M68ICS08JLJKUM/D

M68ICS08JLJK

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.

IBM is a registered trademark of International Business Machines Corporation.

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

© Motorola, Inc., 2000; All Rights Reserved

List of Sections

Section 1. Introduction 13

Section 2. Hardware Installation 19

Section 3. Support Information 35

Section 4. Using the MON08 Interface 43

Appendix A. S-Record Information 49

Appendix B. Technical Reference and Troubleshooting 55

Glossary 83

Index 89

List of Sections

Table of Contents

Section 1. Introduction

1.1 Contents 13

1.2 Overview. 13

1.3 JLJKICS Components. 14
1.3.1 M68ICS08JLJK Hardware. 15
1.3.2 ICS Interface Software. 16

1.4 JLJKICS Features 16

1.5 Hardware and Software Requirements 17

1.6 Specifications 18

1.7 About This Manual 18

1.8 Customer Support 18

Section 2. Hardware Installation

2.1 Contents 19

2.2 Overview. 19

2.3 Configuring the In-Circuit Simulator Board. 20
2.3.1 JLJKICS Limitations 22
2.3.1.1 Bus Frequency. 22
2.3.1.2 Port B0. 22
2.3.2 JLJKICS Configurable Jumper Headers 22
2.3.2.1 JLJKICS Bus Frequency Selection Header (JP1) 24
2.3.2.2 Board Reset Selection Header (JP2) 24
2.3.2.3 Clock Source Selection Jumper Headers (JP4 and JP5) 25
2.3.2.4 Operating Voltage Selection Header (JP3) 26
2.3.2.5 Power Source Selection Jumper Headers (JP6 and JP7). 27
2.3.3 20-Pin DIP Emulation Connector (J1) 28

2.3.4 28-Pin DIP Emulation Connector (J6) 28
2.3.5 MON08 Cable (J3) 30
2.3.6 Blank FLASH Programming Cable (J2) 30
2.3.7 Host Computer to JLJKICS Interconnection (J4) 31
2.3.8 Power Supply 31

2.4 Connecting the Hardware 32

2.5 Installing the Software 33

2.6 Connecting to a Target System 33

Section 3. Support Information

3.1 Contents 35

3.2 MCU Subsystem 35

3.3 Level Translation 36

3.4 JLJKICS Connector Signal Definitions 36

Section 4. Using the MON08 Interface

4.1 Contents 43

4.2 Overview 43

4.3 Target System Header Placement and Layout 43

4.4 Target Requirements for Using MON08 45

4.5 Sample Applications 47

Appendix A. S-Record Information

A.1 Contents 49

A.2 Overview 49

A.3 S-Record Contents 49

A.4 S-Record Types 51

A.5 S Record Creation 52

A.6 S-Record Example 52

A.6.1 S0 Header Record 52
A.6.2 First S1 Record. 53
A.6.3 S9 Termination Record 54
A.6.4 ASCII Characters 54

Appendix B. Technical Reference and Troubleshooting

B.1 Contents 55

B.2 Overview. 55

B.3 JLJKICS Board 56

B.4 Troubleshooting the Startup 57

B.5 Troubleshooting MON08 Mode 63

B.6 Board Layout, Schematic Diagrams, and Parts List 67

Glossary

Glossary 83

Index

Index 89

Table of Contents

List of Figures

Figure	Title	Page
2-1	JLJKICS Board Layout	20
2-2	JLJKICS In-Circuit Simulator Block Diagram	21
2-3	ICS Functional Diagram	21
2-4	MCU Bus Frequency Selection Header (JP1)	24
2-5	Board Reset Selection Header (JP2)	25
2-6	Clock Source Selection Jumper Headers (JP4 and JP5)	25
2-7	Jumper Header Voltage Output Options (JP3)	26
2-8	JLJKICS Power Source Selection Headers (JP6 and JP7)	27
2-9	Host Computer to JLJKICS Interconnection (J4)	31
4-1	Recommended Connector Layout	46
4-2	MON08 Connector Jumpered for Normal Target Operation	46
4-3	Application Designed with a Circuit for MON08	47
4-4	Application for Blank FLASH Programming	47
A-1	S-1 Record Example	54
B-1	JLJKICS Board Layout	67

List of Figures

List of Tables

Table	Title	Page
1-1	M68ICS08JLJK Product Components	15
1-2	Hardware Components	15
1-3	Software Components	16
1-4	JLJKICS Board Specifications	18
2-1	JLJKICS Jumper Header Description	23
2-2	MCU Bus Frequency Formula	24
2-3	Logic Table	25
2-4	JLJKICS Power Source Selection Header Settings	27
2-5	J1 20-Pin DIP Emulation Connector	28
2-6	J6 28-Pin DIP Emulation Connector	29
2-7	MON08 J3 Cable Connector	30
2-8	Blank FLASH Programming Cable Connector J2	30
3-1	Target Connector J1	37
3-2	Target Connector J6	38
3-3	MON08 Connector J3 Pin Assignments	40
3-4	DB-9 Communication Connector J4 Pin Assignments	41
4-1	MON08 Target System Connector	44
A-1	S-Record Fields	50
A-2	S-Record Field Contents	50
A-3	S-Record Types	51
A-4	S0 Header Record	52
A-5	S1 Header Record	53
A-6	S9 Header Record	54
B-1	Parts List	68

List of Tables

Section 1. Introduction

1.1 Contents

1.2 Overview 13

1.3 JLJKICS Components 14
1.3.1 M68ICS08JLJK Hardware 15
1.3.2 ICS Interface Software 16

1.4 JLJKICS Features 16

1.5 Hardware and Software Requirements 17

1.6 Specifications 18

1.7 About This Manual 18

1.8 Customer Support 18

1.2 Overview

This section provides an overview of the Motorola M68ICS08JLJK in-circuit simulator (JLJKICS).

The JLJKICS board, a single 4-inch × 6.5-inch printed circuit board (PCB), is a stand-alone development and debugging aid for designers using MC68HC908JL3, MC68HC908JK3, or MC68HC908JK1 microcontroller unit (MCU) devices. The JLJKICS contains both the hardware and software needed to develop and simulate source code for, and to program, these Motorola microcontrollers.

The JLJKICS and the JLJKICS software form a complete editor, assembler, programmer, simulator, and limited real-time I/O (input/output) emulator for the MC68HC908JL3, MC68HC908JK3, and MC68HC908JK1 MCUs. When the JLJKICS is connected to a host PC (personal computer) and target hardware, the actual inputs and outputs of the target system can be used during simulation of code.

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

- The WINIDE and CASM08Z software may be used as editor and assembler.
- With ICS08JLZ, the JLJKICS is used as a simulator.
- With the PROG08SZ software, the JLJKICS is used to program MCU FLASH memory.
- With the ICD08SZ software, the JLJKICS is used as a limited real-time emulator.

The JLJKICS connects to the target machine via either the provided 20-pin or 28-pin ribbon cable. It connects to the software host via a standard DB-9 serial cable.

Use the JLJKICS with any IBM® Windows 95®-based computer (or later version) with a serial port.

1.3 JLJKICS Components

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

Table 1-1. M68ICS08JLJK Product Components

Part Number	Description
ICS08JL	JLJKICS software development package
ICS08JLZ	ICS simulator
ICD08SZ	ICS debugger
MC68HC908JL3CP and MC68HRC908JL3CP	Two 28-pin dual in-line package (DIP) MCUs
M68ICS08JLJK (JLJKICS)	JLJKICS board

1.3.1 M68ICS08JLJK Hardware

Table 1-2 lists the JLJKICS hardware components.

Table 1-2. Hardware Components

Components	Description
U13	28-pin DIP socket for the MC68HC908JL3
U14	20-pin DIP socket for the MC68HC908JK3 and MC68HC908JK1 MCU devices
U16	28-pin SOIC socket for the MC68HC908JL3
U17	20-pin SOIC socket for the MC68HC908JK3 and MC68HC908JK1 MCU devices
J1	One 2-row × 10-pin, 0.3-inch spacing DIP connector allowing debugging connection to target board through a ribbon cable
J2	2-pin header allowing programming of MCU with 2-pin cable
J3	One 2-row × 8-pin, 0.1-inch spacing header for connecting to a target board with the MON08 debugging interface
J4	DB-9 serial cable connector
J5	dc power jack
J6	One 2-row × 14-pin, 0.6-inch spacing DIP connector allowing debugging connection to target board through a ribbon cable
J7	2-pin external oscillator connector
D4	ICS power LED (green)
D7	MCU power LED (yellow)
SW1	Power switch

1.3.2 ICS Interface Software

Windows®-optimized software components are referred collectively to as the JLJKICS software (part number ICS08JL). The ICS08JL software package is a product of P&E Microcomputer Systems, Inc., and is included in the JLJKICS kit. **Table 1-3** lists these software components.

Table 1-3. Software Components

Components	Description
WINIDE.EXE	Integrated development environment (IDE) software interface for editing and interfacing with the other items listed here
CASM08Z.EXE	CASM08Z command-line cross-assembler
ICS08JLZ.EXE	In-circuit/stand-alone simulator software for the MC68HC908JL3, MC68HC908JK3, and MC68HC908JK1 MCU devices
PROG08SZ.EXE	FLASH memory programming software
ICD08SZ.EXE	In-circuit debugging software for limited, real-time emulation

1.4 JLJKICS Features

The JLJKICS 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
- FLASH memory programming with PROG08SZ
- In-circuit and stand-alone simulation of MC68HC908JL3, MC68HC908JK3, and MC68HC908JK1 MCUs with ICS08JLZ, including:
  - Simulation of all instructions, memory, and peripherals
  - Optional simulator pin inputs from the target
  - Conditional breakpoints, script files, and logfiles

- Limited real-time emulation and debugging with ICD08SZ, including:
  - Loading code into RAM
  - Executing real-time in RAM or FLASH
  - One hardware breakpoint in FLASH
  - 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 allowing:
  - Limited in-circuit emulation
  - In-circuit simulation
  - In-circuit programming

1.5 Hardware and Software Requirements

The JLJKICS software requires this minimum hardware and software configuration:

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

1.6 Specifications

Table 1-4 summarizes the JLJKICS board hardware specifications.

Table 1-4. JLJKICS 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

1.7 About This Manual

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

1.8 Customer Support

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

Section 2. Hardware Installation

2.1 Contents

2.2 Overview 19

2.3 Configuring the In-Circuit Simulator Board 20
2.3.1 JLJKICS Limitations 22
2.3.1.1 Bus Frequency 22
2.3.1.2 Port B0 22
2.3.2 JLJKICS Configurable Jumper Headers 22
2.3.2.1 JLJKICS Bus Frequency Selection Header (JP1) 24
2.3.2.2 Board Reset Selection Header (JP2) 24
2.3.2.3 Clock Source Selection Jumper Headers (JP4 and JP5) 25
2.3.2.4 Operating Voltage Selection Header (JP3) 26
2.3.2.5 Power Source Selection Jumper Headers (JP6 and JP7) 27
2.3.3 20-Pin DIP Emulation Connector (J1) 28
2.3.4 28-Pin DIP Emulation Connector (J6) 29
2.3.5 MON08 Cable (J3) 30
2.3.6 Blank FLASH Programming Cable (J2) 30
2.3.7 Host Computer to JLJKICS Interconnection (J4) 31
2.3.8 Power Supply 31

2.4 Connecting the Hardware 32

2.5 Installing the Software 33

2.6 Connecting to a Target System 33

2.2 Overview

This section explains how to:

- Configure the M68ICS08JLJK in-circuit simulator board (JLJKICS)
- Install the hardware
- Install the software
- Connect the board to a target system

In interactive mode, the JLJKICS is connected to the serial port of a host PC. The actual inputs and outputs of a target system can be used during simulation of source code.

