M68ICS08JBUM/D

M68ICS08JB

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

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 — M68ICS08JB In-Circuit Simulator

List of Figures

List of Tables

Section 1. General Information

1.1 Introduction

This section provides general information about the Motorola M68ICS08JB in-circuit simulator (JBICS).

The M68ICS08JB JBICS 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 MC68HC908JB8 microcontroller unit (MCU).

The JBICS 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 JBICS 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, a MONO8 cable, or one of two DIP emulation cables (low cost alternatives to the flex cable). The ribbon cable or flex cable or DIP cable is used when an MCU is resident on the JBICS for emulation or simulation, and the MONO8 cable is used to debug or program a target system’s MCU, directly, when the MCU resides on the target hardware.

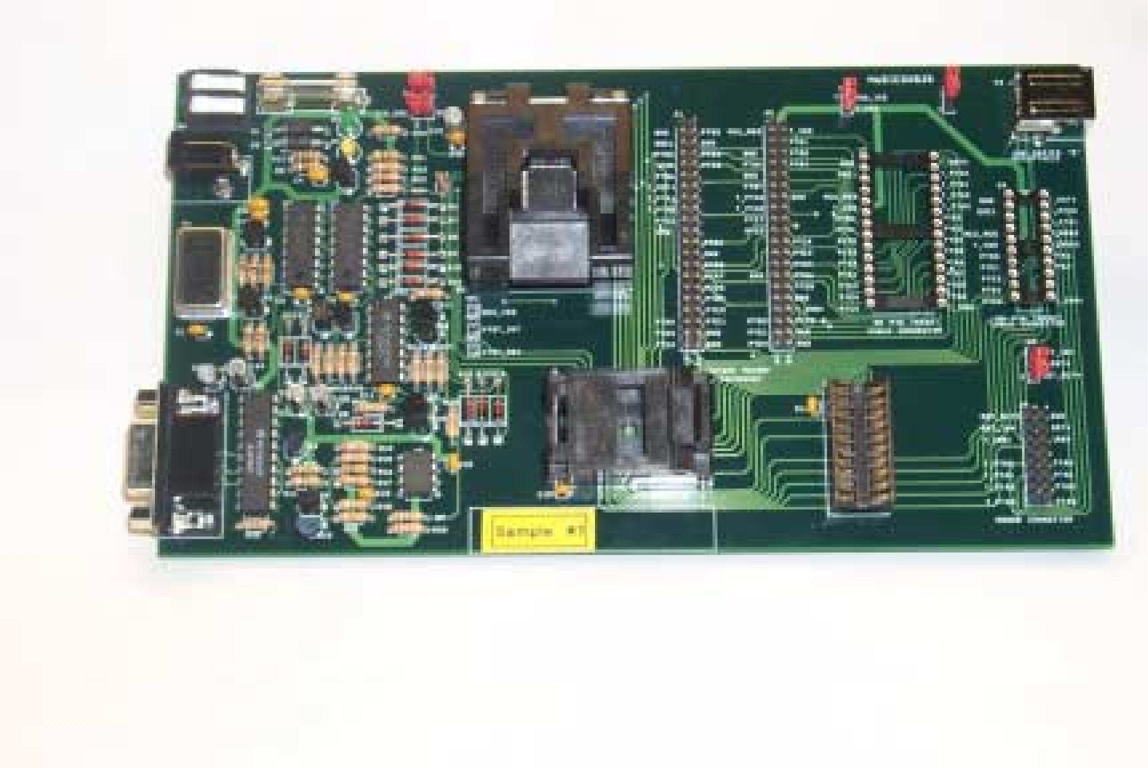


Figure 1-1 M68ICS08JB In-Circuit Simulator Board

The JBICS 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 MC68HC908JB MCUs with ICS08JBZ 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 JBICS as a standalone system without a target board
  - Simulation — using the JBICS as an in-circuit simulator/emulator with a target cable
  - Evaluation - using the JBICS for real-time evaluation of the MCU and to debug user developed hardware and software
  - Programming — using the JBICS 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 ICS08JBZ 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 ICS08JBZ 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 6.0 MHz crystal

1.2 JBICS Components

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

Table 1-1 JBICS Product Components

Part Number	Description
ICS08JB	JBICS software development package
ICS08JBZ	JBICS simulator
ICD08SZ	JBICS debugger/emulation
MC68HC908JB8FB	MCU (44-pin QFP package)
MC68HC908JB8ADW	MCU (SOIC Package)
MC68HC908JB8JP	MCU (PDIP Package)
M68CLB05C	Flex target cable
KRISTA 22-122	Serial cable
FRIWO 11.8999-P5	Power supply
M68ICS08JB	Hardware board
M68ICS08SOM/D	*M68ICS08JB In-circuit Simulator Software Operator's Manual*
M68ICS08JBHOM/D	*M68ICS08 In-circuit Simulator Hardware Operator's Manual*

1.2.1 JBICS Hardware

Table 1-2 lists the JBICS hardware components.

Table 1-2. Hardware Connector Components

Components	Description
XU1	Clam shell test socket for Motorola MC68HC908JB8 MCU; 64-pin QFP (quad flat pack)
XU2	28-pin SOIC test socket for Motorola MC68HC908JB8 MCU
XU3	20-pin PDIP test socket for Motorola MC68HC908JB8
J1 & J2	Two 2-row × 20-pin, 0.1-inch spacing connectors to connect JBICS to a target using M68CLB05C flex cable
J3	Connector to connect 28-pin DIP emulation cable between JBICS and target.
J4	Connector to connect 20-pin DIP emulation cable between JBICS and target.
J5	One 2-row × 8-pin, 0.1-inch spacing connector to connect to target via MON08 debug circuit.
P1	+5 Vdc input voltage (V_{DD})
P2	RS-232 to interface JBICS to host computer serial connector (DEKL-9SAT-F)
P3	Power Terminal
P4	USB Series "B" Receptable to interface JBICS to host computer

1.2.2 ICS Interface Software

Windows-optimized software components are referred to, collectively, as the JBICS software (part number ICS08JB). It is a product of *P&E Microcomputer Systems, Inc.*, and is included in the JBICS 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
ICSO8SZ.EXE	In-circuit/stand-alone simulator software for the MC68HC908JB8 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 JBICS 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 JBICS and the host computer

1.4 Specifications

Table 1-4 summarizes the JBICS hardware specifications.

Table 1-4 JBICS 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 M68ICS08JB (JBICS) for use.

2.2 Hardware Preparation

This paragraph explains:

- Limitations of the JBICS
- Configuration of the JBICS
- Installation of the JBICS
- Connection of the JBICS 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 JBICS Limitations

These sub-paragraphs describe system limitations of the JBICS.

2.2.1.1 Port Bit PTA0

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

2.2.1.2 DDRA0

Setting DDRA0, in the Data Direction Register, 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 PTA1, PTA2, and PTA3

Port bits PTA1, PTA2, and PTA3 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 MONO8 cable.

2.2.2 Configuring JBICS Jumper Headers

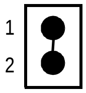
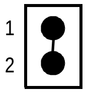
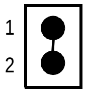
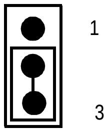
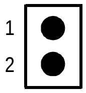
The JBICS supports four configuration options: standalone, simulation, evaluation, and programming.

- Standalone — ICS08JBZ.exe running on the host computer (the JBICS 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 JBICS via the RS-232 cable and ICS08JBZ.exe running on the host computer. This provides access to the M68HC908JB8 MCU, internal registers, and I/O ports.

- Evaluation — Host computer connected to the JBICS and the JBICS 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 JBICS, and the JBICS 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 (port A0 is used for communications, so it is unavailable for emulation).

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

Table 2-1 JBICS Jumper Header Description

Jumper Header	Type (Factory Default Shown)	Description
W1 ICS_OSC or OSC1		**Jumpers W1 and W4 are configured together for the Oscillator Source.** **W1 ON and W4 ON:** JBICS MCU and target board clock signals supplied by the JBICS board oscillator Y1. **W1 ON and W4 OFF:** JBICS MCU clock signal supplied by oscillator Y1 - Target board has its own clock source. **W1 OFF and W4 ON:** JBICS MCU clock signal supplied by target board. **W1 OFF and W4 OFF:** JBICS has no clock source.
W4 OSC1 or TGT_OSC1		
W2 Target System Power		**Jumper:** ICS board system power applied to target cable VDD pin. **No Jumper:** Allows using a separate power supply for target system.
W3 Reset Source		**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.
W5 USB Pull-up Resistor		**Jumper:** Connects 1.5Kohm pull-up resistor from USB D- data line to 3.3V **No Jumper:** 1.5K ohm pull-up is disabled and PTE4/D should only be used as I/O Port

2.2.3 Target Interface Connection Options

There are four ways to connect the JBICS simulator board to your target system:

- Flex cable — low-noise target interface connection
- 20- pin DIP Emulation Cable — low cost flex cable replacement
- 28- pin DIP Emulation Cable — low cost flex cable replacement
- 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 JBICS.

