M68ICS08ABUM/D

M68ICS08AB

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 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

User’s Manual — M68ICS08AB In-Circuit Simulator

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 Hardware Configuration Guide

Glossary

Index

User’s Manual — M68ICS08AB In-Circuit Simulator

List of Figures

List of Tables

Section 1. General Information

1.1 Introduction

This section provides general information about the Motorola M68ICS08AB in-circuit simulator (ABICS)(Figure 1-1).

The ABICS board 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 MC68HC908AB32 microcontroller (MCU).

The ABICS and it's 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.

The ABICS can interface with any IBM® Windows 95®-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 ribbon cable, a Motorola M6CLB05C flex cable, or a MON08 cable. The flex cable is used when an MCU is resident on the ABICS 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.

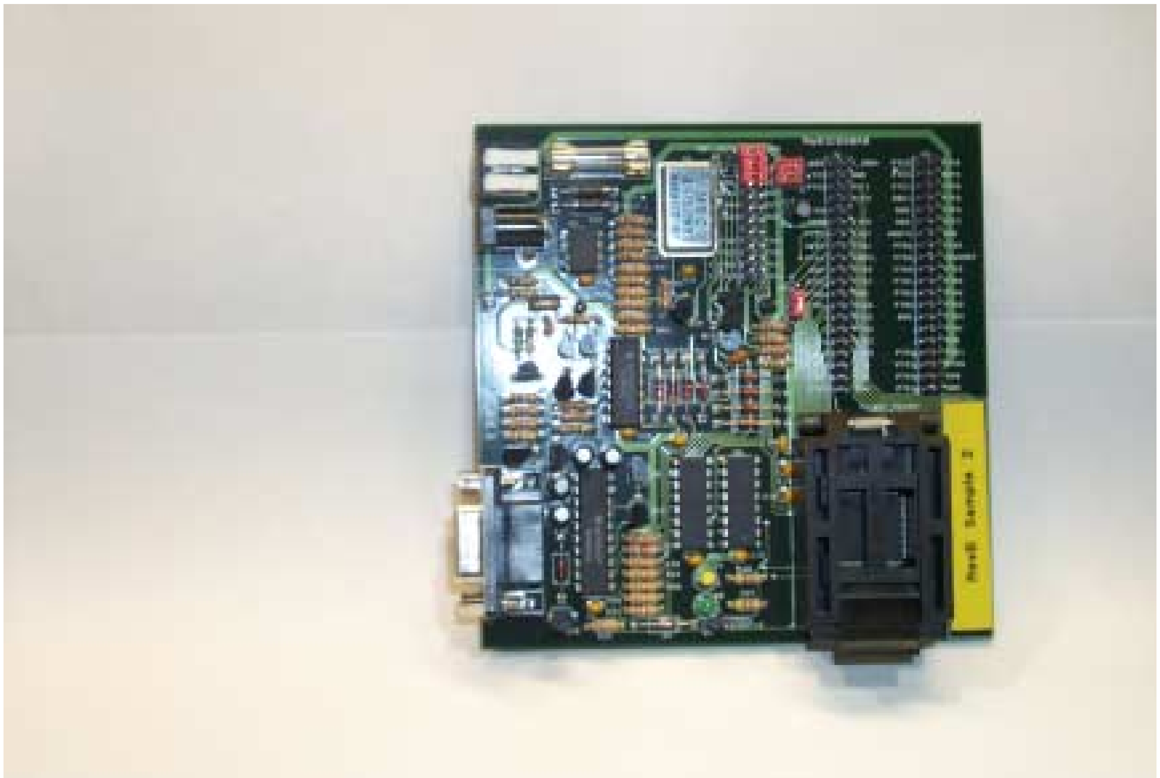


Figure 1-1 M68ICS08AB ABICS In-circuit Simulator Board

The ABICS is a low-cost development system that supports editing, assembling, in-circuit simulation, in-circuit emulation, and FLASH memory programming. Its features include:

- Editing with WINIDE
- Assembling with CASM08Z
- Programming FLASH memory with PROG08SZ
- Simulating in-circuit and stand-alone MC68HC908AB32 MCUs with ICS08ABZ software, providing:
  - Simulation of all instructions, memory, and peripherals
  - Simulation of pin inputs from the target system
  - Installation of conditional breakpoints, script files, and logfiles
- Debugging and emulation (limited real-time) with ICD08SZ, including:
  - Loading code into RAM

  - Executing real-time in RAM or FLASH
  - Placing one hardware breakpoint in FLASH
  - Placing multiple breakpoints in RAM
- On-line help documentation for all software
- Software integrated into the WinIDE environment, allowing function key access to all applications
- MON08 emulation connection to the target system allowing:
  - In-circuit emulation
  - In-circuit simulation
  - In-circuit programming
- Four modes of operation:
  - Standalone — using the ABICS as a standalone system without a target board
  - Simulation — using the ABICS as an in-circuit simulator/emulator with a target cable
  - Evaluation - using the ABICS for real-time evaluation of the MCU and to debug user developed hardware and software
  - Programming — using the ABICS as a programmer
- With the ICD08SZ debugging software, code may be run directly out of the MCU's internal FLASH at real-time speeds.
- With the WinIDE, CASM08Z, editor, simulator, and assembler software - the function is as a limited real-time emulator.
- With the PROG08SZ software - the function is to program MCU FLASH memory.
- With the ICS08ABZ simulation software, the MCU provides the required input/output information that lets the host computer simulate code, performing all functions except for maintaining port values. (The internal FLASH memory on the device is downloaded with a program that generates the appropriate port values.) The ICS08ABZ software on the host computer lets the host computer become a simulator.
- With using the ICD08SZ debugging software, code can be run directly out of the MCU's internal FLASH at real-time speeds.

- Timing is accomplished through a 4.9152 MHz crystal

1.2 ABICS Components

The complete ABICS system includes hardware, software, and documentation. **Table 1-1** lists the ABICS product components.

Table 1-1. ABICS Product Components

Part Number	Description
ICS08AB	ABICS software development package
ICS08ABZ	ABICS simulator
ICD08SZ	ABICS debugger/emulator
MC68HC908AB32	MCU
M68CBL05C	Flex target cable
KRISTA 22-122	Serial cable
FRIWO 11.8999-P5	ABICS Power supply
M68ICS08AB	ABICS Hardware board
M68ICS08SOM/D	In-circuit simulator software operator’s manual
M68ICS08ABUM/D	In-circuit simulator hardware operator’s manual

1.2.1 ABICS Hardware

Table 1-2 lists the ABICS hardware components.

Table 1-2. Hardware Connector Components

Components	Description
XU1	Test socket for the Motorola MC68HC908AB32 MCU: 64-pin QFP (quad flat pack)
J1 &J2	Two 2-row × 20-pin, 0.1-inch spacing connectors to connect the ABICS 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	+5 Vdc input voltage (V_{DD})
P2	RS-232 to interface the ABICS to the host serial connector
P3	Power Terminal

1.2.2 ICS Interface Software

Windows-optimized software components are referred to, collectively, as the ABICS software (part number ICS08AB). It is a product of *P&E Microcomputer Systems, Inc.*, and is included in the ABICS kit **(Table 1-3).**

Table 1-3. Software Components

Components	Description
WINIDE.EXE	Integrated development environment (IDE) software interface for editing and performing software or in-circuit simulation
CASM08Z.EXE	CASM08Z command-line cross-assembler
ICS08ABZ.EXE	In-circuit/stand-alone simulator software for the MC68HC908AB32 MCU
PROG08SZ.EXE	FLASH memory programming software
ICD08SZ.EXE	In-circuit debugging software for limited, real-time emulation

1.3 Hardware and Software Requirements

The ABICS software requires this minimum hardware and software configuration:

- Windows 95 or later version operating system
- Approximately 2 Mbytes of available random-access memory (RAM) and 5 Mbytes of free disk space
- An RS232 serial port for communications between the ABICS and the host computer

1.4 Specifications

Table 1-4 summarizes the ABICS hardware specifications.

Table 1-4. ABICS 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 M68ICS08AB (ABICS) for use.

2.2 Hardware Preparation

This paragraph explains:

- Limitations of the ABICS
- Configuration of the ABICS
- Installation of the ABICS
- Connection of the ABICS 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 ABICS Limitations

These sub-paragraphs describe system limitations of the ABICS.

