M68ICS08ASAZUM/D

M68ICS08ASAZ

In-Circuit Simulator

User's Manual

MOTOROLA

Important Notice to Users

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

Trademarks

This document includes these trademarks:

Motorola and the Motorola logo are registered trademarks
of Motorola, Inc.

Windows and Windows 95/98/NT are registered trademarks of Microsoft Corporation in the U.S. and other countries.

Intel is a registered trademark of Intel Corporation.

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

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

Table of Contents

Section 1. General Information

Section 2. Preparation and Installation

Section 3. Support Information

Section 4. Using the MON08 Interface

Appendix A. S-Record Information

Appendix B. Quick Start for System Devlopment

Lists of Figures

Lists of Tables

Section 1. General Information

1.1 Introduction

This section provides general information about the Motorola M68ICS08ASAZ in-circuit simulator (ASAZICS)(Figure 1-1).

The M68ICS08ASAZ In-Circuit Simulator Kit (ASAZICS Kit) is a stand-alone development and debugging tool. It contains the hardware and software needed to develop and simulate source code and to program Motorola’s MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A microcontrollers (MCU).

The ASAZICS board and ASAZICS software form a complete editor, assembler, programmer, simulator, and limited real-time input/output emulator for the MCU. When connection is made between a host PC (personal computer) and target hardware (your prototype product), actual inputs and outputs of the target system may be used during code simulation.

Depending on the software, the uses of the ASAZICS development package are:

- The WINIDE and CASM08Z software may be used as editor and assembler.
- With ICS08ASAZZ, the ASAZICS is used as a simulator.
- With the PROG08SZ software, the ASAZICS is used to program MCU FLASH memory.
- With the ICD08SZ software, the ASAZICS is used as a limited real-time emulator.

The ASAZICS can interface with any IBM® Windows 95/98/NT®-based computer (or later version) through connection of a single RS-232 serial port using a DB-9 serial cable.

Connection to the target system is accomplished by a Motorola M68CBL05C flex cable, or a MON08 cable. The flex cable is used when an MCU is resident on the ASAZICS for emulation or simulation. And the MON08 cable is used to debug or program a target system's MCU directly, when the MCU resides on the target hardware.

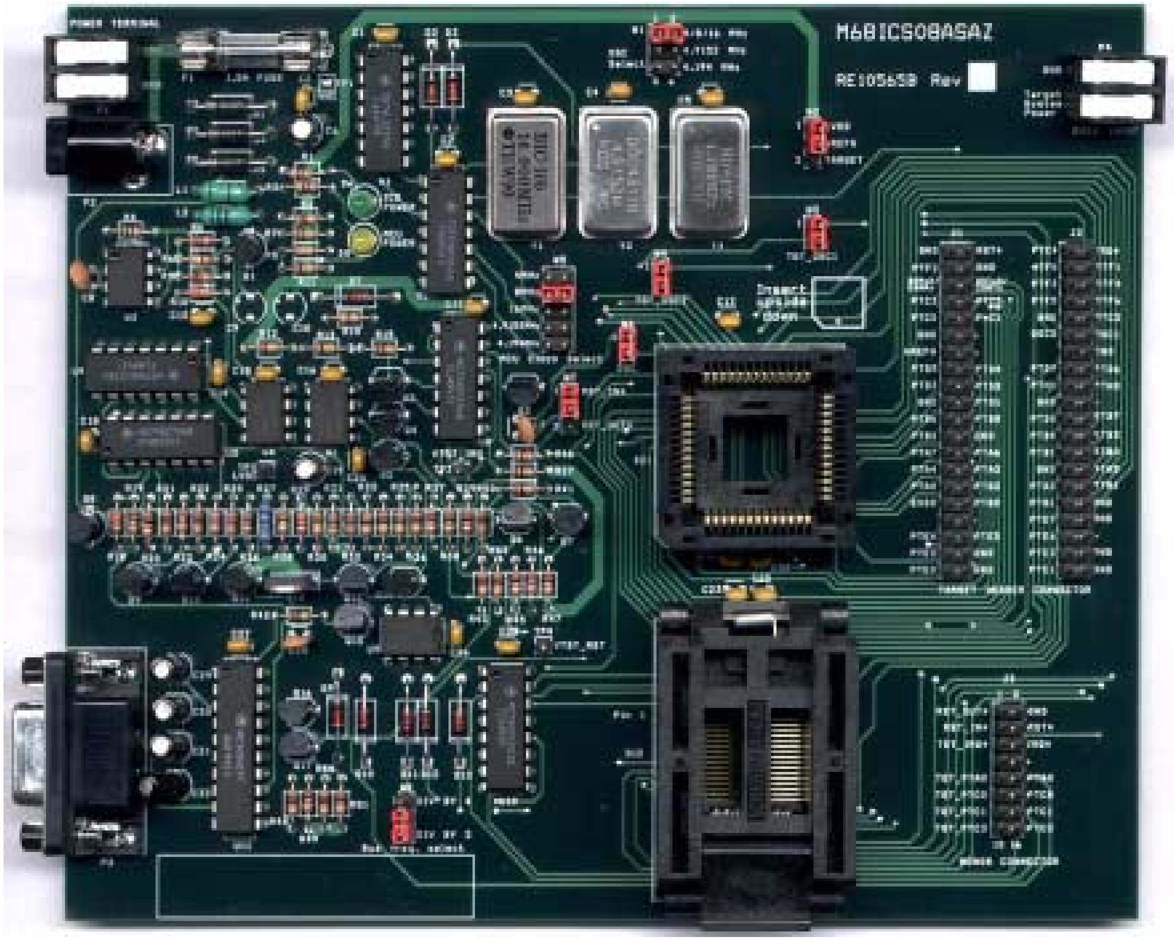


Figure 1-1. M68ICS08ASAZ (ASAZICS) In-circuit Simulator Board

1.2 M68ICS08ASAZ ICS Kit Components

The complete M68ICS08ASAZ In-Circuit Simulator Kit includes hardware, software, and documentation. **Table 1-1** lists the M68ICS08ASAZ In-Circuit Simulator Kit product components.

Table 1-1. M68ICS08ASAZ In-Circuit Simulator Kit Product Components

Part Number	Description
ICS08ASAZ	ASAZICS software development package
ICS08ASAZZ	ASAZICS simulator
ICD08SZ	ASAZICS debugger
MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A	MCU
KRISTA 22-122	Serial cable
FRIWO 11.8999-P5	ASAZICS Power supply
M68ICS08ASAZ (ASAZICS)	ASAZICS Hardware board
M68ICS08SOM/D	In-circuit simulator software operator’s manual
M68ICS08ASAZUM/D	In-circuit simulator hardware operator’s manual

1.2.1 ASAZICS Hardware

Table 1-2 lists the ASAZICS hardware components.

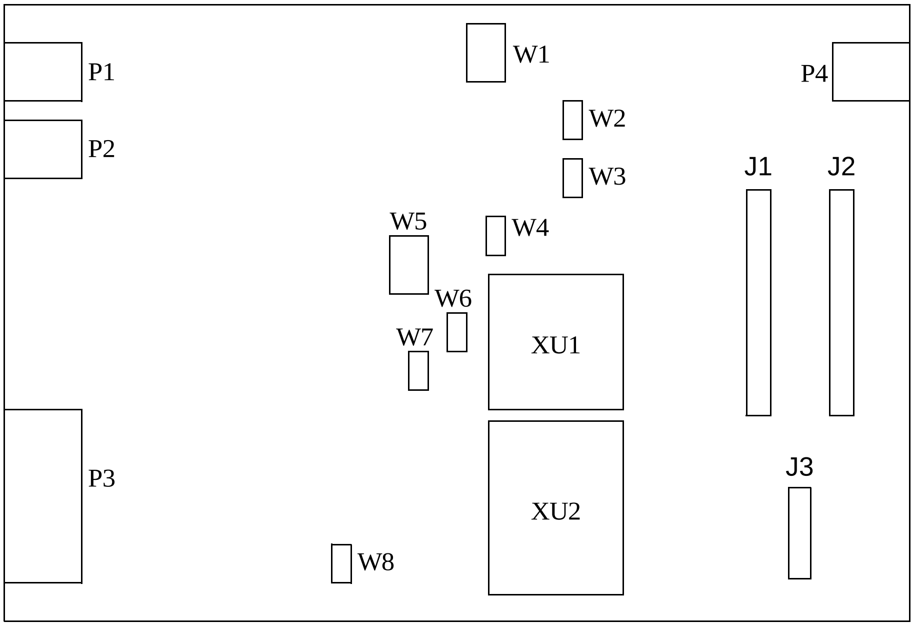


Figure 1-2. Board layout of M68ICS08ASAZ

Table 1-2. Hardware Connector Components

Components	Description
XU1	Test socket for the Motorola MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A MCU: 52-pin PLCC (Plastic Leaded Chip Carrier)
XU2	Test socket for the Motorola MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A MCU: 68-pin QFP (Quad Flat Pack)
J1 & J2	Two 2-row × 20-pin, 0.1-inch spacing connectors to connect the ASAZICS to a target using the M68CBL05C flex cable
J3	One 2-row × 8-pin, 0.1-inch spacing connector to connect to a remote target via the MON08 debug circuit.
P1	Power Terminal
P2	+5 Vdc input voltage (V_{DD})