Table 2-2. Cable/Connector Options for MCUs

MCU	Flex Cable	DIP Emulation Cable	DIP Emulation Cable	MON08 Cable
MC68HC90 8JB8	J1 and J2	J3	J4	J5

2.2.4 Host Computer - JBICS Interconnection (P2)

The host computer to JBICS 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 JBICS kit, a DB9-male-to-female, 6-ft. (3 m) long serial cable.

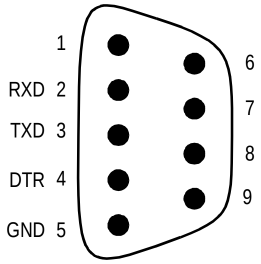


Figure 2-1. P2 Host Computer to JBICS Interconnection

2.2.5 Power Connector (P1)

Connect +5-Vdc power directly to the JBICS via connector P1 (**Figure 2-2**) using the provided power supply.

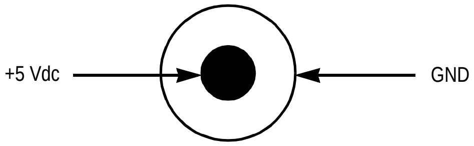


Figure 2-2. P1 Power Connector

2.2.6 USB Connector

The USB connector is directly connected to the MCU USB interface through the USB data pins (D+ and D-). If the development starts without the target board, the user can connect the USB interface to the host by using a USB standard detachable cable (Series "A" plug to Series "B" plug). In this case, a shunt should be place in the jumper W5 to connect the 1.5Kohm pull-up resistor.

Refer to Section 3, Support Information for pin assignment information.

2.3 Connecting the JBICS

The following steps provide instructions for connecting the JBICS 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 (common).*

a. Configure the jumpers W-1 through W-5 (Table 2-1) on the JBICS for your application.

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

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 JBICS board.

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

g. The JBICS power LED (green) lights.

2.4 Connecting the JBICS to the Target System

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

- Emulating using a flex cable for low-noise connection

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.

- Emulating using a 20-pin DIP emulation cable

  When emulating, connect the 20-pin DIP cable to the connector labeled J4 on the simulator board. Attach the other end of the cable to the appropriate connector on the target system. Target head adapters are available.

- Emulating using a 28-pin DIP emulation cable

  When emulating, connect the 28-pin DIP cable to the connector labeled J3 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 JBICS.*

Connect the MON08 debug interface cable to the appropriate MON08 debug interface connector, J5, 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 can be useful in the design, installation, and operation of your application.

3.2 MCU Subsystem

The MCU subsystem consists of the MC68HC908JB8 microcontroller, clock generation and selection, monitor mode control logic that places and holds the JBICS 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 three operating modes:

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

3.3 JBICS Functional Description

NOTE: *For the following discussion on the theory of operation of the JBICS, refer to the schematic diagrams in Paragraph 3.6, Figures 3-2 to 3-9 of this section.*

3.3.1 ICS08JB Board

The core component of the board is the MC68HC908JB8 MCU (Figure 3-5). This MCU resides either on the ICS08JB board or on a target system.

When the MCU resides on the JBICS, the board may be used as an in-circuit emulator or simulator for the MC68HC908JB8. For this configuration, a target cable is run from the board to the target system.

The JBICS supports three kinds of target cables:

A 20-pin ribbon cable (Figure 3-8) terminated in 20-pin male DIP (Dual In-line Package) headers

A 28-pin ribbon cable (Figure 3-8) terminated in 28-pin male DIP headers

A flexible target head adapter cable (Figure 3-7) (Motorola part number M68CBL05C), terminating in connectors for target head adapter. For a 44-pin QFP-package MCU on the target system, use Motorola THA model number M68TC08JBFB44.

Using a target cable is recommended but optional; the board may be utilized with flying leads to other circuits. The MCU may be any one of the MC68HC908JB8FB, MC68HC908JB8DW, or MC68HC908JB8JP MCUs. On the JBICS board, socket XU1 supports the 44-pin QFP package, socket XU2 supports the 28-pin SOIC package, and socket XU3 supports the 20-pin DIP package.

When the MCU resides on a target system, the ICS08JB board can communicate with the MCU over a 16-pin MON08 cable. The MON08 interface is intended for in-system debugging and programming of an MCU in the target system.

The ICS08JBZ simulation software simulates the operation of an MCU in the PC while communicating with an external MCU to provide I/O functions. The PC executes code simulating the MCU, then sends or requests MCU port data, allowing for a real- world interface for the in-circuit simulator. The simulated MCU operation is much slower than the actual MCU performance, but the simulator allows the actual processing steps to seen and followed, while sill allowing the MCU to interface to all necessary signals within the target system.

The PROG08SZ programming software is used to program the FLASH memory on an MCU. Only one part may be programmed at a time. The MCU to be programmed may be socketed on the JBICS, or it may reside on a target board that sypports MON08.

The JBICS board also provides +5 Vdc power, +8 Vdc power for the VTST_INT and VTST_RST voltage required to enter monitor mode, a 6MHz clock signal, and host PC RS-232 level translation.

When using the ICD08SZ debugging software, your code can be run directly out of the MCU's internal FLASH at real-time speeds.

3.3.2 Clock Selection

The JBICS contains a 6-MHz crystal oscillator (Figure 3-3). When the remote target connection is made, the user may opt to feed the output from the JBICS crystal (MCU_OSC) to the external clock input (OSC1) of the JBICS via W4, a 2-pin shunt (Figure 3-7).

3.3.3 Board Reset

The JBICS includes two reset sources:

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

The host system resets the JBICS (Figure 3-14) 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 JBICS is both an input and an output. The JBICS drives its RESET pin low after encountering several different exception conditions. W3 (Figure 2-7) is provided to allow you to select whether the target system can reset the MCU on the JBICS (jumper between pins 1 and 2) or whether the target system receives a reset signal from the JBICS (jumper between pins 2 and 3).

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

3.3.4 Device Configuration Selection

The operation mode of the JBICS processor is selected at the rising edge of the RESET signal. The JBICS requires that the processor operate in monitor mode. To set monitor mode operation, the IRQ* line to the JBICS 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 8.5V, with the highest V_{DD} of 5V, which gives a range of minimum 7.5V and maximum 8.5V.

The JBICS RST* pin is the main mode select input and is pulled to logic 0, then logic 1 (processor V_{DD}), to select 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 mode: read, write, iread, iwrite, readsp, and run. Each command is echoed back through PTA0 for error checking. These commands are described in the *M68ICS08JB IN-circuit Simulator Software Operator’s Manual.*

3.3.5 Level Translation

The JBICS has two operation voltage +3.3 and +5.0 volts while the host development system interface is an RS-232 (com) port. U1 (Figure 3-3) on the ICS converts 5 V logic signals to RS-232 levels. U6 on the ICS converts the 5V to 3.3V. Some transistors are used to translate 5 V logic levels to the MCU operating voltage (3.3V).

3.3.6 Host System Connector

The host system interface is via a 9-pin DB-9 serial connection plug, P2, DEKL-9SAT-F.

3.3.7 USB Interface

A USB series B receptacle, P4, (Figure 3-5) is built in the JBICS board. It can connect to the USB host by USB standard detachable cable (Series “A” plug to Series “B” plug)

3.4 JBICS Connector Signal Definitions

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

3.4.1 Target Flex Cable Interface Connectors J1 and J2

A generic cable (Motorola part number M68CLB05C) 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.

.