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 ABICS Jumper Headers

The ABICS supports four configuration options: standalone, simulation, evaluation, and programming.

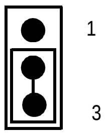
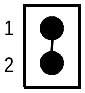
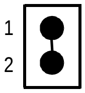
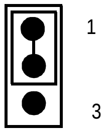
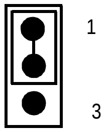
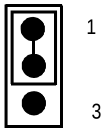
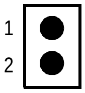
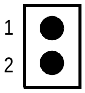
- Standalone — ICS08ABZ.exe running on the host computer (the ABICS is not connected.) Emulation of the MCU CPU, registers, and I/O ports are done within the host computer environment.
- Simulation — Host computer connected to the ABICS via the RS-232 cable and ICS08ABZ.exe running on the host computer. This provides access to the M68HC(9)08AB MCU, internal registers, and I/O ports.

- Evaluation — Host computer connected to the ABICS and the ABICS 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 connected to the ABICS, and the ABICS connected to the target system via the MON08 cable, if the MCU is resident on the target system. Use the PROG08SZ.exe to program the MCU FLASH module. In the programming mode there is limited evaluation.

Eight jumper headers (**Table 2-1**) on the ABICS are used to configure the hardware options.

CAUTION: *The ABICS can be set to operate at a variety of voltages. When configuring the ABICS 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 ABICS Jumper Header Description

Jumper Header (No W1 & W2)	Type (Factory Default Shown)	Description
W3 Target Reset Select		Jumper on position 1&2: RST_IN from target resets on-board ripple counters and MCU. Jumper on position 2&3: ICS RST_OUT (from RST*) resets target.
W4 TGT_OSC		No Jumper: Disconnects OSC1 input to OSC input of target adapter Jumper: Connects OSC1 input to OSC input of target adapter.
W5 OSC		No Jumper: Disconnects on-board oscillator to OSC1 input of MCU Jumper: Selects on-board oscillator.
W6 VREFH		Jumper on position 1&2: Selects on-board V_{DD} as ADC reference high signal. Jumper on position 2&3: Selects VREFH signal from target as ADC reference high signal.
W7 VDDAREF		Jumper on position 1&2: Selects on-board V_{DD} as ADC power supply. Jumper on position 2&3: Selects VDDAREF signal from target as ADC power supply.
W8 AVSS		Jumper on position 1&2: Selects on-board GND (Common) as ADC common and reference voltage. Jumper on position 2&3: Selects AVSS/VREFL signal from target as ADC common and reference voltage.
W9 PWR_ON		No Jumper: DTR turns on regulator. Jumper: Disable DTR control; on-board regulator always on.
W10 EVDD		(On-board regulator reference input) No Jumper: Pin 1 target power used as reference input. Jumper: Pin 2 is GND (Common).

2.2.3 Target Interface Connection Options

There are two ways to connect the ABICS simulator board to your target system:

- Flex cable — low-noise target interface connection
- MON08 cable — target interface connection with MCU FLASH programming and limited emulation

Table 2-2 is a quick reference for defining the cable/connector setup to use with the ABICS.

Refer to Section 3, Support Information for pin assignments and signal definition.

Table 2-2. Cable/Connector Options for MCUs

MCU	Flex Cable	MON08 Cable
MC68HC908AB32	J1 and J2	J3

2.2.4 Host Computer - ABICS Interconnection (P2)

The host computer to ABICS interface is via the single system connector P2, 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 ABICS kit, a DB9-male-to-female, 6-ft. (3 m) long serial cable.

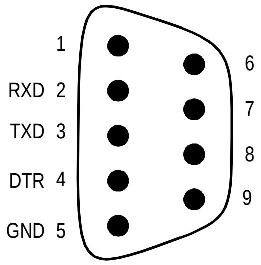


Figure 2-1 P2 Host Computer to ABICS Interconnection

2.2.5 Power Connector (P1)

Connect +5-Vdc power directly to the ABICS via connector J3 (**Figure 2-2**) using the provided power supply.

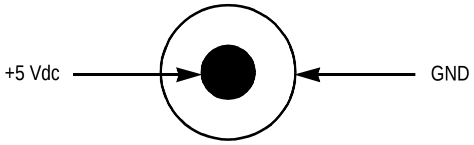


Figure 2-2 P1 Power Connector

2.3 Connecting the ABICS

The following steps provide instructions for connecting the ABICS to the host PC and power connection.

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.*

a. Configure the jumpers W-3 through W-10 (Table 2-1) on the ABICS for your application.

b.Install an MCU into the appropriate socket, for your application, onto the ABICS board.

Note: 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.

c. Plug the serial cable into P2 on the ABICS.

d. Plug the serial cable into the COM port on the host PC.

NOTE: *Steps e. through g. should not be completed until all connections to the target are completed (**Paragraph 2.4**).*

e. Connect the power cable to P1 on the ABICS board.

f. Plug the power cable into an ac power outlet, using one of the country-specific adapters.

g. The ABICS green power LED lights.

2.4 Connecting the ABICS to the Target System

Connect the ABICS to the target system using one of these methods:

- Emulating using a flex cable for low-noise

  When emulating, connect the 80-pin M68CLB05C flex cable to the connectors labeled J1and J2 on the simulator board. Attach the other end of the cable to the appropriate connector on the target system. Target head adapters are available.

- Using a MON08 cable to debug the target system.

NOTE: *An MCU must be installed in the target system. No MCU should on the ABICS.*

Connect the MON08 debug interface cable to the appropriate MON08 debug interface connector, J3, for communication with the target system's MCU. Attach the other end of the cable to the appropriate connector on the target system.

NOTE: *For more detailed information on the MON08, refer to Section 4 of this manual.*

2.5 Installing the Software

For instructions for installing the ICS08 software, refer to *P&E Microcomputer Systems, Inc.*, *M68ICS08HC08 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 may be useful in the design, installation, and operation of your application.

3.2 MCU Subsystem

The MCU subsystem consists of the MC68HC908AB32 microcontroller, clock generation and selection, monitor mode control logic that places and holds the ABICS 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 ABICS socket simple simulation
- In the ABICS socket for programming
- In the ABICS socket connected to the target for emulation
- In the target for MON08 debug operation

3.3 ABICS Functional Description

NOTE: *For the following discussion on the theory of operation of the ABICS, refer to the schematic diagrams in Paragraph 3.6, Figures 3-5 to 3-11 of this section.*

3.3.1 ICS08AB Board

The core component of the board is the MC68HC908AB32 MCU. The MCU may be plugged-in to either the ABICS board or to a target system board for test or evaluation.

When the MCU resides on the ABICS board, the board may be used as an in-circuit emulator or simulator for the MC68HC908AB32. For this configuration, a low-noise flexible target cable is run from the board to the target system. The cable (Motorola part number M68CBL05C), is terminated in connectors for target head adapters.

When the MCU resides on a target system board, communication with the MCU is over a 16-pin MON08 cable (Motorola part number 01-RE91008W01). Either version of the MCU may be supported when using the MON08 cable.

When using the ICD08SZ simulation software, the MCU provides the required input/output information that lets the host computer to simulate code, performing all functions except for maintaining the port values. The internal FLASH/EEPROM memory on the device is downloaded with a program that generates the appropriate port values.

The ICD08SZ software on the host computer allows the host computer to become a simulator. When the ICS requires port data, the computer requests the data through the host's serial connection to the core MCU. The core MCU responds by sending the data to the host via the serial connection. It is this arrangement that allows a real-world interface for the in-circuit simulator. The clock runs the MCU at a 4.9512-MHz external clock rate. However, note that the simulation speed will be slower than this rate, because the host computer is the simulator.

When using the ICD08SW debugging software, code may be run directly out of the MCU's internal FLASH at real-time speeds.

When using the PROG08SZ programming software, the MCU's FLASH/EEPROM memory can be programmed. Socket XU1 supports the 64-pin QFP version of the part. The ICS08AB32 also supports in-circuit programming of either version of the part through the MON08 cable.

The ICS08AB32 board also provides +5 Vdc power, +8.0 Vdc power for the V_{TST} voltage required to enter monitor mode, a 4.9152-MHz clock signal, and host PC RS-232 level translation.

3.3.2 M68HC908AB32 MCU

The MCU is an MC68HC908AB32 and is available in one package, the 64-pin QFP, which mounts in a clam-shell socket.