In software stand-alone mode, the JLJKICS is not connected to the PC. The ICS08JLZ software can be used as a stand-alone simulator running on the PC.

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.*

2.3 Configuring the In-Circuit Simulator Board

The JLJKICS includes a single 4-inch × 6.5-inch printed circuit board (PCB) (M68ICS08JLJK). **Figure 2-1** shows a diagram of the JLJKICS board, **Figure 2-2** shows a block diagram of the JLJKICS board, and **Figure 2-3** is the functional block diagram.

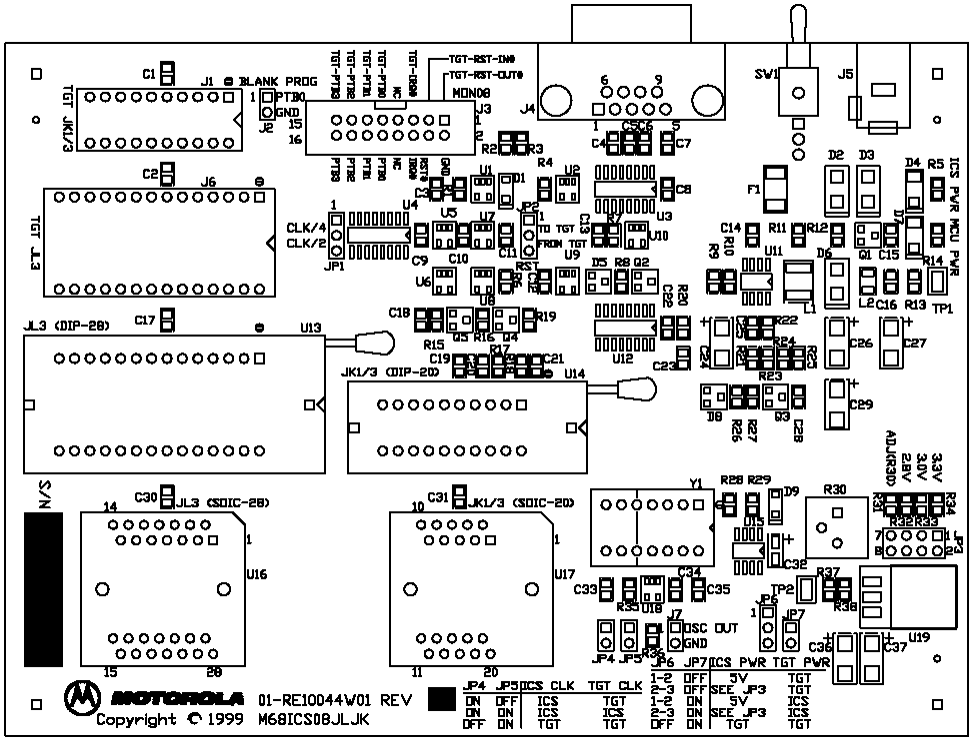


Figure 2-1. JLJKICS Board Layout

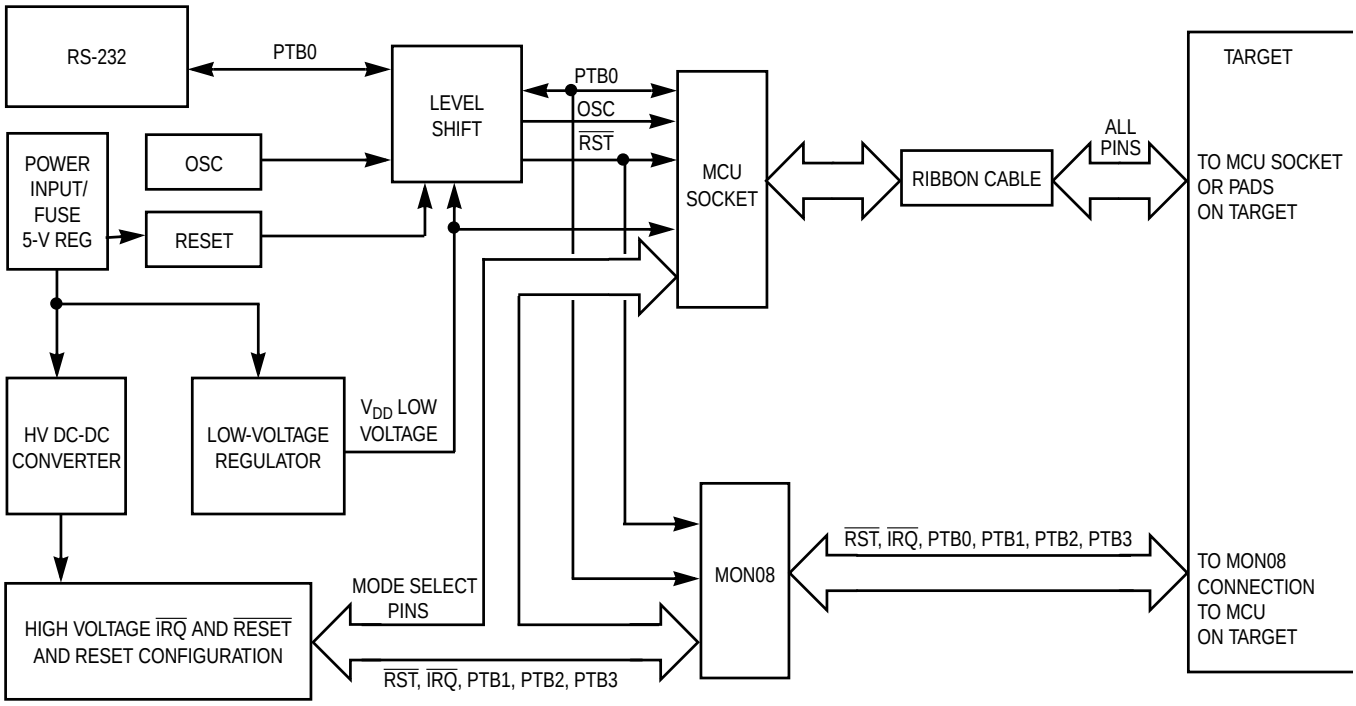


Figure 2-2. JLJKICS In-Circuit Simulator Block Diagram

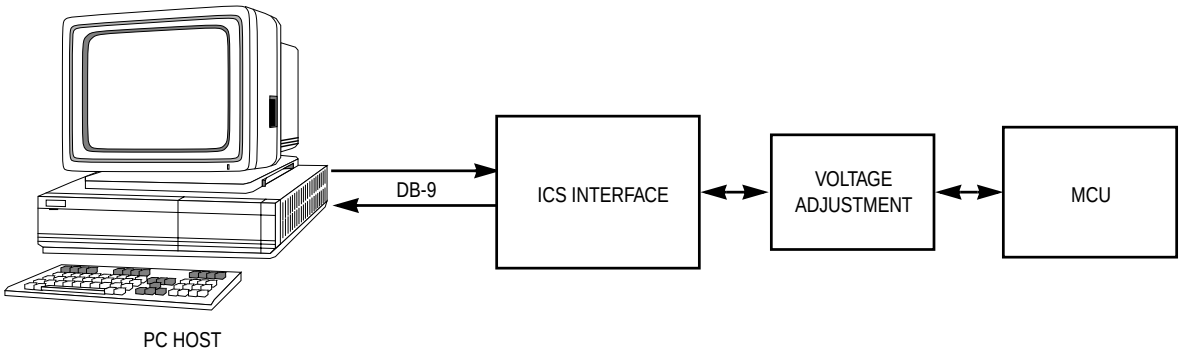


Figure 2-3. ICS Functional Diagram

2.3.1 JLJKICS Limitations

This section describes system limitations of the JLJKICS.

2.3.1.1 Bus Frequency

The RKICS communicates using the MON08 features. This forces the communication rate to $f_{Bus}/256$. Therefore, the bus frequencies are limited by standard baud rates allowed by the host software. See **2.3.2.1 JLJKICS Bus Frequency Selection Header (JP1)** for available options.

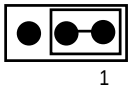
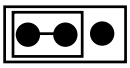
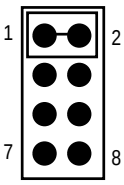
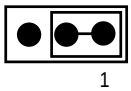
2.3.1.2 Port B0

Port B0 is used for communications, so it is unavailable for emulation.

2.3.2 JLJKICS Configurable Jumper Headers

Seven jumper headers on the JLJKICS are used to reconfigure the hardware options. **Table 2-1** is a quick reference to these optional settings, while subsections **2.3.2.1** through **2.3.2.5** describe jumper header configuration in greater detail.

Table 2-1. JLJKICS Jumper Header Description

Jumper Header	Type	Description		
JP1 Bus frequency selection		Jumper on pins 1 and 2 (default) — MCU bus frequency is set to 2.4576 MHz (OSC ÷ 4) and the I/O baud rate is set to 9600. Jumper on pins 2 and 3 — MCU bus frequency is set to 4.9152 MHz (OSC ÷ 2) and the I/O baud rate is set to 19,200.		
JP2 Board reset		Jumper on pins 1 and 2 — The MCU's reset signal initiates resets to the target system. Jumper on pins 2 and 3 (default) — The target board reset signal initiates resets to the JLJKICS on-board MCU.		
JP3 Operating voltage selection		Jumper on pins 1 and 2 (default) — 3.3 V Jumper on pins 3 and 4 — 3.0 V Jumper on pins 5 and 6 — 2.8 V Jumper on pins 1 and 2 — ADJ (R30)		
JP4 Crystal to MCU JP5 Target clock to MCU	 	**JP4**	**JP5**	
		On	On	The JLJKICS MCU and the target board clock signals are supplied by the on-board oscillator Y1 (default).
		On	Off	The JLJKICS MCU clock signal is supplied by the oscillator Y1 and the target board has its own clock source.
		Off	On	The JLJKICS MCU clock signal is supplied by the target board.
		Off	Off	No clock supplied to JLJKICS MCU
JP6 5 V or regulator JP7 ICS power connected to target	 	**JP6**	**JP7**	
		1-2	Off	5-V supply on JLJKICS, target board power separate
		2-3	Off	ICS voltage controlled by JP3 setting, target power separate
		1-2	On	5 V supplied to both JLJKICS and target board (default)
		2-3	On	Voltage controlled by JP3 supplied to both JLJKICS and target
		Off	On	Target supplies power to JLJKICS MCU

2.3.2.1 JLJKICS Bus Frequency Selection Header (JP1)

Use jumper header JP1 to select the MCU bus speed and the I/O baud rate. Install a jumper on jumper header JP1 pins 1 and 2 to set the MCU bus frequency to 2.4576 MHz and the I/O baud rate to 9600. (Refer to **Figure 2-4**.) Install a jumper on jumper header JP1 pins 2 and 3 to set the MCU bus frequency to 4.9152 MHz and the I/O baud rate to 19,200. Refer to **Table 2-2** for the MCU bus frequency formula.

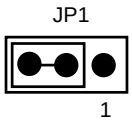


Figure 2-4. MCU Bus Frequency Selection Header (JP1)

Table 2-2. MCU Bus Frequency Formula

JP1 Setting	Formula
1-2	Crystal oscillator frequency ÷ by 4 = bus frequency Bus frequency ÷ by 256 = baud rate
2-3	Crystal oscillator frequency ÷ divided by 2 = bus frequency Bus frequency ÷ by 256 = baud rate

2.3.2.2 Board Reset Selection Header (JP2)

The reset function of the JLJKICS is both an input and an output. The JLJKICS drives its $\overline{\text{RST}}$ pin low after encountering several different exception conditions. JP2 selects whether the target system can reset the MCU on the JLJKICS or whether the target system receives a $\overline{\text{RST}}$ signal from the JLJKICS (refer to **Figure 2-5**). Install a jumper on jumper header JP2 pins 1 and 2 to use the JLJKICS $\overline{\text{RST}}$ signal to reset the JLJKICS MCU and the target system. Install a jumper on jumper header JP2 pins 2 and 3 (factory default) to use the target system reset function to reset the JLJKICS MCU.