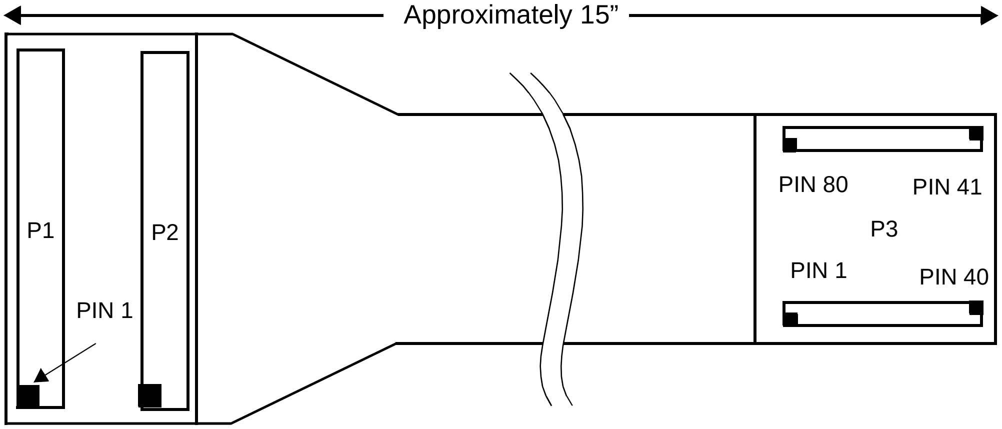


Figure 0-1. FLEX Cable

Table 3-1 J1 Target Flex Connector Pin Assignment Descriptions

Pin No.	Schematic	Direction	Signal Description
1	GND		
2	PTB2	Bidirectional	General Purpose I/O
3	TGT_OSC1	Bidirectional	Clock Signal - JBICS W4 jumper settings determine direction
4	GND		
5	PTB3	Bidirectional	General Purpose I/O
6	PTD0	Bidirectional	General Purpose I/O
7	PTB5	Bidirectional	General Purpose I/O
8	PTD2	Bidirectional	General Purpose I/O
9	PTB6	Bidirectional	General Purpose I/O
10	N/C		
11	TGT_RST	Bidirectional	Logical-level reset - W3 determines direction
12	N/C		
13	TGT_PTA1	Bidirectional	General Purpose I/O
14	N/C		
15	TGT_PTA3	Bidirectional	General Purpose I/O
16	N/C		
17	PTC6	Bidirectional	General Purpose I/O
18	N/C		
19	GND		
20	N/C		
21	N/C		
22	PTE0	Bidirectional	General Purpose I/O
23	N/C		
24	GND		
25	N/C		
26	PTA5	Bidirectional	General Purpose I/O
27	N/C		

Pin No.	Schematic	Direction	Signal Description
28	PTA7	Bidirectional	General Purpose I/O
29	N/C		
30	PTD6	Bidirectional	General Purpose I/O
31	N/C		
32	PTD7	Bidirectional	General Purpose I/O
33	N/C		
34	PTC3	Bidirectional	General Purpose I/O
35	PTD4	Bidirectional	General Purpose I/O
36	PTC1	Bidirectional	General Purpose I/O
37	PTE3	Bidirectional	General Purpose I/O
38	GND		
39	PTC0	Bidirectional	General Purpose I/O
40	GND		

Table 3-2 J2 Target Flex Connector Pin Assignment Descriptions

Pin No.	Schematic	Direction	Signal Description
1	MCU-REG	Out	3.3V output of on-chip voltage regulator
2	TGT-VDD	I/O	5V JBICS W2 jumper settings determine if signal is input or output
3	N/C		
4	PTB1	Bidirectional	General Purpose I/O
5	GND		
6	PTB0	Bidirectional	General Purpose I/O
7	PTB4	Bidirectional	General Purpose I/O
8	PTD1	Bidirectional	General Purpose I/O
9	GND		
10	N/C		
11	PTB7	Bidirectional	General Purpose I/O
12	N/C		

Pin No.	Schematic	Direction	Signal Description
13	TGT-PTA0	Bidirectional	General Purpose I/O
14	GND		
15	TGT_PTA2	Bidirectional	General Purpose I/O
16	N/C		
17	PTC7	Bidirectional	General Purpose I/O
18	N/C		
19	PTC5	Bidirectional	General Purpose I/O
20	N/C		
21	N/C		
22	PTC4	Bidirectional	General Purpose I/O
23	N/C		
24	PTE2	Bidirectional	General Purpose I/O
25	N/C		
26	PTA4	Bidirectional	General Purpose I/O
27	GND		
28	PTA6	Bidirectional	General Purpose I/O
29	N/C		
30	PTD5	Bidirectional	General Purpose I/O
31	N/C		
32	GND		
33	N/C		
34	TGT_IRQ*	In	Interrupt Request from Target
35	PTD3	Bidirectional	General Purpose I/O
36	PTC2	Bidirectional	General Purpose I/O
37	PTE1	Bidirectional	General Purpose I/O
38	GND		
39	PTE4	Bidirectional	General Purpose I/O
40	GND		

Table 3-3 Flex Cable Plug Pin Assignments

Signal	ICS08JB8 Connector P1 Pin Number	ICS08JB8 Connector P2 Pin Number	Target Head Adapter Pin Number P3
TGT_VDD	NA	2	1
MCU_REG	NA	1	2
PTB2	2	NA	3
GND	1	NA	4
PTB1	NA	4	5
NC	NA	3	6
GND	4	NA	7
OSC1	3	NA	8
PTB0	NA	6	9
GND	NA	5	10
PTD0	6	NA	11
PTB3	5	NA	12
PTD1	NA	8	13
PTB4	NA	7	14
PTD2	8	NA	15
PTB5	7	NA	16
NC	NA	10	17
GND	19	NA	18
NC	10	NA	19
PTB6	9	NA	20
NC	NA	12	21
PTB7	NA	11	22
NC	12	NA	23
TGT_RST	11	NA	24
GND	24	NA	25
TGT_PTA0	NA	13	26

Signal	ICS08JB8 Connector P1 Pin Number	ICS08JB8 Connector P2 Pin Number	Target Head Adapter Pin Number P3
NC	14	NA	27
TGT_PTA1	13	NA	28
NC	NA	16	29
TGT_PTA2	NA	15	30
NC	16	NA	31
TGT_PTA3	15	NA	32
NC	NA	18	33
PTC7	NA	17	34
NC	18	NA	35
PTC6	17	NA	36
NC	NA	20	37
PTC5	NA	19	38
NC	20	NA	39
GND	38	NA	40
NC	NA	21	41
PTC4	NA	22	42
NC	21	NA	43
PTE0	22	NA	44
NC	NA	23	45
PTE2	NA	24	46
NC	23	NA	47
GND	40	NA	48
NC	NA	25	49
PTA4	NA	26	50
NC	25	NA	51
PTA5	26	NA	52
GND	NA	9	53
PTA6	NA	28	54

Signal	ICS08JB8 Connector P1 Pin Number	ICS08JB8 Connector P2 Pin Number	Target Head Adapter Pin Number P3
NC	27	NA	55
PTA7	28	NA	56
NC	NA	29	57
PTD5	NA	30	58
PTD6	30	NA	60
NC	29	NA	59
NC	NA	31	61
GND	NA	14	62
NC	31	NA	63
PTD7	32	NA	64
NC	NA	33	65
TGT_IRQ	NA	34	66
NC	33	NA	67
PTC3	34	NA	68
PTD3	NA	35	69
PTC2	NA	36	70
PTD4	35	NA	71
PTC1	36	NA	72
PTE1	NA	37	73
GND	NA	27	74
PTE3	37	NA	75
GND	NA	32	76
PTE4	NA	39	77
GND	NA	38	78
PTC0	39	NA	79
GND	NA	40	80

3.4.2 J3 DIP Connector Pin Assignments

Table 3-4 J3 Connector Pin Assignments

Pin No.	Schematic	Direction	Signal Description
1	GND	Gound	MCU ground
2	TGT_OSC1	I/O	Clock Signal - ICS08JB jumper settings determine if signal is an input or an output W4
3	None	None	None
4	MCU_REG	Output	3.3V ouput of the on-chip voltage regulator
5	TGT_VDD	I/O	5V ICS08JB jumper settings determine if signal is an input or and output W2
6	PTD0	I/O	I/O Port from MCU
7	PTD1	I/O	I/O Port from MCU
8	PTD2	I/O	I/O Port from MCU
9	PTD3	I/O	I/O Port from MCU
10	PTD4	I/O	I/O Port from MCU
11	PTE1	I/O	I/O Port from MCU
12	PTE3	I/O	I/O Port from MCU
13	PTE4	I/O	I/O Port from MCU
14	PTC0	I/O	I/O Port from MCU
15	TGT_IRQ	I/O	Interrupt Request from Target
16	PTD6	I/O	I/O Port from MCU
17	PTD5	I/O	I/O Port from MCU
18	PTA7	I/O	I/O Port from MCU
19	PTA6	I/O	I/O Port from MCU
20	PTA5	I/O	I/O Port from MCU
21	PTA4	I/O	I/O Port from MCU
22	PTA5	I/O	I/O Port from MCU

Pin No.	Schematic	Direction	Signal Description
23	PTE0	I/O	I/O Port from MCU
24	PTA3	I/O	I/O Port from MCU
25	PTA2	I/O	I/O Port from MCU
26	PTA1	I/O	I/O Port from MCU
27	PTA0	I/O	I/O Port from MCU
28	TGT_RST	I/O	Logic-level reset - W3 determines if this signal is an input or an output