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 ICS socket for programming
- In the ICS socket for simple simulation
- In the socket and connected to the target for emulation
- On the target for MON08 debug operation

3.3.3 Clocks

The ABICS contains a 4.1952-MHz crystal oscillator. When the remote target connection is made, the user may opt to feed the output from the ABICS crystal (SP-OSC) to the external clock input (OSC1) of the ABICS via W5, a 2-pin jumper.

3.3.4 Board Reset

The ABICS includes two reset sources:

- An output from the POR (Power-On Reset) circuit via the host system software
- An internal reset exception operation of the processor

The host system resets the ABICS by cycling power to most of the ICS circuitry, including the POR circuit. RS-232 handshake line DTR is used for this purpose.

The reset function of the ABICS is both an input and an output. The ABICS drives its reset pin low after encountering several different exception conditions. W3 is provided to allow you to select whether the target system may reset the MCU on the ABICS or whether the target system receives a reset signal from the ABICS.

`RST*` is not a bidirectional, open-drain signal at the target connectors. Removing the jumper leaves the `RST_IN*` signal pulled up to THE MCU operating voltage.

3.3.5 Device Configuration Selection

The operation mode of the ABICS processor is selected at the rising edge of the RESET signal. The ABICS requires that the processor operate in monitor mode. To set monitor mode operation, the IRQ* line to the ABICS is level shifted to apply V_{HI} to the processor on the rising edge of reset. V_{HI} is a signal name that is specified as minimum V_{DD} +2.5 V and maximum 8 V, with the highest V_{DD} of 3.3 V, yielding a range of minimum 5.8 V and maximum 8 V.

The ABICS RST* pin is the main mode select input and is pulled to logic 0, then logic 1 (processor V_{DD}), to select the MCU monitor mode. The host software must communicate security bytes to the MCU to resume execution out of reset. Communication to the monitor ROM is via standard, non-return-to-zero (NRZ) mark/space data format on PTA0. The MCU maintains monitor mode and disables the COP module through continued application of V_{HI} on either IRQ* or RST*.

Six commands may be issued by the host software in control of the MCU in monitor mod: read, write, iread, iwrite, readsp, readsp, and run. Each command is echoed back through PTA0 for error checking. These commands are described in the *M68ICS08AB In-circuit Simulator Software Operator’s Manual.*

3.3.6 Level Translation

The ABICS has an operation voltage range of +3.0 to +5.0 volts while the host development system interface is an RS-232 (com) port. U2 on the ICS converts

5 V logic signals to RS-232 levels. Transistors Q9-Q10 translate 5 V logic levels to the MCU operating voltage (3.0-5.0 V).

3.3.7 ABICS Operating Voltage, Variable Selector

To provide the ABICS with power input that matches your target environment, the ABICS includes a on-board regulator. The ABICS monitors the user’s target system power via the EVDD pin of the FLEX cable. EVDD pin is connected to the power supply of the user’s target system via the target adapter. If the EVDD pin is floated, the regulator output 5.0Vdc.

The on-board regulator is activated by the RS-232 handshake line DTR. To activate the regulator mannually, set jumper W9.

3.4 ABICS Connector Signal Definitions

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

3.4.1 Target Flex Cable Interface Connectors J1 and J2

Table 3-1 and Table 3-1 describe the pin assignments for the flexible cable ABICS board connectors J1 and J2.

A generic cable (Motorola part number M68CBL05C) connects between the ICS module and target adapter(s) for the different user package targets.

The FLEX cable has two 2 × 40, 0.1-inch center connector (P1, P2) at the end, which connects to the ICS module. At the opposite end, it has two 2 × 20, 0.5-inch center connector (P3), which connects to the target adapter.

Table 3-3 shows the connectivity between the two ends of the FLEX cable and the usage of the lines in this application.

.

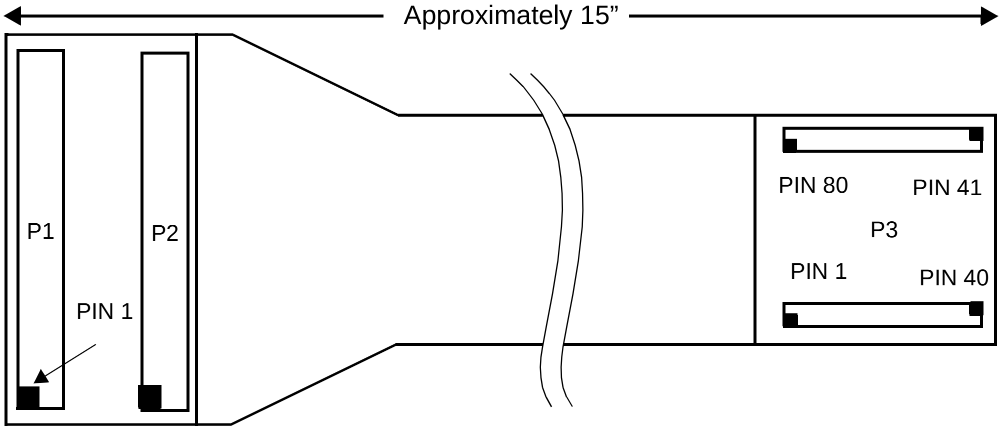


Figure 3-1 FLEX Cable

3.4.2 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	Common		
2	TGT_IRQ*	In	External interrupt request
3	PTC2	Bidirectional	General purpose I/O
4	Common		
5	TGT_PTC0	Bidirectional	General purpose I/O
6	PTF1	Bidirectional	General purpose I/O/ TimerA channel 3
7	N/C		
8	PTF3	Bidirectional	General purpose I/O
9	VDD		Logical chip power supply
10	N/C		
11	LVDD		Voltage supplied by voltage regulator (board or target)
12	PTF5	Bidirectional	General purpose I/O/ Timer B channel 1
13	PTD7	Bidirectional	General purpose I/O
14	PTB7	Bidirectional	General purpose I/O
15	PTD5	Bidirectional	General purpose I/O
16	PTD1	Bidirectional	General purpose I/O
17	PTH1	Bidirectional	General purpose I/O
18	TGT_AVSS/VREFL	In	ADC common & reference voltage
19	Common		
20	PTD3	Bidirectional	General purpose I/O
21	PTB2	Bidirectional	General purpose I/O
22	PTA7	Bidirectional	General purpose I/O
23	PTB4	Bidirectional	General purpose I/O
24	Common		
25	PTB6	Bidirectional	General purpose I/O
26	PTA4	Bidirectional	General purpose I/O

Pin No.	Schematic	Direction	Signal Description
27	Not Connected		
28	PTA2	Bidirectional	General purpose I/O
29	Not Connected		
30	TGT_PTA0	Bidirectional	General purpose I/O
31	PTF6	Bidirectional	General purpose I/O
32	PTG2	Bidirectional	General purpose I/O with key wakeup feature
33	PTE1	Bidirectional	General purpose I/O/ SCI Transmit Data
34	PTG0	Bidirectional	General purpose I/O with key wakeup feature
35	PTE3	Bidirectional	General purpose I/O/ Timer A Channel 1
36	Common		
37	PTE5	Bidirectional	General purpose I/O/ SPI data path
38	Common		
39	PTE7	Bidirectional	General purpose I/O
40	Common		

Table 3-2 J2 Target Flex Connector Pin Assignment Descriptions

Pin No.	Schematic	Direction	Signal Description
1	PTC5	Bidirectional	General purpose I/O
2	PTC4	Bidirectional	General purpose I/O
3	TGT_PTC3	Bidirectional	General purpose I/O
4	RST	In/Out	Reset signal from target Reset signal to target
5	TGT_PTC1	Bidirectional	General purpose I/O
6	PTF0	Bidirectional	General purpose I/O/ Timer A Channel 2
7	OCS1	Bidirectional	External clock in/out
8	PTF2	Bidirectional	General purpose I/O/ Timer B Channel 2
9	Common		
10	PTF4	Bidirectional	General purpose I/O/ Timer B Channel 0
11	Common		