$\overline{\text{RST}}$ is not a bidirectional, open-drain signal at the target connectors. Removing the jumper leaves the $\overline{\text{RST-IN}}$ signal pulled up to MCU operating voltage.

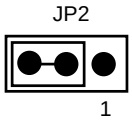


Figure 2-5. Board Reset Selection Header (JP2)

2.3.2.3 Clock Source Selection Jumper Headers (JP4 and JP5)

The JLJKICS contains a 9.8304-MHz crystal oscillator. When the remote target connection is made, the user may opt to feed the output from the JLJKICS crystal (ICS-OCS) to the external clock input (OSC1) of the JLJKICS via jumper headers JP4 and JP5 (refer to **Figure 2-6**). Refer to **Table 2-3** for a description of the clock source options.

Figure 2-6. Clock Source Selection Jumper Headers (JP4 and JP5)

Table 2-3. Logic Table

JP4	JP5	Description
On	On	The JLJKICS MCU and the target board clock signals are supplied by the on-board oscillator Y1 (default).
On	Off	The JLJKICS MCU clock signal is supplied by the oscillator Y1 and the target system has its own clock source.
Off	On	The JLJKICS MCU clock signal is supplied by the target system.
Off	Off	JLJKICS MCU has no clock source.

2.3.2.4 Operating Voltage Selection Header (JP3)

To provide the JLJKICS with power input that matches the target environment, the JLJKICS includes a user-selectable voltage level shift. In the default setting (jumper installed on jumper header JP7), the JLJKICS is connected to provide power to the user's target system, so the operating voltage of the JLJKICS must be adjusted for the desired voltage. (**Remove JP7 if there is any possibility that these voltages are not the same.**) The JLJKICS includes a test point (TP2) for measurement of the processor voltage and a screwdriver adjustable potentiometer (R30).

To set the MCU operating voltage, jumper the set of pins on JP3 (refer to **Figure 2-7**) to the desired voltage. If 5 V is needed, connect JP6 pins 1 and 2. If a voltage different from the default settings is needed, connect JP6 pins 2 and 3 and place jumper JP3 on the ADJUST setting. The potentiometer then can be adjusted to set the desired voltage. The default setting is 3.3 Vdc.

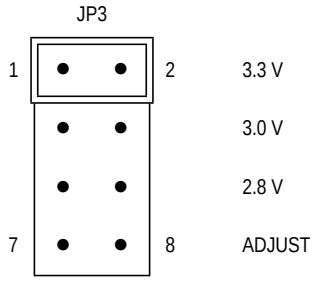


Figure 2-7. Jumper Header Voltage Output Options (JP3)

2.3.2.5 Power Source Selection Jumper Headers (JP6 and JP7)

Use JP6 and JP7 to select the power source for the JLJKICS and target system's MCU and external circuitry (refer to **Table 2-4**).

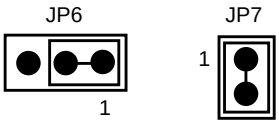


Figure 2-8. JLJKICS Power Source Selection Headers (JP6 and JP7)

Table 2-4. JLJKICS Power Source Selection Header Settings

JP6	JP7	JLJKICS MCU Power	Target Connector Power
1-2	Off	5 V	From target
2-3	Off	Set by JP3	From target
1-2	On	5 V	Same as MCU
2-3	On	Set by JP3	Same as MCU
Off	On	Same as target	From target
Off	Off	No power	From target

CAUTION: *If the JLJKICS power and target system power are different potentials, damage could occur to the ICS or the target system integrated circuits. The target system V_{DD} level must be the same potential as the ICS voltage level in all cases. Remove JP7 if there is any possibility that these voltages are not the same.*

NOTE: *JP7 is not applicable for the MON08 connection. There is no TGT-VDD on the MON08 header, J3.*

2.3.3 20-Pin DIP Emulation Connector (J1)

The J1 connector is a 20-pin dual in-line package (DIP) emulation connector. The pinout is identical to that on the MC68HC908JK3/JK1 part. This allows the JLJKICS to connect to a target board where the MCU would normally be placed and perform emulation and debugging with an MCU on the JLJKICS. This allows boards with no (or incorrect) MON08 implementations to be tested, debugged, and/or modified.

Table 2-5 shows the emulation connector interface pins.

Table 2-5. J1 20-Pin DIP Emulation Connector

Pin	Description	Pin	Description
1	$\overline{\text{TGT-IRQ}}$	20	$\overline{\text{TGT-RST}}$
2	Ground	19	PTD4
3	OSC1	18	PTD5
4	NC	17	PTD2
5	TGT-VDD	16	PTD3
6	PTB7	15	TGT-PTB0
7	PTB6	14	TGT-PTB1
8	PTB5	13	TGT-PTB2
9	PTD7	12	TGT-PTB3
10	PTD6	11	PTB4

2.3.4 28-Pin DIP Emulation Connector (J6)

The J6 connector is a 28-pin DIP emulation connector. The pinout is identical to that of the MC68HC908JL3 MCU. The use of J6 is identical to that of J1, except for the JL3 part.

Table 2-6. J6 28-Pin DIP Emulation Connector

Pin	Description	Pin	Description
1	$\overline{\text{TGT-IRQ}}$	28	$\overline{\text{TGT-RST}}$
2	PTA0	27	PTA5
3	Ground	26	PTD4
4	OSC1	25	PTD5
5	NC	24	PTD2
6	PTA1	23	PTA4
7	TGT-VDD	22	PTD3
8	PTA2	21	TGT-PTB0
9	PTA3	20	TGT-PTB1
10	PTB7	19	PTD1
11	PTB6	18	TGT-PTB2
12	PTB5	17	TGT-PTB3
13	PTD7	16	PTD0
14	PTD6	15	PTB4

2.3.5 MON08 Cable (J3)

The 16-pin MON08 cable connects to J3 on the JLJKICS board and to the target-system board. The MON08 cable allows testing, debugging, and reprogramming of a fully assembled board without removing the MCU.

Refer to **Section 4. Using the MON08 Interface** for a complete description of the MON08 connector and how to properly design target boards to utilize this connector.

Table 2-7 shows the MON08 J3 cable connector pinout.

Table 2-7. MON08 J3 Cable Connector

Pin	Description	Pin	Description
1	$\overline{\text{RST_OUT}}$	2	Ground
3	$\overline{\text{RST_IN}}$	4	$\overline{\text{RST}}$
5	$\overline{\text{TGT_IRQ}}$	6	$\overline{\text{IRQ}}$
7	NC	8	NC
9	TGT_PTB0	10	PTB0
11	TGT_PTB1	12	PTB1
13	TGT_PTB2	14	PTB2
15	TGT_PTB3	16	PTB3

2.3.6 Blank FLASH Programming Cable (J2)

J2 is a 2-pin subset of MON08 (J3). This connection provides the minimum number of connections required for programming blank FLASH of target MCU. **Table 2-8** shows the J2 cable connector pinout.

Table 2-8. Blank FLASH Programming Cable Connector J2

Pin	Description
1	PTB0
2	Ground

2.3.7 Host Computer to JLJKICS Interconnection (J4)

The host computer/JLJKICS interface is via the single system connector J4, which is a 9-pin, D-type, through-hole, female, right angle connector (Amp part number AMP-9726-A) mounted on the top side of the PCB.

To communicate with the JLJKICS, connect a DB-9-compatible host computer to I/O port J4 on the JLJKICS. This connection requires the cable assembly supplied with the JLJKICS kit. This assembly cable is a DB-9 male-to-female, 6-foot (3-meter) long serial cable. **Figure 2-9** shows RS-232C default signal assignments for J4.

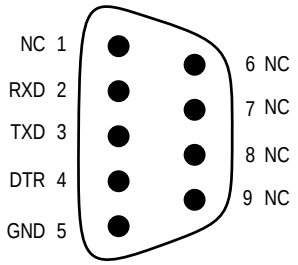


Figure 2-9. Host Computer to JLJKICS Interconnection (J4)

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

2.3.8 Power Supply

The power for the JLJKICS is supplied by the included power supply. The minimum requirements for a supply with the JLJKICS are:

- +5 Vdc regulated
- 2 A
- positive tip polarity
- 2.5 mm barrel jack

2.4 Connecting the Hardware

The following steps describe how to connect the JLJKICS to the PC and to target boards, if desired. These instructions do not apply to MON08 or other situations where the MCU is not on the JLJKICS.

Before beginning, locate these items:

- Serial cable (DB9)
- Power supply

To configure the JLJKICS for use with a host PC:

1. Install an MCU into the JLJKICS board.

   Locate the appropriate socket on the board:

   – U17 for 20-pin SOIC versions of the MC68HC908JK3 and MC68HC908JK1

   – U14 for 20-pin DIP versions of the MC68HC908JK3 and MC68HC908JK1

   – U13 for 28-pin DIP version of the MC68HC908JL3

   – U16 for 28-pin SOIC version of the MC68HC908JL3

   Install the MCU into this socket, observing the pin 1 orientation with the silkscreened dot or the marking on the silkscreen. The top (label side) of the MCU package must be visible when looking at the component side of the board.

2. Verify the jumper settings are correct for the desired JLJKICS usage. See **2.3 Configuring the In-Circuit Simulator Board** for jumper settings and definitions.

3. Connect the JLJKICS to the host PC.

   Locate the 9-pin connector labeled J4 on the board. Using the cable provided, connect it to a serial port on the host PC.

4. If appropriate, attach a target board to the JLJKICS using the cables provided. Verify that the ICS and software work prior to connecting any target boards. If using J1 or J6 to connect to a target, an MCU must be present on the ICS for the software to work. If using the MON08 connector (J3), an MCU should be present on the target board and not on the ICS.

5. Apply power to the board.

   Plug the power supply into an ac power outlet, using one of the country-specific adapters provided. Use SW1 to turn on the power. The ICS power LED on the board should light.

2.5 Installing the Software

For instructions for installing the ICS08 software, refer to a book by P&E Microcomputer Systems, Inc., titled *M68ICS08 68HC08 In-Circuit Simulator Operator's Manual*, Motorola document order number M68ICS08OM/D.

2.6 Connecting to a Target System

The two ways to connect the JLJKICS board to a target system are:

1. With a 20- or 28-pin DIP cable:

   The MCU should be removed from the target board and, if not otherwise present, a connector should be soldered in its place so that the DIP cable can attach to all the processor signals. The JLJKICS should have an identical MCU installed in one of its sockets and the voltage and clock sources should be properly configured (see **2.3 Configuring the In-Circuit Simulator Board**).

   Surface mount adapters are available separately for 20-pin SOIC (M68DIP20SOIC) and 28-pin SOIC (M68DIP20SOIC) footprints on the target board.

2. With the 16-pin MON08 interface:

   This method allows in-circuit FLASH programming and debugging of the target board's MCU. An MCU must be installed in the target board. The JLJKICS must not have an MCU installed.

   Connect the JLJKICS MON08 connector to a compatible MON08 connector on the target system. Instructions for the proper construction and use of this interface on the target board are found in **Section 4. Using the MON08 Interface**.