3.4.3 J4 DIP Connector Pin Assignments

Table 3-5 J4 Connector Pin Assignments

Pin No.	Schematic	Direction	Signal Description
1	GND	Gound	MCU ground
2	TGT_OSC1	I/O	Clock Signal - ICS08JB jumper settings determine if signal is an input or an output W4
3	None	None	None
4	MCU_REG	Output	3.3V ouput of the on-chip voltage regulator
5	TGT_VDD	I/O	5V ICS08JB jumper settings determine if signal is an input or and output W2
6	PTD0	I/O	I/O Port from MCU
7	PTE1	I/O	I/O Port from MCU
8	PTE3	I/O	I/O Port from MCU
9	PTE4	I/O	I/O Port from MCU
10	PTC0	I/O	I/O Port from MCU
11	TGT_IRQ	I/O	Interrupt Request from Target
12	PTA7	I/O	I/O Port from MCU
13	PTA6	I/O	I/O Port from MCU
14	PTA5	I/O	I/O Port from MCU

Pin No.	Schematic	Direction	Signal Description
15	PTA4	I/O	I/O Port from MCU
16	PTA3	I/O	I/O Port from MCU
17	PTA2	I/O	I/O Port from MCU
18	PTA1	I/O	I/O Port from MCU
19	PTA0	I/O	I/O Port from MCU
20	TGT_RST	I/O	Logic-level reset - W3 determines if this signal is an input or an output

3.4.4 Target MONO8 Interface Connector J5 Pin Assignments

The MONO8 interface connector, J3 (Table 2-6), is used when the MCU is mounted on the target. Refer to **Section 4 Using the MON08** for detailed information.

Table 3-6 J3 MONO8 Target Connector Pin Assignment Descriptions

Pin No.	Schematic	Direction	Signal Description
1	RST_OUT*	Out	Reset signal to target
2	GND		
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	N/C		
8	N/C		
9	TGT_PTA0	Bidirectional	General purpose I/O
10	PTA0	Bidirectional	General purpose I/O

Pin No.	Schematic	Direction	Signal Description
11	TGT_PTA1	Bidirectional	General purpose I/O
12	PTA1	Bidirectional	General purpose I/O
13	TGT_PTA2	Bidirectional	General purpose I/O
14	PTA2	Bidirectional	General purpose I/O
15	TGT-PTA3	Bidirectional	General purpose I/O
16	PTA3	Bidirectional	General purpose I/O

3.4.5 Power Connector (P1)

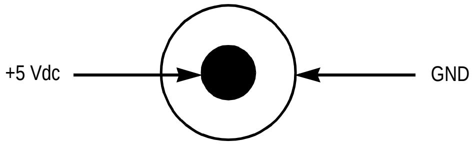


Figure 3-1. P1 Power Connector

Table 3-7 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 JBICS logic circuits
2	GND	Common
3	GND	Common

3.4.6 Host Computer - JBICS Interconnection (P2)

The host computer to JBICS 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 3-6).

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

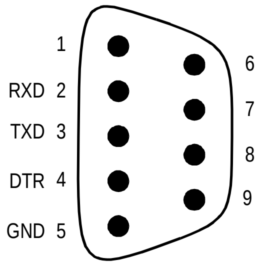


Figure 3-2. P2 Host Computer to JBICS Interconnection

Table 3-8 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

Table 3-9 USB Connector P4 Pin Assignment Descriptions

Pin No.	Mnemonic	Signal
1	VDDS	N/C
2	D -	PTE4/D- - Bidirectional General Purpose I/O or USB data pin
3	D +	PTE3/D+ - Bidirectional General Purpose I/O or USB data pin
4	GND	Common

3.5 Parts List

Table 3-10. JBICS Parts List (Sheet 1 of 2)

Reference Designator	Description	Manufacturer	Part Number
C12	Capacitor, 82pF, ceramic	AEC	Z5U-820
C2, C3, C7, C8, C19	Capacitor, 10uF	Truth	85C +80-20%
C1, C5, C9, C10, C11, C13, C14, C15, C16, C17, C18, C20	Capacitor, 0.1 uF	AVX	SR215E104MAA
D1	1A 20V Schottky Rectifier	MOTOROLA	1N5817
D2	Zener Transient Voltage Suppressors	MOTOROLA	SA 5.0A
D3, D4	LED	KINGBRIGHT	L34HT
D5, D6, D7, D8, D9, D10, D11, D12, D13, D14, D15, D16	Diode	SEMTECH	1N4148
F1	FUSE	BUSSMAN	GMA 1.5A 250V
J2, J1	Header, 2x20, 100, Target Head	MOBICON	PHDS-40G1
J3	28-pin DIP socket, 28-pin Target Head	MOBICON	28PMC
J4	20-pin DIP socket, 20-pin Target Head	MOBICON	20PMC
J5	Header, 2x8, 100, MONO8	MOBICON	PHDS-16G1
P1	Power Jack	WEALTH	DS-210A
P2	Connector DB9	MOBICON	DB9SR
P3	Power Terminal	RDI	2SV-02
P4	USB series B receptacle	BETAMAX	USB-022
Q1	Transistor, NPN	MOTOROLA	MPS2369A
Q2, Q3, Q4, Q8, Q9, Q10, Q11, Q12,	Transistor, NPN	MOTOROLA	BC547B
Q5	Transistor, NPN	MOTOROLA	2N3906
Q6	Transistor, PNP	MOTOROLA	BC557B
R1, R5, R23, R29	Resistor, 1K, 5%	YAGEO/DIGIKEY	CR-25-B-1K

Table 3-10. JBICS Parts List (Sheet 2 of 2)

Reference Designator	Description	Manufacturer	Part Number
R2	Resistor, 330R, 5%	YAGEO/DIGIKEY	CR-25-B-330
R3, R4	Resistor, 1K, 5%	YAGEO/DIGIKEY	CR-25-B-47K
R8, R9, R10, R11, R12, R13, R14, R15, R17, R18, R19, R22, R24, R25, R26, R27, R28, R30, R32, R33, R34, R35, R36	Resistor, 10K, 5%	YAGEO/DIGIKEY	CR-25-B10K
R16	Resistor, 100K, 5%	YAGEO/DIGIKEY	CR-25-B-100K
R20, R21	Resistor, 470R, 5%	YAGEO/DIGIKEY	CR-25-B-470
R37	Resistor, 1K5, 5%	YAGEO/DIGIKEY	CR-25-B-1K5
U1	EIA-232/V.28 CMOS Driver/Receiver	MOTOROLA	MC145407P
U2	CMOS Quad 2-input Analog Multiplexer	MOTOROLA	MC14551
U3	Monolithic WFR, Binary Counter	MOTOROLA	MC74HC4020AN
U5	Single Supply Operational Amplifier	MOTOROLA	MC33172P
U6	MicroPower Voltage Regulator	MOTOROLA	LP2950ACZ-3.3
U7	Non-Inverting 3-State Buffer	MOTOROLA	MC74HC125AN
W1, W2, W4, W5	Header, 2x1, 100	3M	2402-6112TG
W3	Header, 3x1, 100	3M	2403-6112TG
XU1	Socket, 68HC908JB8FB44	YAMAICHI	1C51-0444-825
XU2	Socket, 68HC908JB8DW28	YAMICHI	1C51-0282-334
XU3	Socket, 68HC908JB8P20	WELLS-CTI	613-0200316
Y1	6MHz Oscillator	HOSONIC	HO-12C6M

3.6 JBICS Board Layout and Schematic Diagrams

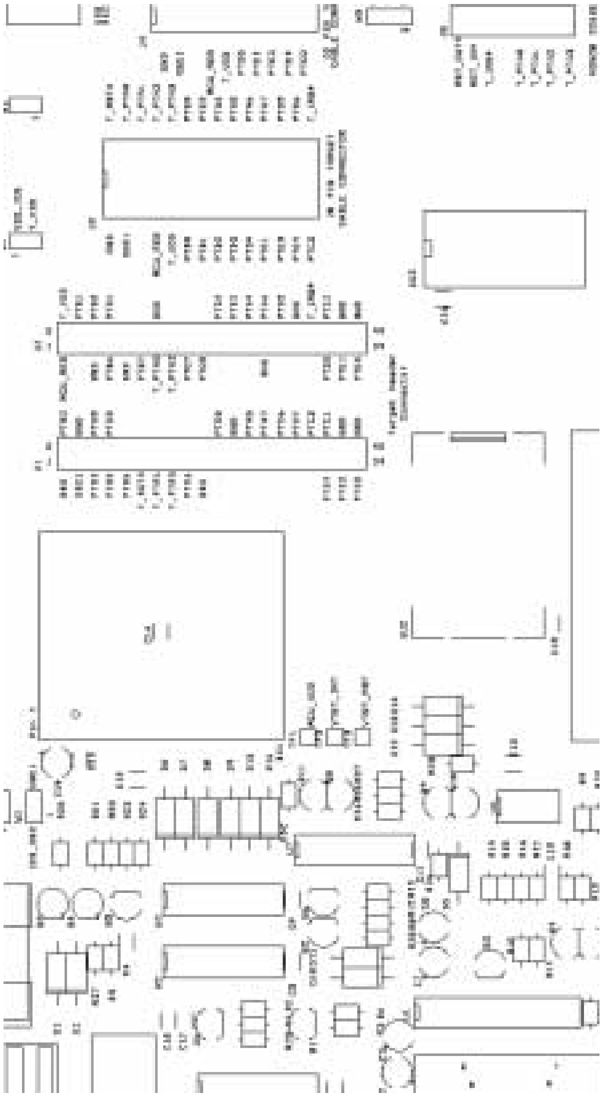


Figure 3-3 JBICS Board Layout

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.