Pin No.	Schematic	Direction	Signal Description
12	PTF7	Bidirectional	General purpose I/O
13	TGT_VREFH	In	ADC reference voltage
14	Common		
15	PTD6	Bidirectional	General purpose I/O/ Timer External Input clock
16	PTD0	Bidirectional	General purpose I/O
17	PTD4	Bidirectional	General purpose I/O
18	TGT_VDDAREF	In	ADC Power Supply
19	PTH0	Bidirectional	General purpose I/O withe key wakeup feature
20	PTD2	Bidirectional	General purpose I/O
21	PTB1	Bidirectional	General purpose I/O
22	PTB0	Bidirectional	General purpose I/O
23	PTB3	Bidirectional	General purpose I/O
24	PTA6	Bidirectional	General purpose I/O
25	PTB5	Bidirectional	General purpose I/O/ ADC Channel
26	PTA5	Bidirectional	General purpose I/O
27	Common		
28	PTA3	Bidirectional	General purpose I/O
29	Not Connected		
30	PTA1	Bidirectional	General purpose I/O
31	Not Connected		
32	Common		
33	PTE0	Bidirectional	General purpose I/O/ SCI Transmit Data
34	PTG1	Bidirectional	General purpose I/O with keyboard wakeup feature
35	PTE2	Bidirectional	General purpose I/O/ SCI Transmit Data
36	EVDD	In	Target supplied power
37	PTE4	Bidirectional	General purpose I/O/ SPI Slave select
38	Common		
39	PTE6	Bidirectional	General purpose I/O/ SPI data path
40	Common		

Table 3-3 FLEX Cable Connectors

Single	ICS08AB32 Connector P1 Pin Number	ICS08AB32 Connector P2 Pin Number	Target Head Adapter Pin Number
PTC4	NA	2	1
PTC5	NA	1	2
TGT_IRQ*	2	NA	3
GND	1	NA	4
TGT_RST*	NA	4	5
TGT_PTC3	NA	3	6
GND	4	NA	7
PTC2	3	NA	8
PTF0	NA	6	9
TGT_PTC1	NA	5	10
PTF1	6	NA	11
TGT_PTC0	5	NA	12
PTF2	NA	8	13
OSC1	NA	7	14
PTF3	8	NA	15
NC	7	NA	16
PTF4	NA	10	17
GND	19	NA	18
NC	10	NA	19
VDD	9	NA	20
PTF7	NA	12	21
GND	NA	11	22
PTF5	12	NA	23
LVDD	11	NA	24
GND	24	NA	25

Table 3-3 FLEX Cable Connectors

Single	ICS08AB32 Connector P1 Pin Number	ICS08AB32 Connector P2 Pin Number	Target Head Adapter Pin Number
VERFH	NA	13	26
PTB7	14	NA	27
PTD7	13	NA	28
PTD0	NA	16	29
PTD6	NA	15	30
PTD1	16	NA	31
PTD5	15	NA	32
VDDAREF	NA	18	33
PTD4	NA	17	34
VERFL	18	NA	35
PTH1	17	NA	36
PTD2	NA	20	37
PTH0	NA	19	38
PTD3	20	NA	39
GND	38	NA	40
PTB1	NA	21	41
PTB0	NA	22	42
PTB2	21	NA	43
PTA7	22	NA	44
PTB3	NA	23	45
PTA6	NA	24	46
PTB4	23	NA	47
GND	40	NA	48
PTB5	NA	25	49
PTA5	NA	26	50
PTB6	25	NA	51
PTA4	26	NA	52

Table 3-3 FLEX Cable Connectors

Single	ICS08AB32 Connector P1 Pin Number	ICS08AB32 Connector P2 Pin Number	Target Head Adapter Pin Number
GND	NA	9	53
PTA3	NA	28	54
NC	27	NA	55
PTA2	28	NA	56
NC	NA	29	57
PTA1	NA	30	58
NC	29	NA	59
TGT_PTA0	30	NA	60
NC	NA	31	61
GND	NA	14	62
PTF6	31	NA	63
PTG2	32	NA	64
PTE0	NA	33	65
PTG1	NA	34	66
PTE1	33	NA	67
PTG0	34	NA	68
PTE2	NA	35	69
EVDD	NA	36	70
PTE3	35	NA	71
GND	36	NA	72
PTE4	NA	37	73
GND	NA	27	74
PTE5	37	NA	75
GND	NA	32	76
PTE6	NA	39	77
GND	NA	38	78
PTE7	39	NA	79

Table 3-3 FLEX Cable Connectors

Single	ICS08AB32 Connector P1 Pin Number	ICS08AB32 Connector P2 Pin Number	Target Head Adapter Pin Number
GND	NA	40	80

3.4.3 Target MON08 Interface Connector J3

The MON08 interface connector, J3 (Table 3-4), is used when the MCU is mounted on the target. Refer to Section 4 Using the MON08 for detailed information.

Table 3-4 J3 MON08 Target Connector Pin Assignment Descriptions

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

3.4.4 Power Connector P1 Pin Assignments

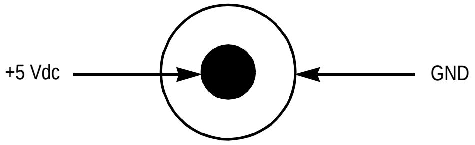


Figure 3-2. P1 Power Connector

Table 3-5. Power Connector P1 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 ABICS logic circuits
2	GND	Common
3	GND	Common

3.4.5 Host Computer - ABICS Interconnection (P2)

The host computer to ABICS interface is via the single system connector P2 (Figure 3-2), which is a 9-pin, D-type connector (Amp part number AMP-9726-A) (Table 3-7).

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

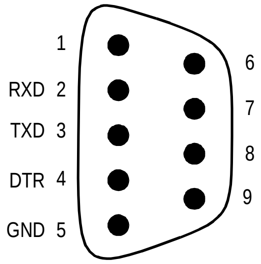


Figure 3-3. P2 Host Computer to ABICS Interconnection

Table 3-6 RS-232C Communication Connector P2 Pin Assignment Descriptions

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 3-7. ABICS Parts List (Sheet 1 of 2)

Reference Designator	Description	Manufacturer	Part Number
C1	Capacitor, 100pF	AEC	Z5U-101
C2, C3, C4, C7, C17	Capacitor, 10uF	Truth	85C +80-20%
C5, C6, C9, C10, C11, C13, C14, C15, C16, C18	Capacitor, 0.1 uF	AVX	SR215E104MAA
C12	Capacitor,.001uF, ceramic	AEC	Z5U-102
D1	1A 20V Schottky Rectifier	MOTOROLA	1N5817
D2	Zener Transient Voltage Suppressors	MOTOROLA	SA 5.0
D3, D4, D5, D6, D9, D10, D11, D12, D13, D14	Diode	SEMTECH	1N4148
D7	LED	KINGBRIGHT	L-934YD
D8	LED	KINGBRIGHT	L-934GD
F1	FUSE	BUSSMAN	GMA 1.5A 250V
J2, J1	Header, 2x20, 200, Target Head	MOBICON	PHDS-40G1
J3	Header, 2x8, 100, MON08	MOBICON	PHDS-16G1
P1	Power Jack	WEALTH	DS-210A
P2	Connector DB9	MOBICON	DB9SR
P3	Power Terminal	RDI	2SV-02
Q1, Q2, Q8	Transistor, PNP	MOTOROLA	BC557B
Q3, Q5, Q6, Q7, Q9, Q10, Q11, Q12, Q13, Q15	Transistor, NPN	MOTOROLA	BC547B
Q4	TMOS Power FET, N-Channel Enhancement-Mode Silicon Gate	MOTOROLA	MTD3055EL
Q14	Transistor, NPN	MOTOROLA	MPS2369A
R1	resistor, 0.1R, 5%	UNION	1/4W 5% MF

Table 3-7. ABICS Parts List (Sheet 2 of 2)

Reference Designator	Description	Manufacturer	Part Number
R2, R3, R6, R7, R8, R9, R16	Resistor, 100K, 5%	UNION	1/4 W 5% CF
R4, R5, R34, R35, R41	Resistor, 1K, 5%	UNION	1/4 W 5% CF
R10	Resistor, 1M, 5%	UNION	1/4 W 5% CF
R13, R14, R15, R17, R18, R19, R20, R21, R24, R25, R26, R30, R31, R32, R33, R37, R38, R39, R40, R42	Resistor, 10K, 5%	UNION	1/4 W 5% CF
R22	Resistor, 2K, 5%	UNION	1/4 W 5% CF
R28, R29	Resistor, 470K, 5%	UNION	1/4 W 5% CF
U1	Single Operational Amplifier	MOTOROLA	MC33172
U2	EIA-232/V.28 CMOS Driver/Receiver	MOTOROLA	MC145407
U5	CMOS Quad 2-input Analog Multiplexer	MOTOROLA	MC14551
U6	Monolithis WFR, Binary Counter	MOTOROLA	MC74HC4020
U7	Non-inverting 3- State Buffer	MOTOROLA	MC74HC125
W3	Header, 3x1, 100	3M	2403-6112TG
W4, W5, W6, W7, W8, W9, W10	Header, 2x1, 100	3M	2402-6112TG
XU1	Socket, 68HC908AB32FU64	YAMAICHI	1C51-0644-892
Y1	4.9152 MHz Oscillator	HOSONIC	HO-12C4.9152M
XF1	Fuse Clip, 5mm Diameter	LITTLEFUSE	100-54
Z1-Z4	Rubber Adhesive Bumpers	SPC TECHNOLOGY	2565
Z5-Z10	.100 Jumper Shunt	3M	929951-00

