC) Registers Map

PHYSICAL ADDRESS	REGISTER NAME	REGISTER DESCRIPTION
$FFFA0000	SCCB-RR0 SCCB-WR0	Port B read register 0 Port B write register 0
$FFFA0001	SCCB-RDR SCCB-TDR	Port B received data register Port B transmitted data register
$FFFA0002	SCCA-RR0 SCCA-WR0	Port A read register 0 Port A write register 0
$FFFA0003	SCCA-RDR SCCA-TDR	Port A received data register Port A transmitted data register

In the SCC, register addressing is direct for the data registers only. In all other cases (with the exception of SCCx-WR0 and SCCx-RR0), accessing the internal SCC read and write registers requires a sequence of two operations. The first operation is a write to SCCx-WR0 with the four least significant bits that point to the selected register. If the second operation is a write, then the selected write register is accessed. On the other hand, if the second operation is a read, then the selected read register is selected. The pointer bits are automatically cleared after the second read or write operation so that SCCx-WR0 (or SCCx-RR0) is addressed again on the next access. Refer to the Z8530 Serial Communications Controller data sheet (listed in Chapter 1 herein) for details on programming and using the SCC.

3.4 SOFTWARE INITIALIZATION OF THE MVME133XT

Motorola provides an operating system that runs on the MVME133XT module, VERSAdos, as well as a debugging package with diagnostics, MVME133XTBug.

For users who do not want to use this operating system, the following information gives the proper sequence to follow when initializing the MVME133XT module.

Upon reset, the MVME133XT module tries to fetch the initial stack pointer from the first four bytes of ROM/PROM/EPROM/EEPROM installed in bank 1 (XU31 even, XU12 odd) and the initial program counter from the next four bytes of bank 1. Therefore, the first two longwords of the ROM/PROM/EPROM/EEPROM in bank 1 must contain the desired values for the stack pointer and the program counter.

Use the following sequence to initialize the MVME133XT from a reset:

1. Initialize all necessary exception vectors.

2. Initialize the MFP GPIO pins for proper input/output direction, and to latch required inputs such as [LOCKVBE*] and [LOCKLTO].

3. Set up all timers in the MFP. Note that timer C is used for the debug port baud rate generator.

4. Set up the serial debug port.

5. Set up the two serial ports of the SCC.

6. Initialize the real-time clock if its oscillator has been turned off.

7. Enable the master interrupt enable control bit [IE*].

Refer to Chapter 4 herein, and to the MC68901 MFP, Z8530 SCC, and MK48T02 real-time clock data sheets (listed in Chapter 1 herein) for instructions on programming these devices.

Appendixes C and D contain some programming examples for the Z8530 SCC serial port B and for the MC68901 MFP timer A, respectively.

THIS PAGE INTENTIONALLY LEFT BLANK.

CHAPTER 4 - FUNCTIONAL DESCRIPTION

4.1 INTRODUCTION

This chapter provides the overall block diagram level description for the MVME133XT module. The general description provides an overview of the module, followed by a detailed description of each section of the module. The simplified block diagram of the MVME133XT is shown in Figure 4-1.

4.2 GENERAL DESCRIPTION

The MVME133XT is a VMEbus CPU module. The MVME133XT has a 25 MHz MC68020 MPU, a 25 MHz MC68882 Floating Point Coprocessor (FPC), 4Mb of shared dynamic RAM (accessible from the VMEbus), a battery backup real-time clock, 2Kb of battery backup SRAM, an RS-232C serial debug port, two multiprotocol serial ports (one with RS-232C interface and one with RS-485 interface), three 8-bit timers, four 28-pin ROM/PROM/EPROM/EEPROM sockets, an A32/D32 VMEbus interface, a simple VMEbus interrupter, a seven-level VMEbus interrupt handler, and the VMEbus system controller functions.

4.2.1 Data Bus Structure

The data bus structure on the MVME133XT is arranged to accommodate the 8-bit, 16-bit, 32-bit, and 16-/32-bit ports that reside on the module. The data bus structure of the MVME133XT is shown in Figure 4-2.

4.2.2 Memory Map

The operation of the map decoder and a detailed discussion of the various memory maps in the MVME133XT are given in Chapter 3. This includes the main memory map, coprocessor interface register map, and shared memory map.

4.2.3 MVME133XT Timing

General characteristics of MVME133XT module timing are given in the following paragraphs and Table 4-1.

4.2.3.1 DRAM Cycle Times

MPU accesses to onboard DRAM require four MPU clock cycles (three minimum + one wait cycle). MPU multiple-address RMW cycles to onboard DRAM can require more than four MPU clock cycles if the MVME133XT does not already have the VMEbus mastership and J3 pins 1-2 are connected. Refer to paragraph 4.3.5.1 for more details on local accesses.

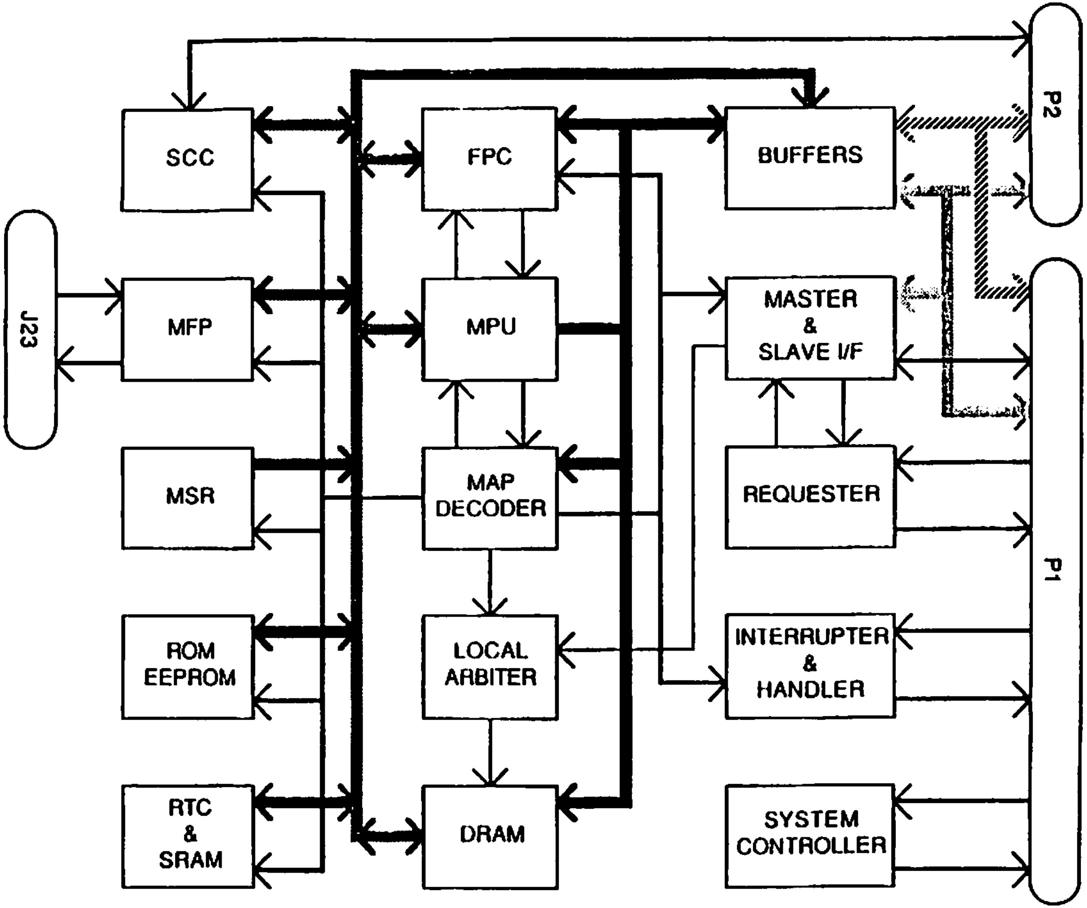


FIGURE 4-1. MVME133XT Block Diagram

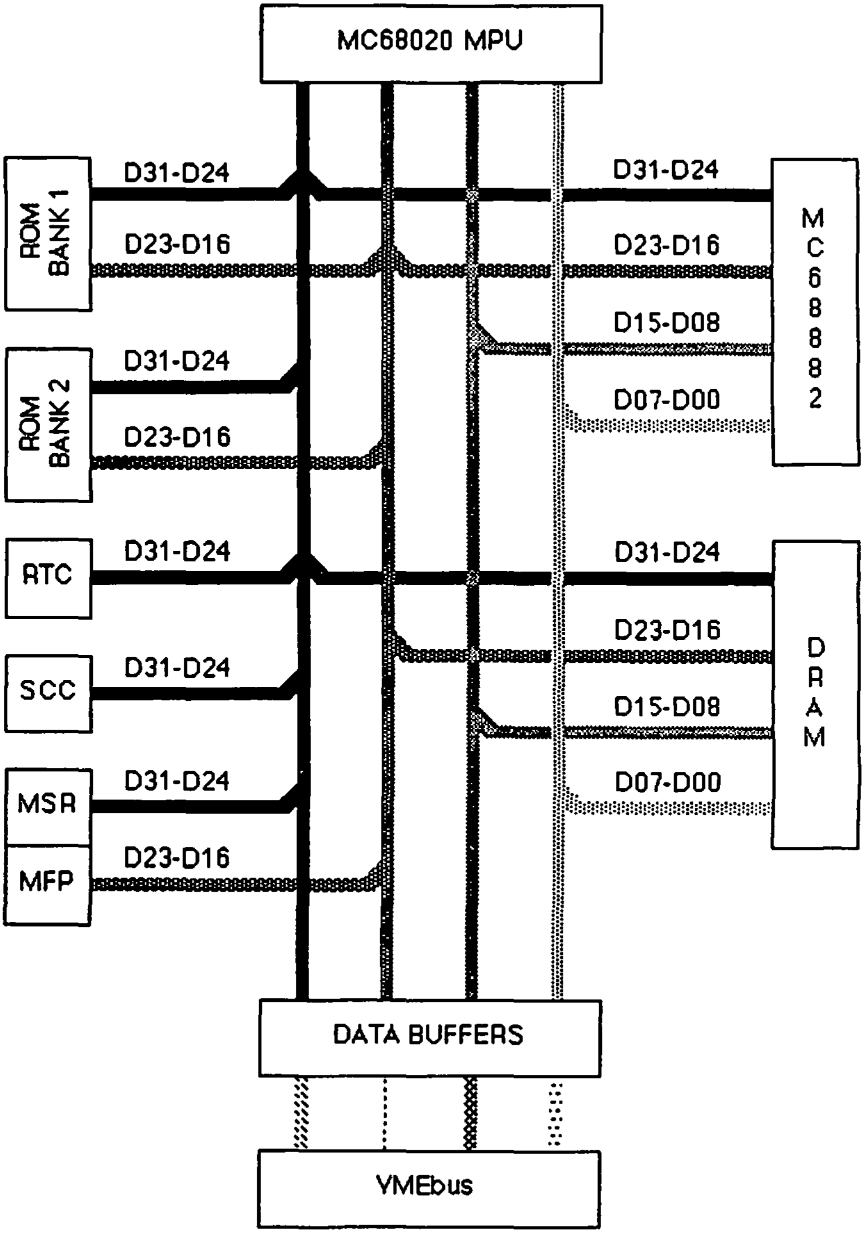


FIGURE 4-2. MVME133XT Data Bus Structure

TABLE 4-1. MVME133XT Timing

TYPE OF ACCESS	MVME133XT READ	MVME133XT WRITE	NOTES
MPU to MC68882 FPC	3 cycles	3 cycles	1,8
MPU to local DRAM	4 cycles	4 cycles	1,2
MPU to local ROM/PROM/EPROM/EEPROM	7 cycles	7 cycles	1,3
VMEbus to local DRAM	9 cycles	8 cycles	4,5
MPU to global RAM (on a slave MVME133XT)	13 cycles	13 cycles	5,6
MPU to global RAM (on a memory module)	9 cycles	7 cycles	6,7

NOTES: 1. No arbitration overhead.

2. Except for RMW cycles where MVME133XT is required to obtain VMEbus mastership before RAM cycle can be started.

3. Device access time must be 200 ns or less at 25 MHz.

4. DS0*/DS1* activated to DTACK* time.

5. Typical values. Actual values may be greater or less depending on the state of the slave MVME133XT.

6. Assume the master MVME133XT is the current VMEbus master.

7. The total number of clock cycles = 5 + (Ta/T) for a read and 6 + (Ta/T) for a write, where Ta = DS0*/DS1* to DTACK* time in nanoseconds and T = MPU clock cycle time in nanoseconds. The result should be rounded up to the nearest integer.

8. Except for read accesses to Response or Save Coprocessor Interface Register (CIR) which take 5 MPU clock cycles.

4.2.3.2 VMEbus Access Time to Onboard DRAM

The onboard DRAM access time from the VMEbus (activation of DS0*/DS1* to activation of DTACK*) is typically eight MPU clock periods (320 ns) for writes and nine MPU clock periods (360 ns)for reads including local bus arbitration overhead. The MVME133XT performs local bus arbitration for every DRAM access from the VMEbus.

4.2.3.3 ROM/PROM/EPROM/EEPROM Cycle Times

All ROM/PROM/EPROM/EEPROM accesses require seven MPU clock cycles (three minimum + four wait cycles) to complete.

4.2.3.4 VMEbus Cycle Times

The following formula assumes that the MVME133XT is the current VMEbus master and that all slaves have released DTACK* and BERR*. The time from the activation of DS0*/DS1* to the activation of DTACK* is Tac in nanoseconds, T is the MPU clock period in nanoseconds, and N is the total number of MPU clock periods required to complete a VMEbus cycle. N must always be rounded up to the next integer.

For read accesses	N = 5 + [Tac / T]	typical
For write accesses	N = 6 + [Tac / T]	typical

The following formula assumes that the MVME133XT is not the current VMEbus master, but that it is the system controller. Also, it assumes that all previous slaves have released DTACK* and/or BERR* when the MVME133XT receives VMEbus mastership. The delay from BR3* low (driven by MVME133XT) to BBSY* high and AS* high is Tr. The time from the activation of DS0*/DS1* to the activation of DTACK* is Tac in nanoseconds, T is the MPU clock period in nanoseconds, and N is the total number of MPU clock periods required to complete a VMEbus cycle. N must always be rounded up to the next integer.

For read accesses	N = 8 + [(Tac + Tr) / T]	typical
For write accesses	N = 9 + [(Tac + Tr) / T]	typical

The following formula assumes that the MVME133XT is not the current VMEbus master, and it is not the system controller. Also, it assumes that all previous slaves have released DTACK* and/or BERR* when the MVME133XT receives VMEbus mastership. The delay from BRX* low (driven by MVME133XT) to BGXIN* low and AS* high is Tg. The time from the activation of DS0*/DS1* to the activation of DTACK* is Tac in nanoseconds, T is the MPU clock period in nanoseconds, and N is the total number of MPU clock periods required to complete a VMEbus cycle. N must always be rounded up to the next integer.

For read accesses	N = 8 + [(Tac + Tg) / T]	typical
For write accesses	N = 9 + [(Tac + Tg) / T]	typical

4.2.3.5 VMEbus Arbitration Time

When the MVME133XT is configured as the system controller and is not requesting VMEbus mastership, the delay from BBSY* high and BR3* low to BG3OUT* low is three MPU clock periods (120 ns) typical and four MPU clock periods (160 ns) maximum.

When the MVME133XT is not configured as the system controller and is not requesting VMEbus mastership, the delay from BGXIN* low to BGXOUT* low is 2.5 MPU clock periods (100 ns) typical and 3.5 MPU clock periods (140 ns) maximum.

4.2.4 System Considerations

4.2.4.1 Sources of BERR*

There are three sources of bus error exceptions on the MVME133XT. They are: Local Bus Time-out (LTO), VMEbus bus error (VBE), and read-modify-write deadlock bus error (RMW-LOCK).

Local bus time-out (LTO) occurs whenever an MPU access does not complete within 524000 MPU clock periods (21 ms for 25 MHz operation). If the system is configured properly, this should only happen if: software accesses a non-existent location within the onboard range, or something prevents this module from becoming the VMEbus master. LTO status is encoded in the LOCKLTO and LOCKVBE status bits. (Refer to paragraph 4.3.11.3.) The bus error source was LTO if LOCKLTO = 1 and LOCKVBE = 0.

VMEbus Bus Error (VBE) occurs when the BERR* signal line is activated on the VMEbus while the MVME133XT is the VMEbus master performing a VMEbus access. VMEbus BERR* should occur only if: an initialization routine samples to see if a device is present on the VMEbus and it is not, software accesses a non-existent device within the VMEbus range, software tries to access a device on the VMEbus incorrectly (such as driving LWORD* low to a 16-bit module), a hardware error occurs on the VMEbus, or a VMEbus slave reports an access error (such as parity error). VBE status is encoded in the LOCKLTO and LOCKVBE status bits. (Refer to paragraph 4.3.11.3.) The bus error source was VBE if LOCKLTO = 0 and LOCKVBE = 1.