Section 3. Support Information

3.1 Contents

3.2 MCU Subsystem 35

3.3 Level Translation 36

3.4 JLJKICS Connector Signal Definitions 36

3.2 MCU Subsystem

The MCU subsystem consists of:

- Microcontroller
- Clock generation and selection
- Monitor mode control logic that places and holds the JLJKICS in monitor mode
- Bus voltage level translation buffers
- Processor operating voltage variable regulator

The MCU is one of these types:

- MC68HC908JK3 20-pin DIP or 20-pin SOIC
- MC68HC908JK1 20-pin DIP or 20-pin SOIC
- MC68HC908JL3 28-pin DIP or 28-pin SOIC

All four sockets are available on the board, although only one can be used at a time.

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

3.3 Level Translation

The JLJKICS has an operation voltage range of +2.7 to +5.5 volts maximum. In order to communicate with the host PC, U3 converts 5-V signals on the JLJKICS to the levels required for RS-232, and vice versa. U2, U8, U10, U18, Q4, and Q5 are all used to translate between the different voltages present on the JLJKICS (2.8-V to 5-V MCU supply, 7.5-V $V_{DD}+V_{hi}$ supply, and constant 5-V supply).

3.4 JLJKICS Connector Signal Definitions

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

NOTE: *The signal descriptions in the following tables are for quick reference only. The MC68HC908JL3 User's Manual, Motorola document order number MC68HC908JL3/D, contains a complete description of the MC68HC908JL3 MCU signals.*

Table 3-1. Target Connector J1

Pin No.	Mnemonic	Schematic NET	Direction	Signal Description
1	$\overline{IRQ}$	$\overline{TGT\text{-}IRQ}$	Input	LOGIC-LEVEL $\overline{IRQ}$ SIGNAL
2	V_{SS}	GND	Ground	MCU GROUND
3	OSC1	TGT-OSC	I/O	CLOCK SIGNAL — JLJKICS jumper settings determine if signal is an input or an output (JP5).
4	NC	NC	NC	NO CONNECT
5	V_{DD}	TGT-VDD	I/O	2.8 V TO 5 V — JLJKICS jumper settings determine if signal is an input or an output (JP7).
6	PTB7	PTB7	I/O	I/O PORT FROM MCU
7	PTB6	PTB6	I/O	I/O PORT FROM MCU
8	PTB5	PTB5	I/O	I/O PORT FROM MCU
9	PTD7	PTD7	I/O	I/O PORT FROM MCU
10	PTD6	PTD6	I/O	I/O PORT FROM MCU
11	PTB4	PTB4	I/O	I/O PORT FROM MCU
12	PTB3	TGT-PTB3	I/O	I/O PORT FROM MCU — Signals will be disconnected by the JLJKICS during reset.
13	PTB2	TGT-PTB2	I/O	I/O PORT FROM MCU — Signals will be disconnected by the JLJKICS during reset.
14	PTB1	TGT-PTB1	I/O	I/O PORT FROM MCU — Signals will be disconnected by the JLJKICS during reset.
15	PTB0	TGT-PTB0	NC	PTB0 is used for serial communications with MCU and is not transmitted over the connector to avoid signal contention problems.
16	PTD3	PTD3	I/O	I/O PORT FROM MCU
17	PTD2	PTD2	I/O	I/O PORT FROM MCU
18	PTD5	PTD5	I/O	I/O PORT FROM MCU
19	PTD4	PTD4	I/O	I/O PORT FROM MCU
20	$\overline{RST}$	$\overline{TGT\text{-}RST}$	I/O	LOGIC-LEVEL RESET — JP2 determines if this signal is an input or an output.

Table 3-2. Target Connector J6

Pin No.	Mnemonic	Schematic NET	Direction	Signal Description
1	$\overline{IRQ}$	$\overline{TGT\text{-}IRQ}$	Input	LOGIC-LEVEL $\overline{IRQ}$ SIGNAL
2	PTA0	PTA0	I/O	I/O PORT FROM MCU
3	V_{SS}	GND	Ground	MCU GROUND
4	OSC1	TGT-OSC	I/O	CLOCK SIGNAL — JLJKICS jumper settings determine if signal is an input or an output (JP5).
5	NC	NC	NC	NO CONNECT
6	PTA1	PTA1	I/O	I/O PORT FROM MCU
7	V_{DD}	TGT-VDD	I/O	2.8 V TO 5 V — JLJKICS jumper settings determine if signal is an input or an output (JP7).
8	PTA2	PTA2	I/O	I/O PORT FROM MCU
9	PTA3	PTA3	I/O	I/O PORT FROM MCU
10	PTB7	PTB7	I/O	I/O PORT FROM MCU
11	PTB6	PTB6	I/O	I/O PORT FROM MCU
12	PTB5	PTB5	I/O	I/O PORT FROM MCU
13	PTD7	PTD7	I/O	I/O PORT FROM MCU
14	PTD6	PTD6	I/O	I/O PORT FROM MCU
15	PTB4	PTB4	I/O	I/O PORT FROM MCU
16	PTD0	PTD0	I/O	I/O PORT FROM MCU
17	PTB3	TGT-PTB3	I/O	I/O PORT FROM MCU — Signals will be disconnected by the JLJKICS during reset.
18	PTB2	TGT-PTB2	I/O	I/O PORT FROM MCU — Signals will be disconnected by the JLJKICS during reset.
19	PTD1	PTD1	I/O	I/O PORT FROM MCU
20	PTB1	TGT-PTB1	I/O	I/O PORT FROM MCU — Signals will be disconnected by the JLJKICS during reset.
21	PTB0	TGT-PTB0	NC	PTB0 is used for serial communications with MCU and is not transmitted over the connector to avoid signal contention problems.
22	PTD3	PTD3	I/O	I/O PORT FROM MCU

Table 3-2. Target Connector J6 (Continued)

Pin No.	Mnemonic	Schematic NET	Direction	Signal Description
23	PTA4	PTA4	I/O	I/O PORT FROM MCU
24	PTD2	PTD2	I/O	I/O PORT FROM MCU
25	PTD5	PTD5	I/O	I/O PORT FROM MCU
26	PTD4	PTD4	I/O	I/O PORT FROM MCU
27	PTA5	PTA5	I/O	I/O PORT FROM MCU
28	$\overline{\text{RST}}$	$\overline{\text{TGT-RST}}$	I/O	LOGIC-LEVEL RESET — JP2 determines if this signal is an input to or an output from the JLJKICS.

Table 3-3. MON08 Connector J3 Pin Assignments

Pin No.	Mnemonic	Signal
1	$\overline{\text{RST-OUT}}$	TARGET SYSTEM RESET — Active-low, logic-level reset output from the JLJKICS to the target system; its use is controlled by JP2
2	GND	GROUND
3	$\overline{\text{RST-IN}}$	JLJKICS SYSTEM RESET — Active-low, logic-level reset input from the target system that is interfaced with the 7.5-V $\overline{\text{RST}}$ signal on the JLJKICS; its use is controlled by JP2
4	$\overline{\text{RST}}$	RESET — Active-low signal that ranges from 0 to 7.5 V; intended to only go to the MCU on the target system
5	$\overline{\text{TGT-IRQ}}$	TARGET INTERRUPT REQUEST — Active-low, logic-level input signal from the target that provides an interrupt to the MCU
6	$\overline{\text{IRQ}}$	INTERRUPT REQUEST — 0- to 7.5-V signal controlled by $\overline{\text{TGT-IRQ}}$; intended to only go to the MCU on the target system
7	NC	NO CONNECT
8	NC	NO CONNECT
9	TGT-PTB0	TARGET PORT B (bit 0) — General-purpose I/O signal from the target system. Not connected on the JLJKICS. See PTB0 for the reason.
10	PTB0	PORT B (bit 0) — PTB0 is used for serial communications with the MCU during monitor mode operations. This signal should only be connected to the MCU to avoid contention problems.
11	TGT-PTB1	TARGET PORT B (bit 1) — General-purpose I/O signal from the target system
12	PTB1	PORT B (bit 1) — I/O signal connected to the MCU on the target board
13	TGT-PTB2	TARGET PORT B (bit 2) — General-purpose I/O signal from the target system
14	PTB2	PORT B (bit 2) — General-purpose MCU I/O signal
15	TGT-PTB3	TARGET PORT B (bit 3) — General-purpose I/O signal from the target system
16	PTB3	PORT B (bit 3) — General-purpose MCU I/O signal

Table 3-4. DB-9 Communication Connector J4 Pin Assignments

Pin No.	Mnemonic	Signal
1	NC	NO CONNECT
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 controlling the power status of the MCU. When DTR is inactive, the processor is turned off.
5	GND	GROUND
6	NC	NO CONNECT
7	NC	NO CONNECT
8	NC	NO CONNECT
9	NC	NO CONNECT

Section 4. Using the MON08 Interface

4.1 Contents

4.2 Overview 43
4.3 Target System Header Placement and Layout 43
4.4 Target Requirements for Using MON08 45
4.5 Sample Applications 47

4.2 Overview

The MON08 interface may be used to debug and program a target system's MCU directly. The target system must be connected to the JLJKICS in-circuit simulator board's MON08 interface connector. This section explains what must be present on a target board to make use of the MON08 capabilities.

4.3 Target System Header Placement and Layout

Two headers must be placed on the target board:

- A 16-pin header such as Berg Electronics part number 67997-616
- A 1-pin header such as Berg Electronics part number 68001-601

Table 4-1 describes the target system connections required on the 16-pin header. The 1-pin header should be connected to $\overline{\text{RST}}$ and be placed next to pin 1 of the 16-pin header. See **Figure 4-1** for the recommended layout of the MON08 connection on the target board. **Figure 4-2** shows the target MON08 header jumpered for normal target operation. Additional information about the connections on the JLJKICS board can be found in **Appendix B. Technical Reference and Troubleshooting**.

Table 4-1. MON08 Target System Connector

Pin No.	M68ICS08JLJK Label	Direction	Target System Connection
1	$\overline{\text{RST-OUT}}$	Out to target	Connect to reset inputs on target board. Provides logic-level reset to target board peripherals while using MON08.
2	GND	Ground	Connect to ground (V_{SS}).
3	$\overline{\text{RST-IN}}$	In from target	Connect to any target board peripherals that should reset the MCU. Interfaces logic level reset from target board with high voltage $\overline{\text{RST}}$ used with MON08.
4	$\overline{\text{RST}}$	Bidirectional	Connect to MCU $\overline{\text{RST}}$ and the 1-pin connector. There must be no other target board connections to this signal. It swings from 0 to +7.5 Vdc during MON08 operations.
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. There must be no other target board connections to this signal. It will swing from 0 to +7.5 Vdc during MON08 operations.
7	NC	NC	Not connected
8	NC	NC	Not connected
9	TGT-PTB0	Bidirectional	Connect to target board circuitry that should be connected to PTB0 on the MCU.
10	PTB0	Bidirectional	Connect to MCU PTB0 pin. There must be no other target board connections to this signal.
11	TGT-PTB1	Bidirectional	Connect to target board circuitry that should be connected to PTB1 on the MCU.
12	PTB1	Bidirectional	Connect to MCU PTB1 pin. There must be no other target board connections to this signal.
13	TGT-PTB2	Bidirectional	Connect to target board circuitry that should be connected to PTB2 on the MCU.
14	PTB2	Bidirectional	Connect to MCU PTB2 pin. There must be no other target board connections to this signal.
15	TGT-PTB3	Bidirectional	Connect to target board circuitry that should be connected to PTB3 on the MCU.
16	PTB3	Bidirectional	Connect to MCU PTB3 pin. There must be no other target board connections to this signal.

