CHAPTER 1 - GENERAL INFORMATION

1•1   INTRODUCTION

This manual provides general information, preparation for use, installation and operating instructions, functional description, and support information for the M68302FADS Application Development System board.

1•2   GENERAL FEATURES

- A complete system development board for 3 processors. The MC68302, MC68LC302 and MC68PM302.
- Expansion connectors providing all 302 processor signals.
- Logic analyzer connectors similar to the expansion connectors.
- Single 5V supply.
- MC68302 core running at 25MHz.
- MC68LC302 core running at 20MHz.
- MC68302 core running at 20MHz.
- 512K byte, zero wait state static RAM, expandable up to 1M byte. (16 bit orientation)
- 1M byte FLASH. (16 bit orientation)
- 2K byte EEPROM. (8 bit orientation)
- Application Development Interface (ADI) port connector.
- MC68681 DUART, with wo RS232 serial ports.
- RESET and ABORT controls.
- RUN and HALT status indicators. (LED's).
- Supports PCMCIA card standard 2.01. (For the MC68PM302)
- Board power can be controlled from the PCMCIA connector of an host PC. (Important when the ADS is used for developing a PCMCIA applications)
- An extender board for plugging into PCMCIA port is supplied with the M68302FADS ADS board.
- Bus expansion connector pin out, is compatible with the 302ADS.

Freescale Semiconductor, Inc.

1•3 SPECIFICATIONS

The M68302FADS specifications and cooling requirements are given in Table 1-1.

Table 1-1 M68302FADS Specifications

CHARACTERISTICS	SPECIFICATIONS
Power requirements (no other boards attached)	+5Vdc @ 0.8 A (typical), 3.15 A (maximum)
Microprocessor	MC68302, MC68LC302, MC68PM302 - MC68302 @ 25 MHz - MC68LC302 @ 20 MHz - MC68PM302 @ 20 MHz
302 processor addressing Total address range. SRAM FLASH memory EEPROM	16M byte (4Mb external, for the LC302 and Pchip). 512K byte, 16 bit wide. (Expandable to 1M bytes) 1M byte, 16 bits wide. 2K byte, 8 bit wide.
Operating temperature	0 degrees to 30 degrees C ambient air temperature
Storage temperature	-25 degrees to 85 degrees C
Relative humidity	5% to 90% (non-condensing)
Dimensions Height Depth Thickness	9.196 inches (233.58 mm) 8.387 inches (213.03 mm) 0.063 inches (1.6 mm)

1•4 COOLING REQUIREMENTS

The M68302FADS is specified designed, and tested to operate reliably with ambient air temperature range from 0 to 70 degrees C. Dynamic Burn-in is performed while the board is table mounted with no other boards attached to it. Test software is executed as the board is subjected to temperature variations. If the board is attached to other boards, the thermal conditions may worsen and it is recommended that operating temperature should not exceed 30 degrees C.

1•5   GENERAL DESCRIPTION

The M68302FADS is a development tool for the MC68302, MC68LC302 and MC68PM302 devices. This board is used for hardware and software development of applications using any member of the 302 family. The M68302FADS has logic analyzer connectors as well as expansion connectors, providing physical connection to all processors' pins. The logic analyzer connectors, enable to monitor bus activity, by providing a direct connection to HP or other logic analyzers. The expansion connectors let the user, to attach hardware applications and to use board resources, to verify the design.

1•6   RELATED DOCUMENTATION

The following publications are applicable to the M68302FADS and may provide additional helpful information.

- MC68302 IMP User's Manual.
- MC68LC302 IMP User's Manual.
- MC68PM302 IMP User's Manual.
- PCMCIA specifications paper

1•7   ABBREVIATIONS USED IN THE DOCUMENT

- IMP - The MC68302 integrated communication processor.
- LC302 or LCIMP - A low cost version of the MC68302 integrated communication processor.
- PM302 or Pchip - MC68302 integrated with a PCMCIA card interface.
- 302 processor - Any of the IMP, LC302 and PM302.
- ADS - Application Development System for the 302 family processor.
- ADI - Application Development Interface.
- UART - Universal Asynchronous Receiver/Transmitter.
- spec - Engineering specification document.
- nsec - nano second.
- μsec - micro second.
- NMI - Non Maskable Interrupt.

1•8   REQUIRED EQUIPMENT
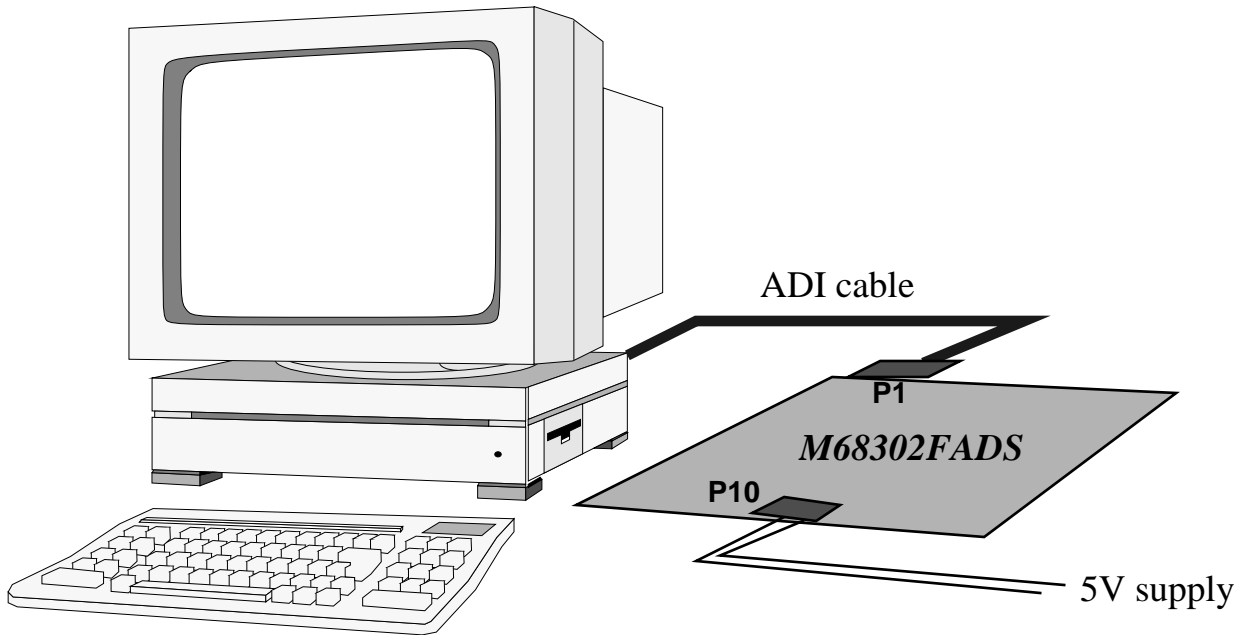
The M68302FADS can be operated in two working environments:

- Host controlled
- Stand-alone

Freescale Semiconductor, Inc.

FIGURE 1-1 describes the setup of an host controlled mode.

The required equipment in this mode is as follows:

- +5V/3A power supply.
- Host Computer, one of the following:
  - o Sun - 4 (Sbus interface)
  - o IBM-PC/XT/AT
- ADI board - compatible with the host computer.
- 37 line flat cable with female 37 pin D-type connectors on each end.

FIGURE 1-1 Host Controlled Configuration

ADI cable

P1

M68302FADS

P10

5V supply

1•8•1   Running the debugger

When the board is connected as shown in FIGURE 1-1, turn on the 5V power supply and run the host software, by typing:

host <ADI sbus slot No.> <ADS ADI address>

The manufacturer settings for ADS ADI address is 0. For more details on ADI addresses setting, please see section 2•3•2.

For example, to run the host on a sun machine with an ADI card installed in its Sbus slot number 1 and with ADS ADI address, set to 3, you should type, at command line prompt:

host 1 3

M68302FADS User's Manual
For More Information On This Product,
Go to: www.freescale.com

The prompt you get, depends on the processor installed, as listed in the following table

Table 1-2 Debugger prompt

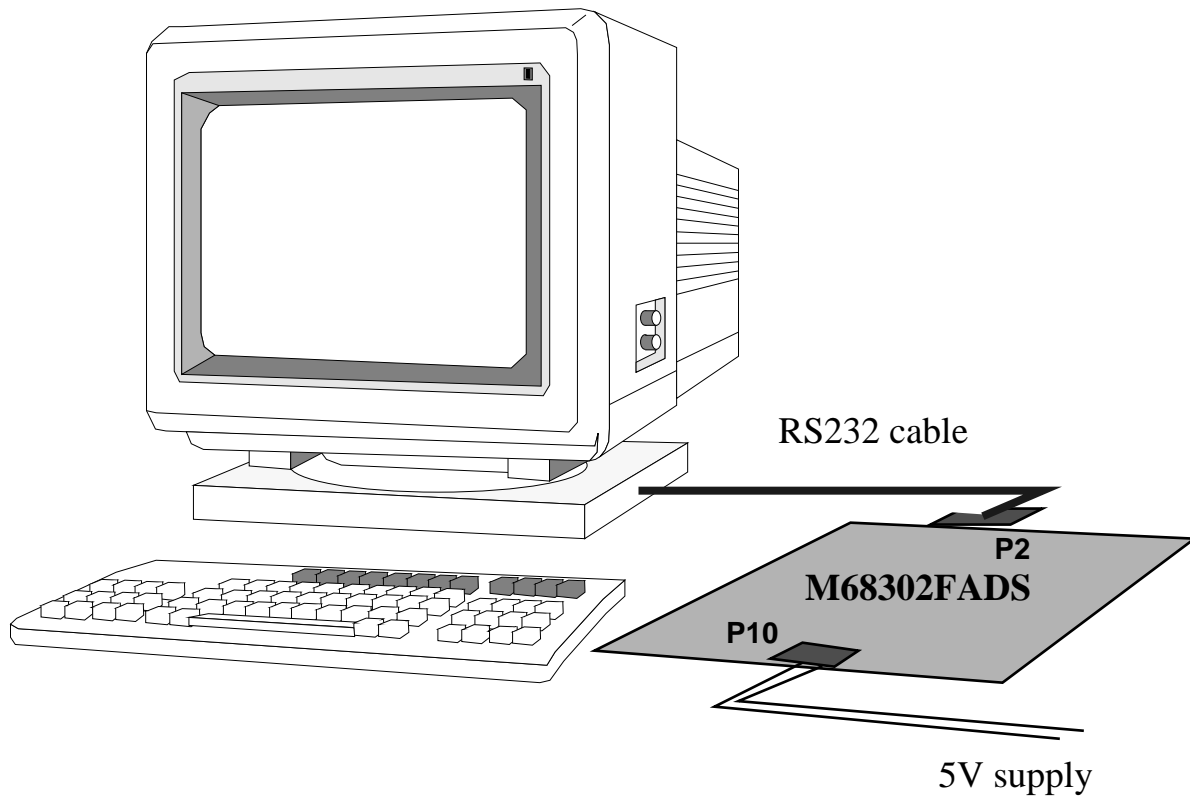| Processor	Prompt
MC68302	IMP Monitor/Debugger - Version 0.0 (C) Copyright 1995 by Motorola Inc. Cold Start IMPbug>
MC68LC302	LC302 Monitor/Debugger - Version 0.0 (C) Copyright 1995 by Motorola Inc. Cold Start LC302bug>
MC68PM302	PCHIP Monitor/Debugger - Version 0.0 (C) Copyright 1995 by Motorola Inc. Cold Start PCHIPbug>

Freescale Semiconductor, Inc.

1•8•2 Stand alone setup

FIGURE 1-2 describes the setup of a stand-alone operating mode.

The required equipment in this mode is:

- +5V/3A power supply.
- VT100 compatible terminal.
- RS-232 cable with male 9 pin D-type connector on the ADS side.

FIGURE 1-2   Stand-alone Configuration

1•8•2•1 Terminal settings

The terminal connected to a M68302FASDS board, should be initialized to work with the following settings:

- Speed - 9600 bps.
- Character size - 8 bits.
- Stop bit length - 1 bit. (1.5 or 2 is fine but 1 is preferred for being a bit faster)
- No parity.

 These settings are written shortly as - 9600, 8, 1 N.

After setting the terminal as specified and turning on the power, a debugger prompt as in Table 1-2, should appear on terminal screen.

1•8•2•2   Parallel ADI / Serial port, priority.

There is no control on board, to select between the parallel ADI or the serial port communication. The debugger decides automatically, with which it should communicate, by giving a priority to the serial port, in case, a terminal is detected. The serial port has some signals, by with the debugger can detect the presence of a terminal. If it is present, then the ADI connection is ignored and all board communication, goes to the serial port. When a terminal is not detected on the serial port, all communication, goes to the ADI port.

Freescale Semiconductor, Inc.

**For More Information On This Product,
Go to: www.freescale.com**

Freescale Semiconductor, Inc.

CHAPTER 2 - HARDWARE PREPERATION AND INSTALLATION

2•1 INTRODUCTION

This chapter provides unpacking instructions, hardware preparation, and installation instructions for the M68302FADS.

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 M68302FADS board, changes of the Dip-Switch/Jumpers settings may be required before installation. Switches, LEDs, Dip-Switches, and connectors are illustrated in FIGURE 1-3 M68302FADS Location diagram on page 11. The board has been factory tested and is shipped with Dip-Switch settings as described in the following paragraphs. Parameters can be changed for the following conditions:

- On board resources enable (SRAM, Flash, EEPROM, ADI port and DUART)
- Bus width.
- PLL ON/OFF.
- Processor type selection.
- ADI port address selection.
- Pchip PCMCIA enable.
- Pchip PCMCIA block power source.

FIGURE 1-3   M68302FADS Location diagram

2•3•1 DS1 DIP switch configuration.

FIGURE 1-4   Dip-Switch DS1

	OFF	DS1	ON	
1				PCEN
2				PCVCCEN
3				RAMEN
4				FLSHEN
5				EEPREN
6				ADIEN
7				DUARTEN
8				ADSA0
9				ADSA1
10				ADSA2

Table 1-3 describes the function of switches 1 - 7.

Table 1-3 Dip-Switch DS1 Description

Switch	Name	Function	Default setting
1	PCEN	Direct control over the Pchip signal PCEN. When set to On, the internal Pchip PCMCIA interface is disabled.	ON
2	PCVCCEN	When set to ON, the board is powered unconditionally, with respect to PCMCIA host power. When set to OFF, the board is powered, only when the host connected to the board through the PCMCIA port, is turned on.	ON
3	RAMEN	When set to ON, on board SRAM is enabled. (1Mbyte space)	ON
4	FLSHEN	When set to ON, on board FLASH memory is enabled.	ON
5	EEPREN	When set to ON, on board EEPROM is enabled.	ON
6	ADIEN	When set to ON, ADI interface is enabled.	ON
7	DUARTEN	When set to ON, DUART MC68681 along with its RS232 port are enabled.	ON

2•3•2 ADI Port Address Selection (Dip-Switch DS1 switches 8-10)

Each M68302FADS can have eight possible ADI port addresses, enabling up to eight M68302FADS boards to be connected to the same ADI card in a host computer. Address selection is done by setting switches 8, 9, 10 in DS1 Dip-Switch. Switch 10 is the most significant bit of the address while switch 8 is the least. A switch set to its 'ON' state, stands for logical '0'. The default setting is ADI address - 7.

Table 2-1 describes the switch settings for each slave address:

Table 2-1 ADI Address Selection

ADDRESS	Switch 10	Switch 9	Switch 8
0	ON	ON	ON
1	ON	ON	OFF
2	ON	OFF	ON
3	ON	OFF	OFF
4	OFF	ON	ON
5	OFF	ON	OFF
6	OFF	OFF	ON
7	OFF	OFF	OFF

2•3•3   DS2 DIP switch configuration (for LC302/PM302).

FIGURE 1-5   Dip-Switch DS2

Table 2-2 Dip-Switch DS2 description.

Switch	Name	Function	Default
1	IOIS16	Set IOIS16 PCMCIA signal level. IOIS16, is not supported by the Pchip. When set to on, PCMCIA I/O transfers are 8 bits width. When set to off, PCMCIA I/O transfers, are 16 bits width.	ON
2	MODCLK	Select PLL multiplication factor when PLL is enabled. (See J1, J2 jumpers) ON - selects x4 multiplication factor and OFF - selects x401 for 32KHz crystal.	ON
3	BUSW	Controls BUSW input of the processor. Selects 16 bit bus when OFF and 8 bit bus width when set to ON.	OFF
10	CHIPTYPE	Selects between IMP and LC302/Pchip mode. ON - IMP, OFF - LC302 and Pchip.	ON = IMP OFF = LC302/PM302 (*)

* FIGURE 1-5, shows default settings for the LC302 and PM302. When the IMP is plugged in, switch 10 should be set to ON. All other switches are the same as for the LC302/PM302.

2•4   INSTALLATION INSTRUCTIONS

When the M68302FADS has been configured as desired by the user, it can be installed according to the required working environment as follows:

2•4•1   +5V Power Supply Connection

The M68302FADS requires +5Vdc @ 1A max, power supply for operation. The ADS has 3.15A fuses on the +5V line, and it is protected against reverse connection of the power supply. Connect the +5V power supply to connector P10 as shown below:

FIGURE 1-6 P18: +5V Power Connector

P10 is a 3 terminal block power connector with power plug. The plug is designed to accept 14 to 22 AWG wires. It is recommended to use 14 to 18 AWG wires. To insure solid ground, two Gnd terminals are supplied. It is recommended to connect both Gnd wires to the common of the power supply, while VCC is connected with a single wire.