RMW-LOCK occurs when there is a VMEbus deadlock during an MPU RMW cycle. As noted in paragraph 4.3.5.2, whenever a VMEbus deadlock occurs, the multiport arbiter breaks the lock by activating both [BERR*] and [HALT*] at the same time. This sequence indicates to the local bus master (MPU) that it should abort the current cycle. Once the local bus master aborts the current cycle, it relinquishes local bus mastership to the VMEbus, which in turn executes a RAM cycle. However, if the MC68020 happens to be executing an RMW cycle when the VMEbus deadlock occurs, it will not relinquish local bus mastership until it completes all portions of the RMW cycle. When the multiport arbiter detects a VMEbus deadlock condition and [RMC*] signal from the MPU is activated, it activates [BERR*] without activating [HALT*] to force the onboard MPU to relinquish the local bus, thus breaking the RMW-LOCK condition, but causing a "benign" bus error exception. RMW-LOCK status is encoded in the LOCKLTO and LOCKVBE status bits. (Refer to paragraph 4.3.11.3.) The bus error source was RMW-LOCK if LOCKLTO = 1 and LOCKVBE = 1.

Because different conditions can cause bus error exceptions, the software must be able to distinguish the source. To aid in this, the MVME133XT provides two bus error status bits: LOCKVBE and LOCKLTO. Refer to Appendix E for details of how these bits are interpreted and processed.

4.2.4.2 Use of RMW Instructions

The MC68020 RMW instructions are TAS, CAS, and CAS2. These instructions cause indivisible cycle sequences to occur on the MC68020 local bus. TAS and single address CAS perform one read and then one write to the same address. Multiple address CAS and CAS2 perform reads and writes to multiple addresses. The VMEbus defines single address indivisible cycles as READ-MODIFY-WRITE cycles. The VMEbus does not define multiple address indivisible cycles. A scheme has been devised to allow indivisible multiple address cycles on the VMEbus. It is not part of the VMEbus specification. It is implemented on the MVME133XT when J3 pins 1-2 are connected. The scheme has the following rules:

1. Locations that are accessed by multiple address indivisible cycles are called Multiple Address Interlock (MAI) locations.

2. All devices that access MAI locations must use indivisible cycle instructions (that is, CAS2 of MC68020).

3. Any device that executes an indivisible cycle instruction must obtain VMEbus mastership before executing the first cycle of the instruction. In addition, it must retain VMEbus mastership until it has completed the last cycle of the instruction.

Rule number 1 is a definition, rule number 2 is a software requirement, and rule number 3 is taken care of automatically by the MVME133XT requester if J3 pins 1-2 are connected.

The MVME133XT does not support the above scheme when J3 pins 1-2 are not connected. In this configuration, the MVME133XT does not obtain VMEbus mastership before executing multiple-address indivisible cycle instructions. In fact, J3 pins 1-2 must only be disconnected if the software never executes RMW cycles within the VMEbus range. The advantage of using this jumper option is that, when it is used properly, RMW-LOCKs never occur.

NOTE

The bus error handler must be able to handle RMW-LOCK bus error. (Refer to paragraph 4.2.4.1.)

4.3 DETAILED DESCRIPTION

The following paragraphs describe in detail the theory of operation for the MVME133XT module. During this discussion, sheets referenced belong to the schematic diagram for the MVME133XT module. See Figure 5-2.

4.3.1 Clocks, Local Time-out, and Retry (Sheet 6)

For the MVME133XT, the frequency of master crystal oscillator Y2 is 50.0 MHz.

The local bus time-out generator aborts any cycle that does not complete within 21 ms for the MVME133XT, by driving [BERR*] active low to the MPU. Refer to paragraph 4.2.4.1, sources of bus error, for details.

4.3.2 MPU and Front Panel Indicators (Sheet 7)

The MVME133XT runs with a 25 MHz MC68020 MPU. However, lower operating frequency is possible by changing the master clock crystal oscillator Y2.

The MC68020 is a full 32-bit processor with 32-bit registers, 32-bit data, and 32-bit addresses. Its advanced architecture, enhanced addressing modes, and on-chip cache are advancements over its predecessors in the MC68000 family of chips.

The FAIL, HALT, RUN, and SCON front panel LED indicators are described in Chapter 3.

4.3.3 MC68882 Floating Point Coprocessor (FPC) (Sheet 8)

The MVME133XT is designed to operate with a 25 MHz MC68882 Floating Point Coprocessor (FPC). The FPC, however, may operate at a different clock frequency from the MPU. Jumper header J12 provides the option of operating the FPC at the MPU clock frequency or at 16 MHz. (Refer to Chapter 2.)

The MC68882 FPC is a full implementation of the IEEE standard for binary floating-point arithmetic. It provides a logical extension to the MC68020 MPU. The MC68882 appears as a 32-bit data port to the MPU. When it operates at the same frequency as the MPU (25 MHz on the MVME133XT), it imposes no delays on CE* or on [DSACK0*] or [DSACK1*] between the MPU and the MC68882. Hence, accesses to the MC68882 (other than a read to the response or save CIR) require three MPU clocks (no wait cycle). Reading the response or save CIR is performed in five MPU clock cycles.

Refer to Chapter 3 for the memory map of the registers in the MC68882 FPC. Refer to the MC68882 Floating-Point Coprocessor User's Manual (Chapter 1 herein) for details on programming and utilizing the FPC.

4.3.4 Map Decoder and DSACKs Generator (Sheet 9)

The operation of the map decoder and a detailed discussion of the various memory maps in the MVME133XT are given in Chapter 3. This includes the main memory map, coprocessor interface register map, and shared memory map.

4.3.5 Local Bus Multiport Arbiter, Refresh, and Dynamic RAM Control (Sheets 10 and 11)

The 4Mb of onboard dynamic RAM (DRAM) is accessible by the local MPU, the refresh circuitry, and the VMEbus. Each of these three things requests and is granted the DRAM by the multiport arbiter.

Because the local address and data busses are used to access the onboard DRAM, any device that uses the DRAM must become the local bus master first. The MPU arbitration logic (BR*, BG*, BGACK*) is utilized by the multiport arbiter to transfer local bus mastership from the current master to the next. The MPU is the default local bus master and has the lowest priority.

4.3.5.1 Local MPU to DRAM Accesses

The local MPU is the default local bus master. Therefore, it has control of the local bus when no one else is using the bus. The DRAM array appears as a 32-bit port to the local MPU. MPU to DRAM accesses are completed in four MPU clock cycles (3 + 1 wait). Multiple-address Read-Modify-Write (RMW) cycles to onboard DRAM may take more than four clock cycles because the DRAM sequencer requires that the MVME133XT has the VMEbus mastership before the first access of the multiple-address RMW cycle can begin. Paragraph 4.2.4.2 has further details on using RMW instructions.

4.3.5.2 VMEbus to Onboard DRAM Accesses

When the MVME133XT shared memory (VMEbus slave) map decoder detects an onboard DRAM select, it requests local bus mastership from the multiport arbiter. (Refer to Chapter 3 for details of the shared memory map.) The multiport arbiter then requests the MPU for the local bus. Once the local bus is released, the multiport arbiter grants local bus mastership to the VMEbus slave interface. At this time, a DRAM read or write cycle is performed. If the VMEbus master is executing an RMW cycle to the DRAM, then the multiport arbiter does not restore local bus mastership to the MPU until both the read and write cycles are completed.

If the MPU is the current local bus master and is executing a cycle that requires the VMEbus when the VMEbus slave map decoder requests local bus mastership, then a VMEbus deadlock condition occurs. To break this VMEbus deadlock condition, the multiport arbiter signals a retry to the MPU by activating both BERR* and HALT*. The MPU responds by aborting the current cycle, at which time it relinquishes local bus mastership so that the multiport arbiter can grant it to the VMEbus. Once the VMEbus has finished with the DRAM, the multiport arbiter returns local bus mastership to the MPU. The MPU then retries the aborted cycle.

However, if the MC68020 MPU is executing an RMW cycle, it does not release the bus once it has initiated an RMW operation. Therefore, instead of indicating a retry, the multiport arbiter must activate [BERR*] to break the deadlock condition. This creates some software implications, which are covered in paragraph 4.2.4.1, sources of BERR*.

The onboard DRAM appears to the VMEbus as a 16-bit port for transfers with LWORD* deactivated, and as a 32-bit port for transfers with LWORD* activated. The MVME133XT supports misaligned transfers to and from the local DRAM by the VMEbus.

4.3.5.3 DRAM Refresh

The dynamic RAMs require that each of their 512 rows be refreshed once every 8 ms. To accomplish this, the refresh timer requests the DRAM sequencer to perform a CAS-before-RAS refresh cycle once every 10.2 us for 25 MHz operation. The DRAM sequencer waits until the current DRAM cycle is finished before initiating the refresh cycle. If the current local bus master begins a new DRAM access during the refresh cycle, the DRAM sequencer delays the access until the refresh cycle is completed.

Note that a DRAM refresh cycle may be executed in concurrence with other MPU or VMEbus slave activities. This way, the DRAM is never starved from refresh.

4.3.6 Onboard Local DRAM Array (Sheets 12 and 13)

The onboard dynamic RAM (DRAM) uses thirty-two 1 Megabit x 1 dynamic RAM ZIPs (zigzag-inline-packages), making a total of 4Mb of local DRAM. It is accessible by the local MPU, the refresh circuitry, and the VMEbus (by another VMEbus master), as described in paragraph 4.3.5. Note that there is no parity checking on the MVME133XT.

4.3.7 VMEbus Master Interface and Address and Data Buffers (Sheets 15 & 16)

The MVME133XT has an A32/D32 VMEbus master interface for buffering of data, address, and control; for word data manipulation to accommodate MC68020 and VMEbus data handling differences; and for interrupt handling and control of misaligned transfers. However, it may be used with devices that have A24 or A32, and D16 or D32 interfaces by the use of jumpers on headers J6 and J17 (refer to Chapter 2) and by observing the following requirements. Refer also to the description of data bus structure in paragraph 4.2.1 and Figure 4-2.

4.3.7.1 VMEbus Address Size

The MVME133XT lets the user select a 32-bit or 24-bit address option for VMEbus references. By properly jumpering J6 (on sheet 9), the user configures the MVME133XT to operate in a mixed 32-bit/24-bit address system, or in a fully 32-bit address system. Refer to paragraph 2.3.6 for details. The MVME133XT VMEbus memory map is directly affected by the address option. The mixed system is shown in Table 4-2. The A32 system is shown in Table 4-3. (Refer also to Table 3-3.)

TABLE 4-2. VMEbus Memory Map in a Mixed A24/A32 System
(J6 Pins 1-2 Connected)

ADDRESS RANGE	VMEbus ACTIVITY TYPE
$00000000-$003FFFFF	No VMEbus activity, onboard DRAM area
$00400000-$00EFFFFF	VMEbus standard (24-bit) address space
$00F00000-$FFEFFFFF	VMEbus extended (32-bit) address space
$FFF00000-$FFFEFFFF	No VMEbus activity, local resource area
$FFFF0000-$FFFFFFFF	VMEbus short I/O (16-bit) address space

TABLE 4-3. VMEbus Memory Map in an A32 System
(J6 Pins 1-2 Not Connected)

ADDRESS RANGE	VMEbus ACTIVITY TYPE
$00000000-$003FFFFF	No VMEbus activity, onboard DRAM area
$00400000-$FFEFFFFF	VMEbus extended (32-bit) address space
$FFF00000-$FFFEFFFF	No VMEbus activity, local resource area
$FFFF0000-$FFFFFFFF	VMEbus short I/O (16-bit) address space

4.3.7.2 VMEbus Data Width

As a VMEbus master, the MVME133XT performs 32-bit data transfers only on longword-aligned accesses and only if VMEbus is a 32-bit data system. J17 (on sheet 11) is jumpered to indicate that the system is 16-bit or 32-bit data.

In a system where there are both 16-bit and 32-bit data, the user may configure J17 so that MPU address line LA24 dynamically indicates to the MVME133XT master interface the data width of VMEbus. In this case, VMEbus is assumed to be 16-bit data when LA24 is high, and 32-bit data when LA24 is low. Thus, the user can place all 32-bit data devices in any of the 128 16Mb blocks where LA24 is low ($00000000 to $00FFFFFF, $02000000 to $02FFFFFF, $04000000 to $04FFFFFF, ..., $FC000000 to $FCFFFFFF, or $FE000000 to $FEFFFFFF); and all 16-bit data devices at any of the other 128 16Mb blocks where LA24 is high ($01000000 to $01FFFFFF, $03000000 to $03FFFFFF, ..., $FF000000 to $FFFFFFFF). Paragraph 2.3.15 shows proper configuration of J17.

4.3.7.3 Accessing the VMEbus

Whenever the MVME133XT executes a VMEbus cycle (read, write, or interrupt acknowledge) and its VMEbus requester has obtained VMEbus mastership, it drives the VME address bus with its local address bus and the VMEbus address modifiers to indicate proper address space. It also activates IACK* if this is an interrupt acknowledge cycle. It activates LWORD* on longword-aligned transfers only if J17 indicates that the VMEbus is a 32-bit port (either statically or dynamically with LA24). (For cycle types and responding devices, refer to paragraph 3.3 and Table 3-2.)

Once A01-A31, AM0-AM5, IACK*, and LWORD* are driven to their appropriate levels on the VMEbus, the MVME133XT activates AS*. The WRITE* line is driven low for write accesses and high for read accesses. After the data bus is driven according to the access cycle, it activates DS0* and/or DS1* appropriately. (Refer also to data bus structure in paragraph 4.2.1 and Figure 4-2.)

If the cycle terminates normally with DTACK* driven to low, then the onboard DSACKs generator circuit activates both [DSACK1*] and [DSACK0*] if LWORD* is low, or only [DSACK1*] if LWORD* is high. If the cycle terminates with BERR* driven to low, then the BERR generator circuit activates [BERR*] to the local processor. Once the handshake has occurred (either DTACK* or BERR*), the MPU removes [AS*], [DS*] and the MVME133XT completes the cycle by disabling the data bus drivers and removing DS0*/DS1* and AS*.

The above sequence is altered slightly when the MPU executes RMW cycles. When the MPU starts an RMW cycle, the VMEbus master interface checks to see if it is a single or multiple address RMW by examining SIZ1 and SIZ0. If it is a multiple address RMW cycle, then the VMEbus master interface operates normally. (VMEbus requester operation is altered as shown in paragraph 4.3.8.) If it is a single address RMW cycle, then the VMEbus master interface keeps AS* active during the entire time from the beginning of the RMW read cycle to the end of the RMW write cycle. This makes a single address RMW cycle from the MPU appear on the VMEbus as a VMEbus-defined read-modify-write cycle.

4.3.8 VMEbus Requester (Sheet 17)

The VMEbus requester is used to obtain and relinquish mastership of the VMEbus. It can request VMEbus mastership on any one of the four request levels depending on the configurations of J8 and J9, and it fully supports the bus-grant daisy-chain. It requests mastership of the VMEbus any time the MVME133XT is not the current VMEbus master and the map decoder or the interrupt handler indicates that the local processor is executing a cycle that requires the VMEbus. It also requests mastership of the VMEbus when the MVME133XT is not the current VMEbus master and the MPU is starting to execute a multiple-address RMW sequence to the onboard DRAM with J3 pins 1-2 connected.

The VMEbus requester operates in the Release-On-Request (ROR) mode. Once the MVME133XT has obtained VMEbus mastership, the VMEbus requester maintains mastership until another VMEbus module requests VMEbus mastership and then only if an RMW sequence is not in process. It releases the VMEbus in one of two different ways, depending on the state of the MVME133XT at the time.

If the MVME133XT is in the middle of a VMEbus cycle (AS* already activated) when the VMEbus requester decides to relinquish VMEbus mastership, it releases BBSY* immediately. The transfer of VMEbus mastership occurs when the VMEbus master interface (refer to paragraph 4.3.7.3) deactivates and releases AS*.

If the MVME133XT is not in the middle of a VMEbus cycle when the VMEbus requester decides to relinquish VMEbus mastership, the VMEbus master interface (refer to paragraph 4.3.7.3) releases all of the VMEbus lines, after which the VMEbus requester releases BBSY* to complete the transfer of VMEbus mastership.

4.3.9 VMEbus System Controller and Interrupter (Sheet 18)

4.3.9.1 System Controller and SYSRESET*

The system controller implements global VMEbus time-out that drives BERR*, global SYSCLK (16 MHz), level 3 VMEbus arbiter, and IACK* daisy-chain driver. All of these MVME133XT system controller functions and the SYSRESET* driver are enabled/disabled by header J4. The position of the jumper on J4 appears as the SYSCON bit in the Module Status Register. (Refer to paragraph 4.3.11.4.) Also, the SCON LED (DS4, sheet 7) is lit if the MVME133XT is configured as the system controller.

The global bus time-out circuit starts the timing upon detecting activation of DS0* and/or DS1*. If DS0* and/or DS1* are activated longer than the time-out period, it drives BERR* low. At 25 MHz, the time-out count can be set for 72 to 82 us (J20 pins 1-2 connected) or for infinity (J20 pins 1-2 open).

The SYSCLK driver drives a periodic 16 MHz clock onto the SYSCLK line on the VMEbus if the system controller on the MVME133XT is enabled.

The level 3 arbiter is designed to meet the VMEbus specification requirements. It is designed to re-arbitrate if no VMEbus master responds to a grant within 72 to 82 us when the MVME133XT operates at 25 MHz. Note that if the MVME133XT is the system controller, then all potential VMEbus masters in the system must be configured to request VMEbus mastership on level three only.

The IACK* daisy-chain driver is designed to meet the VMEbus specification requirements. Note that for the IACK* daisy-chain driver to function properly, the MVME133XT must be in the leftmost slot in the chassis if it is the system controller.