4.4 Target Requirements for Using MON08

The purpose of the MON08 connector is to provide a method of isolating the MCU on the target from the remainder of the target board. All MCU signals going to the MON08 connector should not go anywhere else on the target board (except for ground). When the MON08 connection is not being used, jumper shunts should be placed between MON08 connector pins 3-4, 5-6, 7-8, 9-10, 11-12, 13-14, 15-16, and from MON08 pin 1 to the 1-pin header. This fully connects the target MCU to the target board and allows normal operation of the target system. When the MON08 cable is connected between the JLJKICS and the target board, the target MCU can be reprogrammed and experimented with while it is still in place on the target board.

The JLJKICS connects PTB1..3 to TGT-PTB1..3, except during reset. During reset, PTB1..3 have voltages placed on them that configure the processor to enter the proper mode when coming out of reset. Refer to Section 9, Monitor ROM, in the Motorola MC68HC908JL manual for more information. PTB0 will not be connected to TGT-PTB0 because this signal is used for serial communications with the debugging software.

Any pullups on the reset and $\overline{\text{IRQ}}$ signals should be on the target board side of the MON08 connector and not on the MCU side. Any connections to $\overline{\text{RST}}$ or $\overline{\text{IRQ}}$ may cause MON08 debugging to fail and may damage components on the target since these signals will go up to 7.5 Vdc. See **4.5 Sample Applications** for a recommended schematic diagram for connecting the MCU to the MON08 connector and the remainder of the target board.

The connector may be laid out as an 8-row x 2-row of 0.100-inch spaced holes with traces connecting the proper signals so that products later debugged with MON08 can also serve as production boards without needing shunts installed on all MON08 signals. To then debug with MON08, the traces would be cut and a header installed for connection to the JLJKICS. To resume normal operation, shunts would be installed. See **Figure 4-1** and **Figure 4-2** for the layout guides.

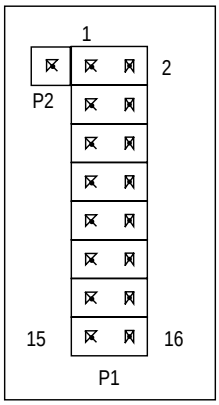


Figure 4-1. Recommended Connector Layout

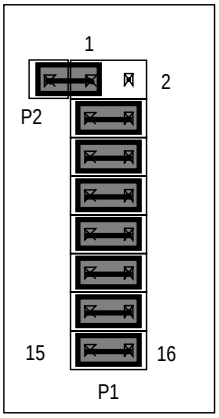


Figure 4-2. MON08 Connector Jumpered for Normal Target Operation

4.5 Sample Applications

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

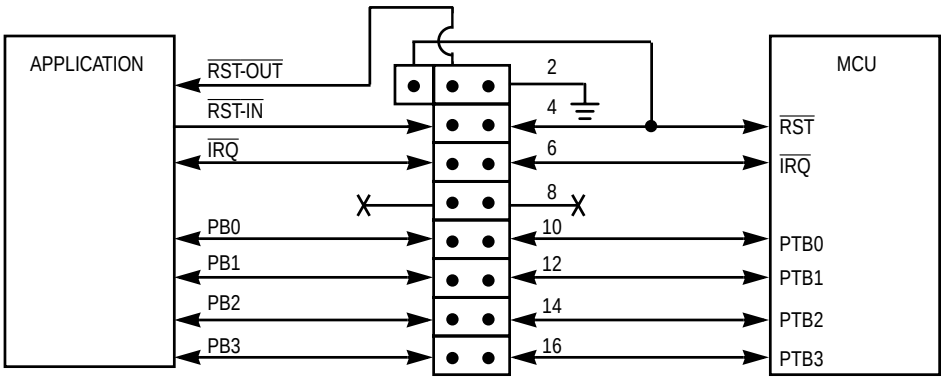


Figure 4-3. Application Designed with a Circuit for MON08

Figure 4-4 shows a simple implementation for blank FLASH programming.

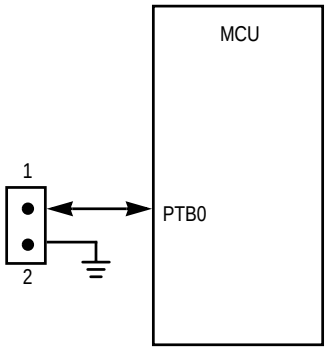


Figure 4-4. Application for Blank FLASH Programming

Use the supplied 2-pin cable to connect the header (depicted in **Figure 4-4**) to the blank programming header, J2, of the JLJKICS. If the target designer reserves PTB0 for FLASH programming serial communication, application isolation will not be necessary. Application isolation of PTB0 will be necessary (as shown in **Figure 4-3**) if the application drives PB0.

Appendix A. S-Record Information

A.1 Contents

A.2 Overview 49
A.3 S-Record Contents 49
A.4 S-Record Types 51
A.5 S Record Creation 52
A.6 S-Record Example 52
A.6.1 S0 Header Record 52
A.6.2 First S1 Record 53
A.6.3 S9 Termination Record 54
A.6.4 ASCII Characters 54

A.2 Overview

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.3 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 the line number generated by some time-sharing systems.

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

A.4 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 ICS08JLJKZ 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. S-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 JLJKICS
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.5 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.6 S-Record Example

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

```
S00600004844521B
S1130000285F245F2212226A00042429008237C2A
S113001000020008000825290018538123410018 13
S113002041E900084#42234300182342000824A952
S107003000144ED492
S9030000FC
```

In the example, the format consists of:

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

A.6.1 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; 0s

Table A-4. S0 Header Record (Continued)

Field	S-Record Entry	Description
Code/Data	48 44 52	Descriptive information identifies these S1 records: ASCII H D R — “HDR”
Checksum	1B	Checksum of S0 record

A.6.2 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.6.3 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
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.6.4 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 in **Figure A-1**.

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

Figure A-1. S-1 Record Example

Appendix B. Technical Reference and Troubleshooting

B.1 Contents

B.2 Overview 55
B.3 JLJKICS Board 56
B.4 Troubleshooting the Startup 57
B.5 Troubleshooting MON08 Mode 63
B.6 Board Layout, Schematic Diagrams, and Parts List 67

B.2 Overview

This appendix provides technical support information for the M68ICS08JLJK in-circuit simulator (JLJKICS) kit, including:

- Functional description of the kit
- Troubleshooting the startup procedure
- Troubleshooting MON08 mode
- Parts list
- Board layout diagram
- Schematic diagrams

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 printed circuit boards are extremely sensitive to electrostatic discharge (ESD). Wear a grounding wrist strap whenever handling any printed circuit board. This strap provides a conductive path for safely discharging static electricity to ground.*

B.3 JLJKICS Board

The core component of the board is the MC68HC908JLJK MCU. This MCU resides either on the JLJKICS board or on a target system.

When the MCU resides on the ICS board, the board may be used as an in-circuit emulator or simulator for the MC68HC908JLJK. For this configuration, a target cable is run from the board to the target system.

The JLJKICS supports three kinds of target cables:

- A 20-pin ribbon cable terminated in 20-pin male DIP (dual in-line package) headers
- A 28-pin ribbon cable terminated in 28-pin male DIP headers
- A 16-pin ribbon cable terminated in 16-pin dual-row insulation displacement connectors (IDC)

Using a target cable is recommended but optional; the board may be utilized with flying leads to other circuits. The MCU can be any of the MC68H(R)C908JL3, MC68H(R)C908JK3, or MC68H(R)C908JK1 chips in any of the available footprints. On the JLJKICS board, socket U13 supports the 28-pin DIP package, socket U14 supports the 20-pin DIP package, socket U16 supports the 28-pin small outline integrated circuit (SOIC) package, and socket U17 supports the 20-pin SOIC package.

When the MCU resides on a target system, the JLJKICS board can communicate with the MCU over a 16-pin MON08 cable (Motorola part number 01-RE91008W01). The MON08 interface is intended for in-system debugging and programming of an MCU in the target system.

The ICS08JLZ 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 as would 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 be seen and followed, while still 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 JLJKICS, or it may reside on a target board that supports MON08. Blank parts may also be programmed on a target board through the blank programming port (J2) on the JLJKICS. This is especially beneficial for programming parts after all debugging and board assembly have been completed.

The JLJKICS board also provides +5 Vdc power, +7.5 Vdc power for the V_{MON} voltage required to enter monitor mode, a 9.8304 MHz clock signal, and RS-232 level translation to and from the host PC.

When using the ICD08SZ debugging software, code can be run directly out of the MCU's internal memory (both FLASH and RAM) at real-time speeds.

NOTE: *The JLJKICS's emulation of the MC68HC908JLJK is limited. Port B bit 0 (PTB0) is used for host-to-MCU communication. The port bit is not available for connection to a target system. Setting DDRB bit 0 to a 1 will stop communications with the simulation or debugger software and will require a system reset to regain communication with the MCU. Port bits PTB1, PTB2, and PTB3 are temporarily disconnected from the target system during reset. Emulation of the MC68HC908JLJK's* $\overline{RST}$ *signal is also limited in that the signal is not a bidirectional, open-drain signal. It is emulated as either an input or an output (determined by jumper JP2) when using the target connectors or as two pins (one input and one output) when using the MON08 interface.*

B.4 Troubleshooting the Startup

The installation procedure in **2.4 Connecting the Hardware** describes how to prepare the JLJKICS for use when the MCU is installed on the JLJKICS board.

This involves one of these situations:

- Using the JLJKICS as an in-circuit simulator/emulator with a target cable
- Using the JLJKICS as a programmer
- Using the JLJKICS as a stand-alone system without a target board

If difficulties are experienced when connecting the kit using the instructions in **2.4 Connecting the Hardware**, follow these steps:

1. If any target cables (including MON08) are connected to the JLJKICS, they should be removed. The following troubleshooting steps assume that no target system connections are present.
2. Make sure that the MCU is installed correctly. Verify that only one MCU is installed on the JLJKICS.
3. Check power on the JLJKICS:
   a. Check the power at the output of the adapter. First, turn the power switch off. Disconnect the JLJKICS from the power supply and measure the power at the wall adapter's output connector to confirm that it produces +5 Vdc. The outer barrel of the connector is ground, and the inner sleeve is +5 Vdc. If there is no power at the connector, verify that the adapter is getting power from the ac power outlet.
   b. Check the power at the JLJKICS board. Reconnect the power supply to J5 and turn the power switch on. Check for +5 Vdc on both sides of fuse F1 using the shielding on serial port J4 for a convenient ground connection. If +5 Vdc is not present on either side of fuse F1, doublecheck the power switch. If +5 Vdc is present on both sides of the fuse, go to the next step. Otherwise, if voltage is on only the side of the fuse nearest the power switch, then the fuse has tripped and should be given some time to reset. The fuse will have tripped if more than +6.2 Vdc or a reverse voltage was applied to the JLJKICS, or if some other fault condition occurred (for instance, a short). If the fuse never resets, it will have to be replaced.
   c. Check the JLJKICS MCU PWR. Turn the power switch off then disconnect the JLJKICS from the power supply and from the host PC. Configure the JLJKICS board to the factory defaults. Reconnect the power supply to the JLJKICS. When the power switch is turned on, the MCU power LED should light. If the LED does not light, there may be a problem with the JLJKICS causing too much of a drain on the +5 Vdc supply.
   d. Check the MCU power at test point TP2 (MCU VDD) located just below the variable resistor R30. Using the serial port shielding as the ground reference, check for +3.3 Vdc at TP2 (with factory default jumper settings).

e. Check the JLJKICS board's V_{MON} power with the host disconnected. With the JLJKICS board powered and no host connection to the JLJKICS, the voltage at TP1 should be 0.

f. Check the JLJKICS board's V_{MON} power with the host connected. First, exit any ICS08JLZ software that may be running on the host PC. Then disconnect power from the JLJKICS. Ensure that the JLJKICS board is configured for the factory default settings. Ensure that there is an MCU in one of the ZIF sockets and that it is inserted correctly. Connect the serial cable between the host PC and the JLJKICS. Apply power to the JLJKICS. At this point, the JLJKICS power LED should be lit, and the MCU power LED should be off. If the MCU power LED is on, there may be a problem with the host PC's serial port or the serial cable. See step 5 for communications problems. If the MCU power LED is off, start the ICS08JLZ simulator software as described in **2.4 Connecting the Hardware** while watching the MCU power LED.