REVISIONS

ZONE	REV	DESCRIPTION	DATE	APPROVED
	0	Original Revision	29-Feb-2000	
	A	Modified after testing	29-Mar-2000	
Page 4	B	Add USB receptacle	10-May-2000	

COMPUTER GENERATED DRAWING : DO NOT REVISE MANUALLY

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

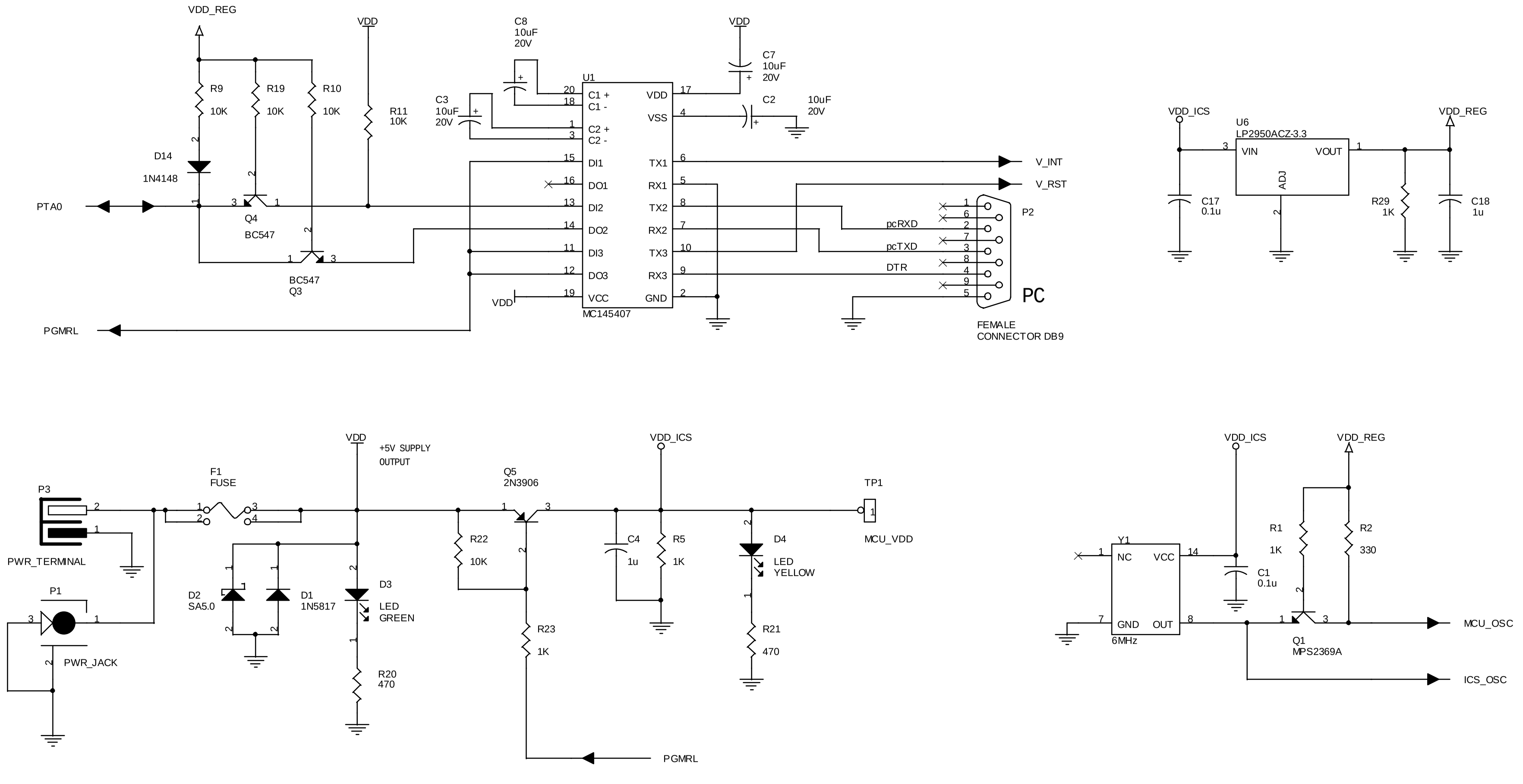
MOTOROLA

DRAWN: Eddie Lee	DATE: 10-May-2000
CHECKED:	DATE:
APPROVED: Edward Chui	DATE: 29-Mar-2000