Although SYSRESET* is not a VMEbus system controller function, the MVME133XT enables/disables its SYSRESET* function at the same time that it enables/disables its system controller functions. When configured as the system controller, the MVME133XT drives the SYSRESET* signal line low when the front panel RESET switch is depressed, when a watchdog time-out occurs, when the RRESET* line is activated, or when a power-up occurs.

NOTE

The MVME133XT does not fully implement SYSRESET* timing of a VMEbus power monitor.

4.3.9.2 VMEbus Interrupter

The VMEbus interrupter provides the value $FF as its status ID byte. It is an 8-bit interrupter and consequently responds to all sizes of interrupt acknowledge cycles. The VMEbus interrupter drives the selected interrupt request line low whenever the MPU performs a read access to a location within $FFFE0000 to $FFFEFFFF. The interrupt level is selected by jumpers on headers J5 and J13. J13 selects the interrupt line for the MVME133XT to drive, and J5 lets the MVME133XT know which level it is interrupting at. Refer to paragraphs 2.3.11 and 2.3.5 for proper level selection for the VMEbus interrupter. The factory configuration is with the MVME133XT interrupter disabled, but with the interrupt handler able to handle all seven levels of VMEbus interrupts (refer to paragraph 4.3.10.1). The state of the interrupter is reflected as the [OIRQ] = GPIO6 bit of the Multifunction Peripheral (MFP) GPIO port. (Refer to paragraph 4.3.11.3.) A typical sequence for interrupting is as follows:

a. Verify that the [OIRQ] bit is 0.

b. Set up the MFP to interrupt the MPU when [OIRQ] transitions from 1 to 0 to indicate that the interrupt has been acknowledged.

c. Perform a read access to physical address $FFFE0000.

d. Continue with other processing until the [OIRQ] interrupt occurs.

e. The VMEbus interrupt has now been acknowledged.

f. Continue with normal processing.

NOTE

Locations $FFFE0000 through $FFFEFFFF are read-only. Write accesses to these locations are not allowed, do not generate VMEbus interrupts, and cause local bus time-outs.

4.3.10 Interrupt Handler, Reset, and Abort (Sheet 19)

4.3.10.1 Interrupt Handler

The interrupt handler gives the onboard MPU the ability to sense and respond to all onboard interrupts, all seven VMEbus interrupts, VMEbus ACFAIL*, VMEbus SYSFAIL*, and the ABORT switch.

All VMEbus interrupts are enabled/disabled using header J13 (on sheet 18), ABORT is enabled/disabled using J1, [BERR*] interrupt is enabled/disabled using J16 (on sheet 10), and all interrupts that go through the MC68901 MFP are enabled in the MC68901. Also, the Z8530 SCC interrupts may be enabled/disabled individually. All interrupts are disabled when the [IE*] bit on the MC68901 MFP is high (logic 1).

When the MPU initiates an interrupt acknowledge cycle, the interrupt handler determines the acknowledge level by examining [A01] - [A03]. Finally, it activates [AVEC*] to indicate to the MPU to generate the interrupt vector internally if the acknowledge cycle was for VMEbus ACFAIL* or the ABORT switch. If the acknowledge cycle is for the Z8530 SCC, the MC68901 MFP, or the VMEbus, then it initiates a vector fetch cycle to the appropriate device. If the IACK is for [BERR*] interrupt, then it provides the MPU with a vector of $FE.

If both onboard and VMEbus interrupts are activated on the same acknowledge level, the interrupt handler acknowledges the onboard interrupt. Note also that VMEbus ACFAIL* and ABORT switch are both on level 7 and have the same interrupt offset vector. Therefore, the software handler routine for this autovector must interrogate the [ACFAIL] bit in the MSR to determine the actual interrupt source.

All the interrupt sources on the MVME133XT (in descending order of priority) and the associated interrupt vectors are summarized in Table 4-4.

TABLE 4-4. Interrupt Sources and Vectors

INTERRUPT SOURCE	VECTOR SOURCE	VECTOR NUMBER	VECTOR OFFSET	LEVEL
Bus Error	direct	$FE	$3F8	7
ABORT*	auto	$1F	$7C	7
VMEbus ACFAIL*	auto	$1F	$7C	7
VMEbus IRQ7*	VMEbus	supplied	4 x vector	7
MC68901 MFP (Multi-function peripheral)	MC68901 MFP	programmable (NOTE 1)	4 x vector	6
VMEbus IRQ6*	VMEbus	supplied	4 x vector	6
Z8530 SCC (Serial Ports)	Z8530 SCC	programmable (NOTE 2)	4 x vector	5
VMEbus IRQ5*	VMEbus	supplied	4 x vector	5
VMEbus IRQ4*	VMEbus	supplied	4 x vector	4
VMEbus IRQ3*	VMEbus	supplied	4 x vector	3
VMEbus IRQ2*	VMEbus	supplied	4 x vector	2
VMEbus IRQ1*	VMEbus	supplied	4 x vector	1

NOTES: 1. Refer to Appendix D for an example of setting up timer A of the MC68901 MFP.

2. Refer to Appendix C for an example of setting up serial port B of the Z8530 SCC.

4.3.10.2 Reset

There are a total of six sources of reset on the MVME133XT, as follows:

a. SYSRESET* (VMEbus system reset) - Resets all onboard devices.

b. Power-Up Reset - Resets all onboard devices and drives SYSRESET* if this module is system controller.

c. Front Panel RESET Switch - Resets all onboard devices and drives SYSRESET* if this module is system controller.

d. Watchdog Time-out - Resets all onboard devices and drives SYSRESET* if this module is system controller.

e. RRESET* (remote reset) - Resets all onboard devices and drives SYSRESET* if this module is system controller.

f. MC68020 RESET Instruction - Resets only the Z8530 SCC and the MC68901 MFP.

All resets wait until the MPU is between cycles before starting.

4.3.10.3 ABORT and RESET Switches

Refer to Chapter 3 for information on these front panel switches.

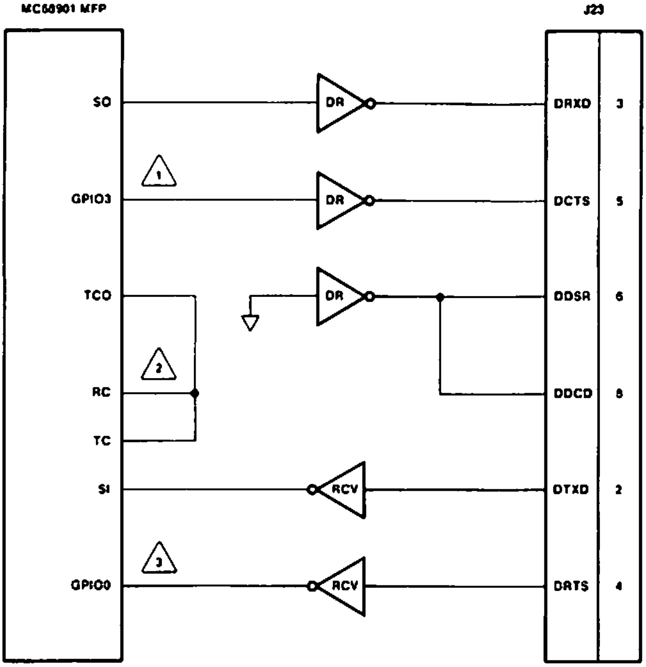
4.3.11 Multifunction Peripheral (MFP): Debug Port, Timers, and Status/Control; and Module Status Register (MSR) (Sheet 20)

The MVME133XT uses the MFP MC68901 chip for its front panel debug port, tick timers, watchdog timer, and the status and control information. The MC68901 has the ability to interrupt the MPU on level 6. (Refer to paragraph 4.3.10.1.) Its interrupt sources are from the timers, the debug port, and the GPIO (status) bits.

The MFP and the Module Status Register (MSR) are combined together to form a 16-bit port to the MPU and are located at a physical base address of $FFF80000. Refer to Chapter 3 for the MFP register map. Refer to the MC68901 data sheet (listed in Chapter 1) for details in programming and using the MFP.

4.3.11.1 Front Panel Serial Debug Port

The front panel debug port (through J23) is a minimal implementation of a to-terminal-only RS-232C serial port. (Refer to Appendix B for a discussion of RS-232C signals.) It uses DRXD as its transmit data output and DTXD as its receive data input. It drives DDCD and DDSR true, controls DCTS with a software bit, and monitors DRTS with another software bit, providing minimal flow control. See Figure 4-3. The baud rate generator for the serial port is timer C of the MC68901 MFP. The XTAL input to the MC68901 is 1.230769 MHz. The baud rates supported are programmed as shown in Table 4-5.



NOTES: △1 GPIO3 PROGRAMMED AS OUTPUT.

△2 TC AND RC ARE DRIVEN BY OUTPUT OF TIMER C.

△3 GPIO0 PROGRAMMED AS INPUT.

FIGURE 4-3. MVME133XT Debug Port (To Terminal Only)

TABLE 4-5. Debug Port Baud Rates Available with XTAL = 1.230769 MHz

DESIRED BAUD RATE	CLOCK MODE	PRE- SCALE	TIMER C COUNT	ACTUAL RATE	PERCENT ERROR
9600	x16	4	1	9615.4	0.16
4800	x16	4	2	4807.7	0.16
2400	x16	4	4	2403.8	0.16
1200	x16	4	8	1201.9	0.16
600	x16	4	$10	601.0	0.16
300	x16	4	$20	300.5	0.16
110	x16	4	$57	110.5	0.47

4.3.11.2 Timers

The MC68901 MFP provides the MVME133XT with four timers, assigned as follows:

TIMER C - Baud rate generator for the front panel serial debug port.

TIMER A - Software tick timer. The tick timer is capable of generating a periodic interrupt. Refer to Appendix D for an example of its setup.

TIMER B - Tick timer overflow/watchdog time-out. The watchdog timer resets the MPU module when timer B output is high after a programmable interval, if J2 pins 1-2 are connected. SYSRESET* is also activated if the MVME133XT is the system controller.

TIMER D - Delay mode only. Unassigned by hardware.

4.3.11.3 MFP Status and Control Register (GPIP)

The MC68901 MFP has eight General Purpose I/O (GPIO) pins. The MVME133XT uses five of these pins as status inputs and three of them as control outputs. All inputs may be latched and generate interrupts. After a reset, the MC68901 MFP makes all of the GPIO pins inputs. Therefore, after each reset, the software should initialize the control bits and make them outputs. MVME133XT hardware defaults the control lines to high when they are not programmed as outputs. GPIO0-GPIO7 pins are assigned as follows:

GPIO0 - Input connected to [DRTS*]. General Purpose I/O Interrupt Port (GPIP) bit 0 is 0 when DRTS is high on the debug RS-232C interface. GPIP bit 0 is 1 when DRTS is low on the debug RS-232C interface. Bit 0 of the Interrupt Pending Register B (IPRB) may be initialized by software to detect the transitions of DRTS.

GPIO1 - Input connected to [LOCKVBE*]. This signal is driven low when a VMEbus access initiated by this module is terminated with BERR* or when an RMW-LOCK condition occurs. Because [LOCKVBE*] always goes back high at the end of the error cycle, GPIP bit 1 always reads as 1 by the time software reads it. However, software must initialize IPRB bit 1 to latch the fact that [LOCKVBE*] has pulsed low. IPRB bit 1 may then be read and cleared by software. Refer to Appendix E, Bus Error Processing, for the use of this status bit.

GPIO2 - Input connected to [LOCKLTO]. This signal is driven high when an MPU access is terminated by the local bus timer or when an RMW-LOCK condition occurs. Because [LOCKLTO] always goes back low at the end of the fault cycle, GPIP bit 2 always reads as 0 by the time software reads it. However, software must initialize IPRB bit 2 to latch the fact that [LOCKLTO] has pulsed high. IPRB bit 2 then may be read and cleared by software. Refer to Appendix E, Bus Error Processing, for the use of this status bit.

GPIO3 - Control connected to [DCTS*]. When bit 3 of GPIP is 0, DCTS is high on the debug RS-232C interface. When bit 3 of GPIP is 1 or when it is programmed as input, DCTS is low on the debug RS-232C interface.

GPIO4 - Control connected to [IE*]. When bit 4 of GPIP is 1 or when it is programmed as input, no interrupt requests reach the MPU. When bit 4 of GPIP is 0, interrupt requests may reach the MPU.

GPIO5 - Control connected to [BRDFAIL]. When bit 5 of GPIP is 1 or when it is programmed as input, the FAIL indicator is lit. Also, if the MVME133XT is not the system controller, it drives the SYSFAIL* line on the VMEbus low during this time. When bit 5 of GPIP is 0, the SYSFAIL* line is not driven by the MVME133XT and the FAIL indicator is not lit.

GPIO6 - Input connected to [OIRQ]. When [OIRQ] is 1, the MVME133XT is driving an interrupt request on the VMEbus. [OIRQ] transitions from 1 to 0 when the MVME133XT interrupt is acknowledged on the VMEbus. Transitions on [OIRQ] may be detected and latched in Interrupt Pending Register A (IPRA) bit 6. [OIRQ] is cleared by reset.

GPIO7 - Input connected to [SYSFAIL]. When SYSFAIL* is low, bit 7 of the GPIP is 1. When SYSFAIL* is high, bit 7 of the GPIP is 0. Transitions on SYSFAIL* may be detected and latched in IPRA bit 7.

4.3.11.4 Module Status Register (MSR)

In addition to the status and control bits that are implemented with the MC68901 MFP, the MVME133XT has eight status bits that are read only, have no latching mechanism, and cause no interrupts (with one exception). Collectively, these bits are called the Module Status Register (MSR). Because of hardware savings, the MSR and the MFP are grouped together and appear as a 16-bit word port to the MPU. (Refer to paragraph 3.3.1 and Table 3-3.) The MSR is located at $FFF80000 through $FFF9FFFE and may be read with either 8-bit or 16-bit transfers. When 16-bit accesses are used, the read data of the MSR appear at the most significant byte of the 16-bit word. Note that even though the MSR ignores all write accesses, a write to the MSR will affect the MFP.

The MC68901 MFP appears on the lower byte of the word, and the MSR appears on the upper byte. The bit assignments for the MSR are:

BIT 15	BIT 14	BIT 13	BIT 12	BIT 11	BIT 10	BIT 9	BIT 8
ACFAIL	SYSCON	PWRUP*	SRBIT4	SRBIT3	SRBIT2	SRBIT1	SRBIT0

ACFAIL - When VMEbus ACFAIL* is low, this bit is 1. When ACFAIL* is high, it is 0. [ACFAIL] is also an input to the interrupt handler.

SYSCON - If this module is the VMEbus system controller, this bit is 1. When it is not the VMEbus system controller, this bit is 0.

PWRUP* - This bit is set to 0 upon Power-Up reset. (Refer to paragraph 4.3.10.2.) It is bit 13 when the MSR is accessed with 16-bit data transfer, and is bit 5 when the MSR is accessed with 8-bit data read. Any access to memory locations $FFFD0000 through $FFFDFFFF and any write access to the MK48T02 or the SCC ($FFFA0000 through $FFFCFFFF) changes this latched bit from logical 0 to 1. A suggested method is to perform a test on location $FFFD0000 (for example, TST.B $FFFD0000).

SRBIT4 - This bit is 0 when J21 pins 9-10 are connected, and is 1 when they are open.

SRBIT3 - This bit is 0 when J21 pins 7-8 are connected, and is 1 when they are open.

SRBIT2 - This bit is 0 when J21 pins 5-6 are connected, and is 1 when they are open.

SRBIT1 - This bit is 0 when J21 pins 3-4 are connected, and is 1 when they are open.

SRBIT0 - This bit is 0 when J21 pins 1-2 are connected, and is 1 when they are open.

4.3.12 Dual Multiprotocol Serial Ports and Z8530 Serial Communications Controller (SCC) (Sheet 21)

The MVME133XT uses the Z8530 SCC to implement its two multiprotocol serial ports. The SCC occupies 128Kb in the MVME133XT memory map and is located at a physical base address of $FFFA0000. Refer to Chapter 3 for map details.

In the SCC, register addressing is direct for the data registers only. In all other cases (with the exception of SCCx-WR0 and SCCx-RR0), accessing the internal SCC read and write registers requires a sequence of two operations. The first operation is a write to SCCx-WR0 with the four least significant bits that point to the selected register. If the second operation is a write, then the selected write register is accessed. On the other hand, if the second operation is a read, then the selected read register is selected. The pointer bits are automatically cleared after the second read or write operation so that SCCx-WR0 (or SCCx-RR0) is addressed again on the next access. Refer to the Z8530 Serial Communications Controller data sheet (listed in Chapter 1 herein) for details on programming and using the SCC.

The SCC provides multifunction support for handling the large variety of serial communications protocols available. The Z8530 can be programmed to satisfy special serial communication requirements as well as standard formats such as byte-oriented synchronous, bit-oriented synchronous, and asynchronous. In addition, protocol variations are supported within each operating mode by checking odd or even parity, character insertion or deletion, CRC generation and checking, break and abort generation and detection, and many other protocol-dependent features.

Port A of the SCC is connected to onboard RS-485 drivers and receivers. Port B of the SCC is connected to onboard RS-232C drivers and receivers. Because of its internal structure, there are several means of obtaining the baud rate clocks for each of the two serial channels. Each channel within the SCC has a programmable baud rate generator. The Baud Rate Generator (BRG) input can be from the RTXC input or from PCLK. The hardware on the MVME133XT allows the RTXC pin for each channel to be connected to an external clock source or to the onboard 1.230769 MHz clock. (Refer to Chapter 2.) The values in the SCC time constant register that are required to create some common baud rates are shown in Table 4-6.