Table 1-2. Hardware Connector Components

Components	Description
P3	RS-232 to interface the ASAZICS to the host serial connector

1.2.2 ASAZICS Software

Windows-optimized software components are referred to, collectively, as the ASAZICS software (part number ICS08ASAZ). It is a product of *P&E Microcomputer Systems, Inc.*, and is included in the ASAZICS kit **(Table 1-3).**

Table 1-3. Software Components

Components	Description
WINIDE32.EXE	Integrated development environment (WINIDE) software interface for editing and performing software or in-circuit simulation
CASM08Z.EXE	CASM08Z command-line cross-assembler
ICS08ASAZZ.EXE	In-circuit/stand-alone simulator software for the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A MCU
PROG08SZ.EXE	FLASH/EEPROM memory programming software
ICD08SZ.EXE	In-circuit debugging software for limited, real-time emulation

1.3 Hardware and Software Requirements

The ASAZICS software requires this minimum hardware and software configuration:

- An IBM-compatible host computer running Windows 95/NT or later version
- Approximately 2 Mbytes of available random-access memory (RAM) and 5 Mbytes of free disk space
- An RS232 serial port for communications between the ASAZICS and the host computer

1.4 Specifications

Table 1-4 summarizes the ASAZICS hardware specifications.

Table 1-4. ASAZICS Board Specifications

Characteristic	Specification
Temperature: Operating Storage	 0° to 40°C –40° to +85°C
Relative humidity	0 to 95%, non-condensing
Power requirement	+5 Vdc, from included ac/dc adapter

1.5 About This Manual

The procedural instructions in this manual assume that the user is familiar with the Windows interface and selection procedures.

1.6 Customer Support

To obtain information about technical support or ordering parts, call the Motorola help desk at 800-521-6274.

Section 2. Preparation and Installation

2.1 Introduction

This section provides information and instruction for configuring, installing, and readying the M68ICS08ASAZ (ASAZICS) for use.

2.2 Hardware Preparation

This paragraph explains:

- Limitations of the ASAZICS
- Configuration of the ASAZICS
- Installing the hardware
- Installing the software
- Connection the board to a target system

ESD CAUTION: *Ordinary amounts of static electricity from clothing or the work environment can damage or degrade electronic devices and equipment. For example, the electronic components installed on the printed circuit board are extremely sensitive to electrostatic discharge (ESD). Wear a grounding wrist strap whenever handling any printed circuit board. This strap provides a conductive path for safely discharging static electricity to ground.*

2.2.1 ASAZICS Limitations

These sub-paragraphs describe system limitations of the ASAZICS.

2.2.1.1 Port A0

Port A0 is used for host to MCU communications, so it is unavailable for emulation.

2.2.1.2 DDRA Bit-0 to 1

Setting DDRA bit-0 to 1 will stop communications with the simulation or debugger software and will require a system reset to regain communication with the MCU.

2.2.1.3 Port bits PTC0, PTC1, and PTC3

Port bits PTC0, PTC1, and PTC3 are temporarily disconnected from the target system during reset.

2.2.1.4 RST signal*

RST* signal is limited because the signal is not a bidirectional, open-drain signal. It is emulated as either an input or output when using the target connectors or as two pins (one input and one output) when using the MON08 cable.

2.2.2 Configuring ASAZICS Jumper Headers

Eight jumper headers (W1 - W8) (**Table 2-1**) on the ASAZICS are used to configure the hardware options. Refer to **Figure 1-2** for the position of jumpers.

CAUTION: *The ASAZICS can be set to operate at a variety of voltages. When configuring the ASAZICS jumper headers, care must be exercised to ensure that the voltages selected for the board match those of the target device. Failure to do so can result in damage to either or both of the pieces of equipment.*

Table 2-1. ASAZICS Jumper Header Description

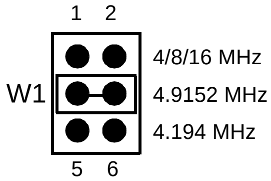
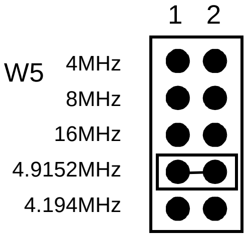
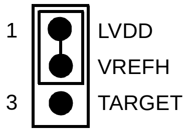
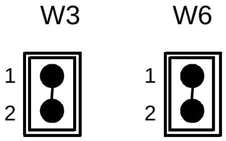
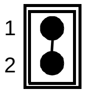
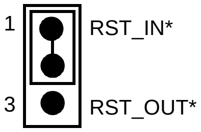
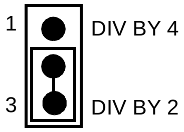
Jumper Header	Type (Factory Default Shown)	Description		
W1 OSC Select W5 MCU Clock Select	 	W1	W5	
		1-2	1-2	The MCU and the target board clock signals are supplied by oscillator Y1. The clock frequency is set to 4MHz.
		1-2	3-4	The MCU and the target board clock signals are supplied by oscillator Y1. The clock frequency is set to 8MHz.
		1-2	5-6	The MCU and the target board clock signals are supplied by oscillator Y1. The clock frequency is set to 16MHz.
		3-4	7-8	The MCU and the target board clock signals are supplied by oscillator Y2. The clock frequency is set to 4.9152MHz.
		3-4	9-10	The MCU and the target board clock signals are supplied by oscillator Y3. The clock frequency is set to 4.194MHz.
W2 VREFH		Jumper on pins 1 and 2 (default): Selects on-board V_{DD} as ADC reference high signal. Jumper on pins 2 and 3: Selects VREFH signal from target as ADC reference high signal.		
W3 TGT_OSC1 W6 OSC		W3	W6	
		On	On	The ASAZICS MCU and the target board clock signals are supplied by the on-board oscillators (Y1, Y2 or Y3, see W1 and W5 for clock signal setting) (default).
		On	Off	The ASAZICS MCU clock signal is supplied by the target board.
		Off	On	The ASAZICS MCU clock signal is supplied by the oscillators (Y1, Y2 or Y3, see W1 and W5 for clock signal setting) and the target board has its own clock signal.
		Off	Off	No clock supplied to ASAZICS MCU

Table 2-1. ASAZICS Jumper Header Description

Jumper Header	Type (Factory Default Shown)	Description
W4 TGT_OSC2		No Jumper: Disconnects OSC2 input to OSC2 input of target adapter Jumper: Connects OSC2 input to OSC2 input of target adapter (Default).
W7 Target Reset Select		Jumper on pins 1 and 2 (default): The target board reset signal initiates resets to the ASAZICS on-board MCU. Jumper on pins 2 and 3: The MCU’s reset signal initiates resets to the target system.
W8 Bus Clock Frequency		Jumper on pins 1 and 2: MCU bus frequency is set to OSC/4. Jumper on pins 2 and 3 (default): MCU bus frequency is set to OSC/2.

2.2.3 Target Flex Cable Interface Connectors J1 and J2

Use connectors J1 and J2 to connect the ASAZICS to the target system via the provided target cable. Connectors J1 and J2 are two 40-pin shrouded headers. Use the optional Motorola M68CBL05B flex cable (which must be purchased separately) to interconnect the ASAZICS to the target system, via connector J1 and J2**. Table 2-2** and **Table 2-3** show the pin assignments of J1 and J2 respectively.

Table 2-2. J1 Pin Assignments

		J1		
Gnd	1	● ●	2	RST*
PTF2	3	● ●	4	Gnd
BDRxD/CANRx	5	● ●	6	BDTxD/CANTx
PTC3	7	● ●	8	PTC5
PTC0	9	● ●	10	PTC1
Gnd	11	● ●	12	Not Conected
VREFH	13	● ●	14	Not Conected
PTD5	15	● ●	16	PTD4
PTD3	17	● ●	18	PTD2
Gnd	19	● ●	20	PTD1
PTB6	21	● ●	22	PTB5
PTB2	23	● ●	24	Gnd
PTA7	25	● ●	26	PTA6
PTA4	27	● ●	28	PTA3
PTA0	29	● ●	30	PTG2
EVDD	31	● ●	32	PTG0
Not Connected	33	● ●	34	Not Connected
PTE4	35	● ●	36	PTE5
PTE2	37	● ●	38	Gnd
PTE0	39	● ●	40	Gnd

Table 2-3. J2 Pin Assignments

		J2		
PTC4	1	● ●	2	IRQ*
PTF0	3	● ●	4	PTF1
PTF3	5	● ●	6	PTF4
PTF5	7	● ●	8	PFT6
Gnd	9	● ●	10	PTC2
OSC2	11	● ●	12	OSC1
Not Connected	13	● ●	14	Gnd
PTD7	15	● ●	16	PTD6
PTH1	17	● ●	18	PTH0
Gnd	19	● ●	20	Not Connected
PTD0	21	● ●	22	PTB7
PTB4	23	● ●	24	PTB3
PTB1	25	● ●	26	PTB0
Gnd	27	● ●	28	PTA5
PTA2	29	● ●	30	PTA1
PTG1	31	● ●	32	Gnd
PTE7	33	● ●	34	Gnd
PTE6	35	● ●	36	Not Connected
PTE3	37	● ●	38	Gnd
PTE1	39	● ●	40	Gnd

2.2.4 Target MON08 Interface Connector J3

The MON08 interface connector, J3 (Table 2-5), is used when the MCU is mounted on the target. Refer to **Section 4. Using the MON08 Interface** for detailed information.