NOTE

Since hardware applications can be connected to the M68302FADS using the expansion connectors P9 and P12, the additional power consumption should be taken into consideration when a power supply is connected to the ADS.

Freescale Semiconductor, Inc.

2•4•2 ADI Installation

For ADI installation on various host computers, refer to APPENDIX A - ADI BOARD INSTALLATION on page 50.

2•4•3 Host computer to ADS connection

The M68302FADS ADI interface connector, P1, is a 37 pin, male, D type connector. The connection between the M68302FADS and the host computer is made by a 37 line flat cable, supplied with the ADI board. FIGURE 1-7 below shows the connector pin arrangement of the connector.

FIGURE 1-7 ADI Port Connector P1

Gnd	20	1	INT_ACK
Gnd	21	2	N.C.
Gnd	22	3	HST_ACK
Gnd	23	4	ADS_ALL
Gnd	24	5	ADS_RESET
Gnd	25	6	ADS_SEL2
(+ 12 v) N.C.	26	7	ADS_SEL1
HOST_VCC	27	8	ADS_SEL0
HOST_VCC	28	9	HOST_REQ
HOST_VCC	29	10	ADS_REQ
HOST_ENABLE~	30	11	ADS_ACK
Gnd	31	12	ADS_INT
Gnd	32	13	HOST_BRK~
Gnd	33	14	ADS_BRK
PD0	34	15	N.C.
PD2	35	16	PD1
PD4	36	17	PD3
PD6	37	18	PD5
		19	PD7

NOTE: Pin 26 on the ADI is connected to +12 v power supply, but it is not used in the ADS.

2•4•4 Terminal to M68302FADS RS-232 Connection. (DCE port)

In a stand-alone operation mode, a VT100 compatible terminal should be connected to the RS-232 connector P2. The RS-232 connector is a 9 pin, female, D-type connector as shown in FIGURE 1-8.

FIGURE 1-8 RS-232 Serial Port Connector P2

CD	1		
TX	2	6	DSR
RX	3	7	RTS
DTR	4	8	CTS
GND	5	9	N.C.

NOTE: The RTS line (pin 7) is not connected in the M68302FADS.

2•4•5 Secondary RS-232 port. (DTE port)

The second RS232 port, is controlled, is not supported by the debugger. The user should write his own drivers to use this port. The connector pin out is shown in FIGURE 1-8.

<u>**FIGURE 1-9 RS-232 Serial Port Connector P3**</u>
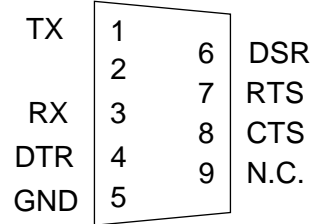
```
TX    1
      2      6   DSR
             7   RTS
RX    3
             8   CTS
DTR   4
             9   N.C.
GND   5
```

NOTE: The CTS line (pin 8) is not connected in the M68302FADS.

Freescale Semiconductor, Inc.

CHAPTER 3 - OPERATING INSTRUCTIONS

3•1   INTRODUCTION

This chapter provides necessary information to use the M68302FADS in host-controlled and in a stand-alone configuration. This includes controls, indicators, memory map details, and software initialization of the board.

3•2   CONTROLS AND INDICATORS

The M68302FADS has the following switches and indicators.

3•2•1   NMI (Abort) Switch SW1

The NMI switch SW2 asserts level 7 interrupt to the processor. The switch signal is debounced, and It is not possible to disable it by software. The NMI (Abort) switch is normally used to abort program execution and return the debugger control.

3•2•2   RESET Switch SW2

The RESET switch SW2, resets all ADS devices and performs reset to the 302 processor. The switch signal is debounced, and it is not possible to disable it by software.

3•2•3   LC302 PLL enable jumper - J1.

The PLL of the LC302, is controlled by applying or preventing PLL block power, on the VCCSYN pin. The selection, is done with J1 jumper, as shown in FIGURE 1-10.

FIGURE 1-10 LC302 PLL enable jumper (J1)

LC302 PLL - Disabled

When the jumper is installed on pins 2 - 3, LC302 PLL is disabled. This actually shorts the LC302 VCCSYN pin, to ground.

LC302 PLL enabled

When the jumper is installed on pins 1 - 2, the LC302 PLL is enabled. This actually connects VCCSYN pin of the LC302 to VCC.

M68302FADS User's Manual

3•2•4   Pchip PLL enable jumper - J2.

The PLL of the Pchip, is controlled by applying or preventing PLL block power, on the VCCSYN pin. The selection, is done with J2 jumper, as shown in FIGURE 1-10.

FIGURE 1-11 Pchip PLL enable jumper (J2)

Pchip PLL - Disabled

When the jumper is installed on pins 2 - 3, the Pchip PLL is disabled. This actually shorts the Pchip VCCSYN pin, to ground.

Pchip PLL
ON   OFF

Pchip PLL enabled

When the jumper is installed between pins 1 - 2, the Pchip PLL is enabled. This actually connects VCCSYN pin of the Pchip to VCC.

Pchip PLL
ON   OFF

3•2•5   PCMCIA power source selection - J3.

The PCMCIA block in the Pchip, can be powered by one of two sources. The first is the board common VCC supply and the second is an external power, supplied through a PC host, plugged into the PCMCIA connector. The selection, is done with J3 jumper, as shown in FIGURE 1-12.

FIGURE 1-12 Pchip PLL enable jumper (J3)

Pchip PCMCIA block is powered by a host PC

When the jumper is installed on pins 2 - 3, the Pchip PCMCIA block, get its power from a host PC, connected to PCMCIA connector - P4.

PCMCIA PWR
INT   PC

Pchip PCMCIA block is powered by board supply board

When the jumper is installed on pins 1 - 2, the Pchip PCMCIA block, get its power from the board common supply.

PCMCIA PWR
INT   PC

For More Information On This Product,
Go to: www.freescale.com

Freescale Semiconductor, Inc.

3•2•6   RUN indicator LD1

This yellow indicator is connected to the 302 processor address strobe (AS~) signal. It lit when AS~ is low (asserted) to indicate a bus activity.

3•2•7   HALT indicator LD2

This red indicator, lit whenever the 302 processor HALT~ pin is low (asserted).

3•2•8   PCOFF indicator LD3

This orange indicator, lit when PCMCIA power control is selected by the DIP-Switch DS1 switch 2 (PCVCCON set to the OFF position) and no external power is detected on the PCMCIA connector - P4. This is a reminder to the user, that the board is not powered, despite power existence, on P10 connector. The board would be powered, upon detection of a turned on PC, on the PCMCIA connector - P4.

3•2•9   Power indicator LD4

This green indicator, lit when the board is powered. Please be aware that suppling 5V to P10 connector does not assure it is powered. When DIP switch DS2 switch number 2, is set to its off position, board power, is PCMCIA power dependent. The on board relay switches the power off, unless a 5V power is detected on the PCMCIA port. When DIP switch DS2 switch number 2, is set to its on position, the board is powered when 5V is supplied trough P10, unconditionally of PCMCIA port status.

3•3   IMP MEMORY MAP

FIGURE 1-13 IMP memory map on page 21, is one of two maps. After a cold or warm reset, the 302 processor's CS0, is automatically enabled for the first block of 8K bytes. In this page the 302 processor sees the first 8K bytes of the 1M bytes FLASH memory installed. No other space than those 8K, is available this time. At this point, the stack pointer and the program counter are fetched from Flash and the debugger starts. One of the debugger first tasks, is to enable the map shown in FIGURE 1-13. It is done by copying the code for the swap task, into the 302 processor's dual port Ram and jump to execute the code. The code redefines CS0, CS1 and CS2 to enable the map shown in FIGURE 1-13 and goes back to the debugger which has been moved due to the new map, to offset 20000H. The swap task is executed from the dual port RAM, so it would not be interrupted by the swap. The swap is done, in order to map a RAM at the 302 processor vector table (start at $0), so the user would be able to change those vectors. (and not a FLASH at $0 - a must after reset)

FIGURE 1-13 IMP memory map

D15	D8	D7	D0	
FFFFFE				FFFFFF
	Reserved for MC68302 internal space and user applications.			
700000				700001
6FFFFE				6FFFFF
				630001
		MC68681 DUART		620001
		2K byte EEPROM		610001
610000	ADI handshake	ADI port data		600001
600000				5FFFFF
5FFFFE				510001
500000				500001
4FFFFE				4FFFFF
				410001
400000				400001
3FFFFE				3FFFFF
300000				300001
2FFFFE	1M byte FLASH memory*			2FFFFF
200000				200001
1FFFFE				1FFFFF
100000				100001
0FFFFE	512K byte static RAM expansion memory			0FFFFF
080000				080001
07FFFE	512K byte static RAM base memory*			07FFFF
000000				000001

```
*Note: The debugger uses 0-7FFF in the Static RAM
       and from 200000 to 21FFFF in the Dynamic RAM.
```

Freescale Semiconductor, Inc.

Each board comes with 1M byte of Flash memory (oriented as 512K words of 16 bits each) and a 512K bytes of static RAM (oriented as 256K words 16 bits each). The RAM can be expanded, on board to a total of 1M bytes. ADI logic, DUART MC68681 and 2K byte EEPROM, occupies each 64K byte which is the minimum division available, by on board chip select logic. Each empty space in the map, is considered available for the user. Any of the on board resources, can be disabled, to free its memory allocation for the user. Please see Table 1-3 Dip-Switch DS1 Description on page 12.

3•3•1   DUART MC68681 registers.

Table 2-3, shows the MC68681 internal registers. This is an eight bit device, connected to D0-D7 and thus located on odd addresses.

Table 2-3 MC68681 internal registers

Register address	Read (R/W~ = 1)	Write (R/W~ = 0)
$620001	Mode register A (MR1A, MR2A)	Mode register A (MR1A, MR2A)
$620003	Status register A (SRA)	Clock select register A (CSRA)
$620005	Do not access *	Command register A (CRA)
$620007	Receiver buffer A (RBA)	Transmitter buffer A (TBA)
$620009	Input port change register (IPCR)	Auxiliary control register (ACR)
$62000B	Interrupt status register (ISR)	Interrupt mask register (IMR)
$62000D	Counter mode: Current MSB of counter (CUR)	Counter/Timer upper register (CTUR)
$62000F	Counter mode: Current LSB of counter (CLR)	Counter/Timer lower register (CTLR)
$620011	Mode register B (MR1B, MR2B)	Mode register B (MR1B, MR2B)
$620013	Status register B (SRB)	Clock select register B (CSRB)
$620015	Do not access *	Command register B (CRB)
$620017	Receiver buffer B (RBB)	Transmitter buffer B (TBB)
$620019	Interrupt vector register (IVR)	Interrupt vector register (IVR)
$62001B	Input port (unlatched)	Output port configuration register (OPCR)
$62001D	Start counter command **	Output port register (OPR) bit set command
$62001F	Stop counter command **	Output port register (OPR) bit reset command

*   This address is used for factory testing and should not be accessed by the user.

** A trigger command.

For more details about the MC68681 DUART, please refer to M68000 family reference.

3•3•2   ADI handshake signals

ADI handshake signals consist of in and out, going signals. In going signals can be read from the MC68681 DUART input port found in address $62001B or from the ADI input port located at address $600000. The second is preferred due to the user ability to map the DUART off. The outgoing signals of the ADI interface are found in the DUART output port latch. In order to let the ADI port, be functional, when the DUART is disabled, all DUART register involved with the ADI outgoing handshake signals, were logically duplicated into the ADI space. So now, the outgoing signals, can be accessed trough two base addresses. The DUART as well as the ADI. Of course the second is preferred from the same reason explained above.

The following tables shows relevant ADI registers as well as the internal bit map of each register.

Table 2-4 ADI registers.

Address	Read (R/W~ = 1)	Write (R/W~ = 0)
$600000	ADI input port	Do not access
$60001B	DUART input port	DUART output port configuration register (OPCR)
$60001D	Do not access	DUART output port (OPR) bit set command
$60001F	Do not access	DUART output port (OPR) bit reset command

To set a DUART output bit, a "1", should be written to the desired bit of the DUART OPR reset command register. To reset a DUART output bit, a "1", should be written into the DUART OPR set command register.

Table 2-5 DUART output port bit set/reset map @$60001D/@$60001F

7	6	5	4	3	2	1	0
ADS ACK	CTS, DSR & CD to the terminal.	N.C.	EEPROM wr enable	ADS INT	ADS REQ	Not used	Int mode

Table 2-6 DUART input port bit map @$60001B

5	4	3	2	1	0
DTR from terminal.	Read as'0'	INT ACK	HST_EN~	HST REQ	HST ACK

Table 2-7 ADI input port bit map @600000

7	6	5	4	3	2	1	0
Read as'0'	Read as'0'	Read as'0'	INT ACK	HST ACK	HST REQ	ADS BRK	ADS_SEL~

ADS_SEL~ signal, is not a member in the ADI protocol, but is a signal, generated on board, to report, the ADS, is selected by the host. For more details on the ADI protocol, please refer to APPENDIX B - ADI PORT HANDSHAKE DESCRIPTION on page 55.

Freescale Semiconductor, Inc.

3•4 The IMP debugger.

Each M68302FADS board, is installed with a debugger, resides on the first two sectors of FLASH memories, U6, U7. Each sector of AM29F040 device, is 64K bytes, thus a total of 128K bytes (1 sectors per FLASH), are allocated for the debugger. Those two sectors are protected by software against, accidental programming.

3•4•1 Debugger upgrade.

The debugger supports an on board upgrade capabilities. A new debugger release, is programmed on board, without the need to take the FLASHes out of their sockets and without the need of a special programmer. Further instruction will be supplied, with each upgrade.

3•4•2 The debugger programming, of internal 302 processor registers.

The initialization done by the debugger to an internal 302 family processor registers, is writing to the base address register (BAR) and programming CS0, CS1 and CS2.

- The BAR is initialized, so the processor register block, resides starting from $700000.
- CS0 is initilized for 1M bytes FLASH starting at 200000H.
- CS1 is initilized for 1M bytes RAM (512K bytes acttualy installed) starting at 000000H.
- CS2 is initilized for 30000H bytes starting at 600000H. CS3 is used for the ADI, EEPROM and DUART. The internal division of this space, is done by an additional logic, outside the302 processor.

3•5   Functional description

FIGURE 1-14 HARDWARE BLOCK DIAGRAM

* Only one of the three
  processors, can be used
  at a time.

3•5•1   MC68LC302 - LCIMP. (Sheet 1)

The LC302 is a low cost version of the original MC68302 - IMP, with an additional on chip PLL.The LC302, lacks a third SCC and has less pins in total. Pins that were taken off, includes the 4 upper address lines (A20-A23), function code (F0-F2), arbitration logic (BR~, BG~, BGACK) and more. For more details please refer to the MC68LC302 user manual. There are 2 packages available for the LC302. An 100 pin TQFP and 132 pin PGA package. The first has less functional pins and it is the low cost package you intended to put in your application. The PGA package is intended primary for development tools and is used in the M68302FADS. The pins added in the PGA package only are - FC0 - FC2, FRZ~, AVEC~ and IAC.

3•5•2   MC68302 - IMP. (Sheet 2)

The IMP is the first chip of the 302 family. It is an integrated multi-protocol processor, using the well known 68000 core.

3•5•3   MC68PM302 - Pchip. (Sheet 3)

The Pchip is integration of the original MC68302 - IMP, with a PCMCIA card logic.As the LC302, it lacks a third SCC and has less pins in total. Pins that were taken off, includes the 4 upper address lines (A20-A23), function code (F0-F2), arbitration logic (BR~, BG~, BGACK) and more. For more details please refer to the MC68PM302 user manual. There are 2 packages available for the Pchip. An 130 pin TQFP and 181 pin PGA package. The first has less functional pins and it is the low cost package you intended to put in your application. The PGA package is intended primary for development tools. The pins added in the PGA package only are - FC0 - FC2, FRZ~, AVEC~ and IAC.

3•5•4   The on board processors, in general

The pins of all 3 processors, are available unbuffered to the user, through the expansion and the logic analyzer connectors. The user can monitor the processor activity and connects the ADS to his own hardware application during its development stage.All processors' pins, are buffered from all devices on the ADS to minimize their load and to enable easy connection to the user hardware application.

3•5•5   Processors' clock (sheet 4)

The system clock can be generated from one of 3 sources. On board clock oscillator, on board crystal circuit and an external clock source. The ADS is shipped with the first option. It consist of a clock oscillator running at twice the processor speed, followed by a divide by two flip flop. The divider assure a duty cycle of almost 50%. The clock oscillator device is mounted on a socket and can be easily replaced to get a different clock speed as requested. In case a crystal circuit or an external clock option is desired, please connect your Motorola dealer.

3•5•6   Indicators and controls (sheet 5)

The ADS reset and abort, control buttons, each equipped with a debounce circuit to prevent oscillations. The board has run, halt power and pc power indicators.

3•5•7   Control logic (sheet 6)

The IMP control logic is programmed into a MACH220 device. It consist of the following units:

- Chip select logic -

  Provides chip select for SRAM, FLASH, EEPROM, ADI, DUART. The chip selects are generated from the IMP CS0, CS1 and CS2. In the first two, the MACH reflect the signal as is. In the case of CS2, the MACH divide its range into three sections, 64K bytes each, for the ADI, EEPROM and the ADI.

- DTACK generator -

  DTACK is returned for all devices, accessed by the chip select logic, except the DUART, which

  has its own DTACK generator and the FLASH which has its CS0 default, for supplying an internal DTACK.