TABLE 4-6. Baud Rates Available with BRG Clock = RTXC Pin = 1.230769 MHz

BAUD RATE	CLOCK MODE	TIME CONSTANT REGISTER VALUE	ACTUAL BAUD RATE	PERCENT ERROR
19200	x16	0	19231	0.16
9600	x16	2	9615	0.16
4800	x16	6	4808	0.16
2400	x16	$E	2404	0.16
1200	x16	$1E	1202	0.16
600	x16	$3E	601	0.16
300	x16	$7E	300	0.16
110	x16	$15E	109	0.67
64000	x1	8	61538	3.85
56000	x1	9	55944	0.10
48000	x1	$B	47337	1.38
38400	x1	$E	38461	0.16

The SCC DPLL input can be either the BRG output or the RTXC pin. The DPLL operates at 32 times the data rate for NRZI and at 16 times the data rate for FM. Some of the data rates that are achievable with the MPU operating at 25 MHz (in the MVME133XT) (PCLK = 3.125 MHz) are given in Table 4-7.

TABLE 4-7. Baud Rates Available with BRG Clock = PCLK = 3.125 MHz (MVME133XT)

BAUD RATE	CLOCK MODE	TIME CONSTANT REGISTER VALUE	ACTUAL BAUD RATE	PERCENT ERROR
N/A	x32	0	24414	N/A
N/A	x32	1	16276	N/A
N/A	x32	2	12207	N/A
48000	x16	0	48828	1.7
N/A	x16	1	32552	N/A
N/A	x16	2	24414	N/A

If other frequencies than the ones available with 1.230769 MHz as the BRG clock source are needed, the frequency of 1.230769 MHz can be changed by reprogramming U28, PALSCON, to divide the 16 MHz by a value other than 13.

NOTE

Note that both ports of the Z8530 SCC and the MC68901 MFP may be using the 1.230769 MHz signal, and changing that frequency may make it impossible to create a desired frequency on the other port of the SCC and/or on the MC68901 MFP debug port.

4.3.12.1 RS-485 Port

Z8530 SCC Port A uses RS-485/RS-422 drivers and receivers. The RS-485 signals are routed to P2 rows A and C. An external cable may be connected to P2 and the user must make a crossover cable to convert from the cable pinout of P2 on MVME133XT to the pinout of the user's serial network. The connector used to interface to the RS-485 network should take shielding into consideration.

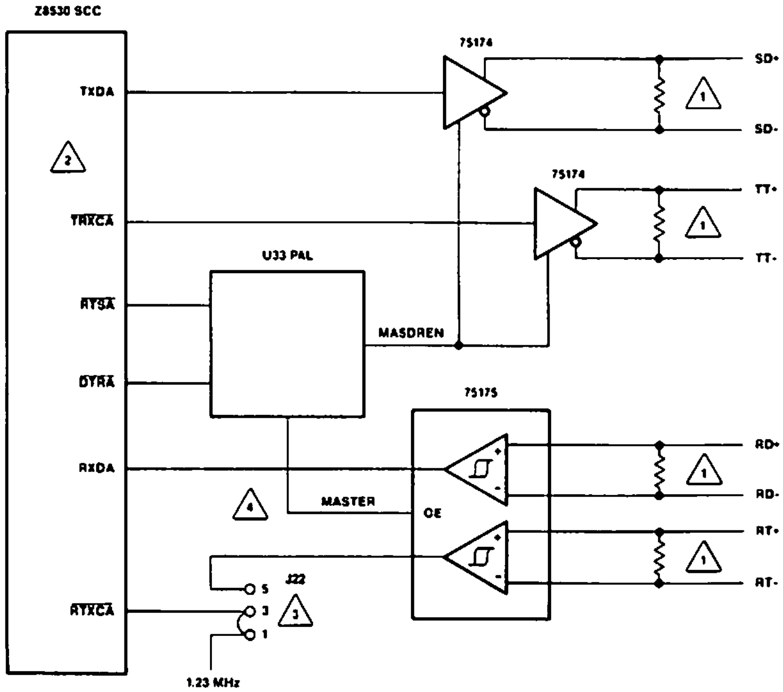
The RS-485 port is configured by software as either master or slave and half or full duplex by controlling DTR/REQA (DTRA) and RTSA of port A. PAL U33 defines the functions of these two control bits. U33 (sheet 14), as factory programmed, defines them: DTRA indicates master when it is high (1) and slave when low (0), and RTSA enables the RS-485 drivers when it is low (0). The user may change the functions of RTSA and DTRA by reprogramming U33. The possible RS-485 port configurations with the default program in U33 are shown in Table 4-8. The port as master is in Figure 4-4. The port as slave is in Figure 4-5. Refer to Appendix A for U33 program details.

TABLE 4-8. RS-485 Port Configurations

DTRA	RTSA	CONFIGURATION	DESCRIPTION
low	low	Slave (drivers on)	TXDA drives RD+/-, SD+/- drives RXDA, TRXCA drives RT+/-, TT+/- drives RTXCA if J22 pins 3-5 are connected.
low	low	Half duplex receive	(TXDA drives RD+/-), SD+/- drives RXDA, (TRXCA drives RT+/-), TT+/- drives RTXCA if J22 pins 3-5 are connected.
low	high	Slave (drivers off)	RD+/- not driven, SD+/- drives RXDA, RT+/- not driven, TT+/- drives RTXCA if J22 pins 3-5 are connected.
low	high	Half duplex receive	RD+/- not driven, SD+/- drives RXDA, RT+/- not driven, TT+/- drives RTXCA if J22 pins 3-5 are connected.
high	low	Master (drivers on)	TXDA drives SD+/-, RD+/- drives RXDA, TRXCA drives TT+/-, RT+/- drives RTXCA if J22 pins 3-5 are connected.
high	low	Half duplex send	TXDA drives SD+/-, (RD+/- drives RXDA), TRXCA drives TT+/-, (RT+/- drives RTXCA if J22 pins 3-5 are connected).
high (NOTE)	high (NOTE)	Master (drivers off) (NOTE)	SD+/- not driven, RD+/- drives RXDA, TT+/- not driven, RT+/- drives RTXCA if J22 pins 3-5 are connected.

NOTE: After a RESET, port A defaults to this configuration with DTRA (DTR/REQA) = high and RTSA = high (Master with drivers off).

MVME133XT has only clocks and data; RS-485 port has no hardware handshakes.



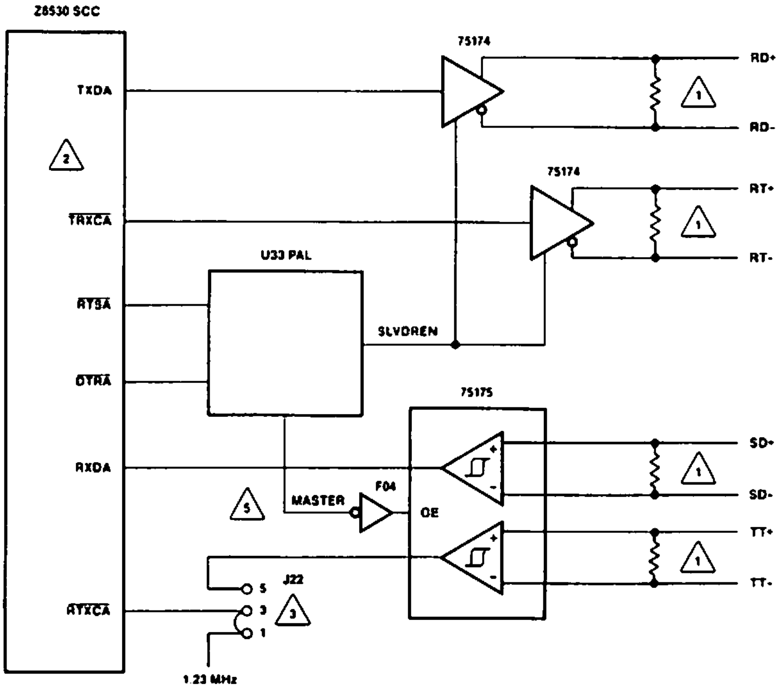
NOTES: △1 PART OF RESISTOR PACK SIP, 120 OHMS.

△2 AUTO-ENABLE BIT FOR PORT 'A' MUST BE CLEARED.

△3 FACTORY CONFIGURATION.

△4 $\overline{\text{DTRA}}$ = 1 TO INDICATE 'MASTER'.

FIGURE 4-4. MVME133XT RS-485 Port Configured as Master (To DCE)



NOTES: 1 PART OF RESISTOR PACK SIP, 120 OHMS.

2 AUTO-ENABLE BIT FOR PORT 'A' MUST BE CLEARED.

3 FACTORY CONFIGURATION.

5 $\overline{\text{DTRA}}$ = 0 TO INDICATE 'SLAVE'.

FIGURE 4-5. MVME133XT RS-485 Port Configured as Slave (To DTE)

4.3.12.2 RS-232C Port

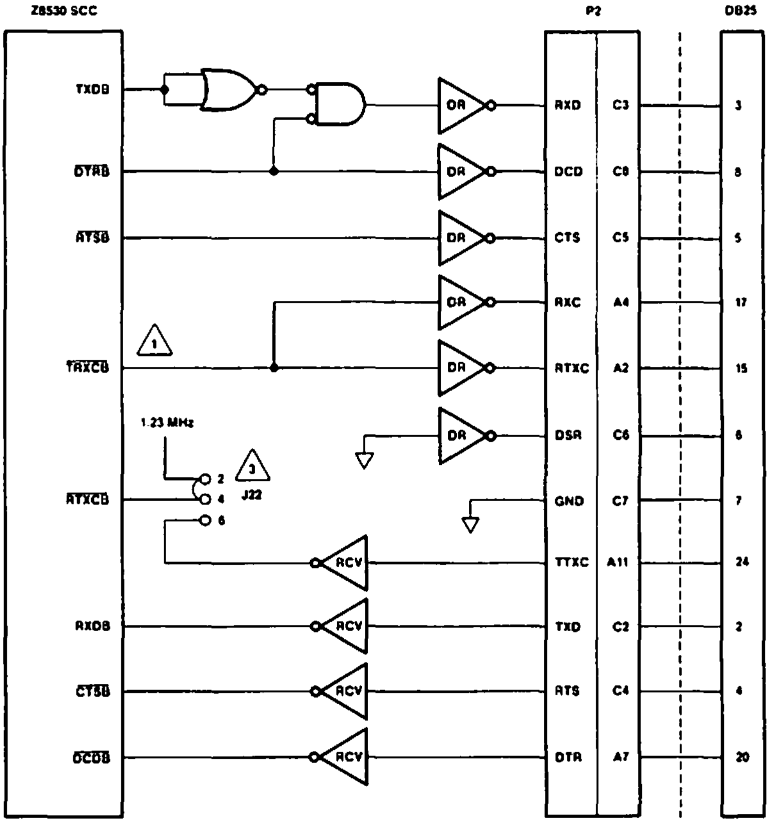
Port B of the SCC uses RS-232C drivers and receivers. All of the buffers and the configuration headers are on the MVME133XT. This port may be configured either as a DTE or as a DCE, by using J18. (Refer to Chapter 2.) The DCE configuration of the port is shown in Figure 4-6. The DTE configuration of the port with RTXC not used is shown in Figure 4-7. The MVME133XT also provides an external ability to disable the TXDB pin of the Z8530 SCC. When the DTRB pin of the Z8530 SCC is high, the TXDB pin is disabled to the RS-232C port. When the DTRB pin is low, the TXDB pin is enabled to the RS-232C port. Note that the DTRB pin is also an RS-232C signal line. DTRB is set high by a reset to the Z8530 SCC.

All the RS-232C lines of this port are routed to P2 rows A and C. An external cable may be connected to P2 and a DB-25 connector crimped directly onto it. For shielding purposes, the DB-25 should be mounted to a back panel that is mounted to the chassis, and connection should be made between the back panel and the shielding metal of the DB-25.

Refer to Appendix C for an example of setting up software for serial port B.

<u>CAUTION</u>

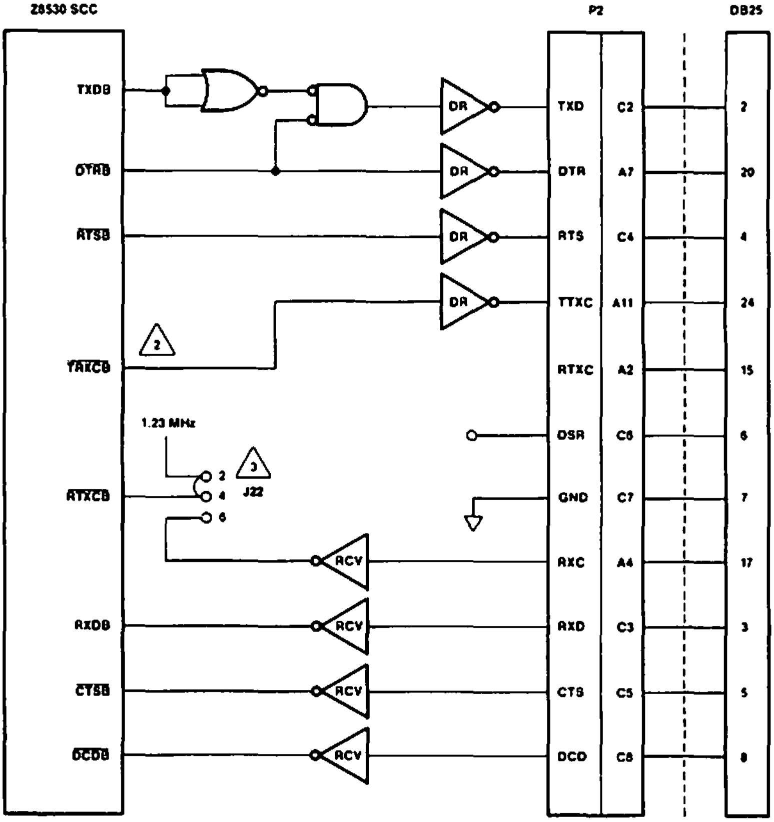
SET UP SERIAL PORT B FOR THE SAME HARDWARE CHARACTERISTICS (PARAGRAPHS 2.3.16 AND 2.3.20) AS USED FOR THE SOFTWARE CHARACTERISTICS (SUCH AS IN APPENDIX C) OR THE PORT WILL NOT OPERATE PROPERLY.



NOTES: △1 TRXCB MUST BE PROGRAMMED AS OUTPUT.

△3 FACTORY CONFIGURATION.

FIGURE 4-6. MVME133XT RS-232C Port Configured as DCE (To Terminal)



NOTES: 2 TRXCB MUST BE PROGRAMMED AS OUTPUT.

3 FACTORY CONFIGURATION.

FIGURE 4-7. MVME133XT RS-232C Port Configured as DTE (To Modem) with RTXC Not Used

4.3.13 ROM/PROM/EPROM/EEPROM Sockets and Battery Backup Real-Time Clock with SRAM (Sheet 22)

4.3.13.1 ROM/PROM/EPROM/EEPROM Sockets

The MVME133XT has four 28-pin ROM/PROM/EPROM/ EEPROM sockets that are organized as two banks with two sockets per bank. They are arranged as follows:

Bank 1: XU31 = even, XU12 = odd;
Bank 2: XU20 = even, XU3 = odd.

Each bank appears as a 16-bit word port to the MPU and can be separately configured for 8K x 8, 16K x 8, 32K x 8, or 64K x 8 ROM/PROM/EPROMs; or for 2K x 8, 8K x 8, or 32K x 8 EEPROMs. Definitions of the ROM/PROM/EPROM/EEPROM socket pins, depending upon the configuration used, are shown in Figure 4-8.

CONFIGURATION								CONFIGURATION				
1	2	3	4	5				5	4	3	2	1
+5V	+5V	A15	NC	A14	1		28	+5V	+5V	+5V	+5V	+5V
A12	A12	A12	A12	A12	2	SOCKET	27	WE*	WE*	A14	A14	VIH
A7	A7	A7	A7	A7	3		26	A13	A13	A13	A13	A13
A6	A6	A6	A6	A6	4		25	A8	A8	A8	A8	A8
A5	A5	A5	A5	A5	5		24	A9	A9	A9	A9	A9
A4	A4	A4	A4	A4	6		23	A11	A11	A11	A11	A11
A3	A3	A3	A3	A3	7		22	OE*	OE*	OE*	OE*	OE*
A2	A2	A2	A2	A2	8		21	A10	A10	A10	A10	A10
A1	A1	A1	A1	A1	9		20	CE*	CE*	CE*	CE*	CE*
A0	A0	A0	A0	A0	10		19	DQ7	DQ7	D7	D7	D7
D0	D0	D0	DQ0	DQ0	11		18	DQ6	DQ6	D6	D6	D6
D1	D1	D1	DQ1	DQ1	12		17	DQ5	DQ5	D5	D5	D5
D2	D2	D2	DQ2	DQ2	13		16	DQ4	DQ4	D4	D4	D4
GND	GND	GND	GND	GND	14		15	DQ3	DQ3	D3	D3	D3

<u>NOTES</u>

SOCKET A14 = BOARD [A15].

SOCKET A15 = BOARD [A16].