Table 2-4. J3 Pin Assignments

J3

RST_OUT*	1	●	●	2	Gnd
RST_IN*	3	●	●	4	RST*
TGT_IRQ*	5	●	●	6	IRQ*
NC	7	●	●	8	NC
TGT_PTA0	9	●	●	10	PTA0
TGT_PTC0	11	●	●	12	PTC0
TGT_PTC1	13	●	●	14	PTC1
TGT_PTC3	15	●	●	16	PTC3

2.2.5 Host Computer - ASAZICS Interconnection (P3)

The host computer to ASAZICS interface is via the single system connector P3, which is a 9-pin, D-type connector (Amp part number AMP-9726-A) (**Figure 2-1**), mounted on the top side of the board.

Connection requires the cable assembly supplied with your ASAZICS kit, a DB9-male-to-female, 6-ft. (3 m) long serial cable.

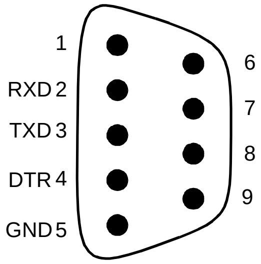


Figure 2-1. P3 Host Computer to ASAZICS Interconnection

2.2.6 Power Connector (P2)

Connect 5Vdc power directly to the ASAZICS via connector P2 (**Figure 2-2**) using the provided power supply.

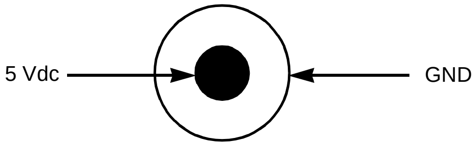


Figure 2-2. P2 Power Connector

2.3 Installing the hardware

The following steps provide installation instructions for the ASAZICS hardware.

Before beginning, locate these items:

- 9-pin RS-232 serial connector on the board, labeled P3
- 5-volt circular power-input connector on the ASAZICS

To prepare the ASAZICS for use with a host PC:

1. Install the MCU into the M68ICS08ASAZ board.

   Locate the appropriate socket on the board:

   – For PLCC, locate XU1

   – For QFP, locate XU2

   Install the MCU into this socket, observing the pin 1 orientation with the silkscreened dot.

   Note: If 64-pin QFP package MCU is used, observe the pin 1 orientation with the silkscreened dot. The top (label side) of the MCU package must be visible when looking at the component side of the board. If 52-pin PLCC package MCU is used, observe the pin 1 orientation with the silksecreened dot. The bottom of the MCU package must be visable when looking at the component side of the board.

2. Configure the jumpers W-1 through W-8 (Table 2-1) on the ASAZICS for your application.
3. Connect the board to the host PC.

   Locate the 9-pin connector labeled P3 on the board. Using the cable provided, connect it to a serial COM port on the host PC.

4. Apply power to the board.

   Connect the 5-volt power supply to the round connector on the board. Plug the power supply into an ac power outlet, using one of the country-specific adapters provided. The ICS power LED on the board should light.

ESD CAUTION: *Ordinary amounts of static electricity from clothing or the work environment can damage or degrade electronic devices and equipment. For example, the electronic components installed on the printed circuit board are extremely sensitive to electrostatic discharge (ESD). Wear a grounding wrist strap whenever handling any printed circuit board. This strap provides a conductive path for safely discharging static electricity to ground.*

2.4 Connecting the ASAZICS to the Target System

There are two ways to connect the M68ICS08ASAZ simulator board to a target system:

1. Using the MCU on the board to break its processor signals out to the target system

   This method allows the on-board MCU to control the target system's hardware. An MCU must be installed on the M68ICS08ASAZ board. The target system's MCU must be removed.

   The processor signals can be routed to the target system by an 80-pin M68CBL05C flex cable for use with the connectors labeled J1 and J2 on the board. Attach the cable to an equivalent connector on the target system. The pin assignments for this connector are given in **Table 2-2** and **Table 2-3**.

2. Using the MON08 debug interface for communication with the target system's MCU

   This method allows in-circuit FLASH programming and debugging of the target system's MCU. An MCU must be installed in the target system. The on-board MCU must be removed. Connect the board's MON08 connector with a compatible MON08 connector on the target system. Complete instructions for constructing this interface on the target board are found in **Section 4. Using the MON08 Interface.**

2.5 Installing the Software

For instructions for installing the ICS08 software, refer to *P&E Microcomputer Systems, Inc., M68ICS08 68HC08 In-Circuit Simulator Operator's Manual*, Motorola document order number M68ICS08SOM/D, and you may refer to http://www.pemicro.com/ics08/index.html#docs.

Section 3. Support Information

3.1 Introduction

This section includes data and information that can be useful in the design, installation, and operation of your application.

3.2 MCU Subsystem

The MCU subsystem consists of the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A microcontrollers, clock generation and selection, monitor mode control logic that places and holds the ASAZICS in monitor mode, the bus voltage level translation buffers, and processor operating voltage variable regulator.

The on-board MCU (the test MCU) simulates and debugs the MCU’s interface to its peripherals and to other devices on the target board through a variety of connections.

Depending on the connection, the MCU is used in one of FOUR operating modes:

- In the ASAZICS socket for simple simulation
- In the ASAZICS socket for programming
- In the ASAZICS socket connected to the target for emulation
- In the target for MON08 debug operation

3.3 ASAZICS Theory of Operation

NOTE: *For the following discussion on the theory of operation of the ASAZICS, refer to the schematic diagrams (Document number : 63A10565S).*

3.3.1 Primary 5VDC (VDD) Supply

Power input to the ASAZICS is via a standard DC power receptacle, with a 2.5mm center pin, P3. Input voltage is 5.0 Vdc.

The power is fused to prevent catastrophic failure by fuse F1 (5x20mm or 3AG fuse).

CAUTION: *Always use a fuse of the proper current and voltage rating. Failure to do so can result in serious equipment damage*

In addition, the input line has a Transient Voltage Suppression (TVS) diode to stop high voltage transients, including electrical static discharge (ESD) events, from damaging the board.

A green LED is provided to indicate that 5 Vdc is being provided to the board.

3.3.1.1 Main Switched Power to the MCU

Power to the MCU sockets and circuitry is switched on and off using the DTR signal into pin 4 of the serial interface (P3). When DTR is low, the power to the MCU sockets is off, and the yellow LED is off. When DTR is asserted high, PGML signal is asserted low to the switch circuit, and LVDD is supplied to the MCU.

The yellow LED lights when there is power to the MCU sockets.

3.3.1.2 Target Power

When the ASAZICS is configured for receiving voltage from the target, EVDD is applied to the regulator to act as the source for LVDD.

3.3.1.3 MCU Analog to Digital Converter (ADC) Power

Power to the ADC circuitry of the MCU is jumper selectable. Jumpers W2 (Table 2-1) select whether ICS board power or target board power is the source for the ADC.

NOTE: *W2 must be configured for proper power operation (Table 2-1).*

3.3.2 Serial Communications

Serial communication, half-duplex mode, to the MCU is via the DB9 connector, P3. Pin 2 is the path for transmit signals and pin 3 is for receive signals. Pin 4, the DTR signal, is utilized as an input to provide the software host with the ability to turn MCU power on and off.

Serial communication to the MCU does not occur until RST* becomes active high.

3.3.3 Clock Selection

The source of timing for the MCU may be either the ASAZICS board's clock Y1/Y2/Y3, or a clock supplied by the target. Selection is by jumper headers W3 and W6 (Table 2-1).

NOTE: *Refer to MCU manual for appropriate MCU bus clock frequency and Monitor Baud Rate. W1 and W5 must be configured together for proper MCU clock signal frequency (Table 2-1).*

3.4 ASAZICS Connector Signal Definitions

The tables in this section describe the pin assignments for the connectors on the ASAZICS board.