   If the MCU power LED does not light at all, there may be a problem with the host PC communicating with the board. Refer to step 5.

   If the MCU power LED flickers a few times and then goes out, the host PC is able to control the power to the JLJKICS board but communications may still not be established with the MCU. As the flickering of the MCU power LED indicates, the host PC is applying and removing power to the JLJKICS board during this period. Use an oscilloscope to view the voltages on TP1 and TP2 as the software tries to establish communication with the MCU. Restart or retry the ICS08JLZ software while looking at the signals. Using the shielding on the serial connector as the ground reference, check that signal TP2 is at 3.3 V (MCU V_{DD}) and that TP1 (V_{MON}) is at 7.5 V. If these voltages are present, the power is good, but communication problems should be investigated as described in step 5.

   If the MCU power LED comes on and stays on, communication probably was established with the MCU. Check for the following voltages, using the shielding on the serial connector as the ground reference:

- Approximately +7.5 Vdc at TP1

- Approximately +3.3 Vdc at TP2

If these voltages are present, the power is good, and the problem lies elsewhere.

4. Make sure that the host PC can communicate with the MCU:

   a. The MCU's PTB0 pin is used for host communications. DDRB bit 0 should never be set to 1 as this disrupts monitor-mode communications. The target connector PTB0 pins (J3 pin 1, J1 pin 2, and J2 pin 7) are never connected to the MCU's PTB0 pin. They are wired only for probing purposes.

   b. Make sure that the serial cable is correctly attached to the JLJKICS and to the correct serial port on the host computer.

   c. Make sure that the cable is a straight-through cable supporting all nine pins of the serial port connection.

   d. Make sure that no hardware security key or other devices are attached to the serial port or cable.

   e. Make sure that the host PC supports the minimum speed requirements of the ICS08JLZ software.

   f. Make sure to use the correct security code to access the MCU. If the security bytes have been programmed previously, the part will not unlock and allow the readback of FLASH memory unless the correct security code is sent to the MCU.

   g. Check for data at the JLJKICS end of the serial cable. Pin 3 of this connector carries RS-232 data into the JLJKICS; pin 2 carries RS-232 data out of the JLJKICS. Pin 4 controls the MCU power. Pin 5 is ground. While the ICS08JLZ software is trying to establish communications, pins 3 and 4 should both toggle between +10 Vdc and –10 Vdc (or +12 Vdc and –12 Vdc). If these signals are not seen at the cable end, the problem is on the PC and cable side of the system. When connected to the JLJKICS, a +10-Vdc signal on pin 4 should activate the MCU V_{DD} test point and the MCU power LED.

   h. Make sure the serial data is getting to the MCU's PTB0 pin. First, exit any ICS08JLZ software that may be running on the host PC. Then disconnect power from the JLJKICS. Ensure that the JLJKICS board is configured for the factory default settings. Ensure that there is an MCU in one of the sockets and that it is

inserted correctly. Connect the serial cable between the host PC and the JLJKICS. Apply power to the JLJKICS. Start the ICS08JLZ simulator software as described in **2.4 Connecting the Hardware**. Probe the PTB0 pin (U13 and U16 pin 21 or U14 and U17 pin 15) for the serial data. Since the board power is turned off and on several times during the connecting phase, the data observed at the MCU's PTB0 pin is affected also.

5. Make sure that the MCU has a good clock source. Use an oscilloscope to check the OSC1 pin at the MCU (U13 and U16 pin 4 and U14 and U17 pin 3). Set the oscilloscope to 0.1 µs per division. The oscillator should run when the MCU power LED is on. Approximately one division per cycle should be observed for a 9.8304-MHz signal, the frequency required for a 9600-baud communications rate. If the clock signal is not present, check to see that a jumper is installed on JP4. This selects the JLJKICS as the source of the OSC1 signal.

6. Make sure that the MCU can enter and remain in monitor mode. For this to happen, the following conditions must occur:

   a. At the rising edge of $\overline{RST}$, $\overline{IRQ}$ must be at V_{MON} (+7.5 Vdc). Using a dual-trace oscilloscope, trigger channel 1 on the rising edge of $\overline{RST}$ (U13/U16 pin 28, U14/U17 pin 20) and read the $\overline{IRQ}$ pin (U13/U14/U16/U17 pin 1) with channel 2. Start the ICS08JLZ software as described in **2.4 Connecting the Hardware** and verify that the $\overline{IRQ}$ signal is approximately +7.5 Vdc when $\overline{RST}$ rises. If $\overline{IRQ}$ is not at +7.5 Vdc, there may be a problem with the JLJKICS board's IRQ circuit. The timing of the reset pulse and the monitor mode control signals on port B is controlled by the 74HC123-based circuit on sheet 3 of the schematics.

   b. At the rising edge of $\overline{RST}$, PTB1, PTB2, and PTB3 must be held at logic values 1, 0, and 1, respectively. Using a dual-trace oscilloscope, trigger channel 1 on the rising edge of $\overline{RST}$ and read the corresponding MCU pin with channel 2. PTB0 is the serial data pin to and from the host PC and should be around +3.3 Vdc at the rising edge of $\overline{RST}$. Analog switch U4 is used to put the default monitor mode configuration onto PTB1, PTB2, and PTB3 at the end of reset. These pins are available for processor I/O at other times. The MCU's PTB0 pin is never connected to the target pins, as it is used for host communication.

c. Either $\overline{RST}$ or $\overline{IRQ}$ must remain at +7.5 Vdc to hold the MCU in monitor mode. The JLJKICS board has an interrupt lock out feature to keep $\overline{IRQ}$ at 7.5 Vdc when the $\overline{RST}$ or $\overline{RST\text{-}IN}$ signal is asserted (low) and to keep it at +7.5 Vdc until after $\overline{RST}$ goes high. The $\overline{TGT\text{-}IRQ}$ signal is allowed to control the $\overline{IRQ}$ signal when $\overline{RST}$ is not asserted.

7. Make sure that external circuitry does not interfere with the monitor mode communications. When connecting external circuitry to the JLJKICS board, use only the target system connectors J1 and J6. This ensures that the target system will not interfere with the communications and setup of the MCU's monitor mode by allowing the JLJKICS to disconnect some target system components during monitor mode entry.

8. Make sure that JP1 (bus clock divisor) is jumpered correctly. JP1 is provided to allow the user to select whether the MCU bus clock is equal to OSC1/2 (JP1 pins 2 and 3 jumpered, default setting) or OSC1/4 (JP1 pins 1 and 2 jumpered). This jumper affects the baud rate in monitor mode. With a 9.8304-MHz crystal, the OSC1/4 gives 9600 baud.

9. When connecting to a target system, observe the setting of JP2 (target $\overline{RST}$ direction). JP2 is provided to allow the user to select whether the target system can reset the MCU on the JLJKICS (jumper between pins 2 and 3) or whether the target system receives a reset signal from the JLJKICS (jumper between pins 1 and 2). $\overline{RST}$ is not a bidirectional, open-drain signal at the target connectors. Removing the jumper leaves the $\overline{RST\text{-}IN}$ signal pulled up to the current MCU V_{DD} voltage setting.

10. When connecting to a target system, observe the setting of JP7 (target V_{DD} disconnect). JP7 is provided to allow the user to select whether the JLJKICS powers the target system's MCU and external circuitry (JP7 jumper on) or whether the target provides the power for its MCU and circuitry (JP7 jumper off). The target system V_{DD} must match the JLJKICS operating voltage in either case.

CAUTION: *Before powering up, remove the JP7 jumper if the target system is powered by a source other than the JLJKICS. Failure to remove the jumper in this case will cause the two power supply outputs to be connected together, possibly causing large currents to flow over the target cable. If the JLJKICS is to provide power to the target system, ensure that the current drain on the target connector's V_{DD} pins is kept under 500 mA.*

B.5 Troubleshooting MON08 Mode

This section describes the troubleshooting steps for the instances where the MCU is installed on a target system and the JLJKICS is used to interact with the target system through the MON08 cable. These instances include in-circuit simulation/emulation and FLASH memory programming through the MON08 cable.

1. Disconnect the target system and make sure that the JLJKICS operates correctly when configured as described in the quick-start instructions. See **2.4 Connecting the Hardware** and **B.4 Troubleshooting the Startup**. If the JLJKICS runs correctly for the quick start, then it should function properly for MON08 aside from problems related to the target board.
2. Prepare the JLJKICS for use with the MON08 cable. Exit the ICS08JLZ software. Turn off the power to the target system. Turn the JLJKICS power switch off and disconnect the power supply. Remove any MCU in the sockets on the JLJKICS.
   a. If the target board is using a 9.8304-MHz clock, skip this step. Otherwise, the baud rate for communications will be incorrect. Do one of the following: Put a 9.8304-MHz clock source on the target board, put a shunt on JP5 and connect J7 (OSC OUT) as a clock source for the target board, or (if available) select a different communications baud rate when using the ICS08JLZ software. See the 68HC908JL/JK manual for the relationship between the clock rate and the baud rate.
   b. Jumper selections on JP7 and JP2 have no effect when using the MON08 cable. Install a jumper on JP1 pins 1 and 2 if using a 9.8304-MHz clock.
3. Connect the 16-pin cable from J3 on the JLJKICS to the target system's MON08 connector. Details on designing a MON08 connector for the target system are given in **Table 4-1. MON08 Target System Connector**. If cutable jumpers were used on the target board, the jumpers must be cut before using the MON08 cable.

4. The target system (including the MCU) must be externally powered. The target system's MCU V_{DD} must match the MCU V_{DD} setting on the JLJKICS to communicate with the JLJKICS. Do not apply power to the target system at this time.

5. Connect the serial cable between the host PC and the JLJKICS. Apply power to the JLJKICS by connecting the wall adapter's output jack to the JLJKICS and turning the power switch on. At this point, the ICS power LED should be lit, and the MCU power LED should be off. If the MCU power LED is on, there may be a problem with the host PC's serial port or the serial cable.

6. Apply power to the target system. At this point, the target MCU should be powered. Check for the appropriate voltage at the MCU's V_{DD} pin. The JLJKICS should leave the target MCU in reset with approximately 0 Vdc at the MCU's $\overline{RST}$ pin. Verify this at the target MCU's $\overline{RST}$ pin. If $\overline{RST}$ floats too high, the MCU may start and execute code out of its FLASH memory. The JLJKICS should reset the MCU again in step 8 when the software is started.

7. Start the ICS08JLZ simulator software as described in **2.4 Connecting the Hardware** while watching the MCU power LED.

   If the MCU power LED does not light, there may be a problem with the host PC communicating with the JLJKICS. Continue with step 8.