SEE SCHEMATIC DIAGRAM, FIGURE 5-2.

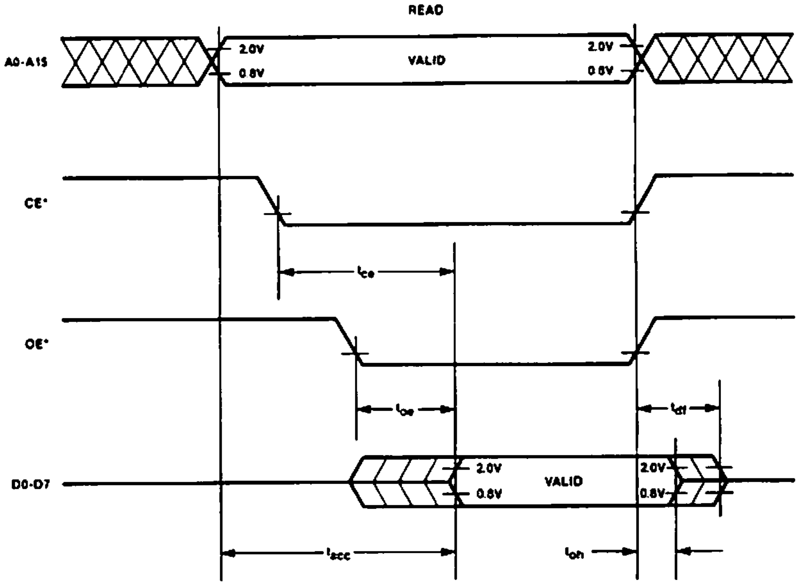
REFER TO PARAGRAPH 2.3.9.

MVME133XTBUG USES 64K x 8 EPROMs.

CONFIGURATION	J10 (BANK 1) OR J11 (BANK 2) CONNECTIONS	DEVICES SUPPORTED
1	1 to 3	8K x 8 or 16K x 8 ROM/PROM/EPROM
2	1 to 3, and 2 to 4	32K x 8 ROM/PROM/EPROM
3	2 to 4, and 3 to 5	64K x 8 ROM/PROM/EPROM (FACTORY CONFIG.)
4	4 to 6	2K x 8 or 8K x 8 EEPROM
5	2 to 3 (wirewrap), and 4 to 6	32K x 8 EEPROM

FIGURE 4-8. ROM/PROM/EPROM/EEPROM Sockets Configurations

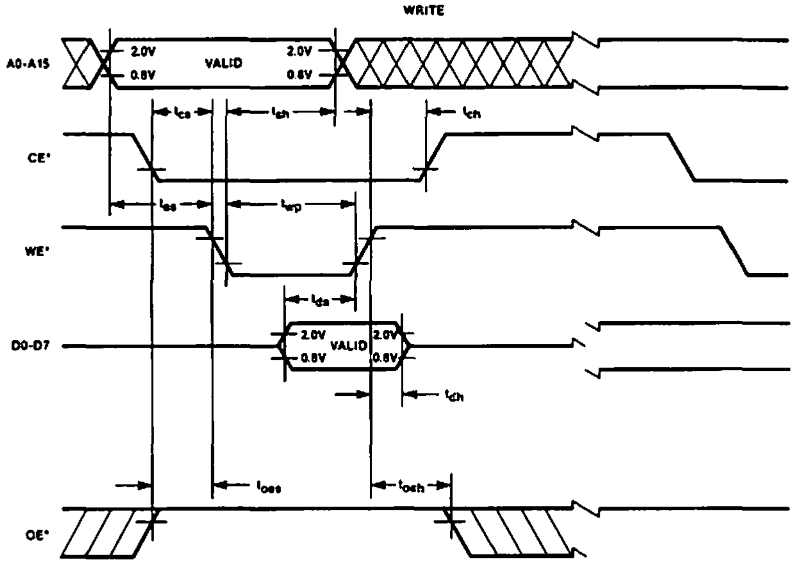
The ROM/PROM/EPROM/EEPROM devices must meet the read timings shown in Figure 4-9. They are guaranteed the write timings shown in Figure 4-10.



SYMBOL	DESCRIPTION	TIME IN NSEC MVME133XT (25 MHz) MIN	 MAX
tacc	Address valid to data valid	---	230
tce	CE* low to data valid	---	215
toe	OE* low to data valid	---	175
toh	Address invalid, CE* or OE* high to data not valid	0	---
tdf	CE* or OE* high to data high impedance	---	40

NOTE: The MVME133XT does not guarantee a maximum transition time on address and data lines during the time that CE* is high.

FIGURE 4-9. ROM/PROM/EPROM/EEPROM Read Timings Required by MVME133XT



SYMBOL	DESCRIPTION	TIME IN NSEC MVME133XT (25 MHz) MIN	 MAX
tas	Address valid to WE* low	45	---
tcs	CE* low to WE* low	30	---
toes	OE* high to WE* low	55	---
tah	Address valid after WE* low	175	---
twp	WE* low pulse width	120	---
tds	Data valid to WE* high	160	---
tdh	WE* high to data not valid	50	---
toeh	WE* high to OE* low	130	---
tch	WE* high to CE* high	30	---

NOTE: The MVME133XT does not guarantee a maximum transition time on address and data lines during the time that CE* is high.

FIGURE 4-10. EEPROM Write Timings Guaranteed by MVME133XT

Consider the following when using EEPROMs on the MVME133XT:

1. The MVME133XT provides no protection against inadvertent writes to EEPROM that might happen during power on/off transitions. Most devices provide some level of internal protection. In order to gain "absolute protection", devices with additional "software protection" are recommended.

2. When a bank is configured for EEPROM, writes to that bank must always be 16-bit wide. This is because any access to one byte of the bank also causes an access to the other byte; thus, byte-wide access causes unintended data to be written to the other byte.

3. There are several different algorithms for erasing/writing to EEPROMs, depending on the manufacturer. The MVME133XT supports only those devices which have a "static RAM" compatible erase/write mechanism.

4. Note that the MVME133XT requires that the EEPROMs must allow wired-OR on the RDY/BSY* pin (for 2K x 8 and 8K x 8 devices). The MVME133XT, however, does not monitor the status of the RDY/BSY* pins.

4.3.13.2 MK48T02 Battery Backup Real-Time Clock with SRAM

The Thompson Components Mostek MK48T02 is utilized by the MVME133XT to provide 2040 bytes of battery backup SRAM and a battery backup real-time clock. The MK48T02 is mapped at a physical base address of $FFFC0000. Its 2 Kb appears redundantly in a 64 Kb block from $FFFC0000 through $FFFCFFFF. Some of the features of the MK48T02 are:

- Integrated ultra-low power SRAM, real-time clock, crystal, power-fail control circuit and battery.
- Byte-wide RAM-like clock access.
- BCD-coded year, month, date, day, hours, minutes, and seconds.
- Software-controlled clock calibration for high accuracy applications.
- Automatic power-fail protection.

The real-time clock (RTC) is provided by the MK48T02. Accessing the RTC is as simple as conventional byte-wide SRAM access via the eight RTC registers located in the upper eight locations of the MK48T02 ($FFFC07F8 through $FFFC07FF). These RTC registers contain year, month, date, day, hours, minutes, and seconds data in 24-hour BCD format. Corrections for 28-, 29- (leap year), 30-, and 31-day months are made automatically. The eighth location is a control register. These RTC registers are not the actual clock counters; instead, they are bi-port read/write SRAM memory locations. The clock control circuit dumps the counters into these bi-port locations once every second. Note that no interrupts are generated by the RTC.

The MVME133XT is shipped with the RTC oscillator stopped to minimize current drain from the on-chip battery on the MK48T02. The battery has a predicted storage life of better than 10 years at storage temperature below 70 degrees C. The backup system life is better than six years at 50% Vcc duty cycle, and better than three years at 0% Vcc duty cycle. If the MVME133XT is going to spend a long period of time on the shelf, then the MK48T02 should be placed in the power-save mode (that is, with the oscillator turned off) to prolong the shelf life of the battery. The MVME133XT is shipped from the factory with the MK48T02 in power-save mode. Before the RTC can be used, its oscillator requires a "kick start" to begin oscillation. Refer to the MK48T02 data sheet (listed in Chapter 1 herein) for calculating and predicting the battery storage life and the backup system life, and for programming and utilizing the RTC.

There are a total of 2040 bytes of non-volatile SRAM available in the MK48T02. The SRAM may be accessed at $FFFC0000 through $FFFC07F7. Because the RTC registers are constructed using bi-port memory cells, access to the rest of the SRAM proceeds unhindered by updates to the RTC registers, even if these RTC registers are being updated at the very moment another location in the memory array is accessed.

Note that a write to the MK48T02 also resets the PWRUP* status flag to a logical 1. (Refer to paragraph 4.3.11.4.)

CHAPTER 5 - SUPPORT INFORMATION

5.1 INTRODUCTION

This chapter provides the interconnection signals, parts list with parts location illustration, and schematic diagram for the MVME133XT module.

5.2 INTERCONNECT SIGNALS

The MVME133XT module interconnects with the VMEbus through connector P1, with the VMEbus and serial ports through connector P2, and with an RS-232C device through connector J23.

5.2.1 Connector P1 Interconnect Signals

Connector P1 is a standard DIN 41612 triple-row, 96-pin male connector. The MVME133XT interconnects with the VMEbus through rows A, B, and C of P1 and through row B of P2. Each pin connection, signal mnemonic, and signal characteristic for the connector is listed in Table 5-1.

5

TABLE 5-1. Connector P1 Interconnect Signals

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
A1-A8	D00-D07	Data bus (bits 0-7) - eight of 32 three-state bidirectional data lines that provide the data path between VMEbus master and slave.
A9	GND	GROUND - connected to the MVME133XT ground plane.
A10	SYSCLK	SYSTEM CLOCK - a 16 MHz free-running clock that is driven by the MVME133XT only when it is configured as system controller.
A11	GND	GROUND - connected to the MVME133XT ground plane.
A12	DS1*	DATA STROBE 1 - signal that indicates which part of the data bus is transferring data. It is driven by the MVME133XT when it is the VMEbus master. It is received by the MVME133XT when it is a VMEbus slave.
A13	DS0*	DATA STROBE 0 - signal that indicates which part of the data bus is transferring data. It is driven by the MVME133XT when it is the VMEbus master. It is received by the MVME133XT when it is a VMEbus slave.

TABLE 5-1. Connector P1 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
A14	WRITE*	WRITE - signal that specifies the direction of data transfers. It is driven by the MVME133XT as a VMEbus master and received by the MVME133XT as a slave.
A15	GND	GROUND - connected to the MVME133XT ground plane.
A16	DTACK*	DATA TRANSFER ACKNOWLEDGE - signal that indicates that valid data is available on the data bus during a read cycle or that it has been accepted during a write cycle. It is received by the MVME133XT as a VMEbus master and driven by the MVME133XT as a VMEbus slave.
A17	GND	GROUND - connected to the MVME133XT ground plane.
A18	AS*	ADDRESS STROBE - the falling edge of this signal indicates that a valid address, address modifier, LWORD*, and IACK* are available on the VMEbus. It is driven by the MVME133XT as a VMEbus master and received by it as a VMEbus slave.
A19	GND	GROUND - connected to the MVME133XT ground plane.
A20	IACK*	INTERRUPT ACKNOWLEDGE - signal that indicates an interrupt acknowledge cycle on the VMEbus. It is driven true by the MVME133XT during an interrupt acknowledge to the VMEbus.
A21	IACKIN*	INTERRUPT ACKNOWLEDGE IN - IACKIN* and IACKOUT* form a daisy-chained signal. The MVME133XT drives IACKOUT* low if there is an activated IACKIN* and the interrupt acknowledge level is not for this module or if it does not have an interrupt pending. Also, when the MVME133XT is configured as system controller, it drives IACKOUT* according to the IACK daisy-chain driver specification.
A22	IACKOUT*	INTERRUPT ACKNOWLEDGE OUT - see IACKIN*.
A23	AM4	ADDRESS MODIFIER (bit 4) - one of the three-state lines that provide additional information about the address bus, such as size, and cycle type. It is driven by the MVME133XT as a master and received by the MVME133XT as a slave.

TABLE 5-1. Connector P1 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
A24-A30	A07-A01	ADDRESS bus (bits 7-1) - seven of 31 three-state lines that specify an address in the memory map. They are driven by the MVME133XT as a master and received by the MVME133XT as a slave.
A31	-12 VDC	-12 Vdc power - used by the RS-232C drivers on the MVME133XT.
A32	+5 VDC	+5 Vdc power - used by the logic circuits on the MVME133XT. Connected to the MVME133XT +5V plane.
B1	BBSY*	BUS BUSY - this signal is driven true by the MVME133XT when it is VMEbus master. When the MVME133XT is system controller, BBSY* is an input to the level 3 arbiter.
B2	BCLR*	BUS CLEAR - not used by the MVME133XT.
B3	ACFAIL*	AC FAILURE - the MVME133XT monitors this signal line to detect ac power failure.
B4	BG0IN*	BUS GRANT IN (level 0) - this signal going true at the input to the MVME133XT indicates that it may become VMEbus master if it is configured for level 0. If the MVME133XT is not requesting VMEbus mastership, then it drives the BG0OUT* signal line low. The other three bus grant lines are tied directly to the corresponding bus grant out lines.
B5	BG0OUT*	BUS GRANT OUT (level 0) - see BG0IN*.
B6	BG1IN*	BUS GRANT IN (level 1) - same as BG0IN* on pin B4.
B7	BG1OUT*	BUS GRANT OUT (level 1) - same as BG0OUT* on pin B5.
B8	BG2IN*	BUS GRANT IN (level 2) - same as BG0IN* on pin B4.
B9	BG2OUT*	BUS GRANT OUT (level 2) - same as BG0OUT* on pin B5.
B10	BG3IN*	BUS GRANT IN (level 3) - same as BG0IN* on pin B4.

TABLE 5-1. Connector Pl Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
B11	BG3OUT*	BUS GRANT OUT (level 3) - same as BG0OUT* on pin B5.
B12	BR0*	BUS REQUEST (level 0) - signal line driven by the MVME133XT when it desires to become VMEbus master (if it is configured for level 0), and received by the MVME133XT to detect whether it should relinquish VMEbus mastership.
B13	BR1*	BUS REQUEST (level 1) - same as BR0* on pin B12.
B14	BR2*	BUS REQUEST (level 2) - same as BR0* on pin B12.
B15	BR3*	BUS REQUEST (level 3) - same as BR0* on pin B12. Also, when the MVME133XT is configured as system controller, BR3* is an input to the level 3 arbiter.
B16-B19	AM0-AM3	ADDRESS MODIFIER (bits 0-3) - same as AM4 on pin A23.
B20	GND	GROUND - connected to the MVME133XT ground plane.
B21	SERCLK	Not used.
B22	SERDAT*	Not used.
B23	GND	GROUND - connected to the MVME133XT ground plane.
B24-B27	IRQ7*-IRQ4*	INTERRUPT REQUEST (7-4) - four of the seven prioritized interrupt request inputs to the MVME133XT. Jumper enabled, level 7 is the highest priority.
B28	IRQ3*	INTERRUPT REQUEST (3) - one of the seven prioritized interrupt request inputs/outputs of the MVME133XT. Jumper enabled as input, and jumper enabled as output.
B29-B30	IRQ2*-IRQ1*	INTERRUPT REQUEST (2-1) - two of the seven prioritized interrupt request inputs to the MVME133XT. Jumper enabled, level 7 is the highest priority.
B31	+5V STDBY	Not used.
B32	+5 VDC	+5 Vdc power - same as +5 VDC on pin A32.

TABLE 5-1. Connector P1 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
C1-C8	D08-D15	DATA bus (bits 8-15) - same as D00-D07 on pins A1-A8.
C9	GND	GROUND - connected to the MVME133XT ground plane.
C10	SYSFAIL*	SYSTEM FAIL - signal driven by the MVME133XT when [BRDFAIL] is true if it is not the system controller. Also can be monitored via the MC68901 MFP.
C11	BERR*	BUS ERROR - signal driven by the MVME133XT bus time-out circuit when it is the system controller and a VMEbus data strobe cycle exceeds 72 to 82 us. Also monitored by the MVME133XT when it is the VMEbus master. It causes a bus error exception in the MC68020 in this case.
C12	SYSRESET*	SYSTEM RESET - signal driven by the MVME133XT when it is configured as system controller during power-up, when the front panel RESET switch is depressed, when a watchdog time-out occurs, or when remote reset becomes true. Also an input to the MVME133XT that causes all of its devices to be reset.
C13	LWORD*	LONGWORD - signal driven true by the MVME133XT when it does a 32-bit data transfer over the VMEbus. Also monitored by the MVME133XT to distinguish 32-bit from 16-bit data accesses to its RAM from the VMEbus.
C14	AM5	ADDRESS MODIFIER (bit 5) - same as AM4 on pin A23.
C15-C30	A23-A08	ADDRESS bus (bits 23-08) - 16 of 31 three-state lines that specify an address in the memory map. They are driven by the MVME133XT as a master and received by the MVME133XT as a slave.
C31	+12 VDC	+12 Vdc power - used by the RS-232C drivers on the MVME133XT.
C32	+5 VDC	+5 Vdc power - used by the logic circuits on the MVME133XT. Connected to the MVME133XT +5V plane.