3.4.1 Target Flex Cable Interface Connectors J1 and J2

Table 3-1. J1 Target Flex Connector Pin Assignment Descriptions

Pin No.	Schematic	Direction	Signal Description
1	Gnd	Ground	MCU GROUND
2	RST*	I/O	LOGIC-LEVEL RESET - W7 determines if this signal is an input or output
3	PTF2	I/O	I/O PORT FROM MCU
4	Gnd	Ground	MCU GROUND
5	BDRxD/CANRx	Input	I/O PORT FROM MCU
6	BDTxD/CANTx	Output	I/O PORT FROM MCU
7	PTC3	I/O	I/O PORT FROM MCU - Signals will be disconnected by ASAZICS during reset.
8	PTC5	I/O	I/O PORT FROM MCU
9	PTC0	I/O	I/O PORT FROM MCU - Signals will be disconnected by ASAZICS during reset.
10	PTC1	I/O	I/O PORT FROM MCU - Signals will be disconnected by ASAZICS during reset.
11	Gnd	Ground	MCU GROUND
12	NC	NC	NO CONNECT
13	VREFH	Input	2.8V TO 5V - W2 determine if signal is an input from target board.
14	NC	NC	NO CONNECT
15	PTD5	I/O	I/O PORT FROM MCU
16	PTD4	I/O	I/O PORT FROM MCU
17	PTD3	I/O	I/O PORT FROM MCU
18	PTD2	I/O	I/O PORT FROM MCU
19	Gnd	Ground	MCU GROUND
20	PTD1	I/O	I/O PORT FROM MCU
21	PTB6	I/O	I/O PORT FROM MCU
22	PTB5	I/O	I/O PORT FROM MCU
23	PTB2	I/O	I/O PORT FROM MCU
24	Gnd	Ground	MCU GROUND
25	PTA7	I/O	I/O PORT FROM MCU

Pin No.	Schematic	Direction	Signal Description
26	PTA6	I/O	I/O PORT FROM MCU
27	PTA4	I/O	I/O PORT FROM MCU
28	PTA3	I/O	I/O PORT FROM MCU
29	PTA0	NC	PTA0 is used for serial communications with MCU and is not transmitted over the connector to avoid signal contention problems.
30	PTG2	I/O	I/O PORT FROM MCU
31	EVDD	Input	2.8V TO 5V - ASAZICS on-board regulator input
32	PTG0	I/O	I/O PORT FROM MCU
33	NC	NC	NO CONNECT
34	NC	NC	NO CONNECT
35	PTE4	I/O	I/O PORT FROM MCU
36	PTE5	I/O	I/O PORT FROM MCU
37	PTE2	I/O	I/O PORT FROM MCU
38	Gnd	Ground	MCU GROUND
39	PTE0	I/O	I/O PORT FROM MCU
40	Gnd	Ground	MCU GROUND

Table 3-2. J2 Target Flex Connector Pin Assignment Descriptions

Pin No.	Schematic	Direction	Signal Description
1	PTC4	I/O	I/O PORT FROM MCU
2	IRQ*	Input	LOGIC-LEVEL IRQ SIGNAL
3	PTF0	I/O	I/O PORT FROM MCU
4	PTF1	I/O	I/O PORT FROM MCU
5	PTF3	I/O	I/O PORT FROM MCU
6	PTF4	I/O	I/O PORT FROM MCU
7	PTF5	I/O	I/O PORT FROM MCU
8	PTF6	I/O	I/O PORT FROM MCU
9	Gnd	Ground	MCU GROUND

Pin No.	Schematic	Direction	Signal Description
10	PTC2	I/O	I/O PORT FROM MCU
11	OSC2	Output	I/O PORT FROM MCU
12	OSC1	I/O	CLOCK SIGNAL - W3 and W6 determine if signal is an input or an output.
13	NC	NC	NO CONNECT
14	Gnd	Ground	MCU GROUND
15	PTD7	I/O	I/O PORT FROM MCU
16	PTD6	I/O	I/O PORT FROM MCU
17	PTH1	I/O	I/O PORT FROM MCU
18	PTH0	I/O	I/O PORT FROM MCU
19	Gnd	Ground	MCU GROUND
20	NC	NC	NO CONNECT
21	PTD0	I/O	I/O PORT FROM MCU
22	PTB7	I/O	I/O PORT FROM MCU
23	PTB4	I/O	I/O PORT FROM MCU
24	PTB3	I/O	I/O PORT FROM MCU
25	PTB1	I/O	I/O PORT FROM MCU
26	PTB0	I/O	I/O PORT FROM MCU
27	Gnd	Ground	MCU GROUND
28	PTA5	I/O	I/O PORT FROM MCU
29	PTA2	I/O	I/O PORT FROM MCU
30	PTA1	I/O	I/O PORT FROM MCU
31	PTG1	I/O	I/O PORT FROM MCU
32	Gnd	Ground	MCU GROUND
33	PTE7	I/O	I/O PORT FROM MCU
34	Gnd	Ground	MCU GROUND

Pin No.	Schematic	Direction	Signal Description
35	PTE6	I/O	I/O PORT FROM MCU
36	NC	NC	NO CONNECT
37	PTE3	I/O	I/O PORT FROM MCU
38	Gnd	Ground	MCU GROUND
39	PTE1	I/O	I/O PORT FROM MCU
40	Gnd	Ground	MCU GROUND

3.4.2 Target MON08 Interface Connector J3

The MON08 interface connector, J3 (Table 2-5), is used when the MCU is mounted on the target. Refer to **Section 4 Using the MON08** for detailed information.

Table 3-3. J3 MON08 Target Connector Pin Assignment Descriptions

Pin No.	Schematic	Direction	Signal Description
1	RST_OUT*	Out	Reset signal to target
2	GND	Ground	GROUND
3	RST_IN*	In	Reset signal from target
4	RST*	Out	Reset signal to MCU
5	TGT_IRQ	Out	Interrupt request to target MCU
6	IRQ*	IN	External interrupt request
7	NC	NC	Not Connected
8	NC	NC	Not Connected
9	TGT_PTA0	I/O	General purpose I/O
10	PTA0	I/O	General purpose I/O
11	TGT_PTC0	I/O	General purpose I/O
12	PTC0	I/O	General purpose I/O
13	TGT_PTC1	I/O	General purpose I/O
14	PTC1	I/O	General purpose I/O
15	TGT-PTC3	I/O	General purpose I/O
16	PTC3	I/O	General purpose I/O

Table 3-4. Power Connector P3 Pin Assignment Descriptions

Pin No.	Mnemonic	Signal
1	VCC	+5 VDC POWER — Input voltage (+5 Vdc @ 1.0 A) from the provided power supply used by the ASAZICS logic circuits
2	GND	Common
3	GND	Common

Table 3-5. RS-232C Communication Connector P3 Pin Assignment

Pin No.	Mnemonic	Signal
2	RXD	RECEIVE DATA — Output for sending serial data to the DTE device
3	TXD	TRANSMIT DATA — Input for receiving serial data output from the DTE device
4	DTR	DATA TERMINAL READY — Input for receiving on-line/in-service/active status from the DTE device
5	GND	Common

3.5 Parts List

Table 3-6. ASAZICS Parts List

Reference Designator	Description	Manufacturer	Part Number
C1, C2, C3, C4, C5, C7, C11, C12, C13, C14, C15, C16, C18, C20, C21, C22, C23, C24, C26, C27, C28	Capacitor, 0.1uF 50V 20% mono	AVX	SR205E104MAA
C6, C19, C29, C30, C31, C32	Capacitor, 10uF 25V +80-20%	TRUTH	GP25V100M4X7
C8	Capacitor, 560pF 50V 20% ceramic	AEC	Z5U-561K
C9, C10	Capacitor, 47uF 16V +80-20%	TRUTH	GP16V470M4X7
C17	Capacitor, 10pF 50V 5% ceramic	AEC	NPO-100J
C25	Capacitor, 100pF 50V 20% ceramic	AEC	Z5U-101K
D1	Zener Transient Voltage Supressors	MOTOROLA	SA5.0
D2, D3, D7, D9, D10, D11, D12, D13	Diode	FAIRCHILD	1N4148
D4, D5	1A 20V Schottky Rectifier	MOTOROLA	1N5817
D6	LED (Green), 3mm	KINGBRIGHT	L-934GD
D8	LED (Yellow) 3mm	KINGBRIGHT	L-934YD
F1	FUSE 1.5A, 5 X 20mm Fast acting	BUSSMANN	GMA 1.5A 250V
J1, J2	Header, 2x20, 100, Target Head	MOBICON	PHDS-40G1
J3	Header, 2x8, 100, MON08	MOBICON	PHDS-16G1
L1	Inductor, 180uH	MEC	EC36-181
L2	Inductor, 10uH	MEC	EC36-100
P1, P4	Power Terminal	RDI	2SV-02
P2	Power Jack, 2.5mm	WEALTH	DS-210A

Table 3-6. ASAZICS Parts List

Reference Designator	Description	Manufacturer	Part Number
P3	Connector DB9	MOBICON	DB9SR
Q1, Q6, Q7, Q13, Q14, Q15, Q16, Q17	Transistor, NPN	MOTOROLA	BC547B
Q2, Q4, Q5	N-Channel MOSFET	MOTOROLA	BS107A
Q3	Transistor, NPN	MOTOROLA	MPS2369A
Q8	Transistor, PNP	MOTOROLA	2N3906
Q9	Transistor, NPN	MOTOROLA	2N3904
Q10	Transistor, PNP	MOTOROLA	BC557B
Q12	TMOS Power FET, N–Channel Enhancement–Mode Silicon Gate	MOTOROLA	MTD3055EL
R1, R3	Resistor, 470, 5% 0.125W	KOA	CF 1/8 471J
R2, R7, R11, R12, R13, R28, R30, R33, R34, R35, R36	Resistor, 100K, 5% 0.125W	KOA	CF 1/8 104J
R4	Resistor, 150 5% 0.125W	KOA	CF 1/8 151J
R5	Resistor, 2.6K 5% 0.125W	KOA	CF 1/8 262J
R6	Resistor, 4.9K 5% 0.125W	KOA	CF 1/8 492J
R8	Resistor, 180K 5% 0.125W	KOA	CF 1/8 184J
R9, R15, R17, R19, R20, R21, R23, R37, R38, R39, R40, R41, R43, R44, R45, R47, R48, R49, R50, R51	Resistor, 10K, 5% 0.125W	KOA	CF 1/8 103J
R10	Resistor, 3.3 5% 0.125W	KOA	CF 1/8 3.3J
R14	Resistor, 470K 5% 0.125W	KOA	CF 1/8 474J
R18, R29	Resistor, 3.3K 5% 0.125W	KOA	CF 1/8 332J
R16, R22, R25, R26, R32, R46	Resistor, 1K, 5% 0.125W	KOA	CF 1/8 102J
R24	Resistor, 2.7K, 5% 0.125W	KOA	CF 1/8 272J