   If the MCU power LED flickers a few times and then goes out, the host PC is able to control the JLJKICS but communications may still not be established with the MCU on the target system. As the flickering of the MCU power LED indicates, the host PC is applying and removing power to the JLJKICS board during this period. The software should give a communications error in this case. Use an oscilloscope to view the voltages on TP1 and another power pin (J6 pin 7) as the software tries to establish communication with the MCU. Restart or retry the ICS08JLZ software while looking at the signals. Using the serial port shielding (or a ground pin on another connector) as the ground reference, check for a signal that varies between 0 and +7.5 Vdc at TP1 ($V_{MON)}$ and between 0 and the set JLJKICS voltage at the second power pin (MCU V_{DD}). If these voltages are present, the power is good, but the MCU is not being placed in monitor mode. Continue with step 8.

If the MCU power LED comes on and stays on and the software does not give a communications error, the software was able to establish communications with the MCU. Check for the following voltages, using the shielding on the serial port as the ground reference:

- Approximately +7.5 Vdc at TP2
- The set JLJKICS voltage at J6 pin 7 or another power pin

If these voltages are present, the JLJKICS power is good. Continue with step 8.

8. Make sure the host PC can communicate with the MCU:
   a. The MCU's PTB0 pin is used for host communications. DDRB bit 0 should never be set to 1, as this disrupts monitor-mode communications. The MON08 pin TGT-PTB0 (J3 pin 9) is never connected to the MCU's PTB0 pin. It is only connected to pins on J1, J3, and J6 for probing purposes. On the MON08 connector J6, pin 8 is wired to the MCU's PTB0 pin. Driving this signal with external logic on the target system will prevent communications.
   b. Make sure that the MON08 cable is installed properly between the JLJKICS and the target system. Pin 1 of each connector on the cable must go to pin 1 of the headers on the JLJKICS and target system.
   c. Make sure that the serial cable is a straight-through cable supporting all nine pins of the serial-port connection and that it is correctly installed between the JLJKICS and the PC.
   d. Make sure that no hardware security key or other device is attached to the serial port or cable.
   e. Make sure that the host PC supports the baud rate requirements of the ICS08JLZ software.
   f. Make sure to use the correct security code to access the MCU. If the security byte has been programmed previously, the part will not unlock and enter monitor mode unless the correct security code is sent to the MCU. Since MON08 cannot cause a power-on reset to the target board, it is necessary to remove power and reattach it to the target board after each security code is attempted.

g. Make sure the serial data is getting to the MCU's PTB0 pin. Restart the ICS08JLZ simulator software as described in steps 3 and 4 of the quick-start instructions. Probe the PTB0 pin of the target MCU for the serial data.

h. Make sure that the target MCU has a good clock source. Use a clock rate that gives a 9600-baud serial communications rate (or another supported baud rate, if necessary) for monitor mode on the target system. Use an oscilloscope to check the OSC1 output at the MCU. If the clock signal is not present, connect the JLJKICS clock to the target board as described in step 2.

9. Make sure that the MCU can enter and remain in monitor mode. For this to happen, these conditions must occur:

   a. At the rising edge of $\overline{RST}$, the target MCU's $\overline{IRQ}$ pin must be at V_{MON} (+7.5 Vdc). Using a dual-trace oscilloscope, trigger channel 1 on the rising edge of the MCU's $\overline{RST}$ pin and read the $\overline{IRQ}$ pin with channel 2. Start the ICS08JLZ software as described in **2.4 Connecting the Hardware** and verify that the $\overline{IRQ}$ signal is approximately +7.5 Vdc when $\overline{RST}$ rises.

   b. At the rising edge of $\overline{RST}$, PTB1, PTB2, and PTB3 must be held at logic values 1, 0, and 1, respectively. The voltage levels should match the power settings of the JLJKICS and the target board. Using a dual-trace oscilloscope, trigger channel 1 on the rising edge of $\overline{RST}$ and read the corresponding MCU pin with channel 2. After the rising edge of $\overline{RST}$, the MCU pins PTB1, PTB2, and PTB3 are connected (by the JLJKICS) to the MON08 connector pins TGT-PTB1, TGT-PTB2, and TGT-PTB3, respectively. The MCU's PTB0 pin is never connected to the target pins because it is used for host communication.

   c. Either $\overline{RST}$ or $\overline{IRQ}$ must remain at +7.5 Vdc to hold the MCU in monitor mode. The JLJKICS board has an $\overline{IRQ}$ lockout feature to keep $\overline{IRQ}$ at 7.5 Vdc when the $\overline{RST}$ or $\overline{RST\text{-}IN}$ signal is asserted (low) and to keep it at +7.5 Vdc until after $\overline{RST}$ goes high. The $\overline{TGT\text{-}IRQ}$ signal is allowed to control the $\overline{IRQ}$ signal when $\overline{RST}$ is not asserted.

10. Make sure that the target circuitry does not interfere with the monitor mode communications. There should be no target board connections to the PTB0, PTB1, PTB2, PTB3, $\overline{IRQ}$ or $\overline{RST}$ pins on the MON08 connector. Any connections to these pins may cause problems with MON08. In particular, connections to $\overline{IRQ}$ and $\overline{RST}$ may cause damage to the target board by supplying circuitry with 7.5 V.

B.6 Board Layout, Schematic Diagrams, and Parts List

Figure B-1 shows the JLJKICS board layout and component locations.

The JLJKICS parts list and schematic diagrams follow.

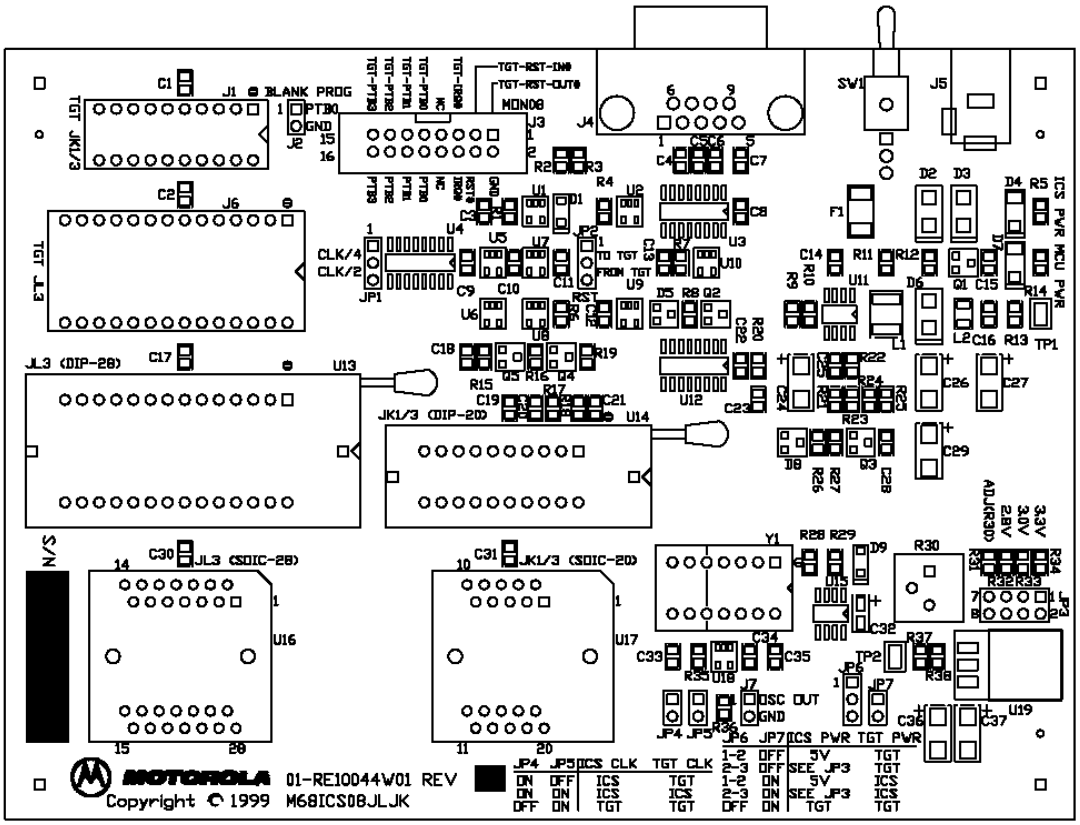


Figure B-1. JLJKICS Board Layout

Table B-1. Parts List (Sheet 1 of 10)

Qty	Reference	Part Description (PII P/N)	Package Type	Disty
14	C1, C2, C4, C5, C6, C7, C8, C16, C17, C19, C23, C25, C30, C31	0.1-µF ceramic capacitor (704-5014-0001)	SMT0805 ceramic, X7R, 10%, 25 V Digi-Key: PCC1828CT-ND Panasonic: ECJ-2VB1E104K	 Digi-Key Digi-Key
1	C14	560-pF ceramic capacitor (704-5071-0001)	SMT0805 ceramic, X7R, 10%, 50 V Digi-Key: PCC561BNCT-ND Panasonic: ECU-V1H561KBN	 Digi-Key Digi-Key
2	C24, C26	47-µF, low-ESR tantalum capacitor (704-5072-0001)	EIA7343 (D) tantalum, low ESR, 20%, 16 V AVX: TPSD476M016R0150	 Kent Milgray Marshall Future Allied
4	C27, C29, C36, C37	10-µF tantalum capacitor (704-5026-0004)	EIA3528 (B) tantalum, 20%, 16 V AVX: TAJB106M016 Digi-Key: PCS3106CT-ND Panasonic: ECS-T1CX106R	 Kent Milgray Marshall Future Allied Digi-Key Digi-Key
14	C3, C9, C10, C11, C12, C13, C15, C18, C20, C21, C22, C28, C34, C35	0.01-µF ceramic capacitor (704-5004-0001)	SMT0805 ceramic, 10%, T&R, 50 V AVX: 08055E103KATMA Digi-Key: PCC103BNCT-ND Panasonic: ECU-V1H103KBG	 Kent Milgray Marshall Future Allied Digi-Key Digi-Key
1	C32	2.2-µF tantalum capacitor (704-5070-0002)	EIA3216 (B) Tantalum, 20%, 16 V AVX: TAJA225M016 Digi-Key: PCS3225CT-ND Panasonic: ECS-T1CY225R	 Kent Milgray Marshall Future Allied Digi-Key Digi-Key
1	C33	180-pF 50-V ceramic capacitor (704-5053-0001)	SMT0805 ceramic, NPO, 5%, 50 V Digi-Key: PCC181CGCT-ND Panasonic: ECUV1H181JCG	 Digi-Key Digi-Key

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

Qty	Reference	Part Description (PII P/N)	Package Type	Disty
2	D1, D9	MBR0520 Schottky diode (716-5039-0001)	SOD-123 Schottky, 500 mA, 20 V Allied: 858-2475 Motorola: MBR0520LT1	Allied Avnet Arrow/ Schweber Future Wyle
1	D2	5.6-V Zener diode (716-5040-0001)	SMB (DO-214AA) 5 W, 5%, 865 mA5.6 V Microsemi: SMBJ5339B	Microsemi Corp.
2	D3, D6	MBRA130 high-power Schottky diode (716-5038-0001)	SMA high-power Schottky rectifier, 1 A, 30 V Motorola: MBRA130	Avnet Arrow/ Schweber Future Wyle
1	D4	Green SMT LED (721-5011-0002)	SMT green, molded reflector carrier, 25 mcd, 130-degree view, 20 mA, 1.7-mm x 2.5-mm Digi-Key: 67-1369-1-ND Lumex: SSL-LXA1725GC-TR	Digi-Key
2	D5, D8	MMBD914 high-speed switching diode (716-5035-0001)	SOT-23 high-speed switching, 200 mA, 4.0 ns, 70 V Digi-Key: FMMD914CT-ND Motorola: MMBD914 Zetex: FMMD914TA	Digi-Key Avnet Arrow/ Schweber Future Wyle
1	D7	Yellow SMT LED (721-5011-0003)	SMT yellow, molded reflector carrier, 12 mcd, 130-degree view, 20 mA, 1.7-mm x 2.5-mm Digi-Key: 67-1370-1-ND Lumex: SSL-LXA1725YC-TR	Digi-Key
1	F1	Resettable fuse, hold = 0.5 A, trip = 1.0 A (719-5004-0001)	SMT 500 mA/1.0 A, 15-V max Digi-Key: MINISMDC050CT-ND Raychem: MINISMDC050-2	Digi-Key OmniPro