5.2.2 Connector P2 Interconnect Signals

Connector P2 is a standard DIN 41612 triple-row, 96-pin male connector. The MVME133XT interconnects with the VMEbus through rows A, B, and C of P1 and through row B of P2. Serial ports A (RS-485) and B (RS-232C) of the Z8530 and remote reset are brought out through rows A and C of P2. (For suggested cables to connect to P2, refer to Chapter 2.) Each pin connection, signal mnemonic, and signal characteristic for the connector is listed in Table 5-2.

TABLE 5-2. Connector P2 Interconnect Signals

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
A1	--	Not used.
A2	RTXC	RS-232C Transmitter Signal Element Timing (DCE) - signal line driven by the TRXCB pin of the Z8530 Serial Communications Controller (SCC) when port B is configured as DCE, and not used when port B is configured as DTE.
A3	--	Not used.
A4	RXC	RS-232C Receiver Signal Element Timing (DCE) - signal line driven by the TRXCB pin of the Z8530 SCC when port B is configured as DCE, and optionally input to the RTXCB pin of the Z8530 SCC when port B is configured as DTE.
A5-A6	--	Not used.
A7	DTR	RS-232C Data Terminal Ready - input to the DCDB pin of the Z8530 SCC when port B is configured as DCE, and output from the DTR/REQB (DTRB) pin of the Z8530 SCC when port B is configured as DTE.
A8-A10	--	Not used.
A11	TTXC	RS-232C Transmitter Signal Element Timing (DTE) - signal line optionally driven by the TRXCB pin of the Z8530 SCC when port B is configured as DTE, and optionally input to the RTXCB pin of the Z8530 SCC when port B is configured as DCE.
A12	--	Not used.

TABLE 5-2. Connector P2 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
A13	SD+	RS-485 Send Data - this signal line is half of the balanced differential pair that includes SD+ and SD-. The pair is buffered to the RXDA pin of the Z8530 SCC when port A is configured as slave, and it is buffered from the TXDA pin of the Z8530 SCC when port A is configured as master.
A14	TT+	RS-485 Terminal Timing - this signal line is half of the balanced differential pair that includes TT+ and TT-. The pair is buffered to the RTXCA pin of the Z8530 SCC (depends on J22) when port A is configured as slave, and it is buffered from the TRXCA pin of the Z8530 SCC when port A is configured as master.
A15	RD+	RS-485 Receive Data - this signal line is half of the balanced differential pair that includes RD+ and RD-. The pair is buffered from the TXDA pin of the Z8530 SCC when port A is configured as slave, and it is buffered to the RXDA pin of the Z8530 SCC when port A is configured as master.
A16	RT+	RS-485 Receive Timing - this signal line is half of the balanced differential pair that includes RT+ and RT-. The pair is buffered from the TRXCA pin of the Z8530 SCC when port A is configured as slave, and it is buffered to the RTXCA pin of the Z8530 SCC (depends on J22) when port A is configured as master.
A17	GND	GROUND - connects to MVME133XT ground plane.
A18-A19	--	Not used.
A20	RRESET*	Remote Reset input - when this signal line is high, no reset function occurs on the MVME133XT. When it is low, the MVME133XT is reset and remains in the reset state until it goes high again. (Note that this signal line is also connected to pin P2-A32.)
A21	--	Not used.
A22	GND	GROUND - connected to MVME133XT ground plane.
A23-A30	--	Not used.

TABLE 5-2. Connector P2 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
A31	GND	GROUND - connected to MVME133XT ground plane.
A32	RRESET*	Remote Reset input - when this signal line is high, no reset function occurs on the MVME133XT. When it is low, the MVME133XT is reset and remains in the reset state until it goes high again. (Note that this signal line is also connected to pin P2-A20.)
B1	+5 VDC	+5 Vdc power - used by the logic circuits on the MVME133XT. Connected to the MVME133XT +5V plane.
B2	GND	GROUND - connected to the MVME133XT ground plane.
B3	Reserved	Not used.
B4-B11	A24-A31	ADDRESS bus (bits 24-31) - eight of 31 three-state lines that specify an address in the memory map. They are driven by the MVME133XT as a master and received by the MVME133XT as a slave.
B12	GND	GROUND - connected to MVME133XT ground plane.
B13	+5 VDC	+5 Vdc power - used by the logic circuits on the MVME133XT. Connected to the MVME133XT +5V plane.
B14-B21	D16-D23	DATA bus (bits 16-23) - eight of 32 three-state bidirectional data lines that provide the data path between VMEbus master and slave.
B22	GND	GROUND - connected to MVME133XT ground plane.
B23-B30	D24-D31	DATA bus (bits 24-31) - eight of 32 three-state bidirectional data lines that provide the data path between VMEbus master and slave.
B31	GND	GROUND - connected to MVME133XT ground plane.
B32	+5 VDC	+5 Vdc power - used by the logic circuits on the MVME133XT. Connected to the MVME133XT +5V plane.
C1	--	Not used.
C2	TXD	RS-232C Transmitted Data - input to the RXDB pin of the Z8530 SCC when port B is configured as DCE, and output from the TXDB pin of the Z8530 SCC when port B is configured as DTE.

TABLE 5-2. Connector P2 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
C3	RXD	RS-232C Received Data - output from the TXDB pin of the Z8530 SCC when port B is configured as DCE, and input to the RDXB pin of the Z8530 SCC when port B is configured as DTE.
C4	RTS	RS-232C Request To Send - input to the CTSB pin of the Z8530 SCC when port B is configured as DCE, and output from the RTSB pin of the Z8530 SCC when port B is configured as DTE.
C5	CTS	RS-232C Clear To Send - output from the RTSB pin of the Z8530 SCC when port B is configured as DCE, and input to the CTSB pin of the Z8530 SCC when port B is configured as DTE.
C6	DSR	RS-232C Data Set Ready - output from the Z8530 SCC that is always high when port B is configured as DCE, and no connect when port B is configured as DTE.
C7	GND	RS-232C Signal Ground/Common Return - connected to the MVME133XT ground plane. NOT connected to chassis ground on the MVME133XT.
C8	DCD	RS-232C Received Line Signal Detector - output from the DTR/REQB (DTRB) pin of the Z8350 SCC when port B is configured as DCE, and input to the DCDB pin of the Z8530 SCC when port B is configured as DTE.
C9-C13	--	Not used.
C14	SD-	RS-485 Send Data - this signal is half of the balanced differential pair that includes SD+ and SD-. Refer to SD+ on pin A13.
C15	TT-	RS-485 Terminal Timing - this signal line is half of the balanced differential pair that includes TT+ and TT-. Refer to TT+ on pin A14.
C16	RD-	RS-485 Receive Data - this signal line is half of the balanced differential pair that includes RD+ and RD-. Refer to RD+ on pin A15.
C17	RT-	RS-485 Receive Timing - this signal line is half of the balanced differential pair that includes RT+ and RT-. Refer to RT+ on pin A16.

TABLE 5-2. Connector P2 Interconnect Signals (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
C18-C19	--	Not used.
C20	GND	GROUND - connected to MVME133XT ground plane.
C21	--	Not used.
C22	GND	GROUND - connected to MVME133XT ground plane.
C23-C28	--	Not used.
C29	GND	GROUND - connected to MVME133XT ground plane.
C30	--	Not used.
C31-C32	GND	GROUND - connected to MVME133XT ground plane.

5.2.3 Connector J23 Interconnect Signals

Connector J23 is a standard RS-232C DB-25 25-pin female connector. J23 provides the interconnection for the MC68901 Multifunction Peripheral (MFP) debug port of the MVME133XT. Each pin connection, signal mnemonic, and signal characteristic for the connector is listed in Table 5-3. Note that J23 mates with a 25-pin cable to connect to a terminal. For further details, refer to Appendix B.

TABLE 5-3. RS-232C Connector J23 Interconnect Signals

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1	--	Not used.
2	DTXD	RS-232C Transmitted Data - input to the SI pin of the MC68901 Multifunction Peripheral (MFP).
3	DRXD	RS-232C Received Data - output from the SO pin of the MC68901 MFP.
4	DRTS	RS-232C Request to Send - input to the GPIO0 pin of the MC68901 MFP.
5	DCTS	RS-232C Clear To Send - output from the GPIO3 pin of the MC68901 MFP.
6	DDSR	RS-232C Data Set Ready - output that is always driven high by the MVME133XT. Note that this pin is connected to pin 8.
7	GND	RS-232C Signal Ground/Common Ground - connected to the MVME133XT ground plane. NOT connected to chassis ground by the MVME133XT.
8	DDCD	RS-232C Received Line Signal Detector - output that is always driven high by the MVME133XT. Note that this pin is connected to pin 6.
9-25	--	Not used.

5.3 PARTS LIST

The components of the MVME133XT are listed in Table 5-4. The parts locations are illustrated in Figure 5-1. These parts reflect the latest issue of hardware at the time of printing.

TABLE 5-4. MVME133XT Module Parts List

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
--	84-W8518B01B	Printed wiring board assembly, MVME133XT
C1-C10,C12-C14,C17-C24,C27-C29,C32,C33,C35-C45,C47-C60,C62,C63	21NW9632A03	Capacitor, ceramic, axial lead, 0.1 uF ± 20% @ 50 Vdc
C11	23NW9618A80	Capacitor, electrolytic, radial lead, 10 uF ± 20% @ 50 Vdc
C15,C16,C30,C31,C34,C46,C61	23NW9618A71	Capacitor, electrolytic, radial lead, 47 uF ± 20% @ 10 Vdc
C25,C26	21SW992C042	Capacitor, ceramic, 330 pF ± 5% @ 50 Vdc
CR1,CR2	48NW9616A03	Diode, silicon, 1N4148/1N914
CR3	48NW9607A20	Rectifier, schottky
DS1,DS2	48NW9612A49	LED, red, right angle
DS3,DS4	48NW9612A59	LED, green, right angle
E1,E2,J1-J11,J13-J22	29NW9805C07	Pin, 0.025 inch square, gold, autoinsert, used on E1(1), E2(1), J1(2), J2(2), J3(2), J4(2), J5(6), J6(2), J7(2), J8(6), J9(12), J10(6), J11(6), J13(21), J14(2), J15(12), J16(2), J17(3), J18(22), J19(2), J20(2), J21(10), J22(6)
--	29NW9805B17	Jumper, insulated, shorting (55 req'd) (used with J1-J11, J13-J22)
J12	--	Header pads on the printed wiring board
--	29NW9805B44	Jumper, 2-pin, male (used with J12)

TABLE 5-4. MVME133XT Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
J23	28NW9802G80	Connector, 25-pin, socket, right angle, D-subminiature
--	47NW9405A28	Jackpost assembly, D-subminiature (used with J23)
L1	76NW9810A04	Bead, ferrite, 0.146 inch x 0.126 inch (at P1-B1)
P1,P2	28NW9802E51	Connector, 96-pin, plug, PWB
--	05NW9007A26	Eyelet, 0.089 inch OD x 0.344 inch long (4 req'd) (used with P1 and P2)
R1,R3,R4,R23, R25	51NW9626B56	Resistor network, SIP, nine 10k ohm
R2	51NW9626B80	Resistor network, SIP, five 100 ohm
R5,R22	51NW9626B84	Resistor network, SIP, seven 2.7k ohm
R6	51NW9626B83	Resistor network, SIP, seven 1.0k ohm
R7	51NW9626B50	Resistor network, SIP, three 47 ohm
R8-R10	06SW-124A97	Resistor, film, 100k ohms, 5%, 1/4 W
R11	06SW-124A25	Resistor, film, 100 ohms, 5%, 1/4 W
R12,R16-R19, R21	51NW9626B64	Resistor network, SIP, four 47 ohm
R13	51NW9626B54	Resistor network, SIP, seven 39k ohm
R14,R15,R20	51NW9626B75	Resistor network, SIP, seven 10k ohm
R24	51NW9626A94	Resistor network, SIP, four 120 ohm (Refer to paragraph 2.4.3.)
--	09NW9811A90	Socket, I.C., SIL, 8-pin (Used with R24)
R26	51NW9626B55	Resistor network, SIP, nine 4.7k ohm

TABLE 5-4. MVME133XT Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
S1,S2	40NW9801B70	Switch, push, SPDT, momentary, PC, right-angle, gold
--	38NW9404C11	Cap, switch, black, for B70 (used with S1)
--	38NW9404C12	Cap, switch, red, for B70 (used with S2)
U1,U30	51NW9615F30	I.C. DM74S05N
U2	51NW9615K69	I.C. 74F10PC
U3,U12, U20,U31	--	Customer-supplied ROMs/PROMs/EPROMs/EEPROMs
XU3,XU12, XU20,XU31	09-W4659B14	Socket, I.C., SIL, 14-pin (8 req'd)
U4	(NOTE)	I.C. Programmed PAL
--	09NW9811B45	Socket, I.C., DIL, 24-pin, with capacitor 0.1 uF (alternate is part number 09NW9811B01) (used with U4, U7, U17, U33, U39, U65)
U5,U6,U8,U9, U38,U40,U42, U52,U92,U110	51NW9615T30	I.C. 74F623N
U7	(NOTE)	I.C. Programmed PAL
U10,U46	51NW9615K71	I.C. 74F04PC
U11,U50,U57	51NW9615K59	I.C. 74F175PC
U13,U14	51NW9615R55	I.C. N74F38N
U15	51NW9615U48	I.C. 74F367PC
U16	(NOTE)	I.C. Programmed PAL
--	09NW9811B18	Socket, I.C., DIL, 20-pin, with capacitor 0.1 uF (used with U16, U23, U27, U28, U32, U41, U43, U44, U45, U47, U54, U58, U69, U70)
U17	(NOTE)	I.C. Programmed PAL
U18,U64	51NW9615K73	I.C. 74F00PC

TABLE 5-4. MVME133XT Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
U19,U21,U25, U26,U29,U49, U63,U72,U73	51NW9615J39	I.C. 74F74PC
U22	51NW9615E93	I.C. SN74LS14N
U23	(NOTE)	I.C. Programmed PAL
U24	51NW9615P48	I.C. SN74ALS240AN
U27	(NOTE)	I.C. Programmed PAL
U28	(NOTE)	I.C. Programmed PAL
U32	(NOTE)	I.C. Programmed PAL
U33	(NOTE)	I.C. Programmed PAL
U34	51NW9615V50	I.C. IDT74FCT244AP (alternate is 74F1244N, part number 51NW9615T32)
U35	51NW9615U60	I.C. MK48T02B-25
--	09NW9811A91	Socket, I.C., SIL, 12-pin (2 req'd) (used with U35) (alternate is socket, I.C., DIL, 24-pin, part number 09NW9811A16)
U36	51NW9615V16	I.C. MC68020RC25E (on MVME133XT) (alternate is XC68020RC25E, part number 51NW9615V07)
--	09NW9811B42	Socket, I.C., pin-grid-array, 124-pin (used with U36) (alternate is part number 09NW9811B12)
U37	51NW9615W47	I.C. MC68882RC25 (on MVME133XT) (alternate is XC68882RC25, part number 51NW9615W68)
--	09NW9811B41	Socket, I.C., pin-grid-array, 68-pin (used with U37)
U39	(NOTE)	I.C. Programmed PAL
U41	(NOTE)	I.C. Programmed PAL
U43	(NOTE)	I.C. Programmed PAL

TABLE 5-4. MVME133XT Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
U44	(NOTE)	I.C. Programmed PAL
U45	(NOTE)	I.C. Programmed PAL
U47	(NOTE)	I.C. Programmed PAL
U48,U67	51NW9615K70	I.C. 74F08PC
U51	51NW9615U47	I.C. 74F113PC
U53,U109,U111	51NW9615S83	I.C. MC145406P
U54	(NOTE)	I.C. Programmed PAL
U55,U56	51NW9615K66	I.C. 74F32PC
U58	(NOTE)	I.C. Programmed PAL
U59-U61	51NW9615K60	I.C. 74F158APC
U62	51NW9615T24	I.C. MK68901N-05 (on MVME133XT)
--	09-W4659B24	Socket, I.C., SIL, 24-pin (2 req'd) (used with U62)
U65	(NOTE)	I.C. Programmed PAL
U66	51NW9615P21	I.C. MC74HC4040N
U68	51NW9615F38	I.C. SN74LS393N
U69	(NOTE)	I.C. Programmed PAL
U70	(NOTE)	I.C. Programmed PAL
U71	51NW9615N56	I.C. 74F174PC
U74-U89, U93-U108	51NW9615W60	I.C. M5M41000AL-8 (alternate is TC511000Z-85, part number 51NW9615W52; TC511001Z-85, part number 51NW9615W53; or TC511002Z-85, part number 51NW9615W54)
U90	51NW9615H38	I.C. SN75175N
U91	51NW9615H37	I.C. SN75174N
U112	51NW9615K09	I.C. SN74ALS244AN

TABLE 5-4. MVME133XT Module Parts List (cont'd)

REFERENCE DESIGNATION	MOTOROLA PART NUMBER	DESCRIPTION
U113	51NW9615R68	I.C. Z8530APC
--	09-W4659B20	Socket, I.C., SIL, 20-pin (2 req'd) (used with U114)
Y1	48AW1015B09	Crystal oscillator, 16 MHz ± 0.01%
Y2	48AW1015B25	Crystal oscillator, 50.0 MHz ± 0.01% (on MVME133XT)
--	67NW9415A17	Kit, ejector handle, 6U component
--	64-W5257B01	Panel, front, MVME134
--	33-W5590B58	Nameplate, Scanbe, MVME133XT
--	33-W5089B01	Nameplate, Scanbe, logo
--	42NW9401B14	Captive collar screw (2 req'd)
--	03NW9004B48	Screw, captive, M2.5 (2 req'd)

NOTE: When ordering, use number labeled on part.