3.6 ABICS Board Layout and Schematic Diagrams

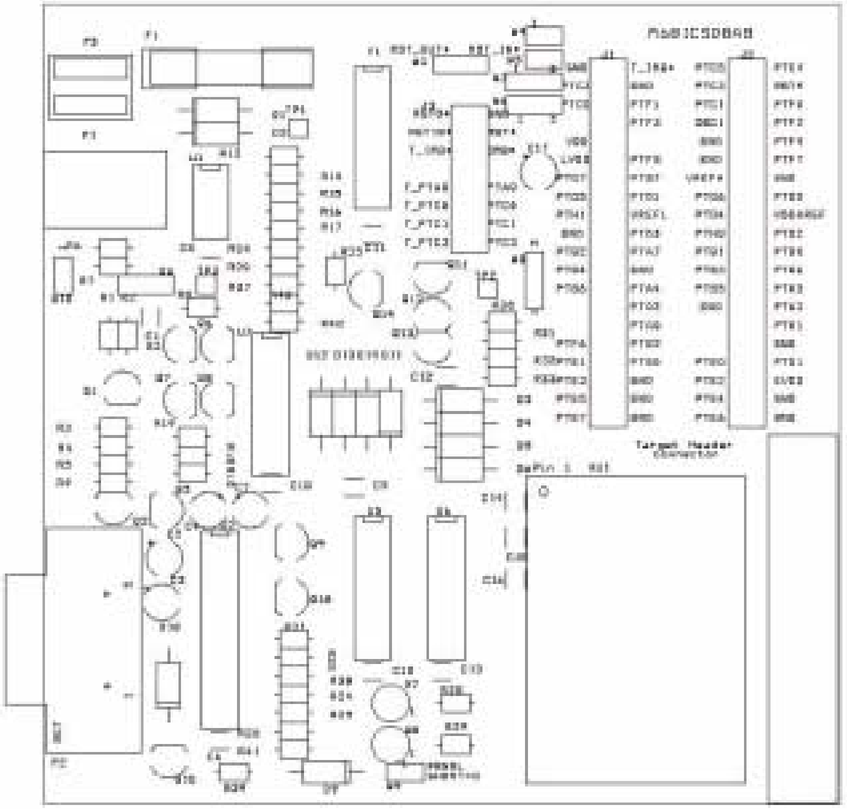


Figure 3-4 ABICS Board Layout

REVISIONS				
ZONE	REV	DESCRIPTION	DATE	APPROVED
	0	Original Revision	3-Dec-1999	
	A	Modified after testing	15-Jan-2000	
	B	Add buffer at MON08 Connector	27-Mar-2000	

NOTES:

1. UNLESS OTHERWISE SPECIFIED:

   RESISTANCE VALUES ARE IN OHMS.
   RESISTORS ARE 1/4 WATT, 5%
   CAPACITANCE VALUES ARE IN
   MICROFARADS

2. INTERRUPTED LINES CODES WITH THE SAME LETTER OR LETTER COMBINATIONS ARE ELECTRICALLY CONNECTED.

3. DEVICE TYPE NUMBER IS FOR REFERENCE ONLY. THE NUMBER VARIES WITH THE MANUFACTURER.

4. SPECIAL SYMBOL USAGE:

   * DENOTES ACTIVE-LOW SIGNAL
   [ ] DENOTE VECTORED SIGANLS

5. INTERPRET DIAGRAM IN ACCORDANCE WITH AMERICAN NATIONAL STANDARDS INSTITUTE SPECIFICATIONS, CURRENT REVISION, WITH THE EXCEPTION OF LOGIC BLOCK SYMBOLOGY.

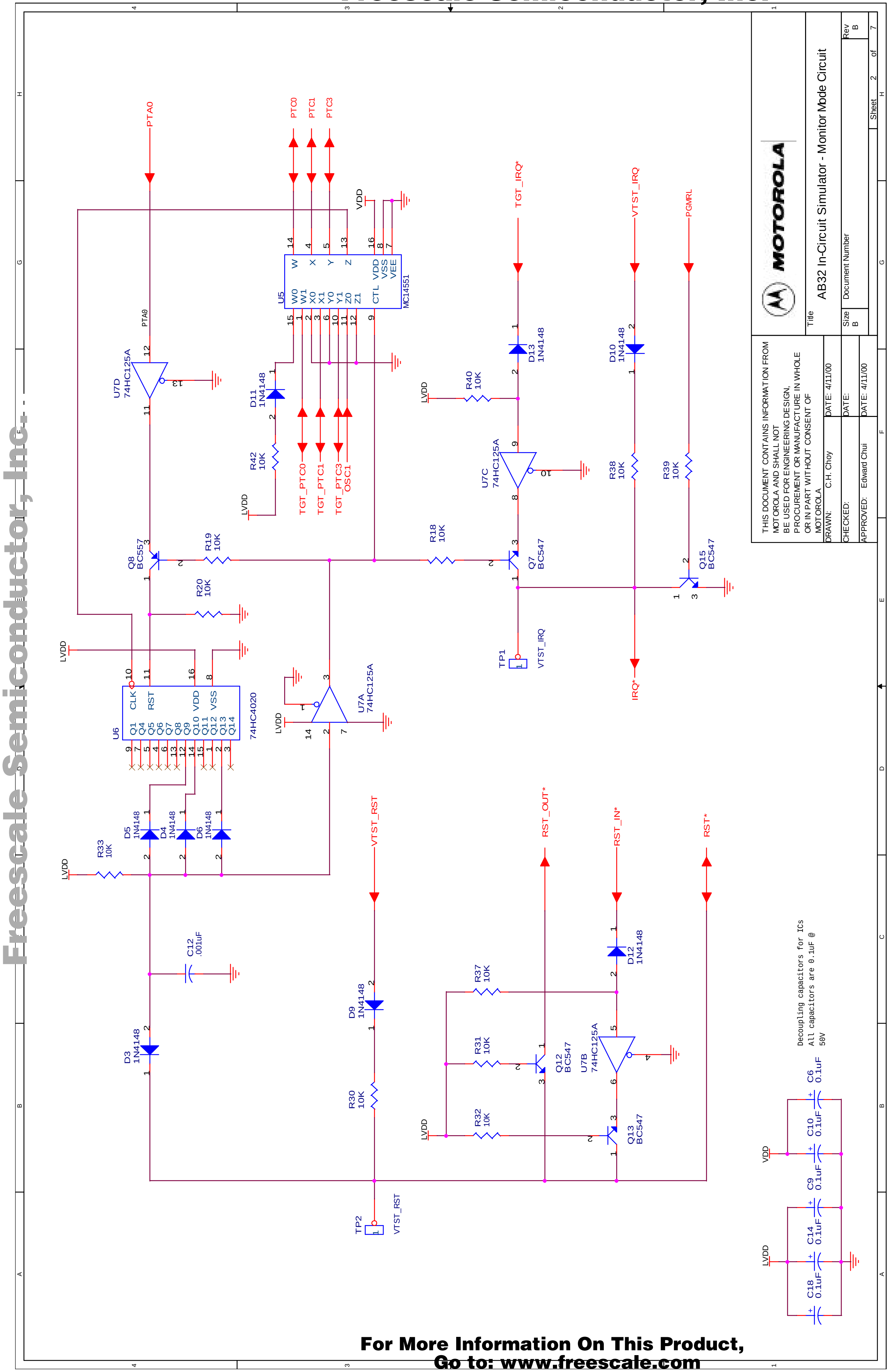
```
CHANGES:
8 JUNE 2000: R13 on page 3,  10K -> 5K, reference ECO20
             Correct port direction of off-page connector RST*, RST_IN*
```