- Interrupt controller -

    There are 3 level 7 (NMI) interrupt sources, on board. Abort switch, ADI abort and an external interrupt line. The interrupt controller, assert a level 7 interrupt with auto vector line and supports both IMP interrupt modes.Those modes are normal and dedicated. The interrupt controller, is reported with DUART output port bit 0, the interrupt operating mode. The debugger initializes the IMP with normal mode.

- resources enable -

    Resources like SRAM, FLASH, etc. can be taken out of the memory map (disabled) by this logic, by a user Dip switches selection.

- wait state logic -

    The wait state logic, delays a DTACK generation by several system clock out pulses, as programed into the MACH device for each resource.

- Chip type selection -

    The bus interface for the LC302 and the Pchip, is different from the original IMP. The first two has a glue less design concept and interface to external devices, through AS~, OE~, WEL~, WEH~ signal. The IMP, on contrary, uses AS~, DS~ and R/W~ signals. The MACH is used to interface those signals, according to the processor installed on board. (A selection between LC302/Pchip or IMP, is done by switch 10 of DIP switch set - DS2)

- buffers control -

    This logic control data buffers in 16 and 8 bits bus width.

3•5•8   ADI control logic (sheet 7)

A MACH100 device, controls the ADI interface. The ADI interface appears on sheet 10. (See APPENDIX B - ADI PORT HANDSHAKE DESCRIPTION, for more details on the ADI interface.

3•5•9   Buffers (sheet 8)

To minimize signal load on the 302 processors pins, all address and data lines are buffered. The buffers support 8 bit bus width as well.

3•5•10   SRAM (sheet 9)

The ADS is supplied with 512 Kilobyte of SRAM, which is implemented by four MCM6229A devices. The devices are organized as word (16 bits wide). The SRAM access time is 35 nsec.

Another bank of four MCM6229A devices can be soldered on the ADS to increase the SRAM memory space up to a 1 Mbyte.

3•5•11   Flash Memory (sheet 9)

Two 512 Kilobyte Flash Memory devices, the 29F040, are used to form 1 Mbyte program storage memory. The devices are organized as word (16 bits wide). The Flash Memory access time is 90 nsec.

The Flash Memory devices are mounted on sockets, enable the user to load code, out of the board.

3•5•12   EEPROM. (sheet 9)

The EEPROM used in the ADS is the AT28C16 device (250 nano-seconds access time, 2K x 8 bit). It appears as a static RAM for read or write cycle, with a write cycle extended up to 1 millisecond. During an internal write cycle, the 68302 must not access the EEPROM, but it may access any other device or performs a data polling on the EEPROM to detect end of a write cycle. The data polling feature of the EEPROM, enables the 68302 to detect the end of an internal write cycle. During an internal write cycle, the 302 processor, may read the last byte written to the EEPROM. If the internal write cycle is in process, then the result is the complement of the written data on bit 7, else the result is the true data. Only data pulling access, will return correct results from the EEPROM, during an internal write cycle. The AT28C16 has

internal hardware protection, against inadvertent writes to the EEPROM, that might happen during power up or power down time.

3•5•13 ADI port (sheet 10)

The ADI interface uses LS TTL drivers to communicate with the host. The ADI parallel port supplies parallel link from the ADS to various host computers. This port is connected via a 37 line cable to a special board called ADI (Application Development Interface) installed in the host computer. It is possible to connect the ADS board to an IBM-PC/XT/AT or to SUN-4 SPARC station, provided that they have an ADI board with the appropriate software drivers installed on them. Each ADS, has 8 possible slave addresses, for its ADI port, enabling up to 8 ADS boards to be connected to the same AD part. The ADS address is selected by the DS1 dip switch found on sheet 3. The ADI port connector P1 is a 37 pin, male, D type connector. (Same connector used for communicating with the on board command converter). The connection between the ADS and the host computer is by a 37 line flat cable, supplied with the ADI board. FIGURE 1-15 below shows the pin configuration of the connector.

FIGURE 1-15 ADI Port Connector

Gnd	20	1	INT_ACK
Gnd	21	2	N.C.
Gnd	22	3	HST_ACK
Gnd	23	4	ADS_ALL
Gnd	24	5	ADS_RESET
Gnd	25	6	ADS_SEL2
N.C.	26	7	ADS_SEL1
HOST_VCC	27	8	ADS_SEL0
HOST_VCC	28	9	HOST_REQ
HOST_VCC	29	10	ADS_REQ
HOST_ENABLE~	30	11	ADS_ACK
Gnd	31	12	ADS_INT
Gnd	32	13	HOST_BRK~
Gnd	33	14	ADS_BRK
PD0	34	15	N.C.
PD2	35	16	PD1
PD4	36	17	PD3
PD6	37	18	PD5
		19	PD7

3•5•13•1 ADI Port Signal Description

In the list below, the directions 'I', 'O' and 'I/O' refer to the ADS board. (I.E. 'I' means input to the ADS)

•ADS_SEL(0:2) -'I'

These three inputs determine the ADS board address. The address being set, would be the only one to which the board respond, when referenced by the host computer. Up to 8 boards can be addressed by one ADI port.

• ADS_ALL -'I'

This input line is used to reset or abort program execution on all ADS development boards that are connected to the same ADI board. When this line is active, the IMP ADI and the on board command converter, are simultaneously referenced.

•HOST_ENABLE~ -'I'

This line is always driven low by the ADI board. The ADS hardware uses this line to determine if a host is connected to the ADI port.

•ADS_BRK -'I'

This line is used in conjunction with the addressing lines or with the ADS_ALL line to generate a non-maskable interrupt (interrupt level 7) to the processor.

- •ADS_RESET -'I'

When a host is connected, this line is used in conjunction with the addressing lines or with the ADS_ALL line to reset the ADS board.

- •HOST_REQ -'I'

This signal initiates a host to ADS write cycle.

- •ADS_ACK -'O'

This signal is the ADS response to the HOST_REQ signal, indicating that the ADS board has detected the assertion of HOST_REQ.

- •ADS_REQ -'O'

This signal initiates an ADS to host write cycle.

- •HST_ACK -'I'

This signal serves as the host's response to an ADS_REQ signal.

- •HOST_BRK~ -'O'

This open-collector signal, generates an interrupt to the host. This signal is common to all ADS boards that are connected to the same ADI.

- •ADS_INT -'O'

This line is polled by the host computer during its interrupt acknowledge cycle to determine which ADS board has generated an interrupt.

- •INT_ACK -'I'

This line is asserted by the host at the end of its interrupt acknowledge cycle. This signal is used by the ADS hardware to negate the HOST_BRK~ signal. The ADS software, must negate the ADS_INT signal upon detecting an assertion of INT_ACK, to support the daisy-chain interrupt structure.

- •HOST_VCC -'I' (three lines)

These +5V power lines, are used by the ADI logic, to determine if a host computer is powered on. The ADS does not use these lines for power supply.

- •PD(0:7) -'I/O'

These eight I/O lines are the parallel data bus. This bus is used to transmit and receive data from the host computer.

3•5•14   MC68681 DUART (sheet 11)

The MC68681 is DUART device connected to the 68302 part that provides the ADS with the following functions:

- • 16 bit programmable counter/timer.
- • ADI Port interface for the 68302 part.
- • 2 RS-232 ports interface.

The debugger initializes port A of the DUART, to work with - 9600bps, 8 bit data, 1 stop bit and no parity.

3•5•14•1 DUART Clock Source

The MC68681 uses a 3.6864Mhz crystal as its clock source, thus its communication rate, is independent of the system clock. The crystal is connected, between X1 and X2 pins, of the DUART.

3•5•15   RS-232 serial ports. (sheet 11)

The RS-232 serial ports of the ADS are connected to channels A and B, of the MC68681 DUART. The ADS can be connected to a VT100 compatible terminal through the primary serial port (P2 connector), which is

the only one supported by the debugger. Both ports has a 9 pin, D-type, female connector as shown in FIGURE 1-8.

FIGURE 1-16 RS-232 Serial Ports Connector

CD 1
TX 2   6 DSR
RX 3   7 RTS (N.C)
DTR 4   8 CTS
GND 5   9 N.C.

**P2 - Primery port
(DCE port)**

CD 1
TX 2   6 DSR
RX 3   7 RTS
DTR 4   8 CTS (N.C)
GND 5   9 N.C.

**P2 - Secoundary port
(DTE port)**

3•5•15•1 Primary RS-232 port Signal Description

In the list below, the directions 'I', 'O' and 'I/O' refer to the ADS board.

(I.E.'I' means input to the ADS)

- CD (O) - Data Carrier Detect. This line is always asserted by the ADS.
- TX (O) - Transmit Data.
- RX (I) - Receive Data.
- DTR (I) - Data Terminal Ready. This signal is used by the software in the ADS to detect if a terminal is connected to the ADS board.
- DSR (O) - Data Set Ready. This line is always asserted by the ADS.
- RTS (I) - Request To Send. This line is not connected in the ADS.
- CTS (O) - Clear To Send. This line is always asserted by the ADS.

* The secondary port has the above signals with the opposite flow direction.

3•5•16   PCMCIA Connector (sheet 12)

The PCMCIA interface pins of the Pchip appears on an Sbus miniature connector, which is interfaced, into a PCMCIA port, trough an extender. The ADS does not use host PCMCIA power, but a relay on the ADS is used to power the ADS, upon power detection, on the PCMCIA connector pins. The user can choose to use this relay by a Dip switch selection.

CHAPTER 4 - SUPPORT INFORMATION

4•1 INTRODUCTION

This chapter provides the interconnection signals, parts list, and schematic diagrams of the M68302FADS board.

4•2 INTERCONNECT SIGNALS

The M68302FADS board interconnects with external devices through the following connectors, listed in Table 2-8.

Table 2-8 Board interconnection

Reference	Type	Function
P1	37 pin, male D type connector	ADI port
P2	9 pin, female D type connector	RS232 terminal port (DCE)
P3	9 pin, female D type connector	RS232 host port (DTE)
P4	68 pin male, miniature Sbus connector	PCMCIA extender interface
P5 - P8	20 pin, male connectors, compatible with HP logic analyzer connector	logic analyzer connectors
P11	96 pin, male DIN connector	logic analyzer connectors
P9, P12	96 pin, female DIN connector	expansion port connectors
P10	3 pin power connector	5v power supply input

4•2•1   Connector P1 Interconnect Signals

P1 is a 37 pin, male D type connector. It is the ADI port of the M68302FADS. P1 signals, are listed in Table 2-9. (In the following table, input and output signals, refer to the ADS board)

Table 2-9 Connector P1 Interconnect Signals

Pin No.	Signal Name	Description
1	INT_ACK	Interrupt Acknowledge input signal.
2	-	Not connected
3	HST_ACK	Host Acknowledge input signal.
4	ADS_ALL	ADS All, input signal.
5	ADS_RESET	ADS Reset input signal.
6	ADS_SEL2	ADS Select 2, input signal. (ADI address bit 2)
7	ADS_SEL1	ADS Select 1, input signal. (ADI address bit 1)
8	ADS_SEL0	ADS Select 0, input signal. (ADI address bit 0)
9	HOST_REQ	HOST Request, input signal.
10	ADS_REQ	ADS Request, output signal.
11	ADS_ACK	ADS Acknowledge, output signal.
12	ADS_INT	ADS Interrupt, output signal.
13	HOST_BRK~	HOST Break open collector, output signal.
14	ADS_BRK	ADS Break, input signal.
15	-	Not connected
16	PD1	Bit 1 of the ADI port data bus
17	PD3	Bit 3 of the ADI port data bus
18	PD5	Bit 5 of the ADI port data bus
19	PD7	Bit 7 of the ADI port data bus
20 - 25	GND	board common ground
26	-	Not connected. The host supplies +12V on this pin, but it is not connected on the M68302FADS
27 - 29	HOST_VCC	HOST VCC input (+5V). The M68302FADS does not use these inputs for power supply.
30	HOST_ENABLE~	HOST Enable, input signal.
31 - 33	GND	Ground signal of the M68302FADS
34	PD0	Bit 0 of the ADI port data bus
35	PD2	Bit 2 of the ADI port data bus
36	PD4	Bit 4 of the ADI port data bus
37	PD6	Bit 6 of the ADI port data bus

4•2•2   Connector P2 Interconnect Signals

P2 is a 9 pin, female D type connector. It is the RS232 serial port of the M68302FADS. P2 signals are listed in Table 2-9. (In the following table, input and output signals, refer to the ADS board)

Table 2-10 Connector P2, Interconnect Signals

Pin No.	Signal Name	Description
1	CD	Carrier Detect output.
2	TX	Transmit Data output.
3	RX	Receive Data input.
4	DTR	Data Terminal Ready input.
5	GND	Ground signal of the ADS.
6	DSR	Data Set Ready output.
7	RTS (N.C.)	Request To Send. This line is not connected in the ADS.
8	CTS	Clear To Send output.
9	-	Not connected

4•2•3   Connector P3 Interconnect Signals

P3 is a 9 pin D type female connector. It is a host secondary RS232 port. Signals are listed in Table 2-13.

Table 2-11 Connector P3, Interconnect Signals

Pin No.	Signal Name	Description
1	TX	Transmit Data input.
2	-	Not connected
3	RX	Receive Data output.
4	DTR	Data Terminal Ready output.
5	GND	Ground signal of the ADS.
6	DSR	Data Set Ready input.
7	RTS	Request To Send output.
8	CTS (N.C)	Clear To Send input. This line is not connected in the ADS
9	-	Not connected

4•2•4   Connector P4 interconnect signals

P4, is 68 pin, SBUS type connector. It is the PCMCIA extender port. P4 signals are listed in Table 2-12.

Table 2-12 Connector P4 interconnect signals

Pin No.	Signal name	I/O	Description
1	GND		Board common ground
2	D3	I/O	PCMCIA data bit 3
3	D4	I/O	PCMCIA data bit 4
4	D5	I/O	PCMCIA data bit 5
5	D6	I/O	PCMCIA data bit 6

Table 2-12 Connector P4 interconnect signals

Pin No.	Signal name	I/O	Description
6	D7	I/O	PCMCIA data bit 7
7	CE1~	I	Card enable 1
8	A10	I	PCMCIA address bit 10
9	OE~	I	PCMCIA output enable
10	A11	I	PCMCIA address bit 11
11	A9	I	PCMCIA address bit 9
12	A8	I	PCMCIA address bit 8
13	A13	I	PCMCIA address bit 13 (Not supported)
14	A14	I	PCMCIA address bit 14 (Not supported)
15	WE~	I	PCMCIA write enable
16	RDY	O	PCMCIA ready output.
17	VCC	-	Host power.
18	VPP1	-	Programming voltage (Not supported)
19	A16	I	PCMCIA address bit 16 (Not supported)
20	A15	I	PCMCIA address bit 15 (Not supported)
21	A12	I	PCMCIA address bit 12 (Not supported)
22	A7	I	PCMCIA address bit 7
23	A6	I	PCMCIA address bit 6
24	A5	I	PCMCIA address bit 5
25	A4	I	PCMCIA address bit 4
26	A3	I	PCMCIA address bit 3
27	A2	I	PCMCIA address bit 2
28	A1	I	PCMCIA address bit 1
29	A0	I	PCMCIA address bit 0
30	D0	I/O	PCMCIA data bit 0
31	D1	I/O	PCMCIA data bit 1
32	D2	I/O	PCMCIA data bit 2
33	IOIS16~	O	I/O is 16. (Set by on board jumper)
34	GND	-	Board common ground
35	GND	-	Board common ground
36	CD1~	O	PCMCIA card detect 1 (Grounded)
37	D11	I/O	PCMCIA data bit 11
38	D12	I/O	PCMCIA data bit 12
39	D13	I/O	PCMCIA data bit 13
40	D14	I/O	PCMCIA data bit 14
41	D15	I/O	PCMCIA data bit 15
42	CE2~	I	PCMCIA card enable 2
43	RFRSH		Not supported
44	IORD	I	PCMCIA I/O read
45	IOWR	I	PCMCIA I/O write

Table 2-12 Connector P4 interconnect signals

Pin No.	Signal name	I/O	Description
46	A17	I	PCMCIA address bit 17
47	A18	I	PCMCIA address bit 18
48	A19	I	PCMCIA address bit 19
49	A20	I	PCMCIA address bit 20
50	A21	I	PCMCIA address bit 21
51	VCC	-	Host VCC
52	VPP2	-	Not supported
53	A22	I	PCMCIA address bit 22
54	A23	I	PCMCIA address bit 23
55	A24	I	PCMCIA address bit 24
56	A25	I	PCMCIA address bit 25
57	RFU		Reserved for future use
58	RST	I	PCMCIA reset
59	WAIT	O	PCMCIA wait
60	IPACK	O	Input port ack
61	REG	I	Reg memory attribute
62	SPKR	-	Not supported
63	SCHG~	I	PCMCIA status changed
64	D8	I/O	PCMCIA data bit 8
65	D9	I/O	PCMCIA data bit 9
66	D10	I/O	PCMCIA data bit 10
67	CD2~	O	PCMCIA card detect 2 (Grounded)
68	GND	-	Board common ground

4•2•4•1 Connector P5 Interconnect Signals

P10 is a dual row, 20 pin, male connector. The upper address bus and control signals of the processor, appear on this connector, for easy monitoring by a logic analyzer. P5 connector signals are listed in Table 2-13.