5.4 SCHEMATIC DIAGRAM

The schematic diagram for the MVME133XT module is illustrated in Figure 5-2.

THIS PAGE INTENTIONALLY LEFT BLANK.

APPENDIX A - U33, U39, AND U65 PROGRAMMABLE ARRAY LOGIC PROGRAM DETAILS

INTRODUCTION

Pinouts and logic equations are given for each of these PALs. The following symbol conventions are used: / means NOT, * means AND, + means OR.

U33

The PAL chip U33 (PALSLV) on schematic sheet 14 is a 24-pin I.C. (a PAL20L8) used for address selection for extended addressing and for defining functions RTSA* and DTRA* of the RS-485 port (port A) of the Z8530 SCC chip.

The pinout for U33 is:

PIN NUMBER	SIGNAL MNEMONIC	PIN NUMBER	SIGNAL MNEMONIC
1	AM5 = I1	13	/MATCH0 = I13
2	/A24SLV = I2	14	/MATCH1 = I14
3	AM1 = I3	15	VDS = O15
4	AM3 = I4	16	/SLVTEST = OE = IO16
5	/DTRA = I5	17	/BUFFEN = IO17
6	/RTSA = I6	18	AM2 = IO18
7	AM0 = I7	19	MASTER = IO19
8	/IACK = I8	20	MASDREN = IO20
9	VDS1 = I9	21	SLVDREN = IO21
10	/MATCH2 = I10	22	/SLVADR = O22
11	VDS0 = I11	23	AM4 = I23
12	GND	24	VCC

Outputs for U33 all depend on OE being true (that is, pin 16 being high, that is, no external test (/SLVTEST) being performed). The output equations are:

```
IF(/SLVTEST)     /VDS = /VDS0 * /VDS1

IF(/SLVTEST)   SLVADR = MATCH0*MATCH1*MATCH2*     )Address match and
                        /AM5*/AM4*AM3*AM1*/AM0*  )ext'd address/program space
                        /IACK*/BUFFEN            )and not IACK and not self.

                      + MATCH0*MATCH1*MATCH2*     )Address match and
                        /AM5*/AM4*AM3*/AM1*AM0*  )ext'd address/data space
                        /IACK*/BUFFEN            )and not IACK and not self.

                      + A24SLV*MATCH2*           )Address match and
                        AM5*AM4*AM3*AM1*/AM0*    )st'd address/program space
                        /IACK*/BUFFEN            )and not IACK and not self.

                      + A24SLV*MATCH2*           )Address match and
                        AM5*AM4*AM3*/AM1*AM0*    )st'd address/data space
                        /IACK*/BUFFEN            )and not IACK and not self.
```

U33 (cont'd)

```
IF(/SLVTEST)  /MASTER = DTRA                  )DTRA* = 1 ==> master.

IF(/SLVTEST) /MASDREN = DTRA + /RTSA          )Not master or disabled by
                                              )RTSA.  RTSA*=0 ==> enable.

IF(/SLVTEST) /SLVDREN = /DTRA + /RTSA         )Not slave or disabled by
                                              )RTSA.  RTSA*=0 ==> enable.
---------------------------------------------------------------------------
```

U39

The PAL chip U39 (PALMAP) on schematic sheet 9 is a 24-pin I.C. (a PAL20L8) used for decoding the memory map addresses for the onboard resources of the MVME133XT. The MPU sees the onboard local DRAM as at physical address $00000000 through $003FFFFF, when U39 uses the default program that follows. (Refer to Chapters 2, 3, and 4 for details.)

The pinout for U39 is:

PIN NUMBER	SIGNAL MNEMONIC	PIN NUMBER	SIGNAL MNEMONIC
1	LA20 = I1	13	FC1 = I13
2	LA24 = I2	14	FC0 = I14
3	LA31 = I3	15	/SHIO = 015
4	LA28 = I4	16	/VMESEL = I016
5	LA27 = I5	17	/AUXSEL = I017
6	LA22 = I6	18	VMED16 = I018
7	LA30 = I7	19	/XXXF = I019
8	LA26 = I8	20	/VECT = I020
9	LA29 = I9	21	/RAMSEL = I021
10	LA25 = I10	22	/A24VME = 022
11	LA21 = I11	23	LA23 = I23
12	GND	24	VCC

```
RAMSEL = /LA31*/LA30*/LA29*/LA28*   )$00000000 to
         /LA27*/LA26*/LA25*/LA24*   )$003FFFFF,
         /LA23*/LA22*/VECT*FC1*/FC0 )program.

       + /LA31*/LA30*/LA29*/LA28*   )$00000000 to
         /LA27*/LA26*/LA25*/LA24*   )$003FFFFF,
         /LA23*/LA22*/VECT*/FC1*FC0 )data.
```

U39 (cont'd)

```
      AUXSEL = LA31*LA30*LA29*LA28*LA27*  )$FFF00000 to
               LA26*LA25*LA24*LA23*LA22*  )$FFFEFFFF,
               LA21*LA20*/XXXF*/VECT*     )program.
               FC1*/FC0

             + LA31*LA30*LA29*LA28*LA27*  )$FFF00000 to
               LA26*LA25*LA24*LA23*LA22*  )$FFFEFFFF,
               LA21*LA20*/XXXF*/VECT*     )data.
               /FC1*FC0

             + VECT                       )Reset vector.

        SHIO = LA31*LA30*LA29*LA28*LA27*  )$FFFF0000 to
               LA26*LA25*LA24*LA23*LA22*  )$FFFFFFFF.
               LA21*LA20*XXXF

      VMESEL = /RAMSEL*/AUXSEL*FC1*/FC0   )Program or
             + /RAMSEL*/AUXSEL*/FC1*FC0   )data.

      A24VME = /LA31*/LA30*/LA29*/LA28*   )$00000000 to
               /LA27*/LA26*/LA25*/LA24*   )$00EFFFFF.
               /LA23

             + /LA31*/LA30*/LA29*/LA28*
               /LA27*/LA26*/LA25*/LA24*
               /LA22

             + /LA31*/LA30*/LA29*/LA28*
               /LA27*/LA26*/LA25*/LA24*
               /LA21

             + /LA31*/LA30*/LA29*/LA28*
               /LA27*/LA26*/LA25*/LA24*
               /LA20

     /VMED16 = /LA31                      )$F0000000 to
             + /LA30                      )$FFFFFFFF are
             + /LA29                      )D16 locations.
             + /LA28
```

U65

The PAL chip U65 (PALADR) on schematic sheet 14 is a 24-pin I.C. (a PAL20L8) used for selecting one 256Mb block within the 4Gb address map range for the MVME133XT. The base address is $00000000 with the default program listed below. Header J15 then selects one of the 64 possible positions within this 256Mb block for the 4Mb of onboard shared DRAM. (Refer to Chapter 2.)

The pinout for U65 is:

PIN NUMBER	SIGNAL MNEMONIC	PIN NUMBER	SIGNAL MNEMONIC
1	AD1 = I1	13	VDS = I13
2	AD0 = I2	14	VA22 = I14
3	VA31 = I3	15	/MATCH0 = O15
4	VA30 = I4	16	/MATCH1 = IO16
5	VA29 = I5	17	/MATCH2 = IO17
6	VA28 = I6	18	/ADRTEST = OE = IO18
7	VA27 = I7	19	AD3 = IO19
8	VA26 = I8	20	AD4 = IO20
9	VA25 = I9	21	AD5 = IO21
10	VA24 = I10	22	DVDS = O22
11	VA23 = I11	23	AD2 = I23
12	GND	24	VCC

Outputs for U65 all depend on OE being true (that is, pin 18 being high, that is, no external test (/ADRTEST) being performed). The output equations are:

```
IF (/ADRTEST)  /DVDS = /VDS

IF (/ADRTEST) MATCH0 = /VA31*/VA30*/VA29*/VA28*   )Base address is $0XXXXXXX.
                       /VA27*/VA26*/AD5*/AD4      )Matching A27 and A26.

                     + /VA31*/VA30*/VA29*/VA28*   )Base address is $0XXXXXXX.
                       /VA27*VA26*/AD5*AD4        )Matching A27 and A26.

                     + /VA31*/VA30*/VA29*/VA28*   )Base address is $0XXXXXXX.
                       VA27*/VA26*AD5*/AD4        )Matching A27 and A26.

                     + /VA31*/VA30*/VA29*/VA28*   )Base address is $0XXXXXXX.
                       VA27*VA26*AD5*AD4          )Matching A27 and A26.

IF (/ADRTEST) MATCH1 = /VA25*/VA24*/AD3*/AD2      )Matching A25 and A24.
                     + /VA25*VA24*/AD3*AD2
                     + VA25*/VA24*AD3*/AD2
                     + VA25*VA24*AD3*AD2

IF (/ADRTEST) MATCH2 = /VA23*/VA22*/AD1*/AD0      )Matching A23 and A22.
                     + /VA23*VA22*/AD1*AD0
                     + VA23*/VA22*AD1*/AD0
                     + VA23*VA22*AD1*AD0
```

APPENDIX B - RS-232C INTERCONNECTIONS

The RS-232C standard is the most widely used interface between terminals and computers or modems, and yet it is not fully understood. This is because all the lines are not clearly defined, and many users do not see the need to conform for their applications. A system should easily connect to any other. Many times designers think only of their own equipment, but the state-of-the-art is computer-to-computer or computer-to-modem operation.

The RS-232C standard was originally developed by the Bell System to connect terminals via modems. Therefore, a number of handshaking lines were included. In many applications these are not needed, but since they permit diagnosis of problems, they are included in many applications.

Table 1 lists the standard RS-232C interconnections. To interpret this information correctly it is necessary to know that RS-232C is intended to connect a terminal to a modem. When computers are connected to computers without modems, one of them must be configured as a terminal and the other as a modem. Because computers are normally configured to work with terminals, they are said to be configured as a modem. Also, the signal levels must be between +3 and +15 volts for a high level, and between -3 and -15 volts for a low level. Any attempt to connect units in parallel may result in out of range voltages and is not allowed by the RS-232C specification.

TABLE 1. RS-232C Interconnections

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
1		Not used.
2	TXD	TRANSMIT DATA - Data to be transmitted is furnished on this line to the modem from the terminal.
3	RXD	RECEIVE DATA - Data that is demodulated from the receive line is presented to the terminal by the modem.
4	RTS	REQUEST TO SEND - RTS is supplied by the terminal to the modem when required to transmit a message. With RTS off, the modem carrier remains off. When RTS is turned on, the modem immediately turns on the carrier.

TABLE 1. RS-232C Interconnections (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
5	CTS	CLEAR TO SEND - Clear to send is a function supplied to the terminal by the modem, and indicates that it is permissible to begin transmission of a message. When using a modem, CTS follows the off-to-on transition of RTS after a time delay.
6	DSR	DATA SET READY - Data set ready is a function supplied by the modem to the terminal to indicate that the modem is ready to transmit data.
7	SIG-GND	SIGNAL GROUND - Common return line for all signals at the modem interface.
8	DCD	DATA CARRIER DETECT - Sent by the modem to the terminal to indicate that a valid carrier is being received.
9-14		Not used.
15	TXC	TRANSMIT CLOCK - This line clocks output data to the modem from the terminal.
16		Not used.
17	RXC	RECEIVE CLOCK - This line clocks input data from a terminal to a modem.
18,19		Not used.
20	DTR	DATA TERMINAL READY - A signal from the terminal to the modem indicating that the terminal is ready to send or receive data.
21		Not used.
22	RI	RING INDICATOR - RI is sent by the modem to the terminal. This line indicates to the terminal that an incoming call is present. The terminal causes the modem to answer the phone by carrying DTR true while RI is active.
23		Not used.
24	TXC	TRANSMIT CLOCK - Same as TXC on pin 15.

TABLE 1. RS-232C Interconnections (cont'd)

PIN NUMBER	SIGNAL MNEMONIC	SIGNAL NAME AND DESCRIPTION
25	BSY	BUSY - A positive EIA signal applied to this pin causes the modem to go off-hook and make the associated phone busy.

NOTES: 1. High level = +3 to +15 volts. Low level = -3 to -15 volts.

2. RS-232C is intended to connect a terminal to a modem. When computers are connected to computers without modems, one of the computers must be configured as a modem and the other as a terminal.

There are several levels of conformance that are appropriate for typical RS-232C interconnections. The bare minimum requirement is the two data lines and a ground. The full version of RS-232C requires 12 lines and accommodates automatic dialing, automatic answering, and synchronous transmission. A middle-of-the-road approach is illustrated in Figure 1.

One set of handshaking signals frequently implemented are RTS and CTS. CTS is used in many systems to inhibit transmission until the signal is high. In the modem application, RTS is turned around and returned as CTS after 150 microseconds. RTS is programmable in some systems to work with the older type 202 modem (half duplex). CTS is used in some systems to provide flow control to avoid buffer overflow. This is not possible if modems are used. It is usually necessary to make CTS high by connecting it to RTS or to some source of +12 volts such as the resistors shown in Figure 1. It is also frequently jumpered to an MC1488 gate that has its inputs grounded (the gate is provided for this purpose). Another signal used in many systems is DCD. The original purpose of this signal was to tell the system that the carrier tone from the distant modem was being received. This signal is frequently used by the software to display a message like CARRIER NOT PRESENT to help the user to diagnose failure to communicate. Obviously, if the system is designed properly to use this signal, and it is not connected to a modem, the signal must be provided by a pullup resistor or gate as described before (see Figure 1). Many modems expect a DTR high signal and issue a DSR. These signals are used by software to help prompt the operator for possible causes of trouble. The DTR signal is used sometimes to disconnect the phone circuit in preparation for another automatic call. It is necessary to provide these signals to talk to all possible modems (see Figure 1). As shown, Figure 1 is a good minimum configuration that almost always works. If the CTS and DCD signals are not received from the modem, the jumpers can be moved to provide the needed signal, artificially. Figure 2 shows a way that an RS-232C connector can be wired to enable a computer to connect to a basic terminal with only three wires. This is based on the fact that most terminals have a DTR signal that is ON, and that can be used to pullup the CTS, DCD, and DSR signals. Two of these connectors wired back-to-back can be used. It must be realized that all the handshaking has been bypassed and possible diagnostic messages do not occur. Also the TX and RX lines may have to be crossed since TX from a terminal is outgoing but the TX line on a modem is an incoming signal.

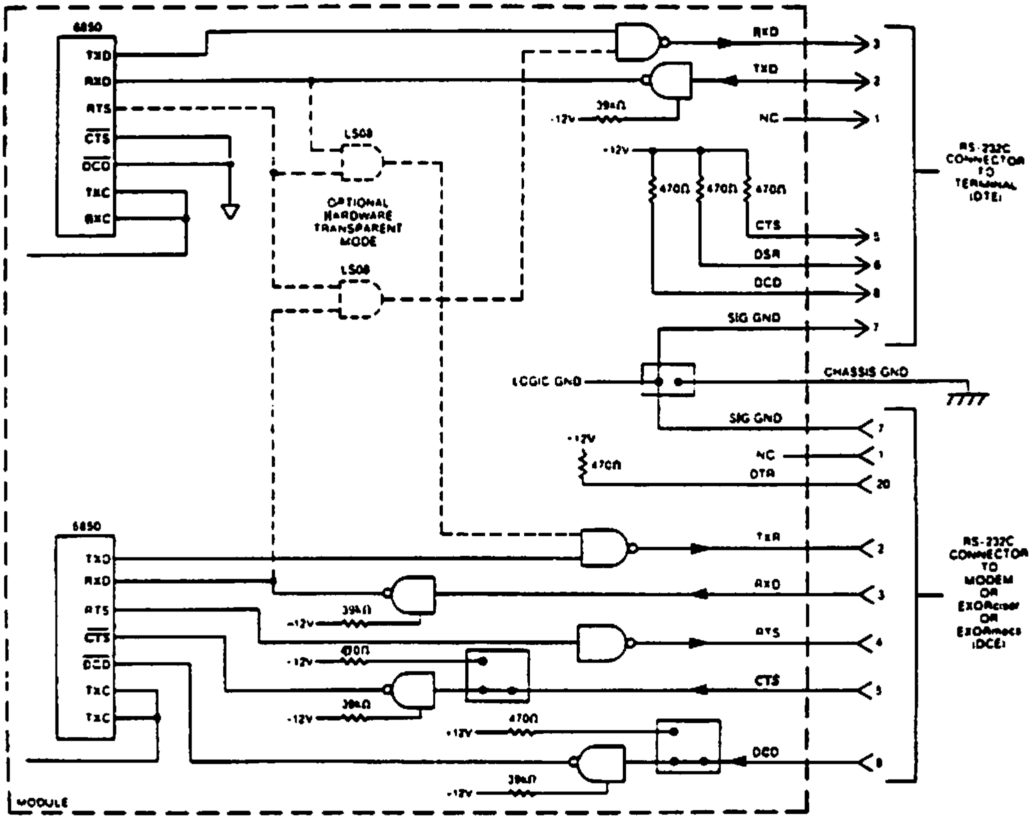


FIGURE 1. Middle-of-the-Road RS-232C Configuration

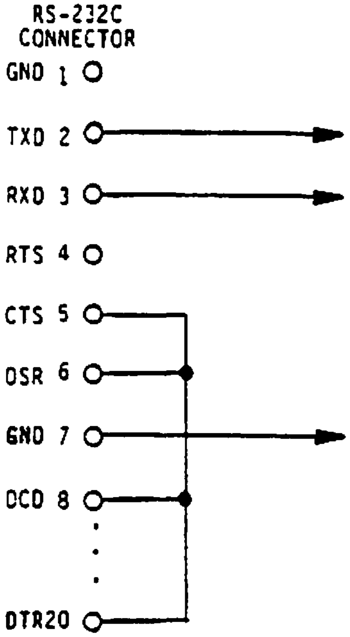


FIGURE 2. Minimum RS-232C Connection

Another subject that needs to be considered is the use of ground pins. There are two pins labeled GND. Pin 7 is the SIGNAL GROUND and must be connected to the distant device to complete the circuit. Pin 1 is the CHASSIS GROUND, but it must be used with care. The chassis is connected to the power ground through the green wire in the power cord and must be connected to the chassis to be in compliance with the electrical code. The problem is that when units are connected to different electrical outlets, there may be several volts difference in ground potential. If pin 1 of the devices are interconnected with a cable, several amperes of current could result. This not only may be dangerous for the small wires in a typical cable, but could result in electrical noise that could cause errors. That is the reason that Figure 1 shows no connection for pin 1. Normally, pin 7 should only be connected to the CHASSIS GROUND at one point, and if several terminals are used with one computer, the logical place for that point is at the computer. The terminals should not have a connection between the logic ground return and the chassis.