Title
JB8 In-Circuit Simulator - JBICS

Size	Document Number	Rev
B		B

Sheet 1 of 7

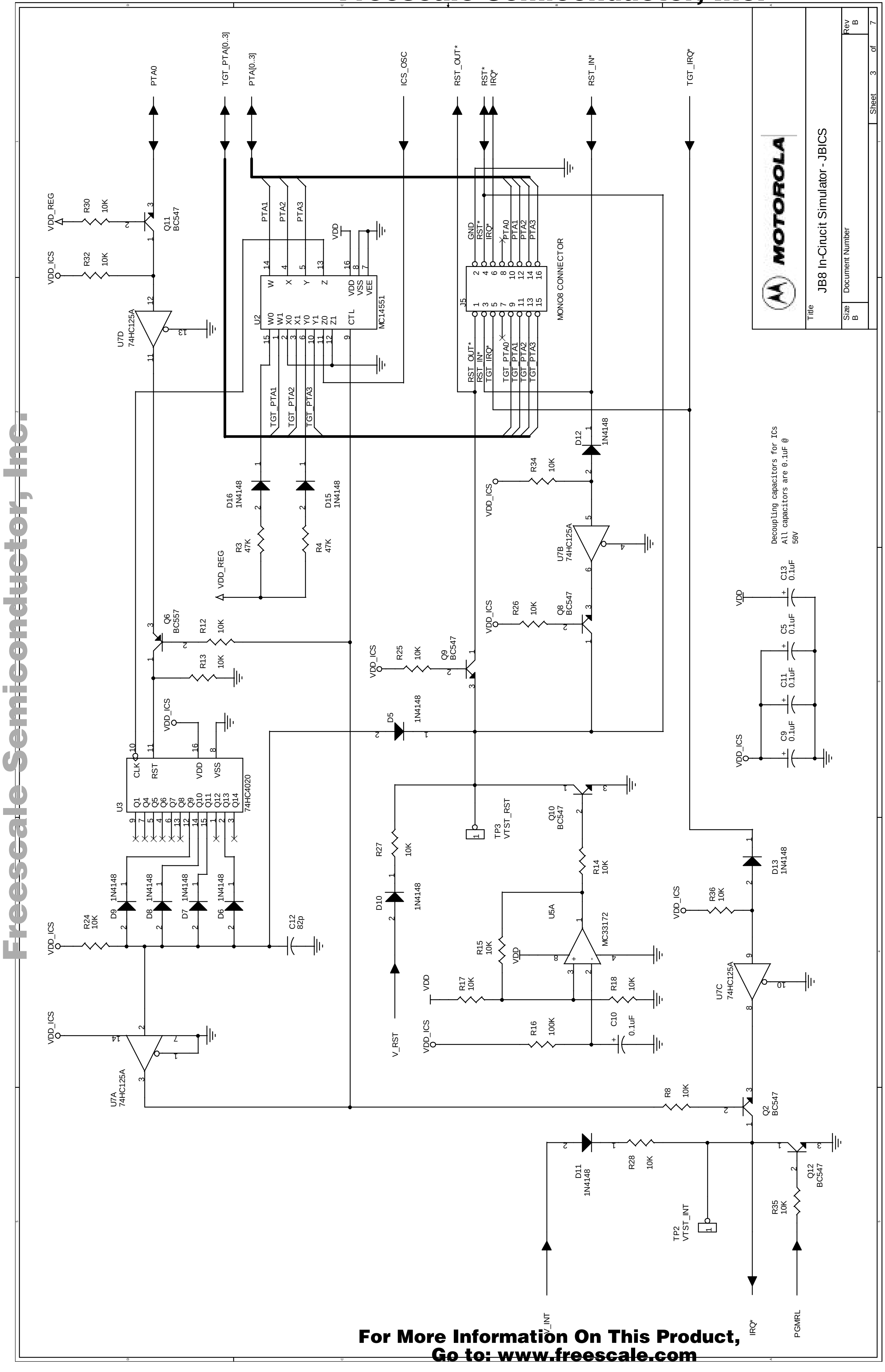


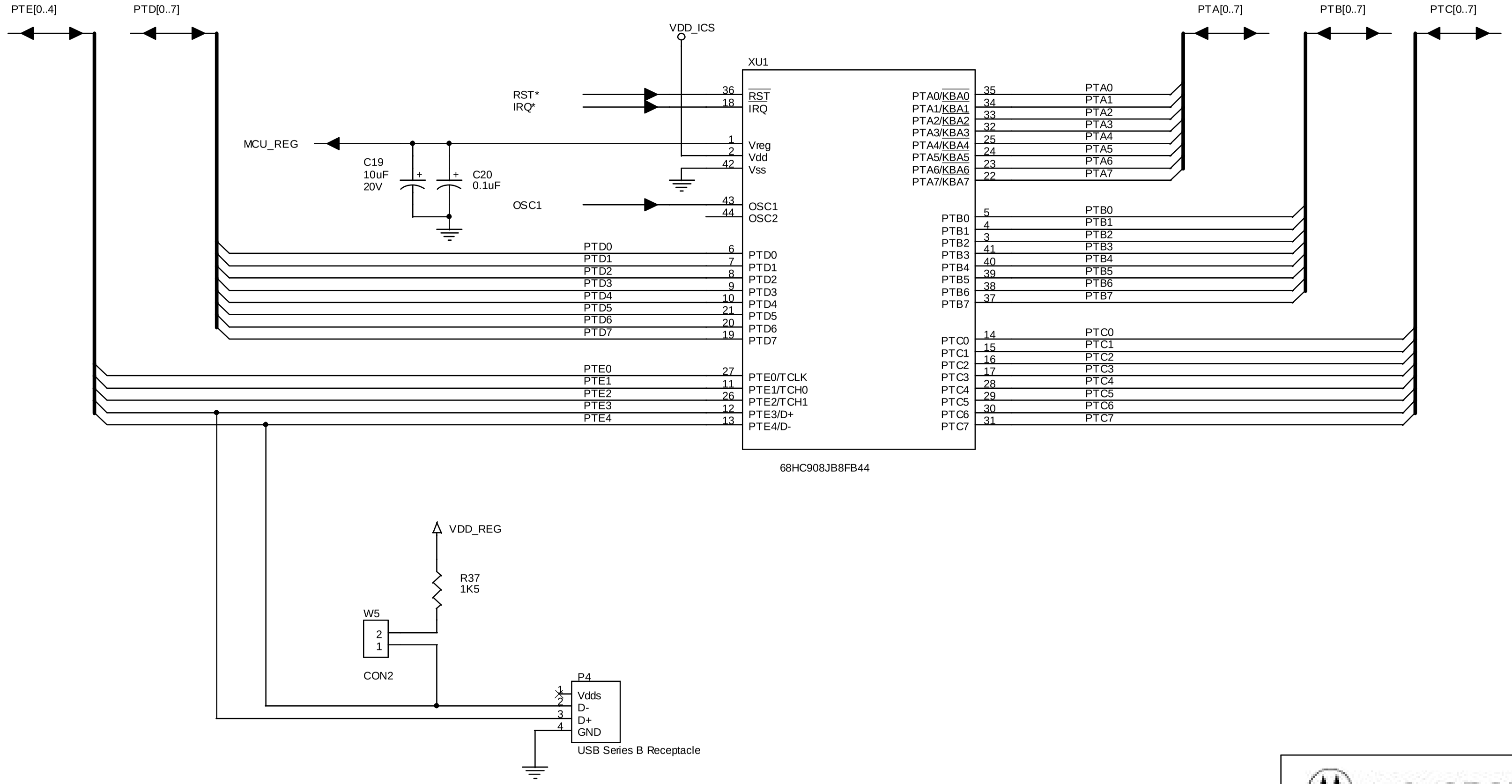
MOTOROLA

Title
JB8 In-Cirucit Simulator - JBICS

Size B | Document Number | Rev B

Sheet 2 of 7


PTA0
TGT_PTA[0..3]
PTA[0..3]
ICS_OSC
RST_OUT*
RST*
IRQ*
RST_IN*
TGT_IRQ*
MOTOROLA
Title
JB8 In-Cirucit Simulator - JBICS
Size
B
Document Number
Rev
B
Sheet 3 of 7
VDD_REG
R30 10K
Q11 BC547
VDD_ICS
R32 10K
U7D 74HC125A
PTA1
PTA2
PTA3
VDD
U2
MC14551
W X Y Z
W0 W1 X0 X1 Y0 Y1 Z0 Z1 CTL
VDD VSS VEE
GND
RST*
IRQ*
PTA0
PTA1
PTA2
PTA3
J5
MONO8 CONNECTOR
RST_OUT*
RST_IN*
TGT_IRQ*
TGT_PTA0
TGT_PTA1
TGT_PTA2
TGT_PTA3
D12 1N4148
R34 10K
VDD_ICS
U7B 74HC125A
D16 1N4148
D15 1N4148
R3 47K
R4 47K
VDD_REG
Decoupling capacitors for ICs
All capacitors are 0.1uF @
50V
C13 0.1uF
C5 0.1uF
C11 0.1uF
C9 0.1uF
VDD
VDD_ICS
Q6 BC557
R12 10K
R13 10K
R26 10K
Q8 BC547
R25 10K
Q9 BC547
VDD_ICS
D5 1N4148
U3
74HC4020
CLK RST VDD VSS
Q1 Q4 Q5 Q6 Q7 Q8 Q9 Q10 Q11 Q12 Q13 Q14
TP3 VTST_RST
Q10 BC547
R27 10K
R14 10K
D10 1N4148
D9 1N4148
D8 1N4148
D7 1N4148
D6 1N4148
R24 10K
C12 82p
U5A
MC33172
R15 10K
VDD
R17 10K
R18 10K
VDD_ICS
R36 10K
D13 1N4148
U7C 74HC125A
V_RST
R16 100K
C10 0.1uF
U7A 74HC125A
R8 10K
Q2 BC547
D11 1N4148
R28 10K
Q12 BC547
TP2 VTST_INT
R35 10K
V_INT
IRQ*
PGMRL