Table 3-6. ASAZICS Parts List

Reference Designator	Description	Manufacturer	Part Number
R27	Resistor, 0.1 1% 0.25W	KOA	MF25D 0010F
R31	Resistor, 100 5% 0.125W	KOA	CF 1/8 101J
R42	Resistor, 1M, 5% 0.125W	KOA	CF 1/8 106J
U1	4-Bit Binary Counter	MOTOROLA	74LS293
U2	Hex Inverter	MOTOROLA	74HC04
U3	DC-to-DC converter control circuit	MOTOROLA	MC34063A
U4, U8	Non-Inverting 3-State Buffer	MOTOROLA	MC74HC125A
U5	Dual D Flip-Flop with Set and Reset	MOTOROLA	74HC74
U6, U7	Universal Voltage Monitor	MOTOROLA	MC34161
U9	Single Supply Operational Amplifier	MOTOROLA	MC33172
U10	Multiplexer/Demultiplexer	MOTOROLA	74HC4053
U11	EIA-232/V.28 CMOS Driver/Receiver	MOTOROLA	MC145407
W1	Header, 3x2, 100	3M	2380-6121TG
W2, W7,W8	Header, 3x1, 100	3M	2340-6111TG
W3, W4, W6	Header, 2x1, 100	3M	2340-6111TG
W5	Header, 5x2, 100	3M	2380-6121TG
XF1	FUSECLIP, 5 mm,	Littelfuse	100-054
XU1	52-Pin PLCC Socket	YAMAICHI	IC120-0524-207
XU2	64-Pin QFP Socket	YAMAICHI	IC51-0644-692
Y1	16MHz Oscillator	KDS	DOC-20NA 16M
Y2	4.9152MHz Oscillator	HOSONIC	HO-12C 4.9152M
Y3	4.194MHz Oscillator	KDS	DOC-20NA 4.194M

3.6 ASAZICS Board Layout

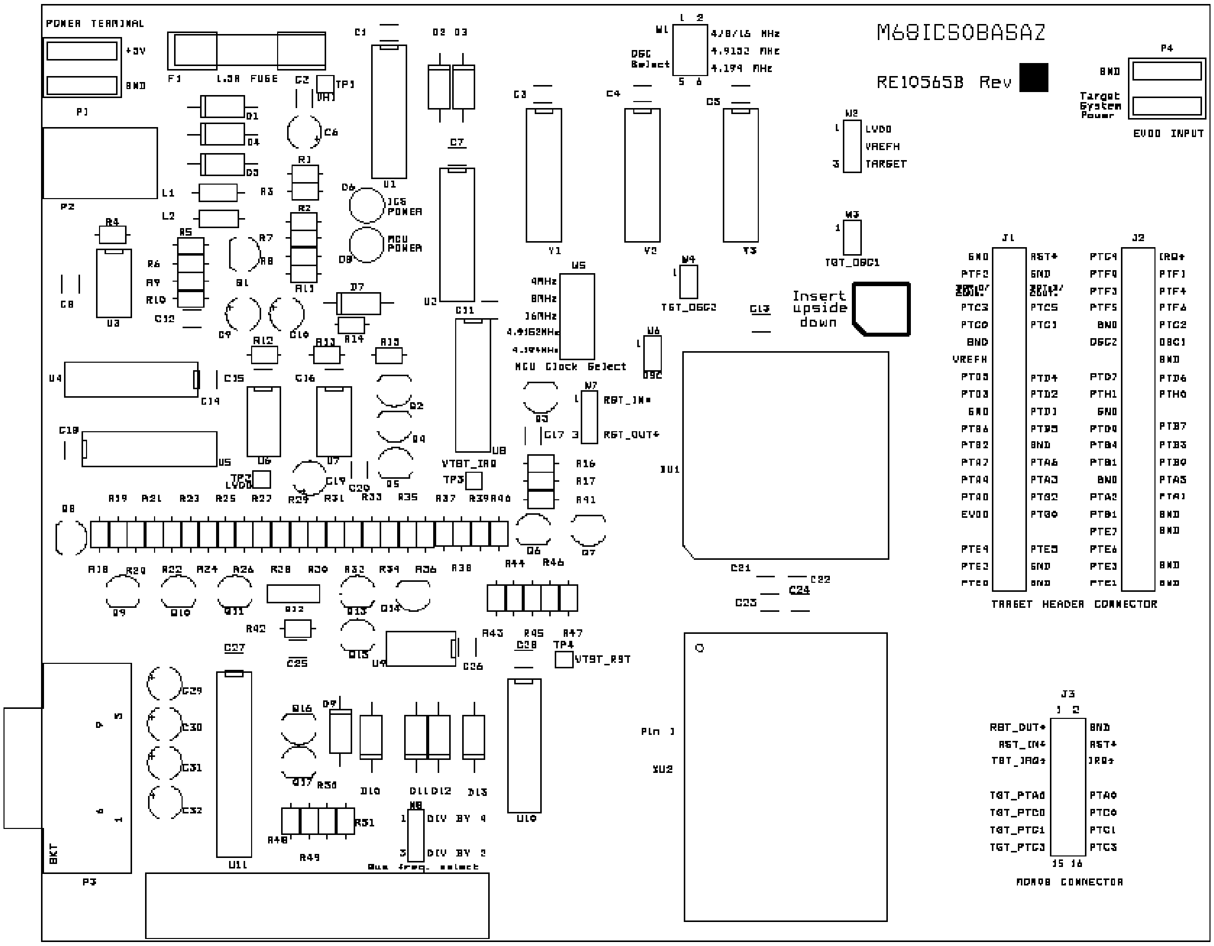


Figure 3-1. ASAZICS Board Layout

Section 4. Using the MON08 Interface

4.1 Introduction

The MON08 debugging interface may be used to debug and program a target system’s MCU directly. The target system must be connected to the ASAZICS board’s MON08 interface connector. This section explains how to connect to the MON08 interface on the target board.

4.2 Target System Header Placement and Layout

Two headers must be placed on the target board:

- P1 — 16-pin header such as Berg Electronics part number 67997-616
- P2 — 1-pin header such as Berg Electronics part number 68001-601

Table 4-1 and **Table 4-2** show the target-system interconnections for P1 and P2.

Table 4-1. MON08 Target System Connector P1

Pin No.	M68ICS08ASAZ Label	Direction	Target System Connection
1	$\overline{\text{RST-OUT}}$	Out to target	Connect to logic that is to receive the $\overline{\text{RST}}$ signal.
2	GND	Ground	Connect to ground (V_{SS}).
3	$\overline{\text{RST-IN}}$	In from target	Connect to all logic that generates resets.
4	$\overline{\text{RST}}$	Bidirectional	Connect to MCU $\overline{\text{RST}}$ pin and P1 pin 1. No other target-system logic should be tied to this signal. It will swing from 0 to +8.5 Vdc.
5	$\overline{\text{TGT-IRQ}}$	In from target	Connect to logic that generates interrupts.
6	$\overline{\text{IRQ}}$	Out to target	Connect to MCU $\overline{\text{IRQ}}$ pin. No other target-system logic should be tied to this signal. It will swing from 0 to +8.5 Vdc.
7	NC	NC	Not connected
8	NC	NC	Not connected

Table 4-1. MON08 Target System Connector P1 (Continued)

Pin No.	M68ICS08ASAZ Label	Direction	Target System Connection
9	TGT-PTA0	Bidirectional	Connect to user circuit that would normally be connected to PTA0 on the MCU. This circuit will not be connected to the MCU when the in-circuit simulator is being used.
10	PTA0	Bidirectional	Connect to MCU PTA0 pin. No other target-system logic should be tied to this signal. Host I/O present on this pin.
11	TGT_PTC0	Bidirectional	Connect to user circuit that would normally be connected to PTC0 on the MCU.
12	PTC0	Bidirectional	Connect to MCU PTC0 pin. No other target-system logic should be tied to this signal. Held at Vdd during reset and for 256 cycles after reset.
13	TGT-PTC1	Bidirectional	Connect to user circuit that would normally be connected to PTC1 on the MCU.
14	PTC1	Bidirectional	Connect to MCU PTC1 pin. No other target-system logic should be tied to this signal. Grounded during reset.
15	TGT-PTC3	Bidirectional	Connect to user circuit that would normally be connected to PTC3 on the MCU.
16	PTC3	Bidirectional	Connect to MCU PTC3 pin. No other target-system logic should be tied to this signal. Grounded during reset.