Table 2-13 P5 connector interconnect signals

Pin NO	Signal name	Description
1	NC	No connection
2	NC	No connection
3	WEH~/UDS~	write enable high/upper data strobe
4	A15	address line 15
5	A14	address line 14
6	A13	address line 13
7	A12	address line 12
8	A11	address line 11
9	A10	address line 10
10	A9	address line 9

Table 2-13 P5 connector interconnect signals

Pin NO	Signal name	Description
11	A8	address line 8
12	A7	address line 7
13	A6	address line 6
14	A5	address line 5
16	A4	address line 4
16	A3	address line 3
17	A2	address line 2
18	A1	address line 1
19	WEH~/UDS~	write enable high/upper data strobe
20	GND	Board common ground

4•2•5 Connector P6 Interconnect Signals

P9 is a dual row, 20 pin, male connector. The upper address bus and bus control signal of the processor, appear on this connector, for easy monitoring by a logic analyzer.P9 connector signals are listed in Table 2-14.

Table 2-14 P6 connector interconnect signals

Pin NO.	Signal name	Description
1	NC	No connection
2	NC	No connection
3	WEL~/LDS~	write enable low/lower data storbe
4	BGACK~	bus grant acknowledge
5	FC2	function code 2
6	FC1	function code 1
7	FC0	function code 0
8	IAC	internal access
9	WEH~/UDS~	write enable high/upper data strobe
10	WEL~/LDS~	write enable low/lower data strobe
11	OE~/R/W~	output enable/read write
12	A23	address line 23
13	A22	address line 22
14	A21	address line 21
15	A20	address line 20
16	A19	address line 19
17	A18	address line 18
18	A17	address line 17
19	A16	address line 16
20	GND	Board common ground

4•2•6   Connector P7 interconnect Signals

P7 is a dual row, 20 pin, male connector. The processor data bus appears on this connector, for easy monitoring by a logic analyzer. P7 signals are listed in Table 2-16.

Table 2-15 P7 interconnect signals

Pin NO	Signal name	Description
1	NC	No connection
2	NC	No connection
3	AS~	address strobe
4	D15	data line 15
5	D14	data line 14
6	D13	data line 13
7	D12	data line 12
8	D11	data line 11
9	D10	data line 10
10	D9	data line 9
11	D8	data line 8
12	D6	data line 7
13	D7	data line 6
14	D5	data line 5
15	D4	data line 4
16	D3	data line 3
17	D2	data line 2
18	D1	data line 1
19	D0	data line 0
20	GND	Board common ground

4•2•7   Connector P8 interconnect Signals

P8 is a dual row, 20 pin, male connector. The processor data bus appears on this connector, for easy monitoring by a logic analyzer. P8 signals are listed in Table 2-16

Table 2-16 P8 interconnect signals

Pin NO	Signal name	Description
1	NC	No connection
2	NC	No connection
3	CLKO	clock out
4	CS3~	chip select line 3
5	CS2~	chip select line 2
6	AVEC~	auto vector
7	BERR~	bus error
8	RST~	reset

Table 2-16 P8 interconnect signals

Pin NO	Signal name	Description
9	CLKO	clock out
10	DTACK~	data transfer acknowledge
11	HALT~	halt control
12	BR~	bus request
13	CS1~	chip select line 1
14	IPL2~	interrupt priority level 2
15	IPL1~	interrupt priority level 1
16	IPL0~	interrupt priority level 0
17	BG~	bus grant
18	CS0~	chip select line 0
19	AS~	address strobe
20	GND	Board common ground

4•2•8   Connector P9 interconnect signals

P9 is a 3 row, 96 pin female DIN connector. All processors signals, are routed directly to P9 and P12, for user hardware applications. P9 signals are listed in Table 2-17.

Table 2-17 Connector P9 interconnect signals

Pin NO	Signal name	Description
A1	GND	Board common ground
A2	GND	Board common ground
A3	GND	Board common ground
A4	A7	address line bit 7
A5	A4	address line bit 4
A6	A9	address line bit 9
A7	A14	address line bit 14
A8	A19	address line bit 19
A9	A16	address line bit 16
A10	A21	address line bit 21
A11	FC2	function code bit 2
A12	RMC~	read modify write cycle
A13	CS3~	chip select 3
A14	PB3	port B bit 3
A15	IAC	internal access
A16	GND	common
A17	PB1	port B bit 1
A18	FC1	function code bit 1
A19	D14	data bit 14
A20	D11	data bit 11
A21	OE~/R/W~	output enable/read write

Table 2-17 Connector P9 interconnect signals

Pin NO	Signal name	Description
A22	D6	data bit 6
A23	DTACK~	data transfer acknowledge
A24	BERR~	bus error
A25	IPL1~	interrupt priority level 1
A26	IPL2~	interrupt priority level 1
A27	PA15	port A bit 15
A28	BGACK~	bus grant acknowledge
A29	FRZ~	freeze
A30	PA13	port A bit 13
A31	BG~	bus grant
A32	VCC	board common +5V supply
B1	GND	Board common ground
B2	A1	address bit 1
B3	GND	Board common ground
B4	A2	address bit 2
B5	A5	address bit 5
B6	A10	address bit 10
B7	A15	address bit 15
B8	A12	address bit 12
B9	A17	address bit 17
B10	A22	address bit 22
B11	FC0	function code 0
B12	CS0~	chip select bit 0
B13	PB11	port B bit 11
B14	PB10	port B bit 10
B15	PB2	port B bit 2
B16	PB8	port B bit 8
B17	PB4	port B bit 4
B18	WEH~/UDS~	write enable high/upper data strobe
B19	D10	data bit 10
B20	D13	data bit 13
B21	AS~	address strobe
B22	D9	data bit 9
B23	D2	data bit 2
B24	AVEC~	auto vector
B25	IPL0~	interrupt priority 0
B26	BR~	bus request
B27	EXTAL	external clock input/crystal
B28	RST~	reset
B29	PA12	port A bit 12

Table 2-17 Connector P9 interconnect signals

Pin NO	Signal name	Description
B30	D3	data bit 3
B31	busw	bus width
B32	VCC	board common +5V supply
C1	GND	board common ground
C2	GND	board common ground
C3	GND	board
C4	A3	address bit 3
C5	A6	address bit 6
C6	A11	address bit 11
C7	A8	address bit 8
C8	A13	address bit 13
C9	A18	address bit 18
C10	A23	address bit 23
C11	A20	address bit 20
C12	CS1~	chip select 1
C13	CS2~	chip select 2
C14	PB6	port B bit 6
C15	PB5	port B bit 5
C16	PB7	port B bit 7
C17	PB9	port B bit 9
C18	PB0	port B bit 0
C19	D15	data bit 15
C20	D12	data bit 12
C21	WEL~/LDS	write enable low/lower data strobe
C22	D8	data bit 8
C23	D7	data bit 7
C24	D1	data bit 1
C25	CLKO	clock out
C26	D5	data bit 5
C27	BCLR~	bus clear
C28	D4	data bit 4
C29	D0	data bit 0
C30	HALT~	halt control
C31	PA14	port A bit 14
C32	VCC	board common +5V supply

4•2•9   Connector P10 Interconnect Signals

P18 is 3 pin connector for 5v power supply. The connector is supplied with 3 pin plug for convenient

connection to the power supply. P10 connector signals, are listed in Table 2-9.

Table 2-18 Connector P10 interconnect Signals

Pin No.	Signal Name	Description
1	VCC	Power supply, +5V connection.
2	GND	Power supply, ground connection.
3	GND	Power supply, ground connection.

4•2•10   Connector P11 Interconnect Signals

P11 is a 3 row, 96 pin, male DIN connector. This connector, together with P5, P6, P7 and P8, has all processor pins for easy interfacing to a logic analyzer. P11 signals, are listed in Table 2-19.

Table 2-19 Connector P11 interconnect signals

Pin NO	Signal name	Description
A1	GND	Board common ground
A2	GND	Board common ground
A3	GND	Board common ground
A4	IPB0	port B bit 0
A5	IPB1	port B bit 1
A6	IPB2	port B bit 2
A7	IPB3	port B bit 3
A8	IPB4	port B bit 4
A9	IPB5	port B bit 5
A10	IPB6	port B bit 6
A11	IPB7	port B bit 7
A12	IPB8	port B bit 8
A13	IPB9	port B bit 9
A14	IPB10	port B bit 10
A15	IPB11	port B bit 11
A16	GND	Board common ground
A17	PDI(8)	port D bit 8
A18	PDI(9)	port D bit 9
A19	PDI(10)	port D bit 10
A20	PDI(11)	port D bit 11
A21	PDI(12)	port D bit 12
A22	PDI(13)	port D bit 13
A23	PDI(14)	port D bit 14
A24	GND	Board common ground
A25	GND	Board common ground
A26	RXD1	SCC1 receive data input
A27	TXD1	SCC1 transmit data output
A28	CD1~	SCC1 carrier detect
A29	CTS1~	SCC1 clear to send

Freescale Semiconductor, Inc.

Table 2-19 Connector P11 interconnect signals

Pin NO	Signal name	Description
A30	RTS1~	SCC1 request to sent
A31	GND	Board common ground
A32	GND	Board common ground
B1	GND	Board common ground
B2	GND	Board common ground
B3	GND	Board common ground
B4	PCD0	PCMCIA data bit 0
B5	PCD1	PCMCIA data bit 1
B6	PCD2	PCMCIA data bit 2
B7	PCD3	PCMCIA data bit 3
B8	N.C.	No connect
B9	GND	Board common ground
B10	PCA3	PCMCIA address bit 3
B11	PCA4	PCMCIA address bit 4
B12	PCA5	PCMCIA address bit 5
B13	GND	Board common ground
B14	PCRDY	PCMCIA ready signal
B15	PCOE~	PCMCIA output enable
B16	PCCE1~	PCMCIA card enable 1
B17	GND	Board common ground
B18	PCREG~	PCMCIA reg signal
B19	PCWAIT~	PCMCIA wait signal
B20	PCSTSCH~	PCMCIA status change signal
B21	GND	Board common ground
B22	GND	Board common ground
B23	GND	Board common ground
B24	GND	Board common ground
B25	GND	Board common ground
B26	GND	Board common ground
B27	GND	Board common ground
B28	GND	Board common ground
B29	GND	Board common ground
B30	GND	Board common ground
B31	GND	Board common ground
B32	GND	Board common ground
C1	GND	Board common ground
C2	GND	Board common ground
C3	GND	Board common ground
C4	PA0	port A bit 0
C5	PA1	port A bit 1

Table 2-19 Connector P11 interconnect signals

Pin NO	Signal name	Description
C6	PA2	port A bit 2
C7	PA3	port A bit 3
C8	PA4	port A bit 4
C9	PA5	port A bit 5
C10	PA6	port A bit 6
C11	PA7	port A bit 7
C12	PA8	port A bit 8
C13	PA9	port A bit 9
C14	PA10	port A bit 10
C15	PA11	port A bit 11
C16	PA12	port A bit 12
C17	PA13	port A bit 13
C18	PA14	port A bit 14
C19	PA15	port A bit 15
C20	GND	Board common ground
C21	BRG1~	baud rate generator 1 output
C22	GND	Board common ground
C23	RCLK1	SCC 1 receive clock
C24	GND	Board common ground
C25	TCLK1	SCC 1 receive clock
C26	GND	Board common ground
C27	CTS3~	SCC3 clear to send
C28	RTS3~	SCC3 request to send
C29	CD3~	SCC 3 carrier detect
C30	GND	Board common ground
C31	GND	Board common ground
C32	GND	Board common ground

4•2•11   Connector P12 interconnect signals

P12 is a 3 row, 96 pin female DIN connector. Processor signals, are directly routed to P9 and P12 connectors, for user hardware applications. P19 connector signals, are listed in Table 2-20.

Table 2-20 Connector P12 interconnect signals

Pin NO	Signal name	Description
A1	VCC	Board common +5V supply
A2	VCC	Board common +5V supply
A3	GND	Board common ground
A4	PB0	port B bit 0
A5	PB1	port B bit 1

Freescale Semiconductor, Inc.

Table 2-20 Connector P12 interconnect signals

Pin NO	Signal name	Description
A6	PB2	port B bit 2
A7	PB3	port B bit 3
A8	PB4	port B bit 4
A9	PB5	port B bit 5
A10	PB6	port B bit 6
A11	PB7	port B bit 7
A12	PB8	port B bit 8
A13	PB9	port B bit 9
A14	PB10	port B bit 10
A15	PB11	port B bit 11
A16	GND	Board common ground
A17	PDI8	port D bit 8
A18	PDI9	port D bit 9
A19	PDI10	port D bit 10
A20	PDI11	port D bit 11
A21	PDI12	port D bit 12
A22	PDI13	port D bit 13
A23	PDI14	port D bit 14
A24	GND	Board common ground
A25	GND	Board common ground
A26	RXD1	SCC1 receive data
A27	TXD1	SCC1 transmit data
A28	CD1~	SCC1 carrier detect
A29	CTS1~	SCC1 clear to send
A30	RTS1~	SCC1 request to send
A31	GND	Board common ground
A32	GND	Board common ground
B1	VCC	Board common +5V supply
B2	EXTNMI~	external level 7 interrupt source
B3	GND	Board common ground
B4	PCD0	PCMCIA data bit 0
B5	PCD1	PCMCIA data bit 1
B6	PCD2	PCMCIA data bit 2
B7	PCD3	PCMCIA data bit 3
B8	N.C.	No connect
B9	GND	Board common ground
B10	PCA3	PCMCIA address bit 3
B11	PCA4	Board common ground
B12	PCA5	Board common ground
B13	GND	Board common ground

Table 2-20 Connector P12 interconnect signals

Pin NO	Signal name	Description
B14	PCRDY	PCMCIA ready signal
B15	PCOE~	PCMCIA output enable
B16	PCCE1~	PCMCIA card enable 1
B17	GND	Board common ground
B18	PCREG~	PCMCIA reg signal
B19	PCWAIT~	PCMCIA wait signal
B20	PCSTSCHG~	PCMCIA status change
B21	GND	Board common ground
B22	BCLR~	bus clear
B23	RMC~	read modify write cycle
B24	GND	Board common ground
B25	DISCPU	disable CPU
B26	GND	Board common ground
B27	GND	Board common ground
B28	GND	Board common ground
B29	GND	Board common ground
B30	GND	Board common ground
B31	VCC	Board common +5V supply
B32	VCC	Board common +5V supply
C1	VCC	Board common +5V supply
C2	GND	Board common ground
C3	GND	Board common ground
C4	PA0	port A bit 0
C5	PA1	port A bit 1
C6	PA2	port A bit 2
C7	PA3	port A bit 3
C8	PA4	port A bit 4
C9	PA5	port A bit 5
C10	PA6	port A bit 6
C11	PA7	port A bit 7
C12	PA8	port A bit 8
C13	PA9	port A bit 9
C14	PA10	port A bit 10
C15	PA11	port A bit 11
C16	PA12	port A bit 12
C17	PA13	port A bit 13
C18	PA14	port A bit 14
C19	PA15	port A bit 15
C20	GND	Board common ground
C21	BRG1	baud rate generator 1

Freescale Semiconductor, Inc.

Table 2-20 Connector P12 interconnect signals

Pin NO	Signal name	Description
C22	GND	Board common ground
C23	RCLK1	SCC 1 receive clock
C24	NC	no connect
C25	TCLK1	SCC 1 transmit clock
C26	GND	Board common ground
C27	CTS3~	SCC 3 clear to send
C28	RTS3~	SCC 3 request to send
C29	CD3|	SCC 3 carrier detect
C30	GND	Board common ground
C31	GND	Board common ground
C32	VCC	board common +5V supply

For More Information On This Product,
Go to: www.freescale.com

Table 2-21 Part list

Reference	Part NO/Value	Manufacturer	total
C1 (Alternative to C1A)	220uF/25V - Alum.	Nippon	1
C1A (Alternative to C1)	220uF/10V SMD - TPSE227K010R	AVX	1
C2 (Alternative to C2A)	47uF/16V - Alum.	Nippon	1
C2A (Alternative to C2)	47uF/10V SMD - TAJD476K010	AVX	1
C3 C4 C5 C6 (C7) C8 C9 C10 C11 C12 C13 C14 C15 C16 C17 C19 C20 C21 C22 C23 C24 C25 C26 C27 C28 C29 C30 C31 C32 C34 C35 C36 C38 C40 C41 C42 C45 C46 C47 C48 C49 C50 C51 C52 C53 C54 C56 C61 C66 C67C68 C69 C70 C71 C72 C73 C74 C76 C77 C78 C82 C83 C84 C85 C86	0.1uF/50V 1206	AVX	65
C18 C33 C55 C57 C58 C59 C60 C62 C63 C64 C65 C75	10uF/20V 10% 5A-100KAT00J	Nippon	12
C37 C43	1.5nF/50V 1206	AVX	2
(C39) (C44)	10pF/50V 10% 1206	AVX	2
C79	1nF/50V 10% 1206	AVX	1
C80	4.7pF/50V 10% 1206-5A4R7KAT00J	AVX	1
C81	15pF/50V 10% 1206-5A150KAT00J	AVX	1
D1 D2 (Alternative to D1A, D2A)	1N4148		2
D1A D2A (Alternative to D1, D2)	LL4148	TSC	2
D3	MBRD620CT	Motorola	1
D4	1SMC50AT3	Motorola	1
DS1 DS2	DIP SW 10PST 90 HBW10S	Grayhill	2
F1	Fuse 3.155A/250V Fast 5x20mm 2170005 + PCB Fuse holder PTF15		1 1
J1 J2 J3	3 pin jumper		3
LD1	Yellow LED LY-T670-HK	Siemens	1
LD2	Red LED LR-T670-HK	Siemens	1
LD3	Orange LED LO-T670-HK	Siemens	1
LD4	Green LED LG-T670-HK	Siemens	1
P1	D-type 37p male 90' w clip conn.	Keltron	1
P2 P3	D-type 9p female 90' w clip conn.	Keltron	2
P4	68p male 90' PC mini SBUS conn.	ODU	1
P5 P6 P7 P8	20p logic analyzer conn.	Molex	4
P9 P12	DIN96p female 90' w clip	Elco	2

Freescale Semiconductor, Inc.