THIS PAGE INTENTIONALLY LEFT BLANK.

APPENDIX C - Z8530 SCC SERIAL PORT B SETUP EXAMPLE

This example sets up port B (the RS-232C port) of the Z8530 SCC as follows:

9600 baud, asynchronous only
Interrupt on Received Character, Transmitter Buffer Ready, and External Status Change with common interrupt vector.

SETUP:

Move #$30 into SCCB_WR0	($FFFA0000)	Clear receiver error status.
Move #$10 into SCCB_WR0	($FFFA0000)	Clear external status interrupts.
Move #$09 into SCCB_WR0	($FFFA0000)	Select register 9.
Move #$40 into SCCB_WR0	($FFFA0000)	Reset channel B.
Move #$0A into SCCB_WR0	($FFFA0000)	Select register 10.
Move #$00 into SCCB_WR0	($FFFA0000)	Make sure NRZ format is set.
Move #$0E into SCCB_WR0	($FFFA0000)	Select register 14.
Move #$82 into SCCB_WR0	($FFFA0000)	Disable baud rate generator.
Move #$04 into SCCB_WR0	($FFFA0000)	Select register 4.
Move #$44 into SCCB_WR0	($FFFA0000)	Divide by 16, no parity, one stop bit.
Move #$03 into SCCB_WR0	($FFFA0000)	Select register 3.
Move #$C1 into SCCB_WR0	($FFFA0000)	Receiver: eight bits, receiver enabled.
Move #$05 into SCCB_WR0	($FFFA0000)	Select register 5.
Move #$EA into SCCB_WR0	($FFFA0000)	Transmitter: eight bits, transmitter enabled, RTS on, DTR on.
Move #$0C into SCCB_WR0	($FFFA0000)	Select register 12.
Move #$02 into SCCB_WR0	($FFFA0000)	Lower byte of time constant.
Move #$0D into SCCB_WR0	($FFFA0000)	Select register 13.
Move #$00 into SCCB_WR0	($FFFA0000)	Higher byte of time constant.
Move #$0B into SCCB_WR0	($FFFA0000)	Select register 11.
Move #$56 into SCCB_WR0	($FFFA0000)	RX clock = BR Generator output, TX clock = BR Generator output, TRXC = output = BR Generator output.
Move #$0E into SCCB_WR0	($FFFA0000)	Select register 14.
Move #$81 into SCCB_WR0	($FFFA0000)	BR Generator clock source = RTXC pin.

```
Move #$01 into SCCB_WR0  ($FFFA0000)   Select register 1.
Move #$11 into SCCB_WR0  ($FFFA0000)   Interrupt on all Received
                                       Character or Special Condition.
                                       Also enable external interrupts.

Move #$0F into SCCB_WR0  ($FFFA0000)   Select register 15.
Move #$80 into SCCB_WR0  ($FFFA0000)   Enable Break/Abort interrupts.

Move #$02 into SCCB_WR0  ($FFFA0000)   Select register 2.
Move #$80 into SCCB_WR0  ($FFFA0000)   Interrupt vector number.
                                       ($80 => vector offset = $200.)

Move #$09 into SCCB_WR0  ($FFFA0000)   Select register 9.
Move #$08 into SCCB_WR0  ($FFFA0000)   Master interrupt enable. Status
                                       information NOT to be included in
                                       the vector passed to the MPU.
```

NOTE

To minimize overhead in the interrupt handling routine, status information may be selected to be included in the vector(s). The vector, then, points directly at the appropriate handling routine according to the interrupt cause. If the Vector-Status-Include (VSI) is set and the content in the vector register is $80, then the vector passed to the MPU is:

$80 (vector offset = $200) for Channel B Transmitter Buffer Empty or
$82 (vector offset = $208) for Channel B External Status Change or
$84 (vector offset = $210) for Channel B Received Character Available or
$86 (vector offset = $218) for Channel B Special Received Character.

For this example, place the address of the common interrupt handler at offset $200 in the vector table.

INTERRUPT HANDLER:

```
Move #$03 into SCCB_WR0  ($FFFA0000)   Select register 3.
Read from SCCB_RR0       ($FFFA0000)   Read the Read Register 3 for
                                       interrupt cause.
```

Investigate the interrupt pending bits to determine the cause.
Branch to the appropriate handling routine.

TRANSMIT A CHARACTER:

If Transmitter Buffer Empty interrupt is desired, it must be enabled before outputting a character or else the interrupt will not occur.

```
Move #$01 into SCCB_WR0  ($FFFA0000)   Select register 1.
Move #$13 into SCCB_WR0  ($FFFA0000)   Enable transmitter interrupt.

Move output character
into SCCB_TDR            ($FFFA0001)   Transmit a character.
```

TRANSMITTER BUFFER EMPTY INTERRUPT HANDLER:

```
    Move #$01 into SCCB_WR0  ($FFFA0000)   Select register 1.
    Move #$11 into SCCB_WR0  ($FFFA0000)   Disable transmitter interrupt.

    Move #$38 into SCCB_WR0  ($FFFA0000)   Reset   highest   Interrupt-Under-
                                           Service (IUS).

    Are there more characters to output?
         If Yes, go do TRANSMIT A CHARACTER.
         If No, return from exception.
```

RECEIVED CHARACTER INTERRUPT HANDLER:

```
    Move #$01 into SCCB_WR0  ($FFFA0000)   Select register 1.
    Read from SCCB_RR0       ($FFFA0000)   Read the Read Register 1 to  check
                                           for status.

    Check for framing error, receiver overrun, and parity errors.

    Read from SCCB_RDR       ($FFFA0001)   Read received character.
    Move #$38 into SCCB_WR0  ($FFFA0000)   Reset highest IUS.
```

EXTERNAL STATUS CHANGE INTERRUPT HANDLER:

```
         Break -- either start of break or end of break.
         CTS   -- a transition has occurred on the CTS input pin.
         DCD   -- a transition has occurred on the DCD input pin.

    Move #$00 into SCCB_WR0  ($FFFA0000)   Select register 0.
    Read from SCCB_RR0       ($FFFA0000)   Read  the  Read  Register  0   for
                                           status.

    Move #$10 into SCCB_WR0  ($FFFA0000)   Reset external status interrupt.

    Take actions as necessary.

         If break bit is low, which is the end of a break, a null character is
         still in the receive buffer.  It should be read and discarded.

    Read data from SCCB_RDR  ($FFFA0001)   Read null character.

    Return from exception.
```

THIS PAGE INTENTIONALLY LEFT BLANK.

APPENDIX D - MC68901 MFP TIMER A SETUP EXAMPLE

The following example sets up the MC68901 MFP timer A (software tick timer) to interrupt the MPU periodically every 10 msec.

SETUP:

Clear bit #5 of MFP_IERA	($FFF80007)	Disable timer A interrupts.
Move #$10 into MFP_TACR	($FFF80019)	Reset and stop timer A.
Move #$7B into MFP_TADR	($FFF8001F)	Load count down value. (Refer to Table 1 in this Appendix.)
Move #$06 into MFP_TACR	($FFF80019)	Delay mode, prescaler = 100.
Move #$68 into MFP_VR	($FFF80017)	Set starting vector at $60. Set software interrupt mode.

NOTE

The vector passed to the MPU for the timer A interrupt is $6D => vector offset = 4 x $6D = $1B4.

Move #$DF into MFP_IPRA	($FFF8000B)	Clear timer A interrupt pending bit (bit #5 of IPRA).
Move #$DF into MFP_ISRA	($FFF8000F)	Clear timer A interrupt-in-service bit (bit #5 of ISRA).
Set bit #5 of MFP_IMRA	($FFF80013)	Unmask timer A interrupts.
Set bit #5 of MFP_IERA	($FFF80007)	Enable timer A interrupts.

TIMER A INTERRUPT HANDLER:

Read MFP_ISRA	($FFF8000F)	Read interrupt-in-service register A.

Investigate MFP_ISRA to determine if it was actually from timer A.

Take actions as necessary.

Move #$DF into MFP_ISRA	($FFF8000F)	Clear timer A interrupt-in-service bit (bit #5 of ISRA).

Return from exception.

COUNTDOWN CALCULATION:

The countdown value used during setup may be calculated using the following equation:

$$CD = (TI \times TO) / PS$$

where: CD = countdown value to be loaded into timer data register.

TI = timer input frequency in Hertz = 1,230,769 Hertz.

TO = tick timer interrupts interval in seconds.

PS = prescaler value (4, 10, 16, 50, 64, 100, or 200).

Table 1 contains the values for PS and CD for some selected interrupts intervals.

TABLE 1. Prescaler and Countdown Values for Selected Interrupts Values

TO		PS	CD	
MSEC	SEC		HEX	DECIMAL
1.0	0.0010	10	$7B	123
5.0	0.0050	50	$7B	123
10.0	0.0100	100	$7B	123
20.0	0.0200	100	$F6	246
40.0	0.0400	200	$F6	246
41.6	0.0416	200	$00	256

APPENDIX E - BUS ERROR PROCESSING

1. INTERPRETATION OF BUS ERROR STATUS FLAGS

Because different conditions can cause bus error exceptions, the software must be able to distinguish the source. To aid in this, the MVME133XT provides two bus error status bits: LOCKVBE and LOCKLTO. Even though LOCKVBE is a low-true signal at the input to the MFP, the MFP IPRB will reflect that LOCKVBE has pulsed low with a 1 in the appropriate bit. The bus error handling routine can investigate these bits to determine the source of the bus error. Table 1 shows the interpretations of these two bus error flags.

TABLE 1. Interpretation of Bus Error Status Flags

LOCKLTO	LOCKVBE	DESCRIPTIONS
0	0	No-flag -- Unknown bus error.
0	1	VBE -- MPU was accessing the VMEbus and the cycle was terminated by the VMEbus with a bus error.
1	0	LTO -- MPU attempted to access a non-existent location and the cycle was terminated by the local bus time-out.
1	1	RMW-LOCK -- MPU was attempting to perform an RMW cycle to the VMEbus or a multiple-address RMW cycle to the onboard DRAM and the MVME133XT detected a bus lock condition.

2. BUS ERROR INTERRUPT

Normally, Table 1 is correct in determining the bus error source. However, there are conditions that create confusion in interpreting the bus error flags on the MVME133XT. These conditions that add complexity to the bus error handling are:

1. An interrupt may occur before the software has a chance to read the bus error status. This interrupt service routine may encounter another bus error. Now, the bus error flags may not reflect the source of the second bus error; therefore, this second bus error may be handled incorrectly. Also, when it is time for the first bus error to be serviced, the bus error information has been lost due to the handling of the first bus error. An example of this situation is as follows:

   a. A particular task encountered VBE. Before the bus error handler was entered, a tick-timer interrupt occurred.

   b. While in tick-timer interrupt service routine, an LTO was encountered.

   c. The bus error flags indicated an RMW-LOCK instead of LTO, causing the second bus error to be handled incorrectly.

   d. After the termination of the tick-timer interrupt service routine, the MPU returned to service the first bus error which was VBE. However, the flags were cleared when the second bus error was handled. The flags now indicated that there was "no" bus error source.

   Note that this condition may be much more complex by having many levels in depth. (Example: the first bus error was interrupted by an interrupt whose bus error was interrupted by a higher interrupt whose bus error was interrupted by an even higher interrupt.)

2. A bus error can occur while the MPU is prefetching an instruction that it does not use. An example of this may be:

```
  .
  .
  .
BNE         somewhere
  .                          The MPU prefetches at this address and encounters
  .                          VBE (due to parity error or end of memory.
  .
```

   If the MPU takes the branch, it does not go into exception handling for the bus error during prefetch. The bus error flags, however, remained set. Thus, the next time that bus error occurs to the MPU, the bus error flags contain incorrect information.

The MVME133XT provides a means to avoid both of the above problems with the Bus Error Interrupt. When the onboard logic detects that an MPU cycle was terminated with bus error, it generates a level 7 interrupt to the MPU with status ID of $FE. The purpose of this is to provide a way to raise the interrupt mask for the bus error handler so that it would be able to interrogate the bus error flags correctly without interruption. Because of case #2 (above), the interrupt mask should only be raised if the MPU is actually servicing the bus error. If the MPU is not servicing the bus error (as in case #2), then all the bus error flags should be cleared and the interrupt mask should not be changed. A suggested interrupt handler for vector $FE (vector offset of $3F8) is as follows:

```
BEINT     MOVEM.L   D0/A0,-(SP)          Save working registers.
          MOVEC     VBR,A0               Get Vector Base Register.
          MOVE.L    $8(A0),D0            Get address of BERR handler.
          CMP.L     $A(SP),D0            Compare with PC on stack.
          BEQ.B     CASE1                If PC is not pointing at the BERR
CASE2     BCLR.B    #1,($FFF8000D).L     handler, then clear LOCKVBE and
          BCLR.B    #2,($FFF8000D).L     LOCKLTO bus error flags.
          BRA.B     DONE                 Done.
CASE1     ORI.W     #$0700,$8(SP)        Raise inter. mask of stacked SR.
DONE      MOVEM.L   (SP)+,D0/A0          Restore registers.
          RTE                            End of bus error interrupt
                                         handler.
```

3. UNKNOWN BUS ERROR SOURCE

It is possible to have the no-flag case. Some of the possible causes for the no-flag case are:

a. Software enters into the bus error handler without a bus error actually having occurred.

b. Bus error interrupt is not used. Refer to bus error interrupt (paragraph 2 in this appendix) for more details.

c. All the bus error flags have been cleared previously (either accidentally or intentionally).

If the no-flag case is allowed to occur, then the bus error handler should simply RTE after making sure the rerun bit in the Special Status Word (SSW) is set when it detects the no-flag condition. Eventually, the faulted cycle is rerun and the flags are set again if the bus error was real.

4. FLOWCHART EXAMPLE FOR HANDLING BUS ERROR

Figure 1 is a flowchart example. The handling routine may be either more or less complex, depending on the system configuration.

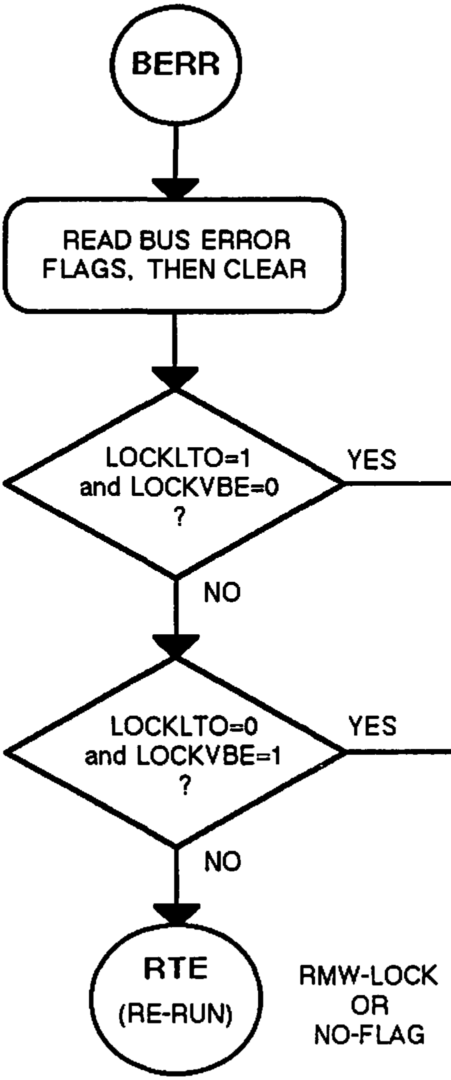

BERR
READ BUS ERROR FLAGS, THEN CLEAR
LOCKLTO=1 and LOCKVBE=0 ?
YES
LTO
NO
LOCKLTO=0 and LOCKVBE=1 ?
YES
VBE
NO
RTE (RE-RUN)
RMW-LOCK OR NO-FLAG

FIGURE 1. Bus Error Exception Handler Flowchart for MVME133XT

INDEX

INDEX