COMPUTER GENERATED DRAWING : DO NOT REVISE MANUALLY

THIS DOCUMENT CONTAINS INFORMATION FROM MOTOROLA AND SHALL NOT BE USED FOR ENGINEERING DESIGN, PROCUREMENT OR MANUFACTURE IN WHOLE OR IN PART WITHOUT CONSENT OF MOTOROLA

MOTOROLA

DRAWN: C.H. Choy	DATE: 4/11/00	Title AB32 In-Circuit Simulator - ABICS		
CHECKED:	DATE:	Size B	Document Number	Rev B
APPROVED: Edward Chui	DATE: 4/11/00			Sheet 1 of 7



Decoupling capacitors for ICs
All capacitors are 0.1uF @
50V

MOTOROLA

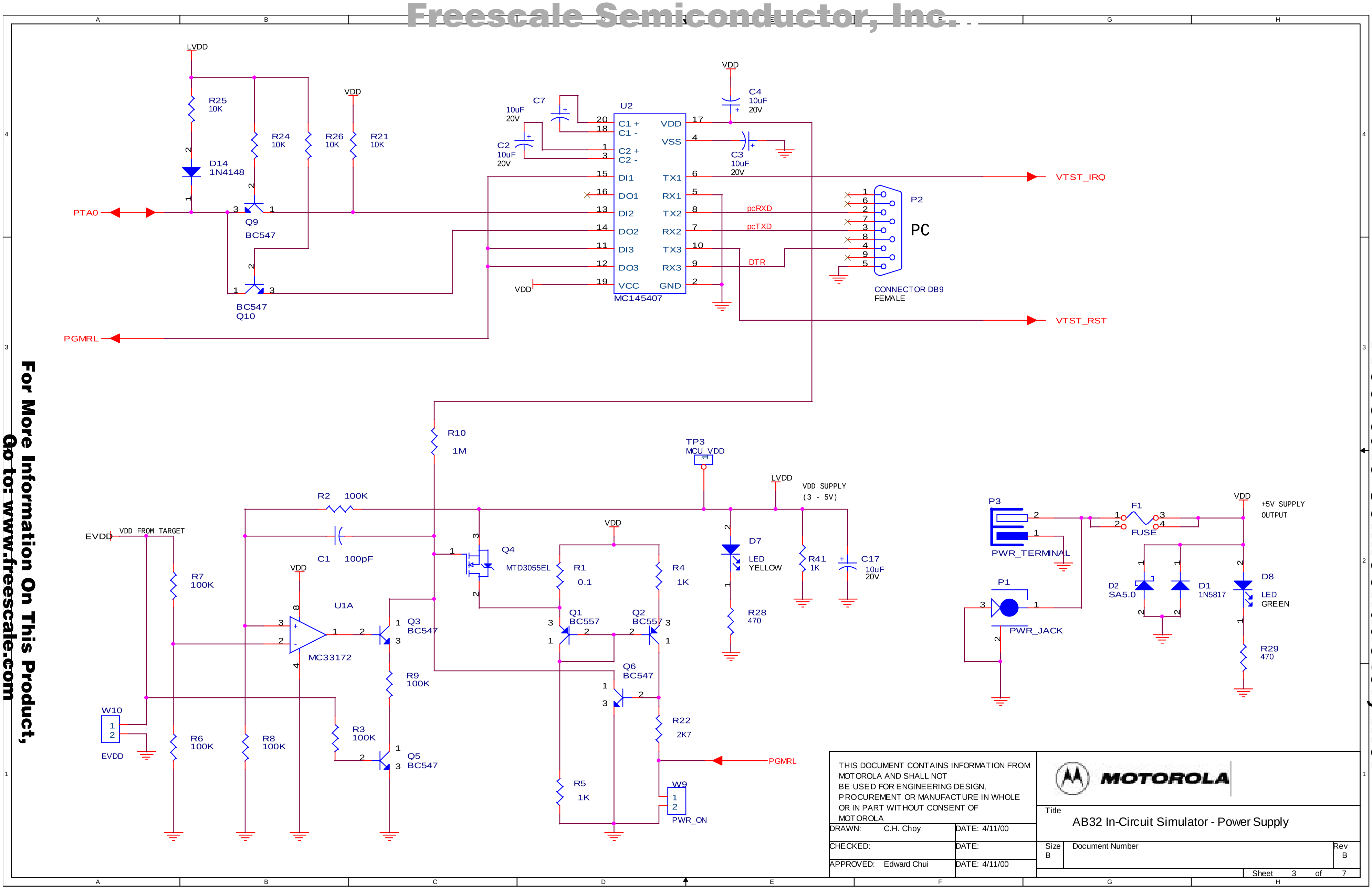
THIS DOCUMENT CONTAINS INFORMATION FROM MOTOROLA AND SHALL NOT BE USED FOR ENGINEERING DESIGN, PROCUREMENT OR MANUFACTURE IN WHOLE OR IN PART WITHOUT CONSENT OF MOTOROLA

DRAWN:	C.H. Choy	DATE: 4/11/00
CHECKED:		DATE:
APPROVED:	Edward Chui	DATE: 4/11/00

Title
AB32 In-Circuit Simulator - Monitor Mode Circuit

Size	Document Number	Rev
B		B

Sheet 2 of 7



THIS DOCUMENT CONTAINS INFORMATION FROM MOTOROLA AND SHALL NOT BE USED FOR ENGINEERING DESIGN, PROCUREMENT OR MANUFACTURE IN WHOLE OR IN PART WITHOUT CONSENT OF MOTOROLA

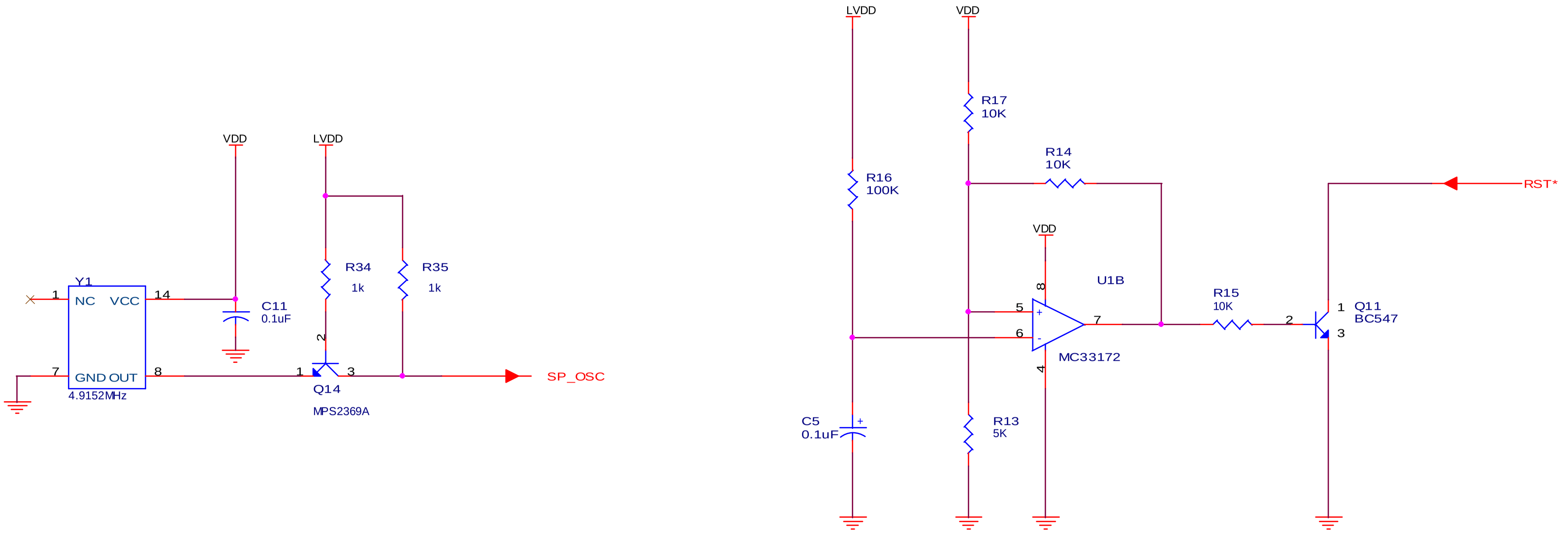
DRAWN:	C.H. Choy	DATE: 4/11/00
CHECKED:		DATE:
APPROVED:	Edward Chui	DATE: 4/11/00