Table 2-21 Part list

Reference	Part NO/Value	Manufacturer	total
P10	3 pin male power conn.	Molex	1
P11	DIN96p m straight 168477096001025	Elco	1
R1 R7	2K 5% 1/4W 1206	AVX	2
(R2) R8	220 5% 1/4W 1206	AVX	2
R3 R9 R18 R21 R22 R23 R25	10K 5% 1/4W 1206	AVX	7
R4 R13 R19	1K 5% 1/4W 1206	AVX	3
(R5) R17	470K 5% 1/4W 1206	AVX	2
(R6) R11 R16 R20	330 5% 1/4W 1206	AVX	4
R10	100 5% 1/4W 1206	AVX	1
R12 R14	100K 5% 1/4W 1206	AVX	2
R15	47K 5% 1/4W 1206	AVX	1
R24	1M 5% 1/4W 1206	AVX	1
RLY1	Relay 5V 8A SPDT 42319005	Finder	1
RN1 RN4 RN5 RN6 RN7 RN8	13x4.7K Res NW DIP SOMC1401472	Dale	6
RN2 RN3	8x22 Res NW 16DIP SOMC1603220J	Dale	2
SW1 SW2	Push button SPDT E121SD1AV2GE + Red cap 8018/3 (For SW2) + Black cap 8018/2 (For SW1)		2 1 1
U1 U5	74LS244D	Motorola	2
U2 U13 U25 U26	74ACT244D	Motorola	4
*U3	MACH110-2oJC + 44p PLCC socket	AMD AMP	1 1
U4	74F543	Motorola	1
*U6 *U7	AM29F040-90 + 32p PLCC socket 205-032-10	AMD AMP	2 2
*U8	AT28C16-15JC + 32p PLCC socket 205-032-10	Atmel AMP	1 1
(U9) (U39)	SPARE		2
**U10	MC68302RC + 132p PGA socket 13x13 PC	Motorola Precidip	1 1
(U11) U12 (U14) U15 (U17) U18 (U20) U21	MCM6229A	Motorola	8
U16 U19 U22	74ACT245	Motorola	3
**U23	MC68LC302 + 132p PGA socket 13x13 PC	Motorola Precidip	1 1
*U24	MACH220-12JC + 68p PLCC socket 205-068-10	AMD AMP	1 1
**U27	MC68PM302RC + 181p PGA socket 15x15 PC	Motorola Precidip	1 1
U28 U29	MC145407	Motorola	2
U30	MC68681FN	Motorola	1

Table 2-21 Part list

Reference	Part NO/Value	Manufacturer	total
U31	74ACT08D	Motorola	1
U32 U35	74ACT00D	Motorola	2
U33	74ACT74D	Motorola	1
*U34	8, 32 or 40MHz Osc. HXO-125532 + 14 pin DIP socket.	Precidip	1 1
U36	74LS05D	Motorola	1
U37	74HC4538D	Motorola	1
U38	74ACT14D	Motorola	1
(Y1)	32KHz or 4MHz crystal HC49U		1
Y2	3.6864MHz crystal HCU46		1
	Rubber leg		5

How to read the table.

(U1)   - Component U1 is not assembled.

*U1   - Component U1 should be mounted on a socket.

**U1   - Only the socket for U1, should be assembled.

Freescale Semiconductor, Inc.

APPENDIX A - ADI BOARD INSTALLATION

A•1   INTRODUCTION

This appendix describes the hardware installation of the ADI board into various host computers.

The installation instructions, cover the following host computers:

1.   IBM-PC/XT/AT
2.   SUN - 4 (SBus interface)

A•2   IBM-PC/XT/AT to M68302FADS Interface

The ADI board should be installed in one of the IBM-PC/XT/AT mother board system, expansion slots. A single ADI can control, up to eight M68302FADS boards. ADI address, in the host computer, is configured to be at I/O addresses space,100-102 (hex), but it may be reconfigure, for an alternate address space.

CAUTION

BEFORE REMOVING OR INSTALLING ANY EQUIPMENT IN THE IBM-PC/XT/AT COMPUTER, TURN THE POWER OFF AND REMOVE THE POWER CORD.

A•2•1   ADI Installation in IBM-PC/XT/AT

Refer to the appropriate Installation and Setup manual of the IBM-PC/XT/AT computer, for instructions on removing computer cover.

The ADI board address block, should be configured at a free I/O address space. The address must be unique and it must not fall within the address range, of another card installed in the computer.

The ADI board address block can be configured to start at one of the three following addresses:

• $100 - This address is unassigned in the IBM-PC
• $200 - This address is usually used for the game port
• $300 - This address is defined as a prototype port

The ADI board is factory configured for address decoding at 100-102 hex in the IBM-PC/XT/AT I/O address map. These are undefined peripheral addresses.

For More Information On This Product,
Go to: www.freescale.com

FIGURE A-1 Physical Location of jumper JG1 and JG2

JG1   JG2

NOTE: Jumper JG2 should be left unconnected.

The following figure, shows the required jumper connection, for each address configuration. Address 0 hex is not recommended and its usage might cause problems.

FIGURE A-2 JG1 Configuration Options

0 hex   100 hex   200 hex   300 hex

To properly install the ADI board, position its front bottom corner in the plastic card guide channel, at the front of the IBM-PC/XT/AT chassis. Keeping ADI board top and any ribbon cables, out of the way, lower the board until its connectors are aligned with the computer expansion slot connectors. Using evenly distributed pressure, press the ADI board straight down until it seats in the expansion slot.

Secure the ADI board to computer's chassis, using a bracket retaining screw. Refer to computer Installation and Setup manual, for instructions on reinstalling, computer cover.

A•3 SUN-4 to M68302FADS Interface

The ADI board should be installed in one of the SBus expansion slots in a Sun-4 SPARCstation. A single ADI, can control up to eight M68302FADS boards.

CAUTION

BEFORE REMOVING OR INSTALLING ANY EQUIPMENT IN THE SUN-4 COMPUTER, TURN THE POWER OFF AND REMOVE POWER CORD.

A•3•1 ADI Installation in a SUN-4 machine

There are no jumper options, on the ADI board, for a Sun-4 machine. The ADI board can be inserted into any available SBus expansion slot, on the mother board.

Refer to the appropriate Installation and Setup manual for the Sun-4 computer for instructions on removing the computer cover and installing the board in an expansion slot.

FIGURE A-3 ADI board for SBus

Following are the Sun's manual, instructions summary:

1. Turn off the power to your system, but keep power cord, plugged in. Be sure to save all open files and follow the steps bellow, to shut down your system:
   - hostname% /bin/su
   - Password: mypasswd
   - hostname# /usr/etc/halt
   - wait for the following messages.

     Syncing file systems... done

     Halted

     Program Terminated

     Type b(boot), c(continue), n(new command mode)
   - When these messages appear, you can safely turn off, your system power.

2. Open the system unit. Be sure to attach a grounding strap, to your wrist and to the power supply, metal casing. Follow the instructions supplied with your system, to gain access to the SBus slots.

3. Remove SBus slot filler panel, from the inner surface of the back panel of your system unit. Note that the ADI board, is a slave only board and thus will function in any available SBus slot.

4. Slide the ADI board at an angle, into the back panel of your system unit. Make sure, the mounting plate on the ADI board, hooks into the holes on the back panel of your system unit.

5. Push the ADI board against the back panel, align the connector with its mate and gently press board corners, to plug the connector firmly.

6. Close your system unit.

7. Connect a 37 pin interface flat cable to the ADI board and secure.

8. Turn on system power and check for proper operation.

Freescale Semiconductor, Inc.

APPENDIX B - ADI PORT HANDSHAKE DESCRIPTION

B•1 INTRODUCTION

The M68302FADS ADI port, can be connected to an ADI board, mounted in a host computer.

There are ADI boards for the following host computers:

1. IBM-PC/XT/AT
2. SUN - 4 (SBus interface)

B•2 ADI Port Concept and Operation Description

Each ADI board, can be connected with up to 8 M68302FADS boards. Each ADS has its own address which is fixed by setting Dip-Switch DS1 switches 8-10, on the board.

The following operations can be performed using the ADI port:

- The host computer can write a byte to the ADS
- The ADS can write a byte to the host computer
- The ADS can interrupt the host computer
- The host computer can interrupt the ADS (interrupt level 7)
- The host computer can reset the ADS

If more than one ADS, is connected to the same ADI board, the host computer can perform the following operations simultaneously on all M68302FADS boards:

- Abort all boards (interrupt level 7)
- Reset all boards

B•3   Handshake Description

Every action between the ADS and the host, is asynchronous and implemented by asserting and negating handshake signals trough software.

All signals have TTL levels. A control signal is asserted, when driven to logic'1' TTL level, and it is negated when driven to logic'0' level.

The connection between a host computer and ADS, is shown in FIGURE B-1 below.

FIGURE B-1 Host Computer (ADI) to M68302FADS Connection

B•3•1   Write Cycle from Host to M68302FADS

The application software in the Host, uses handshake signals, to coordinate data transfer across parallel link. The software installed on the ADS, is responsible for accepting data, and responding to handshake signals. The signals are shown in FIGURE B-2.

The sequence of events during a byte write to a M68302FADS, is as follows:

1. The Host selects the M68302FADS board, by placing the board address, on ADS_SEL(0:2) lines.
2. The Host places a data byte, in a data bus latch (Its output buffer, is in high-impedance state).
3. The Host asserts HOST_REQ signal (data buffer is enabled. data appears on the bus).
4. The ADS, detects the HOST_REQ signal and reads the data byte.
5. The ADS respond by asserting ADS_ACK signal.
6. The Host detects the ADS_ACK signal and negates HOST_REQ signal (data buffer is disabled).
7. The ADS detects HOST_REQ negation and negates ADS_ACK to end the cycle.

FIGURE B-2 Host Write to M68302FADS

B•3•2   Write Cycle from M68302FADS to Host

Signal handshake during an ADS to Host write cycle is shown in FIGURE B-3. The sequence of events, is as follows:

1. The M68302FADS places a data byte on the parallel port data bus (buffer disabled) and asserts ADS_REQ signal. (the ADS_REQ signal does not appear on the port, until the board is selected by the Host)

2. The Host polls each ADS address and detects the ADS_REQ signal from the requesting board. The Host asserts the HST_ACK signal in response, which enables ADS data buffer.

3. The ADS negates ADS_REQ signal. Data appears on the bus, as long as the HST_ACK signal is asserted.

4. The Host reads the data.

5. The Host negates HST_ACK signal, to end the cycle. The ADS ends the cycle.

FIGURE B-3 M68302FADS Write Cycle to Host

B•3•3   M68302FADS Interrupt to the Host

The M68302FADS can generate an interrupt to the Host. Interrupt request and acknowledge sequence, is shown in FIGURE B-4.

The sequence is as follows:

1. The ADS, places a service request code on the parallel port data bus (buffer disabled) asserts ADS_INT and HOST_BRK~ signals. The HOST_BRK~ signal is an open-collector signal, asserted low, common to all ADS boards, that appears immediately on the port. The ADS_INT signal does not appear on the port, until the board is selected by the Host.

2. The Host detects an HOST_BRK~ signal and polls each ADS address to determine the interrupting board.

3. The Host asserts an HST_ACK signal, enabling the ADS data buffer.

4. The Host reads the service request code on the data bus.

5. The Host negates the HST_ACK signal.

6. The Host asserts an INT_ACK signal, which resets the HOST_BRK latch, in the ADS and negates the HOST_BRK~ signal. the HOST_BRK~ signal can remain low (asserted) if another ADS board is driving it low.

7. The selected ADS, detects an INT_ACK signal and negates the ADS_INT signal.

8. The Host negates the INT_ACK signal and ends the cycle.

FIGURE B-4 M68302FADS Interrupt to Host

B•3•4   Host Interrupt to the M68302FADS

The Host can interrupt the ADS (interrupt level 7) to abort program execution, running on the board. This is done by selecting an address of the required ADS and momentarily asserting an ADS_BRK signal, which sets the ADS latch. The latch output, interrupts the processor on the ADS and can be cleared by the interrupt handling software on the ADS.

B•3•5   Host Reset to the M68302FADS

The Host can perform reset to the ADS. Reset is done by selecting the address of the required board and asserting the ADS_RESET signal, for more than 26 microseconds.

B•3•6   Addressing All M68302FADS

The Host can reset or interrupt all M68302FADS boards, which are connected to the same ADI card. The host should assert an ADS_ALL signal, in conjunction with either ADS_RESET or ADS_BRK. The contents of the ADS_SEL(0:2) lines have no affect.

Freescale Semiconductor, Inc.

APPENDIX C - PLD equations.

C•1   IMP control logic - U24.

```
"*******************************************************************************
"* 302 family ADS control logic.                                             *
"* Logic includes CS logic, DTACK generator, interrupt controller, bus       *
"* arbiter, resources enable, buffers control and NMI reg.                    *
"*******************************************************************************
module _302fmlyADS
title  '302 family ADS control logic.


"*******************************************************************************
"* Device declaration.                                                       *
"*******************************************************************************
U01 device 'mach220a';


"*******************************************************************************
"* Pins declaration.                                                         *
"*******************************************************************************


"*******************************************************************************
"* 302 signals.                                                              *
"*******************************************************************************
CLKO                        PIN 15;         "Clock.
AS~                         PIN 16;         "Address strobe.
IAC                         PIN 54;         "Internal access.
A1, A2, A3                  PIN 41, 45, 60; "Address lines. (Int level)
A16, A17, A18, A19          PIN 57,56,40,37;"Address lines and int ack.
CS0~, CS1~, CS2~            PIN 4, 23, 55;  "Chip select lines.
FC0, FC1, FC2               PIN 26, 25, 24; "Function code lines.
WEL~                        PIN 50;         "WEL~/WE~ or LDS~/DS~ in the IMP.
WEH~                        PIN 49;         "WEH~/A0 or UDS~/A0 in the IMP.
OE~                         PIN 20;         "OE~ or R/W~ in the IMP control (in).
BR_W~                       PIN 28;         "Buffered R/W~ (out.)
BUSW                        PIN 17;         "Bus width control.
Rst~                        PIN 47;         "Reset signal.
Halt~                       PIN 67;         "Halt signal.
IPL0~, IPL1~, IPL2~         PIN 38, 36, 39; "Interrupt priority;
AVEC~                       PIN 48;         "Auto vector.


"*******************************************************************************
"* Buffer control signals.                                                   *
"*******************************************************************************
BusL~                       PIN 31;         "Data buffer D0-D7 enable.
BusL2H~                     PIN 30;         "D0-D7 to D8-D15 bus, enable.
BusH~                       PIN 29;         "Data buffer D8-D15 enable.


"*******************************************************************************
"* C/S and R/W signals.                                                      *
"*******************************************************************************
RamLCs~                     PIN 13;         "RAM low CS.
```