Table 4-2. MON08 Target System Connector P2

Pin No.	M68ICS08ASAZ Label	Direction	Target System Connection
1	$\overline{\text{RST}}$	Bidirectional	Connect to MCU $\overline{\text{RST}}$ pin and P2 pin 4. No other target system logic should be tied to this signal. It will swing from 0 to +8.5 Vdc.

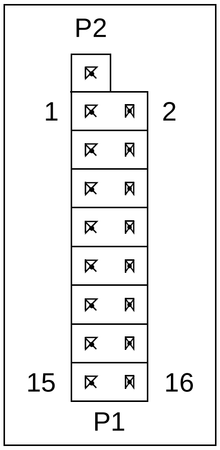


Figure 4-1. MON08 Target System Connector Layout

4.3 Connecting to the In-Circuit Simulator

Using the 16-pin cable provided with the ASAZICS kit, connect one end of the cable to the ASAZICS board at J3. Connect the other end to connector P1 on the target-system board. The pin-1 indicators on each cable end must correspond to the pin-1 indicators on the headers. P2 is not used when connecting to the ASAZICS board.

4.4 Disabling the Target-System Interface

To use the target system in a stand-alone fashion (without the ASAZICS board connected), jumper the pins on the target board’s connectors, as shown in **Figure 4-2**. This reconnects the target MCU to the appropriate circuits on the target system.

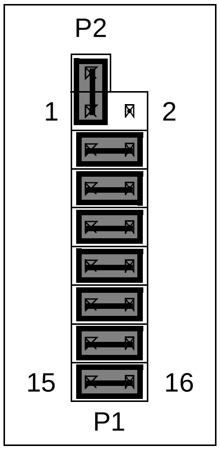


Figure 4-2. Target System Stand-Alone Connection

For production boards, a further enhancement of this scheme would be to include cutable traces between the pins of P1 and P2, as shown in **Figure 4-2**. The traces may be cut when debugging is necessary. To return the board to stand-alone use, jumpers may be installed as shown.

4.5 Sample Application

The circuit shown in **Figure 4-3** intercepts the mode select and communication signals for in-circuit debugging and programming.

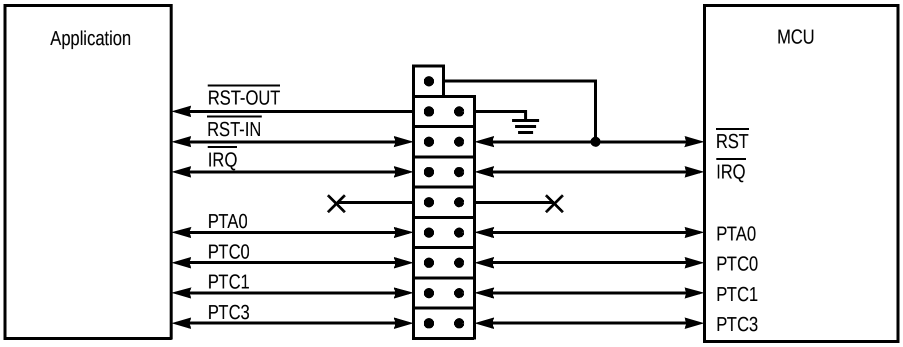


Figure 4-3. Application Designed with a Circuit for MON08

Appendix A. S-Record Information

A.1 Introduction

The Motorola S-record format was devised to encode programs or data files in a printable format for transport between computer platforms. The format also provides for editing of the S records and monitoring the cross-platform transfer process.

A.2 S-Record Contents

Each S record is a character string composed of several fields which identify:

- Record type
- Record length
- Memory address
- Code/data
- Checksum

Each byte of binary data is encoded in the S record as a 2-character hexadecimal number:

- The first character represents the high-order four bits of the byte.
- The second character represents the low-order four bits of the byte.

The five fields that comprise an S record are shown in **Table A-1**.

Table A-1. S-Record Fields

Type	Record Length	Address	Code/Data	Checksum

The S-record fields are described in **Table A-2**.

Table A-2. S-Record Field Contents

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

Each record may be terminated with a CR/LF/NULL. Additionally, an S record may have an initial field to accommodate other data such as line number generated by some time-sharing systems.

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

A.3 S-Record Types

Eight types of S records have been defined to accommodate the several needs of the encoding, transport, and decoding functions. The various Motorola upload, download, and other record transport control programs, as well as cross assemblers, linkers, and other file-creating or debugging programs, utilize only those S records which serve the purpose of the program.

For specific information on which S records are supported by a particular program, consult the user manual for the program.

NOTE: *The ICS08ASAZZ supports only the S0, S1, and S9 record types. All data before the S1 record is ignored. Thereafter, all records must be S1 type until the S9 record, which terminates data transfer.*

An S-record format may contain the record types in **Table A-3**.

Table A-3. Record Types

Record Type	Description
S0	Header record for each block of S records. The code/data field may contain any descriptive information identifying the following block of S records. The address field is normally 0s.
S1	Code/data record and the 2-byte address at which the code/data is to reside.
S2 – S8	Not applicable to ICS08ASAZZ
S9	Termination record for a block of S1 records. Address field may optionally contain the 2-byte address of the instruction to which control is to be passed. If not specified, the first interplant specification encountered in the input will be used. There is no code/data field.

Only one termination record is used for each block of S records. Normally, only one header record is used, although it is possible for multiple header records to occur.

A.4 S Record Creation

S-record format programs may be produced by dump utilities, debuggers, cross assemblers, or cross linkers. Several programs are available for downloading a file in the S-record format from a host system to an 8- or 16-bit microprocessor-based system.

A.5 S-Record Example

A typical S-record format, as printed or displayed, is shown in this example:

Example:

```
S00600004844521B
S1130000285F245F2212226A000424290008237C2A
S11300100002000800082529001853812341001813
S113002041E900084E42234300182342000824A952
S107003000144ED492
S9030000FC
```

In the example, the format consists of:

- An S0 header
- Four S1 code/data records
- An S9 termination record

A.5.1 S0 Header Record

The S0 header record is described in **Table A-4**.

Table A-4. S0 Header Record

Field	S-Record Entry	Description
Type	S0	S-record type S0, indicating a header record
Record Length	06	Hexadecimal 06 (decimal 6), indicating six character pairs (or ASCII bytes) follow
Address	00 00	4-character, 2-byte address field; zeroes
Code/Data	48 44 52	Descriptive information identified these S1 records: ASCII H D R — “HDR”
Checksum	1B	Checksum of S0 record

A.5.2 First S1 Record

The first S1 record is described in **Table A-5**.

Table A-5. S1 Header Record

Field	S-Record Entry			Description	
Type	S1			S-record type S1, indicating a code/data record to be loaded/verified at a 2-byte address	
Record Length	13			Hexadecimal 13 (decimal 19), indicating 19 character pairs, representing 19 bytes of binary data, follow	
Address	0000			4-character, 2-byte address field; hexadecimal address 0000 indicates location where the following data is to be loaded	
Code/Data	Opcode			Instruction	
	28	5F		BHCC	$0161
	24	5F		BCC	$0163
	22	12		BHI	$0118
	22	6A		BHI	$0172
	00	04	24	BRSET	0, $04, $012F
	29	00		BHCS	$010D
	08	23	7C	BRSET	4, $23, $018C
Checksum	2A			Checksum of the first S1 record	

The 16 character pairs shown in the code/data field of **Table A-5** are the ASCII bytes of the actual program.

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

A.5.3 S9 Termination Record

The S9 termination record is described in **Table A-6**.

Table A-6. S9 Header Record

Field	S-Record Entry	Description
Type	S9	S-record type S9, indicating a termination record

Table A-6. S9 Header Record

Field	S-Record Entry	Description
Record Length	03	Hexadecimal 04, indicating three character pairs (three bytes) follow
Address	00 00	4-character, 2-byte address field; zeroes
Code/Data		There is no code/data in an S9 record.
Checksum	FC	Checksum of S9 record

A.5.4 ASCII Characters

Each printable ASCII character in an S record is encoded in binary. **Table A-5** gives an example of encoding for the S1 record. The binary data is transmitted during a download of an S record from a host system to a 9- or 16-bit microprocessor-based system. For example, the first S1 record in **Table A-5** is sent as shown here.

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

Appendix B. Quick Start for System Devlopment

B.1 Introduction

This quick start guide explains the:

- Configuration of the M68ICS08ASAZ In-Circuit Simulator (ASAZICS) Kit
- Installation of software
- Installation of hardware
- Connection of the board to a target system

There are four methods for configuring the ASAZICS development system: standalone, simulation, evaluation, and programming.