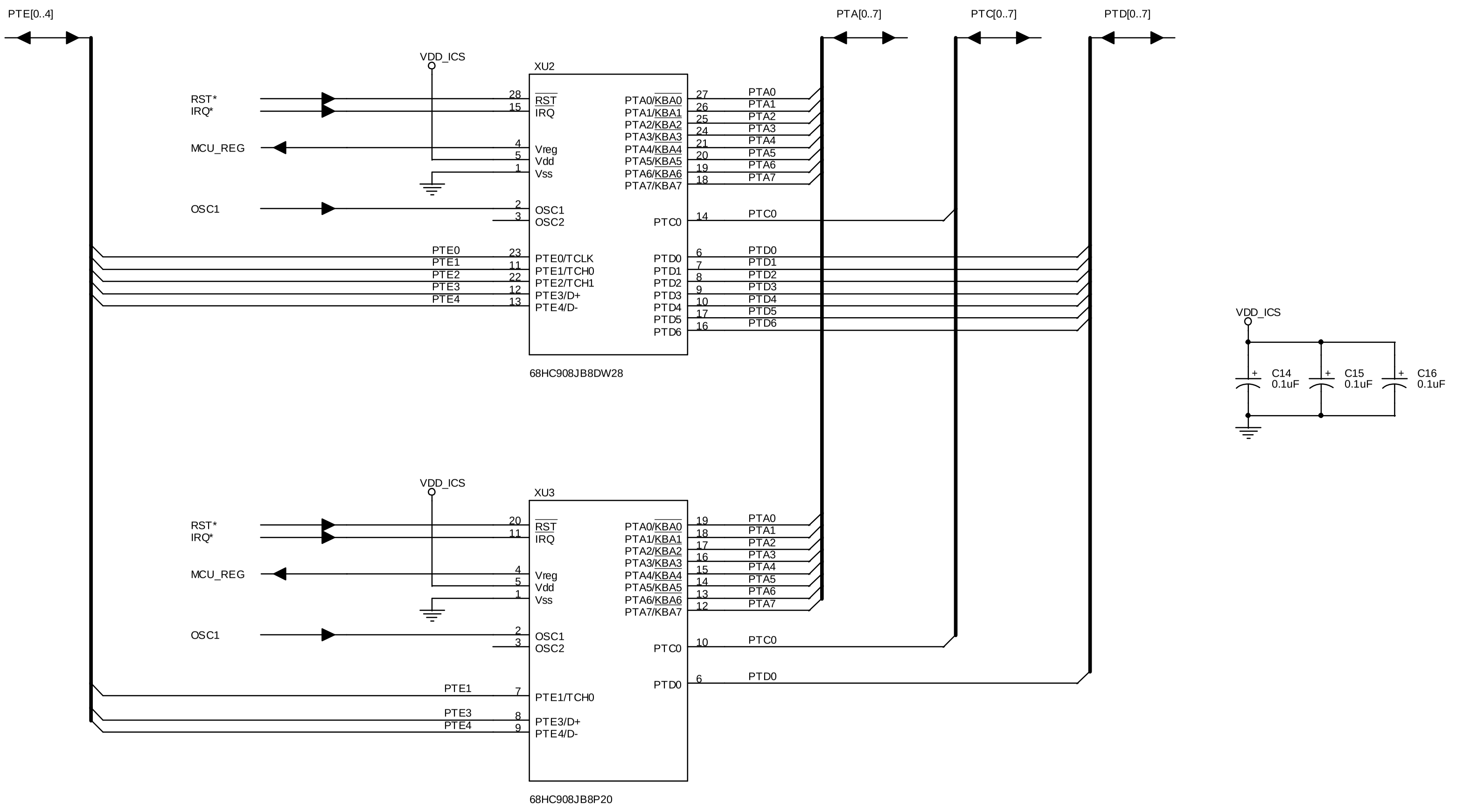


MOTOROLA

Title
JB8 In-Cirucit Simulator - JBICS

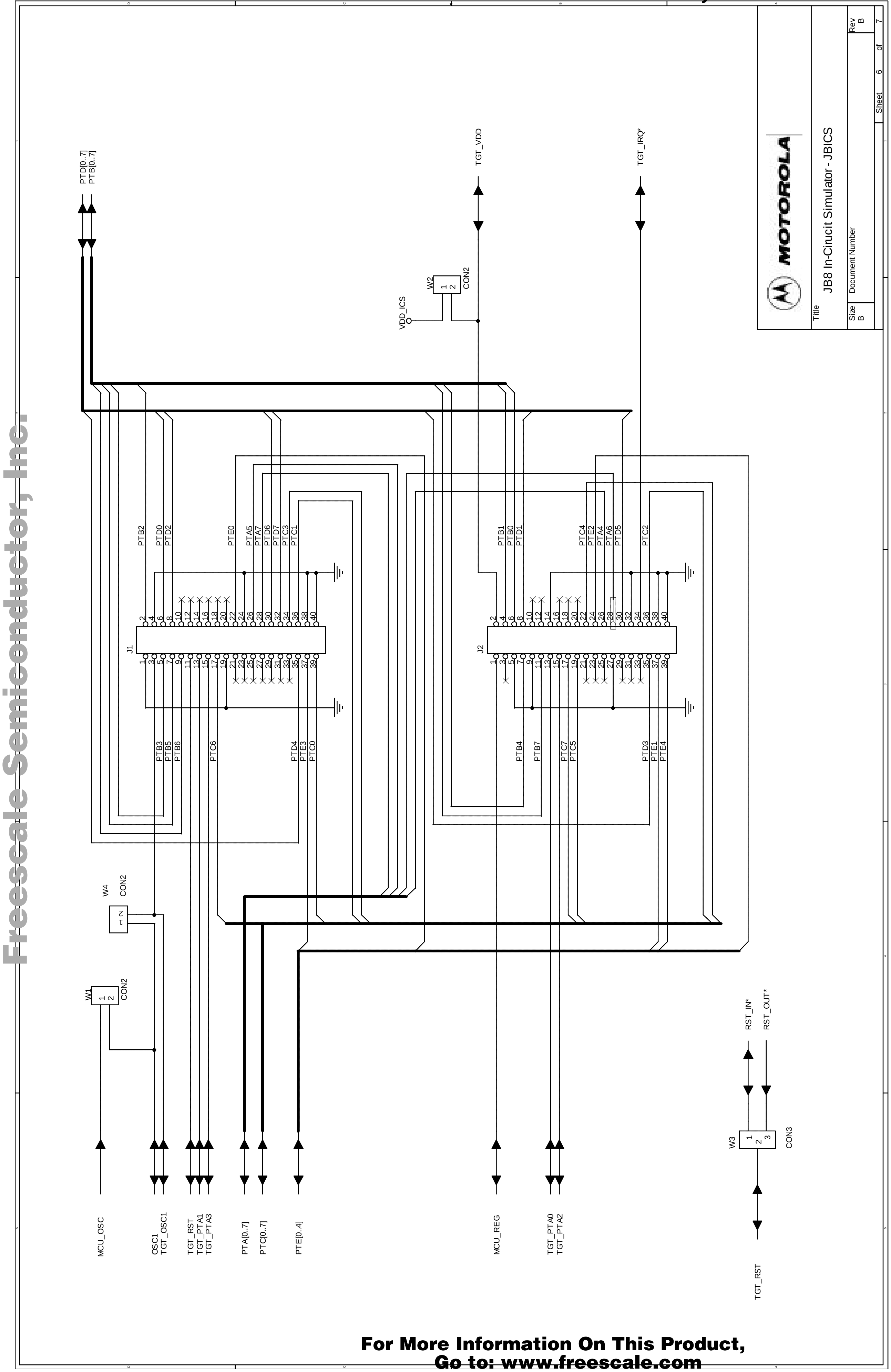
Size	Document Number	Rev
B		B

Sheet 4 of 7

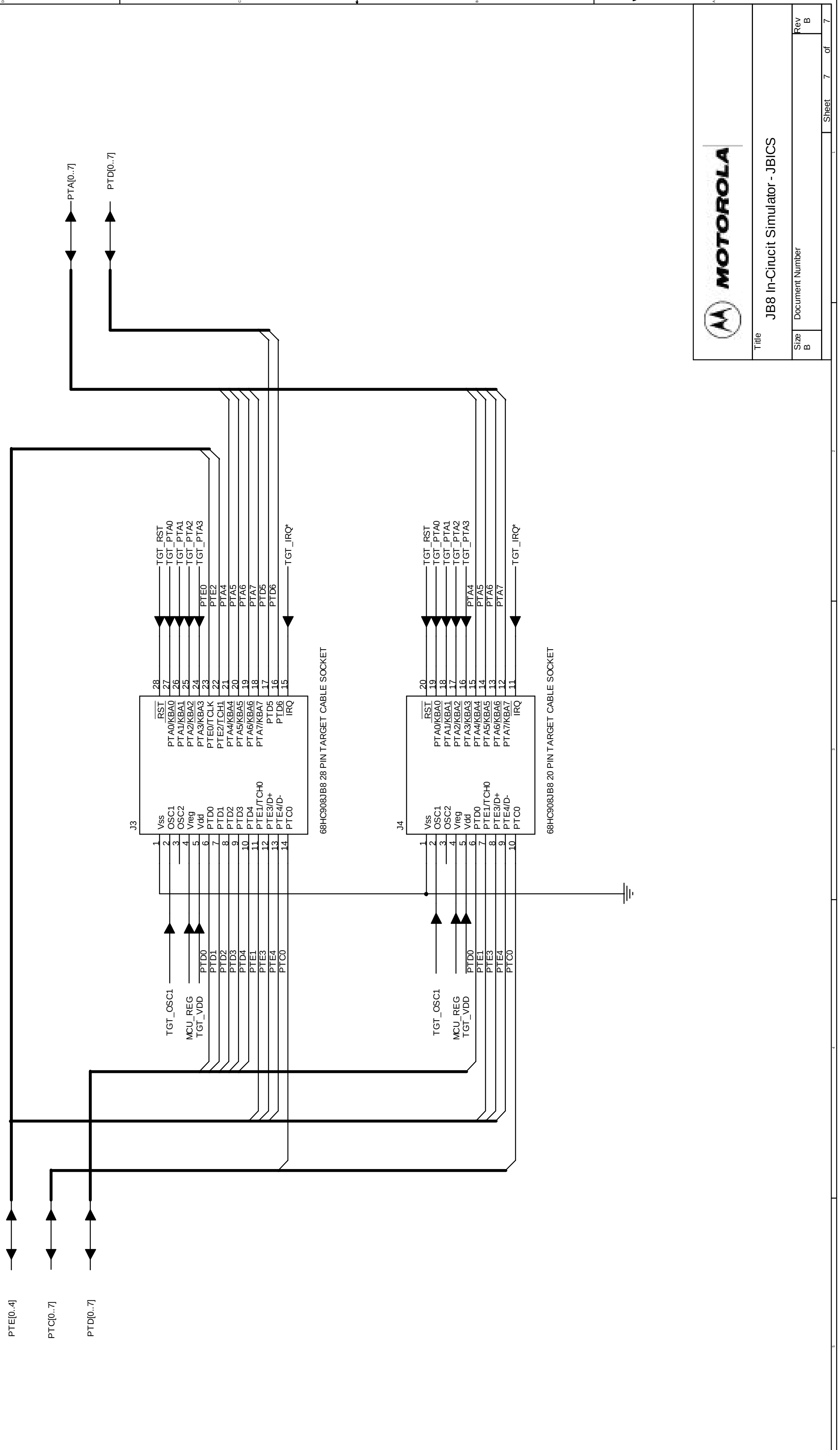


MOTOROLA

Title		
JB8 In-Cirucit Simulator - JBICS		
Size B	Document Number	Rev B
	Sheet 5 of 7	