Freescale Semiconductor, Inc.

```
RamHCs~                     PIN 12;         "RAM high CS.

ExpRamLCs~                  PIN 10;         "Expansion RAM low CS.

ExpRamHCs~                  PIN 11;         "Expansion RAM high CS.

FlashLCs~                   PIN 5;          "FLASH low CS.

FlashHCs~                   PIN 3;          "FLASH high CS.

EepromCs~                   PIN 14;         "EEPROM CS. (Connected to D0-D7)

DuartCs~                    PIN 9;          "DUART CS.

AdiRd~                      PIN 21;         "Adi port, read enable.
AdiWr~                      PIN 22;         "Adi port, write enable.

"Low/High bus R/W control split, is only for deviding signal load.
RdL~                        PIN 7;          "Bus low, read signal.
RdH~                        PIN 2;          "Bus high, read signal.
WrL~                        PIN 6;          "Bus low, write signal.
WrH~                        PIN 32;         "Bus high, write signal.

"*************************************************************************
"* Resources enable signals.                                            *
"*************************************************************************
RamEn~                      PIN 63;         "On board RAM enable signal.
FlashEn~                    PIN 62;         "On board FLASH enable signal.
EepromEn~                   PIN 64;         "On board EEPROM enable signal.
EepromWriteEnable~          PIN 65;         "EEPROM write enable From DUART port.
DuartEn~                    PIN 66;         "On board DUART enable signal.
AdiEn~                      PIN 43;         "On board RAM enable signal.

"*************************************************************************
"* Processor type selection.                                            *
"*************************************************************************
ChipType                    PIN 58;         "Processor type selection.

"*************************************************************************
"* DTACK generator signals.                                             *
"*************************************************************************
Dtack~                      PIN 33;         "DTACK~ to the processor;
DtackOe                     NODE;           "DtackOe, controls Dtack~ output
                              "enable. It is a logical inverse of
                              "the Dtack~ signal, with a delay of
                              "about 15nsec. This way, Dtack~
                              "output enable, stayes active 15nsec
                              "after Dtack~ posage so an active
                              "pullup behavior is obtained.

"*************************************************************************
"* Reset control.                                                       *
"*************************************************************************
ExtRst~                     PIN 51;         "External reset. Reset sources are :
```

```
                                    "Power up, ADI, PCMCIA, or reset
                                    "switch.

"*********************************************************************************
"* Local interrupt controller. An NMI come from 3 sources :                     *
"*      ADI, External and abort switch.                                         *
"*********************************************************************************
ExtNmi~                       PIN 46;        "NMI input.
ExtNmiState0, ExtNmiState1 NODE;            "NMI F.F.
ExtNmiState0, ExtNmiState1 ISTYPE 'reg, buffer';
ExtNmiFlag                    NODE;          "Reports of an external NMI condition.

IntMode                       PIN 44;        "Int. mode selection. (Normal/Dedicated)

ExtNmiAck                     NODE;          "External NMI acknowledge.


"*********************************************************************************
"* Delayed CS2~ for DUART. (Postponed it after R/W~, for the DUART)             *
"*********************************************************************************
CS2~_DLY1                     NODE;          "First delay F.F.
CS2~_DLY1                     ISTYPE 'reg, neg';
CS2~_DLY2                     NODE;          "Second delay F.F.
CS2~_DLY2                     ISTYPE 'reg, neg';



"*********************************************************************************
"* Wait state logic, signals.                                                   *
"*********************************************************************************
WaitState3, WaitState2,
WaitState1, WaitState0    NODE;              "Wait state counter.

WaitState3, WaitState2,
WaitState1, WaitState0    ISTYPE 'reg, buffer';

"*********************************************************************************
"* Constant declaration.                                                        *
"*********************************************************************************
H, L, X, Z = 1, 0, .X., .Z.;
C, D, U    = .C., .D., .U.;

READ  = 1;
WRITE = !READ;

BYTE  = 0;
WORD  = !BYTE;

CS0~_ACTIVE                   = 0;
CS1~_ACTIVE                   = 0;
CS2~_ACTIVE                   = 0;
OE~_ACTIVE                    = 0;
WEL~_ACTIVE                   = 0;
WEH~_ACTIVE                   = 0;
LDS~_ACTIVE                   = 0;
UDS~_ACTIVE                   = 0;
```

```
ADI_RD~_ACTIVE                 = 0;
ADI_WR~_ACTIVE                 = 0;
AS~_ACTIVE                     = 0;
RAM_EN~_ACTIVE                 = 0;
RAM_L_CS~_ACTIVE               = 0;
RAM_H_CS~_ACTIVE               = 0;
EEPROM_EN~_ACTIVE              = 0;
EEPROM_CS~_ACTIVE              = 0;
EEPROM_WRITE_ENABLE~_ACTIVE = 0;
FLASH_EN~_ACTIVE               = 0;
FLASH_L_CS~_ACTIVE             = 0;
FLASH_H_CS~_ACTIVE             = 0;
DUART_EN~_ACTIVE               = 0;
DUART_CS~_ACTIVE               = 0;
ADI_EN~_ACTIVE                 = 0;
IAC_ACTIVE                     = 1;
EXT_NMI~_ACTIVE                = 0;
EXT_RST~_ACTIVE                = 0;
RST~_ACTIVE                    = 0;
EXT_NMI_ACK_ACTIVE             = 1;
DTACK~_ACTIVE                  = 0;


FUNC_INT_ACK_CYCLE = [1, 1, 1];
ADDR_INT_ACK_CYCLE = [1, 1, 1, 1];
EXT_NMI_INT_LEVEL  = [1, 1, 1];


NORMAL_INT_LEVEL_7    = [0, 0, 0];  "Normal mode level 7 int = 0,0,0.
DEDICATED_INT_LEVEL_7 = [0, 1, 1];  "Dedicated mode level 7 int : IPL2~=0.
INT_LEVEL_0           = [1, 1, 1];  "No active interrupt, level.


NORMAL_INT_MODE    = 1;
DEDICATED_INT_MODE = !NORMAL_INT_MODE;


Address      = [A19, A18, A17, A16];
FunctionCode = [FC2, FC1, FC0];
IntLevel     = [A3, A2, A1];
IPL~         = [IPL2~, IPL1~, IPL0~];


WaitState        = [WaitState3, WaitState2, WaitState1, WaitState0];


"Wait state macroes.
ONE_WAIT_STATE        = 1;
TWO_WAIT_STATES       = 2;
THREE_WAIT_STATES     = 3;
FOUR_WAIT_STATES      = 4;
FIVE_WAIT_STATES      = 5;
SIX_WAIT_STATES       = 6;
SEVEN_WAIT_STATES     = 7;
EIGHT_WAIT_STATES     = 8;
NINE_WAIT_STATES      = 9;
TEN_WAIT_STATES       = 10;
ELEVEN_WAIT_STATES    = 11;
TWELVE_WAIT_STATES    = 12;
THIRTEEN_WAIT_STATES = 13;
```

```
FOURTEEN_WAIT_STATES = 14;
FIFTEEN_WAIT_STATES  = 15;


"Processor types.
IMP   = 0;
LC302 = IMP+1;


"Address resolution is 64K bytes.
RAM_ADDRESS     = [0,X,X,X];
EXP_RAM_ADDRESS = [1,X,X,X];
ADI_ADDRESS     = [0,0,0,0];
EEPROM_ADDRESS  = [0,0,0,1];
DUART_ADDRESS   = [0,0,1,0];


ADI_PORT_ACCESS = ( (AdiRd~==ADI_RD~_ACTIVE) # (AdiWr~==ADI_WR~_ACTIVE) );


EXT_NMI_ACK_CYCLE = ( ExtNmiAck.fb==EXT_NMI_ACK_ACTIVE );


"***************************************************************************
"* Vector macroes.                                                        *
"***************************************************************************


"***************************************************************************
"* Enternal NMI machine states.                                           *
"***************************************************************************
ExtNmiState = [ExtNmiState1, ExtNmiState0];


WAIT_FOR_NMI_RELEASE_STATE = 0;
WAIT_FOR_NMI_INSERT_STATE  = WAIT_FOR_NMI_RELEASE_STATE + 1;
NMI_ACTIVE_STATE           = WAIT_FOR_NMI_INSERT_STATE + 1;


"***************************************************************************
"* Vector macroes.                                                        *
"***************************************************************************


"***************************************************************************
"* Equations, state diagrams.                                             *
"***************************************************************************
"*                                                                        *
"* #######                                                                *
"* #           ####    #    #     ##      #####     #     ####    #    #     ####    *
"* #          #    #   #    #    #  #       #       #    #    #   ##   #    #     *
"* #####      #    #   #    #   #    #      #       #    #    #   # #  #     ####    *
"* #          #  # #   #    #   ######      #       #    #    #   #  # #         #   *
"* #          #   #    #    #   #    #      #       #    #    #   #   ##    #     *
"* #######     ### #    ####    #    #      #       #     ####    #    #     ####    *
"*                                                                        *
"***************************************************************************


"***************************************************************************
"* Buffers control.                                                       *
"* BUSL~ enables D0-D7, BUSH~ enable D8-D15, BUSL2H enables the 302 to     *
"* connect to D8-D15 with its lines D0-D7, This feature enable the 302 to  *
"* access 16 bit bus when running in 8 bit width mode.                     *
```

```
"***************************************************************************
equations
   !BusL~    = ((BUSW==WORD) # ((BUSW==BYTE) & (WEH~==1))) & (AS~==AS~_ACTIVE) &
              ( ((CS0~==CS0~_ACTIVE) & (FlashEn~==FLASH_EN~_ACTIVE)) #
        ((CS1~==CS1~_ACTIVE) & (RamEn~==RAM_EN~_ACTIVE)) #
        ((CS2~==CS2~_ACTIVE) &
         (((Address==EEPROM_ADDRESS) & (EepromEn~==EEPROM_EN~_ACTIVE)) #
                  ((Address==ADI_ADDRESS) & (AdiEn~==ADI_EN~_ACTIVE)) #
          ((Address==DUART_ADDRESS) & (DuartEn~==DUART_EN~_ACTIVE)))) );

   !BusL2H~ = (BUSW==BYTE) & (WEH~==0) & (AS~==AS~_ACTIVE) &          "WEH~/A0
              ( ((CS0~==CS0~_ACTIVE) & (FlashEn~==FLASH_EN~_ACTIVE)) #
        ((CS1~==CS1~_ACTIVE) & (RamEn~==RAM_EN~_ACTIVE)) #
        ((CS2~==CS2~_ACTIVE) &
         (((Address==EEPROM_ADDRESS) & (EepromEn~==EEPROM_EN~_ACTIVE)) #
                  ((Address==ADI_ADDRESS) & (AdiEn~==ADI_EN~_ACTIVE)) #
          ((Address==DUART_ADDRESS) & (DuartEn~==DUART_EN~_ACTIVE)))) );

   !BusH~    = (BUSW==WORD) & (AS~==AS~_ACTIVE) &
              ( ((CS0~==CS0~_ACTIVE) & (FlashEn~==FLASH_EN~_ACTIVE)) #
        ((CS1~==CS1~_ACTIVE) & (RamEn~==RAM_EN~_ACTIVE)) #
        ((CS2~==CS2~_ACTIVE) &
         (((Address==EEPROM_ADDRESS) & (EepromEn~==EEPROM_EN~_ACTIVE)) #
                  ((Address==ADI_ADDRESS) & (AdiEn~==ADI_EN~_ACTIVE)) #
          ((Address==DUART_ADDRESS) & (DuartEn~==DUART_EN~_ACTIVE)))) );


"***************************************************************************
"* Delayed CS2~.                                                           *
"* CS2~ is delayed to generate R/W~ before CS2~, for the DUART.            *
"***************************************************************************
equations
   CS2~_DLY1.clk = CLKO;
   CS2~_DLY2.clk = CLKO;

   CS2~_DLY1.ar  = !Rst~;
   CS2~_DLY2.ar  = !Rst~;

   CS2~_DLY1 := CS2~;
   CS2~_DLY2 := CS2~_DLY1;

"***************************************************************************
"* C/S signals.                                                            *
"* C/S for RAM, FLASH, EEPROM, DUART.                                      *
"***************************************************************************
equations
   !RamLCs~    = (CS1~==CS1~_ACTIVE) & (Address==RAM_ADDRESS) &
        (RamEn~==RAM_EN~_ACTIVE) &
        (AS~==AS~_ACTIVE) &
        (IAC==!IAC_ACTIVE) &
        ( (BUSW==WORD) #
          ((BUSW==BYTE) & (WEH~==1)) ); "WEH~ is A0, in 8 bit mode.

   !RamHCs~    = (CS1~==CS1~_ACTIVE) & (Address==RAM_ADDRESS) &
```

```
                  (RamEn~==RAM_EN~_ACTIVE) &
                  (AS~==AS~_ACTIVE) &
                  (IAC==!IAC_ACTIVE) &
                  ( (BUSW==WORD) #
                    ((BUSW==BYTE) & (WEH~==0)) ); "WEH~ is A0, in 8 bit mode.


!ExpRamLCs~ = (CS1~==CS1~_ACTIVE) & (Address==EXP_RAM_ADDRESS) &
                  (RamEn~==RAM_EN~_ACTIVE) &
                  (AS~==AS~_ACTIVE) &
                  (IAC==!IAC_ACTIVE) &
                  ( (BUSW==WORD) #
                    ((BUSW==BYTE) & (WEH~==1)) ); "WEH~ is A0, in 8 bit mode.


!ExpRamHCs~ = (CS1~==CS1~_ACTIVE) & (Address==EXP_RAM_ADDRESS) &
                  (RamEn~==RAM_EN~_ACTIVE) &
                  (AS~==AS~_ACTIVE) &
                  (IAC==!IAC_ACTIVE) &
                  ( (BUSW==WORD) #
                    ((BUSW==BYTE) & (WEH~==0)) ); "WEH~ is A0, in 8 bit mode.


!FlashLCs~  = (CS0~==CS0~_ACTIVE) &
                  (FlashEn~==FLASH_EN~_ACTIVE) &
                  (AS~==AS~_ACTIVE) &
                  (IAC==!IAC_ACTIVE) &
                  ( (BUSW==WORD) #
                    ((BUSW==BYTE) & (WEH~==1)) ); "WEH~ is A0, in 8 bit mode.


!FlashHCs~  = (CS0~==CS0~_ACTIVE) &
                  (FlashEn~==FLASH_EN~_ACTIVE) &
                  (AS~==AS~_ACTIVE) &
                  (IAC==!IAC_ACTIVE) &
                  ( (BUSW==WORD) #
                    ((BUSW==BYTE) & (WEH~==0)) ); "WEH~ is A0, in 8 bit mode.


!EepromCs~  = (CS2~==CS2~_ACTIVE) & (Address==EEPROM_ADDRESS) &
                  (EepromEn~==EEPROM_EN~_ACTIVE) &
                  (AS~==AS~_ACTIVE) &
                  (IAC==!IAC_ACTIVE) &
                  ( (BUSW==WORD) #
                    ((BUSW==BYTE) & (WEH~==1)) ); "WEH~ is A0, in 8 bit mode.


!DuartCs~   = (CS2~_DLY2==CS2~_ACTIVE) &
                  ( ((Address==DUART_ADDRESS) &
                     (DuartEn~==DUART_EN~_ACTIVE)) #
                    ((Address==ADI_ADDRESS) &
                     (AdiEn~==ADI_EN~_ACTIVE) &
                     (A3==1)) ) &
                  (AS~==AS~_ACTIVE) &
                  (IAC==!IAC_ACTIVE) &
                  ( (BUSW==WORD) #
                    ((BUSW==BYTE) & (WEH~==1)) ); "WEH~ is A0, in 8 bit mode.


when (ChipType==IMP) then
    !AdiRd~      = (CS2~==CS2~_ACTIVE) &
```

```
                    (Address==ADI_ADDRESS) &
                    (A3==0) &
                    (AdiEn~==ADI_EN~_ACTIVE) &
                    (AS~==AS~_ACTIVE) &
                    (IAC==!IAC_ACTIVE) &
                    (OE~==READ) &                        "OE~ is R/W~ in the IMP.
                    ( (WEL~==LDS~_ACTIVE) #               "WEL~ is LDS~ in the IMP.
                      (WEH~==UDS~_ACTIVE) );              "WEH~ is UDS~ in the IMP.
         else
            !AdiRd~     = (CS2~==CS2~_ACTIVE) &
                    (Address==ADI_ADDRESS) &
                    (A3==0) &
                    (AdiEn~==ADI_EN~_ACTIVE) &
                    (AS~==AS~_ACTIVE) &
                    (IAC==!IAC_ACTIVE) &
                    (OE~==OE~_ACTIVE);

         when (ChipType==IMP) then
            !AdiWr~     = (CS2~==CS2~_ACTIVE) &
                    (Address==ADI_ADDRESS) &
                    (A3==0) &
                    (AdiEn~==ADI_EN~_ACTIVE) &
                    (AS~==AS~_ACTIVE) &
                    (IAC==!IAC_ACTIVE) &
                    (OE~==WRITE) &                       "OE~ is R/W~ in the IMP.
                    (WEL~==LDS~_ACTIVE) &                "WEL~ is LDS~ in the IMP.
                    ( (BUSW==WORD) #
                      ((BUSW==BYTE) & (WEH~==1)) );      "WEH~ is A0, in 8 bit mode.
         else
            !AdiWr~     = (CS2~==CS2~_ACTIVE) &
                        (Address==ADI_ADDRESS) &
                        (A3==0) &
                        (AdiEn~==ADI_EN~_ACTIVE) &
                        (AS~==AS~_ACTIVE) &
                        (IAC==!IAC_ACTIVE) &
                        (WEL~==WEL~_ACTIVE) &
                        ( (BUSW==WORD) #
                          ((BUSW==BYTE) & (WEH~==1)) );  "WEH~ is A0, in 8 bit mode.

"**************************************************************************
"* RAM, FLASH, EEPROM, read/write control.                               *
"**************************************************************************
equations
"***********************
"* Read low byte signal. *
"***********************
    when (ChipType==IMP) then
        !RdL~ = (OE~==READ) &                            "OE~ is R/W~ in the IMP.
                (WEL~==LDS~_ACTIVE) &                    "WEL~ is LDS~/DS~ in the IMP
                (Rst~==!RST~_ACTIVE) &
                (IAC==!IAC_ACTIVE);
    else
        !RdL~ = (OE~==OE~_ACTIVE) &
                (Rst~==!RST~_ACTIVE) &
```

```
                    (IAC==!IAC_ACTIVE);


"*************************
"* Read high byte signal. *
"*************************
    when (ChipType==IMP) then
        !RdH~ = (OE~==READ) &                       "OE~ is R/W~ in the IMP.
                (WEH~==UDS~_ACTIVE) &                "WEH~ is UDS~/A0 in the IMP
                (Rst~==!RST~_ACTIVE) &
                (IAC==!IAC_ACTIVE);
    else
        !RdH~ = (OE~==OE~_ACTIVE) &
                (Rst~==!RST~_ACTIVE) &
                (IAC==!IAC_ACTIVE);


"*************************
"* Write low byte signal. *
"*************************
    when (ChipType==IMP) then
        !WrL~       = (OE~==WRITE) &                 "OE~ is R/W~ in the IMP.
                (WEL~==LDS~_ACTIVE) &            "WEL~ is LDS~/DS~ in the IMP
                (Rst~==!RST~_ACTIVE) &
                (IAC==!IAC_ACTIVE) &
                !( (EepromWriteEnable~==!EEPROM_WRITE_ENABLE~_ACTIVE) &
                        (EepromCs~==EEPROM_CS~_ACTIVE) );
    else
        !WrL~       = (WEL~==WEL~_ACTIVE) &
                (Rst~==!RST~_ACTIVE) &
                (IAC==!IAC_ACTIVE) &
                !( (EepromWriteEnable~==!EEPROM_WRITE_ENABLE~_ACTIVE) &
                        (EepromCs~==EEPROM_CS~_ACTIVE) );


"*************************
"* Write high byte signal. *
"*************************
    when ( (ChipType==IMP) & (BUSW==WORD) ) then
        !WrH~ = (OE~==WRITE) &                       "OE~ is R/W~ in the IMP.
                (WEH~==UDS~_ACTIVE) &            "WEH~ is UDS~ in the IMP.
                (Rst~==!RST~_ACTIVE) &
                (IAC==!IAC_ACTIVE);
    else when ( (ChipType==IMP) & (BUSW==BYTE) ) then
        !WrH~ = (OE~==WRITE) &                       "OE~ is R/W~ in the IMP.
                (WEL~==LDS~_ACTIVE) &            "LDS~ becomes DS~ in 8 bit mode.
                (Rst~==!RST~_ACTIVE) &
                (IAC==!IAC_ACTIVE);
    else when ( (ChipType!=IMP) & (BUSW==WORD) ) then
        !WrH~ = (WEH~==WEH~_ACTIVE) &
                (Rst~==!RST~_ACTIVE) &
                (IAC==!IAC_ACTIVE);
    else
        !WrH~ = (WEL~==WEL~_ACTIVE) &                "WEL~ becomes WE~ in 8 bit mode.
                (Rst~==!RST~_ACTIVE) &
                (IAC==!IAC_ACTIVE);
```