- Standalone — ICS08ASAZZ.exe is running on the host computer (the ASAZICS board is not connected). Emulation of the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A MCU, registers, and I/O ports is within the host computer environment.
- Simulation — Host computer is connected to the ASAZICS board via the RS-232 cable, and the ICS08ASAZZ.exe is running on the host computer. This provides access to the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A MCU, internal registers, and I/O ports.
- Evaluation — Host computer is connected to the ASAZICS board, and the ASAZICS board is connected to the target system via the flex cable. This method provides limited real-time evaluation of the MCU and debugging user developed hardware and software.

- Programming — Host computer is connected to the ASAZICS board, and the ASAZICS board is connected to the target system via the MON08 cable. Use the PROG08SZ.exe to program the MCU FLASH/EEPROM module. In the programming mode, there is limited evaluation.

B.2 Installing the Software

The ASAZICS software requires this minimum hardware and software configuration:

- An IBM-compatible host computer running Windows 95/NT or later version
- Approximately 2 Mbytes of available random-access memory (RAM) and 5 Mbytes of free disk space
- An RS232 serial port for communications between the ASAZICS and the host computer

For instructions for installing the ICS08 software, refer to *P&E Microcomputer Systems, Inc., M68ICS08 HC08 In-Circuit Simulator Operator's Manual*, Motorola document order number M68ICS08SOM/D, and you may refer to http://www.pemicro.com/ics08/index.html#docs.

B.3 Installing the Hardware

The following steps provide installation instructions for the ASAZICS hardware.

To prepare the ASAZICS for use with a host PC:

1. Install an MCU into the M68ICS08ASAZ board.

   Locate the appropriate socket on the board:

   – For PLCC, locate XU1

   – For QFP, locate XU2

Install an MCU (provided with the ASAZICS package) into the M68ICS08ASAZ board in the appropriate socket. If 64-pin QFP package MCU is used, observing the pin 1 orientation with the silkscreened dot. The top (label side) of the MCU package must be visible when looking at the component side of the board. If 52-pin PLCC package MCU is used, observing the pin 1 orientation with the silkscreened dot. The bottom of the MCU package must be visible when looking at the component side of the board.

2. Configure the jumpers W-1 through W-8 on the ASAZICS for your application. Refer to **B.3.1 ASAZICS Configurable Jumper Headers** for details.
3. Connect the board to the host PC.

   Locate the 9-pin connector labeled P3 on the board. Using the cable provided, connect it to a serial COM port on the host PC.
4. Apply power to the board.

   Connect the 5-volt power supply to the round connector on the board, P2. Plug the power supply into an ac power outlet, using one of the country-specific adapters provided. The ICS power LED on the board lights.

ESD CAUTION: *Ordinary amounts of static electricity from clothing or the work environment can damage or degrade electronic devices and equipment. For example, the electronic components installed on the printed circuit board are extremely sensitive to electrostatic discharge (ESD). Wear a grounding wrist strap whenever handling any printed circuit board. This strap provides a conductive path for safely discharging static electricity to ground.*

B.3.1 ASAZICS Configurable Jumper Headers

Configure the eight jumper headers on the ASAZICS board for your application according to the tables in this section. Refer to **Figure B-1** for the position of jumpers

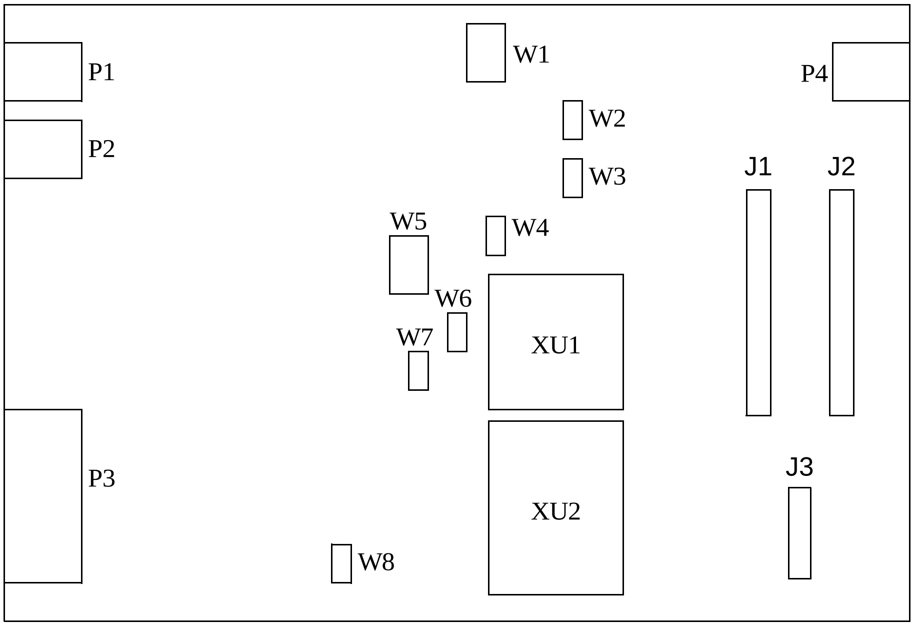
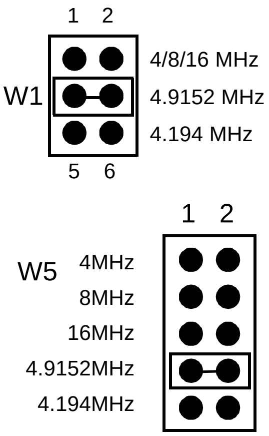
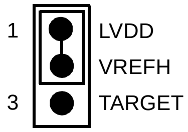
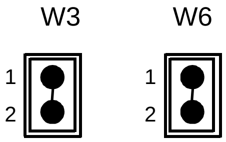
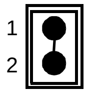
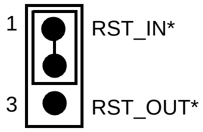
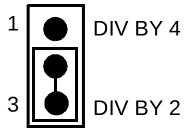


Figure B-1. Board layout of M68ICS08ASAZ

Table B-1. ASAZICS Jumper Header Description

Jumper Header	Type (Factory Default Shown)	Description		
W1 OSC Select W5 MCU Clock Select	 1 2 W1 4/8/16 MHz 4.9152 MHz 4.194 MHz 5 6 1 2 W5 4MHz 8MHz 16MHz 4.9152MHz 4.194MHz	W1	W5	
		1-2	1-2	The MCU and the target board clock signals are supplied by oscillator Y1. The clock frequency is set to 4MHz.
		1-2	3-4	The MCU and the target board clock signals are supplied by oscillator Y1. The clock frequency is set to 8MHz.
		1-2	5-6	The MCU and the target board clock signals are supplied by oscillator Y1. The clock frequency is set to 16MHz.
		3-4	7-8	The MCU and the target board clock signals are supplied by oscillator Y2. The clock frequency is set to 4.9152MHz.
		3-4	9-10	The MCU and the target board clock signals are supplied by oscillator Y3. The clock frequency is set to 4.194MHz.
W2 VREFH		Jumper on pins 1 and 2 (default): Selects on-board VDD as ADC refernce high signal. Jumper on pins 2 and 3: Selects VREFH signal from target as ADC reference high signal.		
W3 TGT_OSC1 W6 OSC		W3	W6	
		On	On	The ASAZICS MCU and the target board clock signals are supplied by the on-board oscillators (Y1, Y2 or Y3, see W1 and W5 for clock signal setting) (default).
		On	Off	The ASAZICS MCU clock signal is supplied by the target board.
		Off	On	The ASAZICS MCU clock signal is supplied by the oscillators (Y1, Y2 or Y3, see W1 and W5 for clock signal setting) and the target board has its own clock signal.
		Off	Off	No clock supplied to ASAZICS MCU
W4 TGT_OSC2		No Jumper: Disconnects OSC2 input to OSC2 input of target adapter Jumper: Connects OSC2 input to OSC2 input of target adapter (Default).		

Jumper Header	Type (Factory Default Shown)	Description
W7 Target Reset Select		Jumper on pins 1 and 2 (default): The target board reset signal initiates resets to the ASAZICS on-board MCU. Jumper on pins 2 and 3: The MCU’s reset signal initiates resets to the target system.
W8 Bus Clock Frequency		Jumper on pins 1 and 2: MCU bus frequency is set to OSC/4. Jumper on pins 2 and 3 (default): MCU bus frequency is set to OSC/2.

B.4 Connecting to a Target System

Connect the simulator board to the target system using one of these methods:

- Using a flex cable

  When emulating an MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A, or MC68HC908AZ32A MCU, connect the 80-pin M68CBL05C flex cable to the connectors labeled J1 and J2 on the simulator board. Attach the other end of the cable to the appropriate connectors on the target system. Target head adapters are available for the 64-pin QFP versions and 52-pin PLCC versions of the MCU.

- Using a MON08 cable

  Connect the MON08 debug interface cable to the MON08 debug interface connector J3 for communication with the target system’s MCU. The MON08 cable lets you program and debug the target system’s MCU FLASH/EEPROM. An MCU must be installed in the target system, and there should be no MCU installed in the ASAZICS.

Glossary

8-bit MCU — A microcontroller whose data is communicated over a data bus made up of eight separate data conductors. Members of the MC68HC908 Family of microcontrollers are 8-bit MCUs.