MOTOROLA

Title
AB32 In-Circuit Simulator - Power Supply

Size B | Document Number | Rev B

Sheet 3 of 7



THIS DOCUMENT CONTAINS INFORMATION FROM MOTOROLA AND SHALL NOT BE USED FOR ENGINEERING DESIGN, PROCUREMENT OR MANUFACTURE IN WHOLE OR IN PART WITHOUT CONSENT OF MOTOROLA

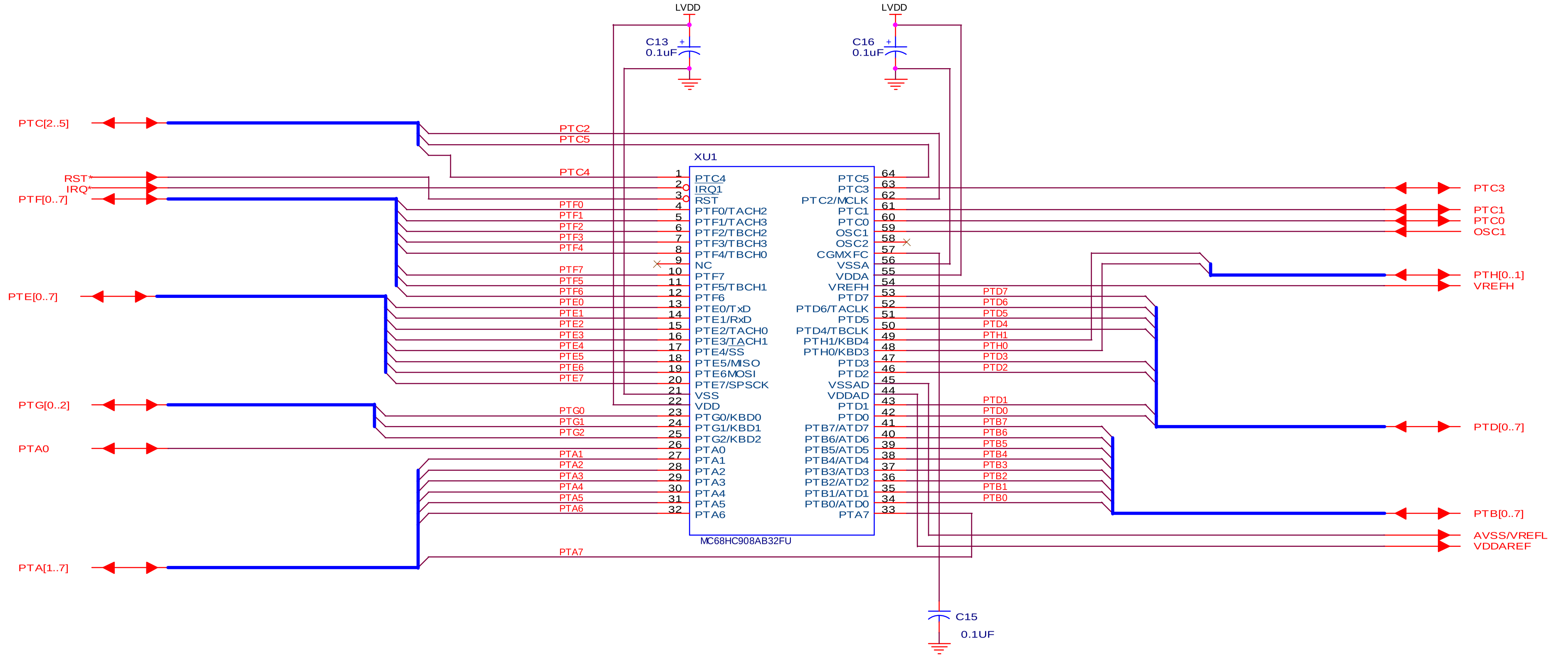
DRAWN:	C.H. Choy	DATE: 4/11/00
CHECKED:		DATE:
APPROVED:	Edward Chui	DATE: 4/11/00

MOTOROLA

Title
AB32 In-Circuit Simulator - Power-On Reset and OSC

Size	Document Number	Rev
B		B

Sheet 4 of 7



THIS DOCUMENT CONTAINS INFORMATION FROM MOTOROLA AND SHALL NOT BE USED FOR ENGINEERING DESIGN, PROCUREMENT OR MANUFACTURE IN WHOLE OR IN PART WITHOUT CONSENT OF MOTOROLA

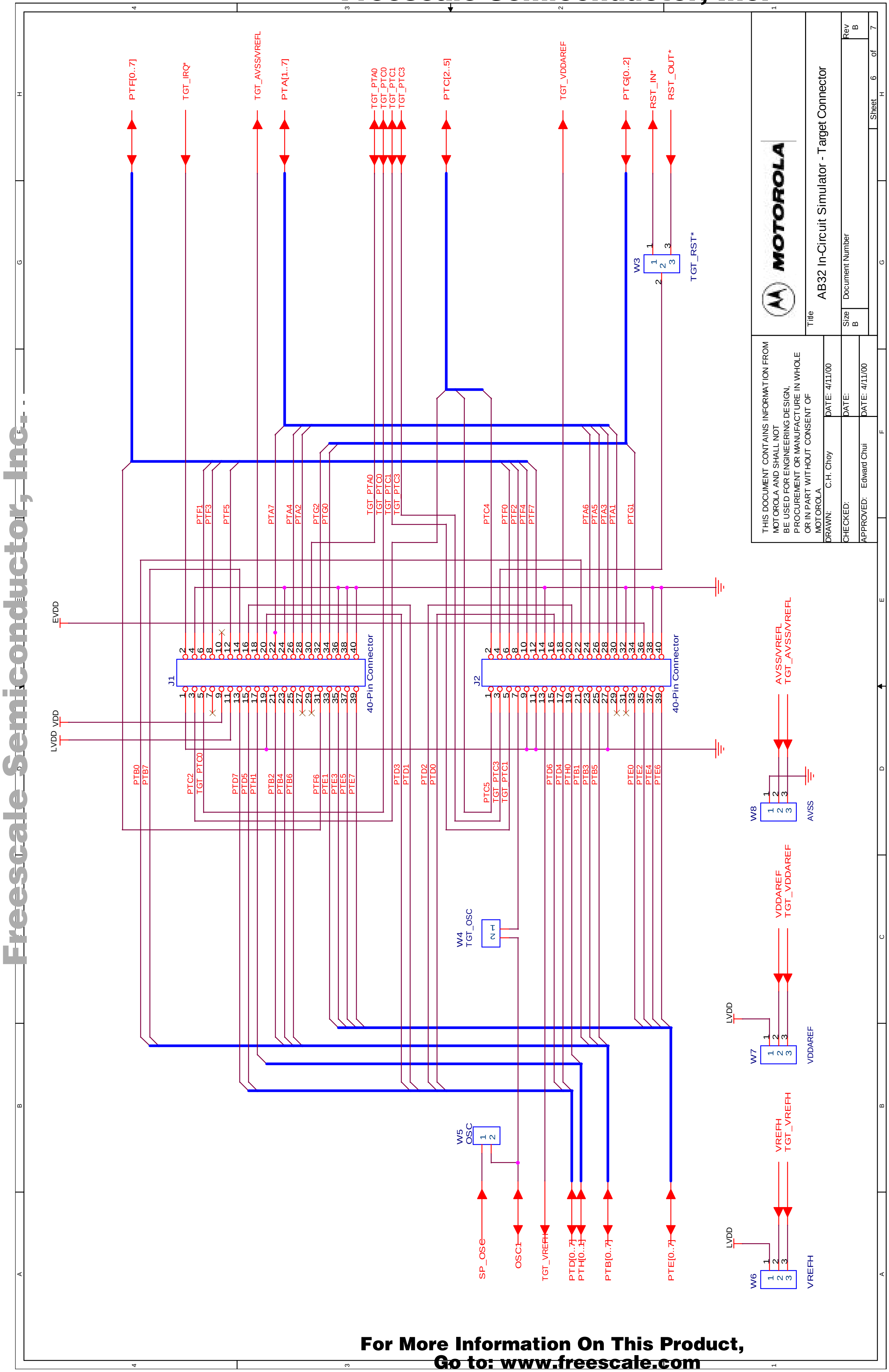
DRAWN:	C.H. Choy	DATE: 4/11/00
CHECKED:		DATE:
APPROVED:	Edward Chui	DATE: 4/11/00