For More Information On This Product,
Go to: www.freescale.com

```
"***************************
"* R/W~ signal.            *
"***************************
   when (ChipType==IMP) then
      BR_W~ = OE~;
   else
      BR_W~ = !OE~;


"*****************************************************************************
"* Dtack output.                                                            *
"*****************************************************************************
equations
   Dtack~.oe = DtackOe;

   when ( ((CS1~==CS1~_ACTIVE) & (RamEn~==RAM_EN~_ACTIVE)) #
          ((CS2~==CS2~_ACTIVE) & (WaitState.fb>=FIVE_WAIT_STATES) &
           (((Address==EEPROM_ADDRESS) & (EepromEn~==EEPROM_EN~_ACTIVE)) #
           ((Address==ADI_ADDRESS) & (AdiEn~==ADI_EN~_ACTIVE)))) ) then
      Dtack~ = DTACK~_ACTIVE;
   else
      Dtack~ = !DTACK~_ACTIVE;

   DtackOe = !Dtack~.fb;


"*****************************************************************************
"* Wait state counter.                                                      *
"* Up to 15 wait state with a 4 bit counter.                                *
"*****************************************************************************
equations
   WaitState.clk = CLKO;
   WaitState.ar  = AS~;

   WaitState := WaitState.fb + 1;


"*****************************************************************************
"* External NMI F.F.                                                        *
"*****************************************************************************
equations
   ExtNmiState.clk = CLKO;
   ExtNmiState.ar  = !Rst~;

state_diagram ExtNmiState
   state WAIT_FOR_NMI_RELEASE_STATE:
      if ( ExtNmi~==EXT_NMI~_ACTIVE ) then
        WAIT_FOR_NMI_RELEASE_STATE
      else
        WAIT_FOR_NMI_INSERT_STATE;

   state WAIT_FOR_NMI_INSERT_STATE:
      if ( ExtNmi~==EXT_NMI~_ACTIVE ) then
        NMI_ACTIVE_STATE
      else
        WAIT_FOR_NMI_INSERT_STATE;
```

```
      state NMI_ACTIVE_STATE:
         if EXT_NMI_ACK_CYCLE then
            WAIT_FOR_NMI_RELEASE_STATE
         else
            NMI_ACTIVE_STATE;


   equations
      ExtNmiFlag = ( ExtNmiState.fb==NMI_ACTIVE_STATE );


"*****************************************************************************
"* 302 reset.                                                               *
"*****************************************************************************
   equations
      Rst~.oe = !Rst~.fb;


      !Rst~    = (ExtRst~==EXT_RST~_ACTIVE);


"*****************************************************************************
"* 302 halt.                                                                *
"*****************************************************************************
   equations
      Halt~.oe = !Halt~.fb;


      !Halt~    = (ExtRst~==EXT_RST~_ACTIVE);


"*****************************************************************************
"* Interrupt priority.                                                      *
"*****************************************************************************
   equations
      IPL0~.oe = !IPL0~.fb;
      IPL1~.oe = !IPL0~.fb;    "IPL0~, IPL1~ have the same function. Only 2 OE
                    "functions are available per MACH block.
      IPL2~.oe = !IPL2~.fb;


      when ExtNmiFlag &
         (IntMode==NORMAL_INT_MODE) &
         (ExtRst~==!EXT_RST~_ACTIVE) then
         IPL~ = NORMAL_INT_LEVEL_7;
      else
         when ExtNmiFlag &
             (IntMode==DEDICATED_INT_MODE) &
            (ExtRst~==!EXT_RST~_ACTIVE) then
            IPL~ = DEDICATED_INT_LEVEL_7;
         else
            IPL~ = INT_LEVEL_0;


"*****************************************************************************
"* External NMI acknowledge.                                                *
"*****************************************************************************
   equations
      ExtNmiAck = ( (IntLevel==EXT_NMI_INT_LEVEL) &
                (FunctionCode==FUNC_INT_ACK_CYCLE) &
                (Address==ADDR_INT_ACK_CYCLE) &
                (AS~==AS~_ACTIVE) );
```

Freescale Semiconductor, Inc.

```
"**********************************************************************
"* Auto vector signal.                                                *
"**********************************************************************
equations
    AVEC~.oe = !AVEC~.fb;

    !AVEC~ = EXT_NMI_ACK_CYCLE & (ExtRst~==!EXT_RST~_ACTIVE);


"**********************************************************************
"* Test vectors                                                       *
"**********************************************************************
"*                                                                    *
"* #######                                                            *
"*    #      ######   ####    #####                                   *
"*    #      #        #           #                                   *
"*    #      #####    ####        #                                   *
"*    #      #           #        #                                   *
"*    #      #        #   #       #                                   *
"*    #      ######   ####        #                                   *
"*                                                                    *
"* #      #                                                           *
"* #      # ######   ####   #####  ####  #####   ####                 *
"* #      # #       #   #   #    # #    # #    # #   #                 *
"* #      # #####   #        #    # #    # #    # ####                 *
"*  #    #  #       #        #    # #    # #####     #                 *
"*   # #    #       #   #    #    # #    # #   #  #   #                 *
"*    #     ######   ####    #     ####  #    #  ####                  *
"*                                                                    *
"**********************************************************************
"test_vectors

end _302fmlyADS
```

C•2   ADI control logic - U3.

```
"*************************************************************************
"* ADI control logic.                                                   *
"* Controls ADI interface for the IMP side and the command converter.   *
"*************************************************************************
module _302ADIctrl
title  'ADI control logic.
        Shlomo Reches - Motorola semiconductor - March 3rd, 1994.'


"*************************************************************************
"* Device declaration.                                                  *
"*************************************************************************
U03 device 'mach110a';


"*************************************************************************
"* Pins declaration.                                                    *
"*************************************************************************
AdiA2, AdiA1, AdiA0        PIN 19, 17, 14; "ADI address.
AdsA2, AdsA1, AdsA0        PIN 9, 5, 4;    "ADS board local address.
AdsAll                     PIN 2;          "ADS all.
AdsBrk                     PIN 13;         "ADS break.
AdsRst                     PIN 26;         "ADS reset.
HostEn~                    PIN 35;         "Host enable.
HostVcc                    PIN 32;         "Host VCC.
HostAck                    PIN 29;         "Host acknowledge.
HostReq                    PIN 36;         "Host request.
IntAck                     PIN 38;         "Interrupt ack from host.
AdsGrp                     PIN 24;         "ADS group. (Not supported)

AdsSel~                    PIN 25;         "IMP ADI interface, selected.

Rst~                       PIN 6;          "Reset IMP.

HstBrkAck~                 PIN 40;         "Reset host break external F.F.

RstSwitch                  PIN 41;         "Reset IMP, switch.
PcmciaRst~                 PIN 21;         "Reset from PCMCIA.
PwrUpRst                   PIN 10;         "Reset from power up circuit.

Nmi~                       PIN 3;          "Common NMI output to the processor.
AbortSwitch                PIN 33;         "User abort switch.
ExtNmi~                    PIN 11;         "External NMI source.

HostRd~                    PIN 20;         "Enable ADS data read by host.


"*************************************************************************
"* Constant declaration.                                                *
"*************************************************************************
H, L, X, Z = 1, 0, .X., .Z.;
C, D, U    = .C., .D., .U.;

HOST_VCC_ACTIVE      = 1;
HOST_EN~_ACTIVE      = 0;
```

```
HOST_ACK_ACTIVE       = 1;
HOST_REQ_ACTIVE       = 1;
ADS_ALL_ACTIVE        = 1;
ADS_BRK_ACTIVE        = 1;
ADS_RST_ACTIVE        = 1;
INT_ACK_ACTIVE        = 1;
RST_SWITCH_ACTIVE     = 1;
PCMCIA_RST~_ACTIVE    = 0;
PWR_UP_RST_ACTIVE     = 1;
RST~_ACTIVE           = 0;
BRK~_ACTIVE           = 0;
HOST_RD~_ACTIVE       = 0;
ADS_SEL~_ACTIVE       = 0;
HST_BRK_ACK~_ACTIVE   = 0;
EXT_NMI~_ACTIVE       = 0;
ABORT_SWITCH_ACTIVE   = 1;

ADS_TO_HOST_DIRECTION = 1;
HOST_TO_ADS_DIRECTION = !ADS_TO_HOST_DIRECTION;

AdsAddr = [AdsA2, AdsA1, AdsA0];
AdiAddr = [AdiA2, AdiA1, AdiA0];


"*****************************************************************************
"* Equations, state diagrams.                                               *
"*****************************************************************************
"*                                                                          *
"* #######                                                                  *
"* #           ####    #    #   ##     #####    #    ####    #    #   ####   *
"* #          #    #   #    #  #  #    #        #   #    #   #   ##   #      *
"* #####      #    #   #    #  #   #   #        #   #    #   #  # #   ####   *
"* #          #  # #   #    #  ######  #        #   #    #   # #  #      #   *
"* #          #   #    #    #  #    #  #        #   #    #   ##   #      #   *
"* #######     ### #    ####   #    #  #        #    ####    #    #   ####   *
"*                                                                          *
"*****************************************************************************


"*****************************************************************************
"* IMP selected from ADI.                                                   *
"*****************************************************************************
equations
   !AdsSel~ = (HostVcc==HOST_VCC_ACTIVE) &
              (HostEn~==HOST_EN~_ACTIVE) &
              ( (AdsAddr==AdiAddr) # (AdsAll==ADS_ALL_ACTIVE) );


"*****************************************************************************
"* IMP reset.                                                               *
"*****************************************************************************
equations
   !Rst~ = (PcmciaRst~==PCMCIA_RST~_ACTIVE) #
           (RstSwitch==RST_SWITCH_ACTIVE) #
           (PwrUpRst==PWR_UP_RST_ACTIVE) #
           ( (AdsRst==ADS_RST_ACTIVE) &
             (AdsSel~==ADS_SEL~_ACTIVE) );
```

```
"****************************************************************************
"* Host break ack. (Reset host break external F.F.)                        *
"****************************************************************************
equations
    !HstBrkAck~ = ( Rst~==RST~_ACTIVE ) #
            ( (IntAck==INT_ACK_ACTIVE) &
              (AdsSel~==ADS_SEL~_ACTIVE) );


"****************************************************************************
"* IMP PD0-PD7 buffer direction control signal.                            *
"****************************************************************************
equations
    when ( (AdsSel~==ADS_SEL~_ACTIVE) &
           (AdsAll==!ADS_ALL_ACTIVE) &
           (HostReq==!HOST_REQ_ACTIVE) &
           (HostAck==HOST_ACK_ACTIVE) ) then
        HostRd~ = HOST_RD~_ACTIVE;
    else
        HostRd~ = !HOST_RD~_ACTIVE;


"****************************************************************************
"* ADS group. (Not supported)                                              *
"****************************************************************************
equations
    AdsGrp.oe = L;


"****************************************************************************
"* Nmi~                                                                     *
"****************************************************************************
equations
    !Nmi~ = ( (AdsBrk==ADS_BRK_ACTIVE) & (AdsSel~==ADS_SEL~_ACTIVE) ) #
            (AbortSwitch==ABORT_SWITCH_ACTIVE) #
            (ExtNmi~==EXT_NMI~_ACTIVE);


end _302ADIctrl
```

NXP

U23
MC68LC302EMU

Freescale Semiconductor, Inc.

MC68302RC

U10

XTAL is left
unconnected
in the IMP

MOTOROLA INC.

PROJECT :	302 family ADS	REV : ENG	SHEET
ENG :	Shlomo Reches	BLOCK :	2 OF 18
CHK :		DESCRIP : IMP	

Freescale Semiconductor, Inc.

MC68PM302RC

U27

MOTOROLA INC.

PROJECT :	302 family ADS
ENG :	Shlomo Reches
CHK :	

REV :	ENG
BLOCK :	Pchip
DESCRIP :	

SHEET 3 OF 18

Mounted
on a socket

U34
50MHz
CLK
OUT
8

R9
10K

VCC

U33
74ACT74

D 2
SET 4
CLK 3
CLR 1
Q 5
Q 6

EXTAL

Parts in the dashed box are an option and should not be mounted on the PC board.

GND

Y1
25MHz

C39 10pF

C44 10pF

R5
470K

R6 330

XTAL

EXTAL

GND

R10
100

C73
0.1uF

External clock
termination

CLKO

GND

R2
220

C7
0.1uF

Clock out
termination

PROJECT : 302 family ADS
ENG :   Shlomo Reches
CHK :

MOTOROLA INC.

REV : ENG
BLOCK :
DESCRIP :

SHEET 4 OF 18

Clock circuit

Freescale Semiconductor, Inc.

Power up reset

Abort debouncer.

Reset debouncer.

PCMCIA reset duration extender.

PC power control indicator.

Power indicator

User and other controls.

MOTOROLA INC.

PROJECT : 302 family ADS
REV : ENG
SHEET 5 OF 18
ENG : Shlomo Reches
BLOCK :
CHK :
DESCRIP : User and other controls.

U24

MACH220

Left pins (input signals):
BR_W 28, FC(0) 26, FC(1) 25, FC(2) 24, ADSWR 23, ADSRD 22, EPRCS 21, RMTCS 14, RMFCS 13, DPRPHCS 12, DPRMLCS 11, UARTCS 10, BRDIC 9, BWRIC 7, FLLCS 6, CST(0) 5, FLHCS 3, BRDH 2, BWRH 4

CLK0 15, AS 16, BUSWDIP 17, OE 20, LDS 49, UDS 50, EXTRST 51, IAC 54, I7

I0/CLK0, I1/CLK1, I2, I3, I4/CLK2, I5/CLK3, I6

I/O0 ... I/O47

BUSH, BUSL2H, BUSL, BWRH, DTACK, IPL(1), A(19), IPL(0), IPL(2), A(18), A(1), ADEN, NR_OE, A(2), NMI, RST, AVEC, CST(2), A(17), A(16), CHIPTYP, U18-59, J6, A(3), FLSHEN, RAMEN, EEPRMEN, EEPWREN, DUARTEN, HALT

MOTOROLA INC.

PROJECT : 302 family ADS
ENG : Shlomo Reches
CHK :
REV : ENG
BLOCK :
DESCRIP :
control logic

SHEET 6 OF 18

U3 MACH110

Pin	Signal		Pin	Signal
10	PWRUPRST		18	U3-18
11	EXTNMI		19	ADIA2
13	ADSBRK		20	HSTRD
32	HSTVCC		21	PCRST
13	I2/CLK0		24	ADSGRP
33	ABORTSW		25	ADSSEC
35	HSTEN		26	ADSRST
	I5/CLK1		27	U3-27
2	ADSALL		28	U3-26
3	NMI		29	HSTACK
4	ADSA0		30	U3-30
5	ADSA1		31	U3-31
6	EXTRST		36	HSTREQ
7	U3-7		37	U3-37
8	U3-8		38	INTACK
9	ADSA2		39	U3-39
14	ADIA0		40	BRKCLR
15	U3-15		41	RSTSW
16	U3-16		42	U3-42
17	ADIA1		43	U3-43

I/O lines: I/O0–I/O31

Freescale Semiconductor, Inc.