A — An abbreviation for the accumulator of the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A MCU.

accumulator — An 8-bit register of the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A CPU. The contents of this register may be used as an operand of an arithmetic or logical instruction.

assembler — A software program that translates source code mnemonics into opcodes that can then be loaded into the memory of a microcontroller.

assembly language — Instruction mnemonics and assembler directives that are meaningful to programmers and can be translated into an object code program that a microcontroller understands. The CPU uses opcodes and binary numbers to specify the operations that make up a computer program. Humans use assembly language mnemonics to represent instructions. Assembler directives provide additional information such as the starting memory location for a program. Labels are used to indicate an address or binary value.

ASCII — American Standard Code for Information Interchange. A widely accepted correlation between alphabetic and numeric characters and specific 7-bit binary numbers.

breakpoint — During debugging of a program, it is useful to run instructions until the CPU gets to a specific place in the program, and then enter a debugger program. A breakpoint is established at the desired address by temporarily substituting a software interrupt (SWI) instruction for the instruction at that address. In response to the SWI, control is passed to a debugging program.

byte — A set of exactly eight binary bits.

C — An abbreviation for carry/borrow in the condition codes register of the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A. When adding two unsigned 8-bit numbers, the C bit is set if the result is greater than 255 ($FF).

CCR — An abbreviation for condition code register in the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A. The CCR has five bits (H, I, N, Z, and C) that can be used to control conditional branch instructions. The values of the bits in the CCR are determined by the results of previous operations. For example, after a load accumulator (LDA) instruction, Z will be set if the loaded value was $00.

clock — A square wave signal that is used to sequence events in a computer.

command set — The command set of a CPU is the set of all operations that the CPU knows how to perform. One way to represent an instruction set is with a set of shorthand mnemonics such as LDA meaning load A. Another representation of an instruction set is the opcodes that are recognized by the CPU.

condition codes register — The CCR has five bits (H, I, N, Z, and C) that can be used to control conditional branch commands. The values of the bits in the CCR are determined by the results of previous operations. For example, after a load accumulator (LDA) instruction, Z will be set if the loaded value was $00.

CPU — Central processor unit. The part of a computer that controls execution of instructions.

CPU cycles — A CPU clock cycle is one period of the internal bus-rate clock. Normally, this clock is derived by dividing a crystal oscillator source by two or more so the high and low times will be equal. The length of time required to execute an instruction is measured in CPU clock cycles.

CPU registers — Memory locations that are wired directly into the CPU logic instead of being part of the addressable memory map. The CPU always has direct access to the information in these registers. The CPU registers in an MC68HC908 are A (8-bit accumulator), X (8-bit index register), CCR (condition code register containing the H, I, N, Z, and C bits), SP (stack pointer), and PC (program counter).

cycles — See CPU cycles.

data bus — A set of conductors that are used to convey binary information from a CPU to a memory location or from a memory location to a CPU; in the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A, the data bus is 8-bits.

development tools — Software or hardware devices used to develop computer programs and application hardware. Examples of software development tools include text editors, assemblers, debug monitors, and simulators. Examples of hardware development tools include simulators, logic analyzers, and PROM programmers. An in-circuit simulator combines a software simulator with various hardware interfaces.

EPROM — Erasable, programmable read-only memory. A non-volatile type of memory that can be erased by exposure to an ultra-violet light source. MCUs that have EPROM are easily recognized by their packaging: a quartz window allows exposure to UV light. If an EPROM MCU is packaged in an opaque plastic package, it is termed a one-time-programmable OTP MCU, since there is no way to erase and rewrite the EPROM.

EEPROM — Electrically erasable, programmable read-only memory.

H — Abbreviation for half-carry in the condition code register of the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A. This bit indicates a carry from the low-order four bits of an 8-bit value to the high-order four bits. This status indicator is used during BCD calculations.

I — Abbreviation for interrupt mask bit in the condition code register of the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A.

index register — An 8-bit CPU register in the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A that is used in indexed addressing mode. The index register (X) also can be used as a general-purpose 8-bit register in addition to the 8-bit accumulator.

input-output (I/O) — Interfaces between a computer system and the external world. For example, a CPU reads an input to sense the level of an external signal and writes to an output to change the level on an external signal.

instructions — Instructions are operations that a CPU can perform. Instructions are expressed by programmers as assembly language mnemonics. A CPU interprets an opcode and its associated operand(s) as an instruction.

listing — A program listing shows the binary numbers that the CPU needs alongside the assembly language statements that the programmer wrote. The listing is generated by an assembler in the process of translating assembly language source statements into the binary information that the CPU needs.

LSB — Least significant bit.

MCU – Microcontroller unit — Microcontroller. A complete computer system including CPU, memory, clock oscillator, and I/O on a single integrated circuit.

MSB — Most significant bit.

N — Abbreviation for negative, a bit in the condition code register of the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A. In two's-complement computer notation, positive signed numbers have a 0 in their MSB (most significant bit) and negative numbers have a 1 in their MSB. The N condition code bit reflects the sign of the result of an operation. After a load accumulator instruction, the N bit will be set if the MSB of the loaded value was a 1.

object code file — A text file containing numbers that represent the binary opcodes and data of a computer program. An object code file can be used to load binary information into a computer system. Motorola uses the S-record file format for object code files.

operand — An input value to a logical or mathematical operation.

opcode — A binary code that instructs the CPU to do a specific operation in a specific way. The MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A CPU recognizes 210 unique 8-bit opcodes that represent addressing mode variations of 62 basic instructions.

OTPROM — A non-volatile type of memory that can be programmed but cannot be erased. An OTPROM is an EPROM MCU that is packaged in an opaque plastic package. It is called a one-time-programmable MCU because there is no way to expose the EPROM to a UV light.

PC — Abbreviation for program counter CPU register of the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A.

program counter — The CPU register that holds the address of the next instruction or operand that the CPU will use.

RAM — Random access memory. Any RAM location can be read or written by the CPU. The contents of a RAM memory location remain valid until the CPU writes a different value or until power is turned off.

registers — Memory locations that are wired directly into the CPU logic instead of being part of the addressable memory map. The CPU always has direct access to the information in these registers. The CPU registers in the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A are A (8-bit accumulator), X (8-bit index register), CCR (condition code register containing the H, I, N, Z, and C bits), SP (stack pointer), and PC (program counter). Memory locations that hold status and control information for on-chip peripherals are called I/O and control registers.

reset — Reset is used to force a computer system to a known starting point and to force on-chip peripherals to known starting conditions.

S record — A Motorola standard format used for object code files.

simulator — A computer program that copies the behavior of a real MCU.

source code — See source program.

SP — Abbreviation for stack pointer CPU register in the

MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A MCU.

source program — A text file containing instruction mnemonics, labels, comments, and assembler directives. The source file is processed by an assembler to produce a composite listing and an object file representation of the program.

stack pointer — A CPU register that holds the address of the next available storage location on the stack.

TTL — Transistor-to-transistor logic.

V_{DD} — The positive power supply to a microcontroller (typically 5 volts dc).

V_{SS} — The 0-volt dc power supply return for a microcontroller.

Word — A group of binary bits. Some larger computers consider a set of 16 bits to be a word but this is not a universal standard.

X — Abbreviation for index register, a CPU register in the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A.

Z — Abbreviation for zero, a bit in the condition code register of the MC68HC908AS60/AS60A, MC68HC908AZ60/AZ60A and MC68HC908AZ32A. A compare instruction subtracts the contents of the tested value from a register. If the values were equal, the result of this subtraction would be 0 so the Z bit would be set; after a load accumulator instruction, the Z bit will be set if the loaded value was $00.

Motorola reserves the right to make changes without further notice to any products herein. Motorola makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Motorola assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters which may be provided in Motorola data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including "Typicals" must be validated for each customer application by customer's technical experts. Motorola does not convey any license under its patent rights nor the rights of others. Motorola products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Motorola product could create a situation where personal injury or death may occur. Should Buyer purchase or use Motorola products for any such unintended or unauthorized application, Buyer shall indemnify and hold Motorola and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Motorola was negligent regarding the design or manufacture of the part. Motorola and Ⓜ are registered trademarks of Motorola, Inc. Motorola, Inc. is an Equal Opportunity/Affirmative Action Employer.

How to reach us:

USA/EUROPE/Locations Not Listed: Motorola Literature Distribution, P.O. Box 5405, Denver, Colorado 80217. 1-303-675-2140
or 1-800-441-2447. Customer Focus Center, 1-800-521-6274

JAPAN: Motorola Japan Ltd.; SPS, Technical Information Center, 3-20-1, Minami-Azabu, Minato-ku, Tokyo 106-8573 Japan.
81-3-3440-8573

ASIA/PACIFIC: Motorola Semiconductors H.K. Ltd.; Silicon Harbour Centre, 2 Dai King Street, Tai Po Industrial Estate,
Tai Po, N.T., Hong Kong. 852-26668334

Mfax™, Motorola Fax Back System: RMFAX0@email.sps.mot.com; http://sps.motorola.com/mfax/;
TOUCHTONE, 1-602-244-6609; US and Canada ONLY, 1-800-774-1848

HOME PAGE: http://motorola.com/sps/

Mfax is a trademark of Motorola, Inc.

MOTOROLA