MOTOROLA

Title
AB32 In-Circuit Simulator - MCU Socket

Size	Document Number	Rev
B		B

Sheet 5 of 7



THIS DOCUMENT CONTAINS INFORMATION FROM MOTOROLA AND SHALL NOT BE USED FOR ENGINEERING DESIGN, PROCUREMENT OR MANUFACTURE IN WHOLE OR IN PART WITHOUT CONSENT OF MOTOROLA

DRAWN: C.H. Choy	DATE: 4/11/00
CHECKED:	DATE:
APPROVED: Edward Chui	DATE: 4/11/00

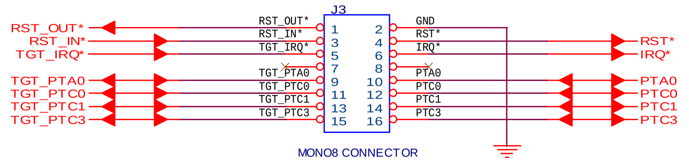
MOTOROLA

Title
AB32 In-Circuit Simulator - Target Connector

Size	Document Number	Rev
B		B

Sheet 6 of 7

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



THIS DOCUMENT CONTAINS INFORMATION FROM
MOTOROLA AND SHALL NOT
BE USED FOR ENGINEERING DESIGN,
PROCUREMENT OR MANUFACTURE IN WHOLE
OR IN PART WITHOUT CONSENT OF
MOTOROLA

MOTOROLA

DRAWN:	C.H. Choy	DATE: 4/11/00
CHECKED:		DATE:
APPROVED:	Edward Chui	DATE: 4/11/00

Title
AB32 In-Circuit Simulator - MON08 Connector

Size	Document Number	Rev
B		B

Sheet 7 of 7

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 ABICS’ 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.	M68ICS08AB 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 common (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 +7.5 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 +7.5 Vdc.
7	NC		
8	NC		

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

Pin No.	M68ICS08AB 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.	M68ICS08AB 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 +7.5 Vdc.

4.3 Connecting to the In-Circuit Simulator

Using the 16-pin cable provided with the ABICS kit, connect one end of the cable to the ABICS 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 ABICS board.

4.4 Disabling the Target-System Interface

To use the target system in a stand-alone fashion (without the ABICS board connected), jumper the pins on the target board's connectors, as shown in **Figure 4-1**. This reconnects the target MCU to the appropriate circuits on the target system.

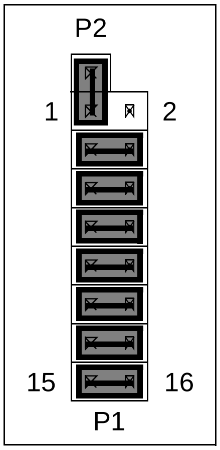


Figure 4-1. Target System Stand-Alone Connection

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 4-1**. The traces may be cut when debugging is necessary. To return the board to stand-alone use, jumpers may be installed as shown.

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 ICS08ABZ 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 ICS08ABZ
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
S113002041E900084#42234300182342000824A952
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 Hardware Configuration Guide

B.1 Introduction

This quick start guide explains the:

- Configuration of the M68ICS08AB in-circuit simulator (ABICS) board
- Installation of the hardware
- Connection of the board to a target system

There are four methods for configuring the ABICS: standalone, simulation, evaluation, and programming.

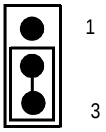
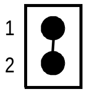
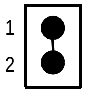
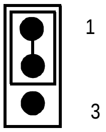
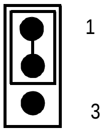
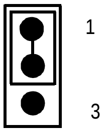
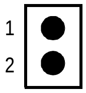
- Standalone — ICS08MRZ.exe is running on the host computer (the ABICS is not connected). Emulation of the MC68HC908AB32 MCU, registers, and I/O ports is within the host computer environment.
- Simulation — Host computer is connected to the ABICS via the RS-232 cable, and the ICS08MRZ.exe is running on the host computer. This provides access to the MC68HC908AB32 MCU, internal registers, and I/O ports.
- Evaluation — Host computer is connected to the ABICS, and the ABICS 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 ABICS, and the ABICS is connected to the target system via the MON08 cable. Use the PROG08SZ.exe to program the MCU FLASH module. In the programming mode there is limited evaluation.

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.1.1 ABICS Configurable Jumper Headers

Configure the seven jumper headers on the ABICS for your application according to the tables in this section.

Table 4-3 ABICS Jumper Header Description

Jumper Header **(No W1 & W2)**	**Type** **(Factory Default Shown)**	**Description**
W3 Target Reset Select		Jumper on position 1&2: RST_IN from target resets on-board ripple counters and MCU. Jumper on position 2&3: ICS RST_OUT (from RST#) resets target.
W4 TGT_OSC		No Jumper: Jumper: Connects OSC1 input to OSC input of target adapter.
W5 OSC		No Jumper: No effect. Jumper: Selects on-board oscillator.
W6 VREFH		Jumper on position 1&2: Selects on-board V_{DD} as ADC reference high signal. Jumper on position 2&3: Selects VREFH signal from target as ADC reference high signal.
W7 VDDAREF		Jumper on position 1&2: Selects on-board V_{DD} as ADC power supply. Jumper on position 2&3: Selects VDDAREF signal from target as ADC power supply.
W8 AVSS		Jumper on position 1&2: Selects on-board GND as ADC ground and reference voltage. Jumper on position 2&3: Selects AVSS/VREFL signal from target as ADC grpimd amd reference voltage.
W9 PWR_ON		No Jumper: DTR turns on regulator. Jumper: Disable DTR control; board always on.

Jumper Header (No W1 & W2)	Type (Factory Default Shown)	Description
W10 EVDD	1 ● 2 ●	No Jumper: Pin 1 is target powere used as reference input. Jumper: Pin 2 is GND.

B.1.2 Target Interface Cable Connections

Below (Table 4-4) is a quick reference for defining the cable/connector setup to use with the MC68HC908AB32

Table 4-4 **Cable/Connector Options for MCUs**

MCU	Flex Cable	MON08 Cable
MC68HC908AB32	J1 and J2	J3

B.1.3 Host Computer — ABICS Interconnection (P2)

Connect the DB9 serial cable. Connect one end of this cable to your host PC and the other end to connector P2 on the ABICS board.

B.2 Installing the Hardware

For installing Motorola development tools, the following steps provide installation instructions for the ABICS hardware.

To prepare the ABICS for use with a host PC:

1. Install an MCU into the M68ICS08AB board.

   Install an MCU (provided with the ABICS package) into the M68ICS08AB board in the appropriate socket, 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.

2. Connect the board to the host PC.

Locate the 9-pin connector labeled P2 on the board. Using the cable provided, connect it to a serial COM port on the host PC.

3. Apply power to the board.

   Connect the 5-volt power supply to the round connector on the board, P1. 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.

B.3 Installing the Software

For instructions for installing the ICS08 software, refer to *P&E Microcomputer Systems, Inc., M68ICS08HC08 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.4 Connecting to a Target System

The two ways to connect the M68ICS08AB simulator board to a target system are via:

- The flex cable
- The MON08 cable

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

- Using a flex cable

  When emulating an MC68HC908AB MCU, connect the 80-pin M68CBL05C flex cable (provided with the kit) to the connectors labeled J1 and J2 on the simulator board. Attach the other end of the cable to the appropriate connector on the target system. Target head adapters are available for the 64-pin QFP 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. An MCU must be installed in the target system, and there should be no MCU installed in the ABICS.

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 MC68HC908AB32 MCU.

accumulator — An 8-bit register of the MC68HC908AB32 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 MC68HC908AB32. 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 MC68HC908AB32. 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 MC68HC908AB32, 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 MC68HC908AB32. 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 MC68HC908AB32.

index register — An 8-bit CPU register in the MC68HC908AB32 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 MC68HC908AB32. 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 MC68HC908AB32 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 MC68HC908AB32.

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 MC68HC908AB32 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 MC68HC908AB32 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 MC68HC908AB32.

Z — Abbreviation for zero, a bit in the condition code register of the MC68HC908AB32. 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.

Index

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/

MOTOROLA