BA(1:23)

A(1:23)

BDSA

BDSL2H

BDSC

BR_W

BD(0:15)

D(0:15)

U13 74ACT244

Pin	Signal
A(1)	2 1A1 1Y1 18 BA(1)
A(2)	4 1A2 1Y2 16 BA(2)
A(3)	6 1A3 1Y3 14 BA(3)
A(4)	8 1A4 1Y4 12 BA(4)
A(5)	11 2A1 2Y1 9 BA(5)
A(6)	13 2A2 2Y2 7 BA(6)
A(7)	15 2A3 2Y3 5 BA(7)
	17 2A4 2Y4 3
GND	1 1G
	19 2G

U22 74ACT245

BD(0)	2 A1 B1 18 D(0)
BD(1)	3 A2 B2 17 D(1)
BD(2)	4 A3 B3 16 D(2)
BD(3)	5 A4 B4 15 D(3)
BD(4)	6 A5 B5 14 D(4)
BD(5)	7 A6 B6 13 D(5)
BD(6)	8 A7 B7 12 D(6)
BD(7)	9 A8 B8 11 D(7)
1 DIR
19 G

U26 74ACT244

A(8)	2 1A1 1Y1 18 BA(8)
A(9)	4 1A2 1Y2 16 BA(9)
A(10)	6 1A3 1Y3 14 BA(10)
A(11)	8 1A4 1Y4 12 BA(11)
A(12)	11 2A1 2Y1 9 BA(12)
A(13)	13 2A2 2Y2 7 BA(13)
A(14)	15 2A3 2Y3 5 BA(14)
A(15)	17 2A4 2Y4 3 BA(15)
GND	1 1G
	19 2G

U19 74ACT245

BD(8)	2 A1 B1 18 D(0)
BD(9)	3 A2 B2 17 D(1)
BD(10)	4 A3 B3 16 D(2)
BD(11)	5 A4 B4 15 D(3)
BD(12)	6 A5 B5 14 D(4)
BD(13)	7 A6 B6 13 D(5)
BD(14)	8 A7 B7 12 D(6)
BD(15)	9 A8 B8 11 D(7)
1 DIR
19 G

U25 74ACT244

A(16)	2 1A1 1Y1 18 BA(16)
A(17)	4 1A2 1Y2 16 BA(17)
A(18)	6 1A3 1Y3 14 BA(18)
A(19)	8 1A4 1Y4 12 BA(19)
	11 2A1 2Y1 9
	13 2A2 2Y2 7
	15 2A3 2Y3 5
	17 2A4 2Y4 3
GND	1 1G
	19 2G

U16 74ACT245

BD(8)	2 A1 B1 18 D(8)
BD(9)	3 A2 B2 17 D(9)
BD(10)	4 A3 B3 16 D(10)
BD(11)	5 A4 B4 15 D(11)
BD(12)	6 A5 B5 14 D(12)
BD(13)	7 A6 B6 13 D(13)
BD(14)	8 A7 B7 12 D(14)
BD(15)	9 A8 B8 11 D(15)
1 DIR
19 G

MOTOROLA INC.

PROJECT :	302 family ADS	REV : ENG	SHEET 8
ENG :	Shlomo Reches	BLOCK :	OF 18
CHK :		DESCRIP : Buffers	

256K word option

MOTOROLA INC.

PROJECT : 302 family ADS
ENG : Shlomo Reches
CHK :

REV : ENG
BLOCK :
DESCRIP : Memory block

SHEET 9 OF 18

MOTOROLA INC.

PROJECT : 302 family ADS
ENG : Shlomo Reches
CHK :
REV :
ENG :
BLOCK :
DESCRIP :
ADI port interface
SHEET 10 OF 18

PCVCC

C75
10uF

GND

U31
74ACT08

RN7 4.7K
RN9 4.7K
RN8 4.7K

VCC

PA(0:15)

PB(0:11)
A(1:23)
PD(0:15)

MOTOROLA INC.

PROJECT :	302 family ADS	REV :	ENG
ENG :	Shlomo Reches	BLOCK :	Logic analyzer connectors.
CHK :			
		DESCRIP :	

SHEET 14 OF 18

Freescale Semiconductor, Inc.

P11

GND

GND

GND

A32 A31 A30 A29 A28 A27 A26 A25 A24 A23 A22 A21 A20 A19 A18 A17 A16 A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1

RTS1 CTS1 CD1 TXD1 RXD1 PDI(14) PDI(13) PDI(12) PDI(11) PDI(10) PDI(9) PDI(8) PB(11) PB(10) PB(9) PB(8) PB(7) PB(6) PB(5) PB(4) PB(3) PB(2) PB(1) PB(0)

B32 B31 B30 B29 B28 B27 B26 B25 B24 B23 B22 B21 B20 B19 B18 B17 B16 B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1

PESTSCHG PCWAIT PCREG PCOE PCCE1 PCRDY PCA5 PCA4 PCA3 PCD3 PCD2 PCD1 PCD0

C32 C31 C30 C29 C28 C27 C26 C25 C24 C23 C22 C21 C20 C19 C18 C17 C16 C15 C14 C13 C12 C11 C10 C9 C8 C7 C6 C5 C4 C3 C2 C1

CD3 RTS3 CTS3 TCLK1 RCLK1 BRG1 PA(15) PA(14) PA(13) PA(12) PA(11) PA(10) PA(9) PA(8) PA(7) PA(6) PA(5) PA(4) PA(3) PA(2) PA(1) PA(0)

PA(0:15) PDI(0:15) PB(0:11)

MOTOROLA INC.

PROJECT :	302 family ADS	REV :	ENG	SHEET	15	OF	18
ENG :	Shlomo Reches	BLOCK :	ENG				
CHK :		BLOCK :	Logic analyzer expansion connector				
		DESCRIP :					

P12

A32	GND
A31	RTS1
A30	
A29	CTS1
A28	CD1
A27	TXD1
A26	RXD1
A25	
A24	PDI(14)
A23	PDI(13)
A22	PDI(12)
A21	PDI(11)
A20	PDI(10)
A19	PDI(9)
A18	PDI(8)
A17	
A16	PB(11)
A15	PB(10)
A14	PB(9)
A13	PB(8)
A12	PB(7)
A11	PB(6)
A10	PB(5)
A9	PB(4)
A8	PB(3)
A7	PB(2)
A6	PB(1)
A5	PB(0)
A4	
A3	
A2	Vcc
A1	

B32	GND
B31	
B30	
B29	
B28	
B27	
B26	DISCPU
B25	
B24	RMC
B23	BCLR
B22	
B21	PCSTSCHG
B20	PCWAIT
B19	PCREG
B18	PCCE1
B17	PCCE
B16	PCRDY
B15	PCA5
B14	PCA4
B13	PCA3
B12	
B11	PCD3
B10	PCD2
B9	PCD1
B8	PCD0
B7	
B6	
B5	
B4	
B3	
B2	EXTNMI
B1	

C32	GND
C31	CD3
C30	RTS3
C29	CTS3
C28	
C27	TCLK1
C26	
C25	RCLK1
C24	
C23	BRG1
C22	
C21	PA(15)
C20	PA(14)
C19	PA(13)
C18	PA(12)
C17	PA(11)
C16	PA(10)
C15	PA(9)
C14	PA(8)
C13	PA(7)
C12	PA(6)
C11	PA(5)
C10	PA(4)
C9	PA(3)
C8	PA(2)
C7	PA(1)
C6	PA(0)
C5	
C4	
C3	
C2	Vcc
C1	

PA(15)
PDI(15)
PB(11)

PA(15)

MOTOROLA INC.

PROJECT :	302 family ADS	REV :	ENG	SHEET	16
ENG :	Shlomo Reches	BLOCK :		OF	
CHK :		DESCRIP :	Expansion connector	18	

P9

MOTOROLA INC.

PROJECT : 302 family ADS
ENG : Shlomo Reches
CHK :

REV : ENG
BLOCK : Expansion connector
DESCRIP :

SHEET 17 OF 18

IPL(0:2)

PB(0:11)

IPL(2)
IPL(1)
IPL(0)
CTS1
CDI
FRZ
BCLK
EXTINT
RXD1
RCLK1
TCLK1
CD3

PB(11)
LDS
CTS3
DTACK
UDS
BGACK
BR
AS
HALT
BERR
OE
AVEC

PA(7)
(BOOT)

BERR

RST

PB(5)

DISCPU

3 state control
signal, (TRIS), on
the Pchip.

VCC
R25
10K

VCC
R24
1M

VCC
R4
1K

GND
R3
10K

GND
R7
2K

GND

U31
74ACT08

U36
74LS05_PWR

U36
74LS05_PWR

U35
74ACT00_PWR

U35
74ACT00_PWR

U35
74ACT00_PWR

U38
74ACT14_PWR

U38
74ACT14_PWR

PWR5V

SPARE
22V10A
U39
CK/11
I/Q10
I/O9
I/O8
I/O7
I/O6
I/O5
I/O4
I/O3
I/O2
I/O1

SPARE
22V10A
U9
CK/11
I/Q10
I/O9
I/O8
I/O7
I/O6
I/O5
I/O4
I/O3
I/O2
I/O1

MOTOROLA INC.

PROJECT :	302 family ADS	REV :	ENG
ENG :	Shlomo Reches	BLOCK :	
CHK :		DESCRIP :	Pullups and spare gates.

SHEET 18 OF 18

1.0   M68302 FADS Addendum to CPU32BUG Debug Monitor User's Manual:

1.1   Overview

The following is an addendum to the CPU32BUG Debug Monitor User's Manual. The CPU32Bug debugger (V0.4) for the M68302 Family Application Development System is almost compatible with the M68CPU32BUG Debug Monitor. The CPU32BUG Debug Monitor User's Manual is the primary documentation. The version of the CPU32Bug that runs on the M68302 FADS board is referred to as the FADSbug. The CPU32Bug or FADSbug is sometimes used as abbreviations to refer to the debugger that runs on the M68302 FADS board.

   NOTE: The debugger prompt on the M68302FADS board will indicate which part is plugged into the board:

   LCIMPbug> is used for the MC68LC302

   PCHIPbug> is used for the MC68PM302

   ENIMPbug> is used for the MC68EN302

   IMPbug> is used for the MC68302.

   NOTE: The FADSbug debugger uses locations through 0x8000. You should always start any programs you run in DRAM from address 0x8000.

1.2   Changes to the FADSbug

There are some enhancements and some changes to the FADSbug software and a brief summary is included below.

1.2.1   Features that are in M68CPU32Bug but not currently on the FADSbug version running on the M68302 FADS board

1.2.1.1   SD:

The SD - Switch Directories feature is not included with the FADSbug version running on the M68302 FADS board.

1.2.1.2   PA/NOPA:

The Printer Attach/Detach feature is not included with the FADSbug version running on the M68302 FADS board.

1.2.1.3   PF:

The Port Format feature is not included with the FADSbug version running on the M68302 FADS board.

1.2.1.4   TM:

The Transparent Mode feature is not included with the FADSbug version running on the M68302 FADS board.

1.3 Enhancements to CPU32Bug (FADSbug) running on the MC68360FADS board

1.3.1 HELP Command:

The enhancement consists in adding the syntax of each individual help.

For example:

```
   >HE GT
    GT     Go to Temporary Breakpoint
    GT <ADDR> [:<COUNT>]
```

The syntax is taken as is from the Manual. This enhancement is not
displayed when the full list of help commands is requested (i.e., when
you type: HE)

1.3.2 Macro Expansion Listing Status (MALS):

This allows the user to interrogate whether the debugger Macro
Expansion is currently on or off (the output is: ON or OFF). Remember that MAL and
NOMAL set/reset the macro expansion.

1.3.3 MAP Command

This is a menu driven direct interface to the registers of the MC68XX302 Family Periph-
erals.

1.3.3.1 Exit

The board exits the user-interface module and enters the FADSbug monitor/debugger.

1.3.3.2 Flip r/w

When set (a highlighted '+' sign appears), chosen structures are read and then written;
if not set, the structures are just read.

1.3.3.3 SCC tx

The two SCC transmit BD tables are displayed in a concise format.

1.3.3.4 SCC rx

The two SCC transmit BD tables are displayed in a concise format.

1.3.3.5 General Purpose DMA

All registers of the general purpose DMA are displayed. These are the DMA mode
register, status register, source and destination addresses, count and FC registers.

Freescale Semiconductor, Inc.

1.3.3.6 Chip Select

All registers of the chip select block are displayed. These are the base and option registers for each of the four chip select areas.The debugger gives you the felixility to write to all registers and chip selects(CS) . Be careful not to write to an area you do not want to change. For example, you can modify CS0, CS!, and CS2, however it is suggested that you do not modify these chip selects on the FADS board because these chip selects are used to decode memory on the FADS board.

1.3.3.7 Parallel I/O

All registers of the parallel I/O block are displayed. These are the control, direction and data registers for both parallel port A, B and D (D in the Pchipbug only).

1.3.3.8 Interrupt Controller

All registers of the interrupt controller block are displayed. These are the interrupt mode, pending, in-service and mask registers.

1.3.3.9 Timers

The registers of the timer block are displayed. They are the mode, status, reference, capture and count registers for timers 1 and 2, and the reference and count registers for the watchdog timer.

1.3.3.10 SCC1 registers

This option displays all the registers associated with SCC channel 1. These are the channel's configuration, mode, sync, event, mask, and status registers

1.3.3.11 SCC2 registers

This option displays all the registers associated with SCC channel 2. These are the channel's configuration, mode, sync, event, mask, and status registers

1.3.3.12 SCC3 registers (Pchipbug ONLY)

This option displays all the registers associated with SCC channel 3. These are the channel's configuration, mode, sync, event, mask, and status registers

1.3.3.13 PCMCIA registers (Pchipbug ONLY)

This option displays all the registers associated with PCMCIA interface.

1.3.3.14 PCMCIA CARD registers (Pchipbug ONLY)

This option displays all the registers associated with PCMCIA Card Configuration interface.

1.3.3.15 16550 registers (Pchipbug ONLY)

This option displays all the registers associated with 16550 emulation.

1.3.3.16 MC68XX302 Commands

This option may be used to writing commands directly to the command register of the MC68XX302. By entering '?', a help screen containing all defined commands is displayed; this screen may be scrolled and the desired command selected from this menu.

1.3.3.17 SCP block

This option displays the data structures associated with the SCP
(Serial Communications Port).

1.3.3.18 SMC block

This option displays the data structures associated with the SMC
(Serial Management Controller).

1.3.3.19 Serial interface

This option displays the data structures associated with the serial
interface block.

1.3.3.20 DRAM Refresh

This option displays the parameters associated with this block. Note that they are overlaid
on the last two buffer descriptors in the SCC2 Transmit BD Table. If you wish to use the
DRAM Refresh Controller, you must only use up to 6 BD's in this table

1.3.4 <u>Symbols:</u>

The user can define (currently up to 30) symbols which are equated to
values of expressions, in the following manner:

```
SYM [<symbol>] [<expression>]
NOSYM [<symbol>]
```

Options:
    (1) SYM
        displays all defined symbols and their value both in
        hexadecimal and in decimal).
    (2) SYM <symbol>
        displays the symbol and its value.
    (3) SYM <symbol> <expression>
        evaluates the expression and gives its value to the
        symbol.

Usage:
    The symbols can be used at the debugger's command level
    and with the in-line assembler.

The symbol name is limited to 30 characters in length.

Example:
    (1) FADSbug> SYM DPRBASE 20000
        FADSbug> SYM REGBASE DPRBASE+1000
        FADSbug> SYM PEPAR REGBASE+16
        FADSbug> md PEPAR

    (2) FADSbug> mm 410;di
       00000410 1000 move.b d0,d0 ? move.l #pepar, d0
       FADSbug>md 410:1 ; di
       00000410 203c0002 1016 MOVE.L   #$21016,D0

For More Information On This Product,
Go to: www.freescale.com

1.3.5 Reprogramming the Flash EPROMS:

There is a new command added that enables users to reprogram
the flash EPROMs on the FADS board.

The command is called LOF.
Syntax:

"LOF <range> <addr>[;B|W|L]"

The LOF command loads from the memory addresses defined by
<range> to another place in the flash memory, beginning at <addr>.

The option field is only allowed when <range> is specified using a
count. In this case, the B, W, or L defines the size of data to
which the count is referring. For example, a count of four with
an option of L would mean to load four long words (or 16 bytes)
from DRAM to the <addr> location in flash. If the range beginning
address is greater than the end address an error results. An error
also results if an option field is specified without a count in
the range.

In order to erase and program the flash 12v is needed. If 12v is
not applied to the board an error results.

Example:

Suppose you want to load your own program to the flash starting
at address (0x200000). The program length is about 0x20000 bytes.

The program must be linked to address 0x200000. The next step is
to load that program to the DRAM. There are two ways to do it:
    1) Use the parallel port (pdl)
    or
    2) Use the serial port. (lo)
Since the program was linked to 0x200000 you must use the LO
command (serial load) or the PDL command (parallel load) with
their offset option to load it to an address in the DRAM. For
 example with the parallel load the command executed is:

    > psrec -sub 1F0000 program.srx 100

The last step is to type the LOF command. For example if the
 program was loaded to 0x10000(DRAM address) then the
 starting address location of the program is at 0x200000, so
 the ending address would be approximately 0x220000. The
 starting address of flash memory is 200000, so the LOF command is:

    > lof 10000 30000 200000

1.4 Changes to the CPU32Bug running on the M68360FADS board

1.4.1 Macros:

The user can define 8 macros of 128 chars each, instead of an unlimited
 number of macros when the total number of characters is only 512 chars.

1.4.2   Addressing Modes:

There have been some minor changes in the assembler addressing modes
format.

The following addressing modes :

- An+Xn+bd
- .PC+Xn+addr

can be represented in the CPU32Bug in the two following syntaxes:

   I: (bd,An,Xi) - (addr,PC,Xi)

   II: bd(An,Xi) - addr(PC,Xi)

In the CPU32Bug running on the M68360FADS board (FADSbug), only the second
syntax is acceptable.

The CPU32Bug running on the M68360FADS board also allows you to
skip or omit entries with the following addressing modes:
- Address register indirect with index, base displacement.
- Program counter indirect with index, base displacement.

The M68CPU32BUG does not.

1.5   Differences between 302bug and CPU32bug compliant debuggers:

The TRAP routines are different fromthe 302bug to the new debuggers (LCIMPbug, PCHIPbug, IMPbug, and EN-IMPbug). The TRAP #15 handler which allows system calls from the user programs has a different format than for 302bug. Refer to the chapter on System Calls in the CPU32 Debug Monitor User's Manual for details on how to format the TRAP command in the user program.

1.6   Chip Selects and Registers

The debugger gives you the felixibility to write to all registers and chip selects(CS) . Be careful not to write to an area you do not want to change. For example, you can modify CS0, CS!, and CS2, however it is suggested that you do not modify these chip selects on the FADS board because these chip selects are used to decode memory on the FADS board.