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


MOTOROLA
Title
JB8 In-Cirucit Simulator - JBICS
Size
B
Document Number
Rev
B
Sheet 7 of 7
PTA[0..7]
PTD[0..7]
TGT_RST
TGT_PTA0
TGT_PTA1
TGT_PTA2
TGT_PTA3
TGT_IRQ*
PTE0
PTE2
PTA4
PTA5
PTA6
PTA7
PTD5
PTD6
J3
Vss
OSC1
OSC2
Vreg
Vdd
PTD0
PTD1
PTD2
PTD3
PTD4
PTE1/TCH0
PTE3/D+
PTE4/D-
PTC0
RST
PTA0/KBA0
PTA1/KBA1
PTA2/KBA2
PTA3/KBA3
PTE0/TCLK
PTE2/TCH1
PTA4/KBA4
PTA5/KBA5
PTA6/KBA6
PTA7/KBA7
PTD5
PTD6
IRQ
1 2 3 4 5 6 7 8 9 10 11 12 13 14
28 27 26 25 24 23 22 21 20 19 18 17 16 15
68HC908JB8 28 PIN TARGET CABLE SOCKET
J4
Vss
OSC1
OSC2
Vreg
Vdd
PTD0
PTE1/TCH0
PTE3/D+
PTE4/D-
PTC0
RST
PTA0/KBA0
PTA1/KBA1
PTA2/KBA2
PTA3/KBA3
PTA4/KBA4
PTA5/KBA5
PTA6/KBA6
PTA7/KBA7
IRQ
1 2 3 4 5 6 7 8 9 10
20 19 18 17 16 15 14 13 12 11
68HC908JB8 20 PIN TARGET CABLE SOCKET
TGT_OSC1
MCU_REG
TGT_VDD
PTD0
PTD1
PTD2
PTD3
PTD4
PTE1
PTE3
PTE4
PTC0
TGT_OSC1
MCU_REG
TGT_VDD
PTD0
PTE1
PTE3
PTE4
PTC0
PTE[0..4]
PTC[0..7]
PTD[0..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 JBICS in-circuit simulator 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.	M68ICS08JB8 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 V.
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 V.
7	N/C		Not connected
8	N/C		Not connected

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

Pin No.	M68ICS08JB8 Label	Direction	Target System Connection
9	TGT-PTA0	Bidirectional	Connect to user circuit that normally would be connected to PTB0 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 PTB0 pin. No other target-system logic should be tied to this signal. Host I/O present on this pin.
11	TGT-PTA1	Bidirectional	Connect to user circuit that normally would be connected to PTA1 on the MCU.
12	PTA1	Bidirectional	Connect to MCU PTA1 pin. No other target-system logic should be tied to this signal. Held at +3.3V during reset.
13	TGT-PTA2	Bidirectional	Connect to user circuit that normally would be connected to PTA2 on the MCU.
14	PTA2	Bidirectional	Connect to MCU PTA2 pin. No other target-system logic should be tied to this signal. Grounded during reset.
15	TGT_PTA3	Bidirectional	Connect to user circuit that normally would be connected to PTA3 on the MCU
16	PTA3	Bidirectional	Connect to MCU PTA3 pin. No other target-system logic should be tied to this signal. Held at 3.3V during reset.

Table 4-2. MON08 Target System Connector P2

Pin No.	M68ICS08GR8 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 V.

4.3 Connecting to the In-Circuit Simulator

Using the 16-pin cable provided with the JBICS kit, connect one end of the cable to the JBICS board at J5. Connect the other end to appropriate connector 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 JBICS board.

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 ICS08JBZ 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 ICS08JBZ
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 M68ICS08JB in-circuit simulator (JBICS) board
- Installation of the hardware
- Connection of the board to a target system

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

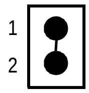
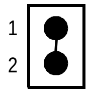
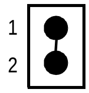
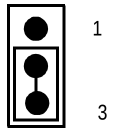
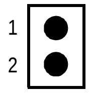
- Standalone — ICS08JBZ.exe is running on the host computer (the JBICS is not connected). Emulation of the M68ICS08JB8 MCU, registers, and I/O ports is within the host computer environment.
- Simulation — Host computer is connected to the JBICS via the RS-232 cable, and the ICS08JBZ.exe is running on the host computer. This provides access to the M68ICS08JB8 MCU, internal registers, and I/O ports.
- Evaluation — Host computer is connected to the JBICS, and the JBICS 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 JBICS, and the JBICS 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 JBICS Configurable Jumper Headers

Configure the five jumper headers on the JBICS for your application according to the tables in this section.

Table 4-3 JBICS Jumper Header Description

Jumper Header	Type (Factory Default Shown)	Description
W1 ICS_OSC or OSC1		**Jumpers W1 and W4 are configured together for the Oscillator Source.** **W1 ON and W4 ON:** JBICS MCU and target board clock signals supplied by the JBICS board oscillator Y1. **W1 ON and W4 OFF:** JBICS MCU clock signal supplied by oscillator Y1 - Target board has its own clock source. **W1 OFF and W4 ON:** JBICS MCU clock signal supplied by target board. **W1 OFF and W4 OFF:** JBICS has no clock source.
W4 OSC1 or TGT_OSC1		
W2 Target System Power		**Jumper:** ICS board system power applied to target cable VDD pin. **No Jumper:** Allows using a separate power supply for target system.
W3 Reset Source		**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.
W5 USB Pull-up Resistor		**Jumper:** Connects 1.5Kohm pull-up resistor from USB D- data line to 3.3V **No Jumper:** USB is disabled and PTE4/D used as I/O port

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 MC68HC908JB8

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

MCU	Flex Cable	DIP Emulation Cable	DIP Emulation Cable	MON08 Cable
MC68HC908JB32	J1 and J2	J3	J4	J5

B.2 Installing the Hardware

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

To prepare the JBICS for use with a host PC:

1. Install an MCU into the M68ICS08JB board.

   Install an MCU (provided with the JBICS package) into the M68ICS08JB 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 Connecting to a Target System

The four ways to connect the ABICS simulator board to a target system are via:

- The low-noise flex cable
- The 20-pin DIP cable

- The 28-pin DIP cable
- The MON08 cable

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

- Using a low-noise flex cable

  When emulating an MC68HC908JB8 MCU, connect an 80-pin M68CLB05C flex cable 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 versions of the MCU.

- Using a 20-pin DIP emulation cable.

  When emulating an MC68HC908JB8 MCU, connect the 20-pin cable to the connector labeled J4 on the simulator board. Attach the other end of the cable to the appropriate connector on the target system.

- Using a 28-pin DIP emulation cable.

  When emulating an MC68HC908JB8DW28 MCU, connect the 28-pin cable to the connector labeled J3 on the simulator board. Attach the other end of the cable to the appropriate connector on the target system. A target head adapter is available for the SOIC package.

- Using a MON08 cable

  Connect the MON08 debug interface cable to the MON08 debug interface connector J5 for communication with the target system's MCU. Attach the other end of the cable to the appropriate connector on the target system. 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 JBICS.

B.4 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.

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 MC68HC908GR8 MCU.

accumulator — An 8-bit register of the MC68HC908GR8 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 MC68HC908GR8. 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 MC68HC908GR8. 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 MC68HC908GR8, 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 MC68HC908GR8. 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 MC68HC908GR8.

index register — An 8-bit CPU register in the MC68HC908GR8 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 MC68HC908GR8. 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 MC68HC908GR8 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 MC68HC908GR8.

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 MC68HC908GR8 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 MC68HC908GR8 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 MC68HC908GR8.

Z — Abbreviation for zero, a bit in the condition code register of the MC68HC908GR8. 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/

Mfax is a trademark of Motorola, Inc.

M68ICS08JBUM/D