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

Qty	Reference	Part Description (PII P/N)	Package Type	Disty
1	J1	20-pin machine-pin DIP socket (726-0004-0002)	DIP-20 machine pin, open frame, tin/gold Digi-Key: ED3320-ND Mill-Max: 110-93-320-41-001	Digi-Key
5	J2, J7, JP4, JP5, JP7	CON2M 2-pin male header (705-0030-0001)	0.100-TH male breakaway Molex: 22-28-4023	Arrow Kent TTI Sterling Richey-Cypress
1	J3	CON8x2M 16-pin dual-row shrouded header (705-0082-0001)	0.100 TH male, dual-row, shrouded, 30 µ-inch gold 3:M: 2516-6002UB Digi-Key: MHB16K-ND Molex: 70246-1622	Richey-Cypress Milgray Norvell Digi-Key Arrow Kent TTI Sterling Richey-Cypress
1	J4	DB9F Right angle PCB mount DB-9F (705-0011-0002)	TH female right angle PCB, grounding board locks, with jjack screws installed Keltron: DNR-09SCJB-SG Kycon: K22-E9S-NJ	OmniPro Sager High Tech Sales
1	J5	3-pin, 5.5-mm x 2.5-mm dc power jack (705-0013-0001)	TH male right angle mates 5.5-mm OD x 2.5-mm ID x 12-mm plug Cui Stack: PJ-002B Digi-Key: SC1152-ND Kycon: KLD-0202-B Switchcraft: RAPC712	Digi-Key Digi-Key Sager High Tech Sales Richey-Cypress Arrow

Table B-1. Parts List (Sheet 4 of 10)

Qty	Reference	Part Description (PII P/N)	Package Type	Disty
1	J6	28-pin machine-pin 0.600 DIP socket (726-0006-0003)	0.600 DIP-28 machine pin, open frame, tin/gold Digi-Key: ED3628-ND Mill-Max: 110-93-628-41-001	Digi-Key
3	JP1, JP2, JP6	CON3M 3-pin male header (705-0031-0001)	0.100-TH male breakaway Molex: 22-28-4033	Arrow Kent TTI Sterling Richey-Cypress
1	JP3	CON4x2M dual-row 8-pin male header (705-0083-0001)	0.100-TH male dual-row breakaway Molex: 10-89-1081	Arrow Kent TTI Sterling Richey-Cypress
1	L1	180-µH, 120-mA SMT inductor (720-5019-0001)	SMT1812 inductor, 120 mA Murata: LQH4N181K04	Future Avnet Milgray TTI Sterling Allied Pioneer
1	L2	10-µH, 100-mA SMT inductor (720-5018-0001)	SMT1206 inductor, 100 mA Murata: LQH1N100K04	Future Avnet Milgray TTI Sterling Allied Pioneer

Table B-1. Parts List (Sheet 5 of 10)

Qty	Reference	Part Description (PII P/N)	Package Type	Disty
2	Q1, Q3	MMBT3906 PNP general-purpose BJT (728-5005-0001)	SOT-23 PNP BJT, 200 – 350 mW, 200 mA, 40 V Digi-Key: FMMT3906CT-ND Digi-Key: MMBT3906DICT-ND Diodes, Inc.: MMBT3906 Motorola: MMBT3906LT1 Zetex: FMMT3906	Digi-Key Digi-Key Avnet Arrow/ Schweber Future Wyle
1	Q2	MMBT3904 NPN general-purpose BJT (728-5002-0001)	SOT-23 NPN BJT, 200 – 350 mW, 200 mA, 40 V Digi-Key: FMMT3904CT-ND Digi-Key: MMBT3904DICT-ND Diodes, Inc.: MMBT3904 Motorola: MMBT3904LT1 Zetex: FMMT3904	Digi-Key Digi-Key Avnet Arrow/ Schweber Future Wyle
2	Q4, Q5	MMBF0201 Low RDSon SMT MOSFET (728-5006-0001)	SOT-23 N-channel MOSFET, RDSon = 1.0 Ω, 225 mW, 300 mA, 20 V Motorola: MMBF0201NLT1	Avnet Arrow/ Schweber Future Wyle
14	R1, R2, R3, R4, R6, R7, R11, R12, R15, R16, R20, R27, R28, R29	10-kΩ, 5% SMT0805 resistor (733-10K0-2301)	SMT0805 thick film SMT, 5%, 1/10 W, 150 V Digi-Key: P10KACT-ND Panasonic: ERJ-6GEYJ103V	Digi-Key Digi-Key
1	R10	59.0-kΩ, 1% SMT0805 resistor (733-57K6-4301)	SMT0805 thick film SMT, 1%, 1/10 W, 150 V Digi-Key: P59.0KCCT-ND Panasonic: ERJ-6ENF5902V	Digi-Key Digi-Key

Table B-1. Parts List (Sheet 6 of 10)

Qty	Reference	Part Description (PII P/N)	Package Type	Disty
3	R13, R17, R18	3.3-kΩ, 5% SMT0805 resistor (733-3K30-2301)	SMT0805 thick film SMT, 5%, 1/10 W, 150 V Digi-Key: P3.3KACT-ND Panasonic: ERJ-6GEYJ332V	 Digi-Key Digi-Key
1	R14	560-Ω, 5% SMT0805 resistor (733-560R-2301)	SMT0805 thick film SMT, 5%, 1/10 W, 150 V Digi-Key: P560ACT-ND Panasonic: ERJ-6GEYJ561V	 Digi-Key Digi-Key
1	R19	100-kΩ, 5% SMT0805 resistor (733-100K-2301)	SMT0805 thick film SMT, 5%, 1/10 W, 150 V Digi-Key: P100KACT-ND Panasonic: ERJ-6GEYJ104V	 Digi-Key Digi-Key
3	R21, R23, R24	10.0-Ω, 1% SMT0805 resistor (733-10R0-4301)	SMT0805 thick film SMT, 1%, 1/10 W, 150 V Digi-Key: P10.0CCT-ND Panasonic: ERJ-6ENF10R0V	 Digi-Key Digi-Key
1	R22	150-Ω, 1% SMT0805 resistor (733-150R-4301)	SMT0805 thick film SMT, 1%, 1/10 W, 150 V Digi-Key: P150CCT-ND Panasonic: ERJ-6ENF1500V	 Digi-Key Digi-Key
1	R30	10-kΩ, single turn, 3/8-inch square potentiometer (708-0003-0001)	TH single-turn cermet, vertical, 10%, 1/2 W Bourns: 3386P-1-103 Digi-Key: 3386P-103-ND	 TTI Pioneer Digi-Key Digi-Key
1	R31	1.1-kΩ, 5% SMT0805 resistor (733-1K10-2301)	SMT0805 thick film SMT, 5%, 1/10 W, 150 V Digi-Key: P1.1KACT-ND Panasonic: ERJ-6GEYJ112V	 Digi-Key Digi-Key
1	R32	1.3-kΩ, 1% SMT0805 resistor (733-1K30-4301)	SMT0805 thick film SMT, 1%, 1/10 W, 150 V Digi-Key: P1.30KCCT-ND Panasonic: ERJ-6ENF1301V	 Digi-Key Digi-Key
1	R33	1.87-kΩ, 1% SMT0805 resistor (733-1K87-4301)	SMT0805 thick film SMT, 1%, 1/10 W, 150 V Digi-Key: P1.87KCCT-ND Panasonic: ERJ-6ENF1871V	 Digi-Key Digi-Key

Table B-1. Parts List (Sheet 7 of 10)

Qty	Reference	Part Description (PII P/N)	Package Type	Disty
1	R34	3.57-kΩ, 1% SMT0805 resistor (733-3K57-4301)	SMT0805 thick film SMT, 1%, 1/10 W, 150 V Digi-Key: P3.57KCCT-ND Panasonic: ERJ-6ENF3571V	 Digi-Key Digi-Key
1	R35	33-Ω, 5% SMT0805 resistor (733-33R0-2301)	SMT0805 thick film SMT, 5%, 1/10 W, 150 V Digi-Key: P33ACT-ND Panasonic: ERJ-6GEYJ330V	 Digi-Key Digi-Key
1	R36	47-Ω, 5% SMT0805 resistor (733-47R0-2301)	SMT0805 thick film SMT, 5%, 1/10 W, 150 V Digi-Key: P47ACT-ND Panasonic: ERJ-6GEYJ470V	 Digi-Key Digi-Key
1	R37	825-Ω, 1% SMT0805 resistor (733-825R-4301)	SMT0805 thick film SMT, 1%, 1/10 W, 150 V Digi-Key: P825CCT-ND Panasonic: ERJ-6ENF8250V	 Digi-Key Digi-Key
1	R38	412-Ω, 1% SMT0805 resistor (733-412R-4301)	SMT0805 thick film SMT, 1%, 1/10 W, 150 V Digi-Key: P412CCT-ND Panasonic: ERJ-6ENF4120V	 Digi-Key Digi-Key
1	R5	300-Ω, 5% SMT0805 resistor (733-300R-2301)	SMT0805 thick film SMT, 5%, 1/10 W, 150 V Digi-Key: P300ACT-ND Panasonic: ERJ-6GEYJ301V	 Digi-Key Digi-Key
3	R8, R25, R26	1.0-kΩ, 5% SMT0805 resistor (733-1K00-2301)	SMT0805 thick film SMT, 5%, 1/10 W, 150 V Digi-Key: P1.0KACT-ND Panasonic: ERJ-6GEYJ102V	 Digi-Key Digi-Key
1	R9	10.2-kΩ, 1% SMT0805 resistor (733-10K2-4301)	SMT0805 thick film SMT, 1%, 1/10 W, 150 V Digi-Key: P10.2KCCT-ND Panasonic: ERJ-6ENF1022V	 Digi-Key Digi-Key
1	SW1	process-sealed SPDT toggle switch (709-0005-0001)	0.100 TH sealed, toggle right-angle, vertical 3 A, 120 Vac or 28 Vdc C&K: ET01MD1AVQE	Altair Sterling
2	TP1, TP2	SMT test point (705-5033-0001)	SMT test point Digi-Key: 5015KCT-ND Keystone: 5015	Digi-Key

Table B-1. Parts List (Sheet 8 of 10)

Qty	Reference	Part Description (PII P/N)	Package Type	Disty
2	U1, U6	7SZ05 single open-drain inverter (701-5017-0001)	SOT23-5 tiny UHS, open-drain inverter 1.8 – 5.5 V Digi-Key: NC7SZ05M5CT-ND Fairchild: NC7SZ05	Digi-Key
1	U11	MC34063A dc-to-dc converter control circuit (714-5001-0001)	SO-8 dc-to-dc control, <40 V Motorola: MC34063A	Avnet Arrow/ Schweber Future Wyle
1	U12	74HC123 dual retriggerable monostable multivibrator (701-5048-0001)	SOIC-16 HC logic 2 – 6 V Digi-Key: MM74HC123AM-ND Fairchild: MM74HC123AM	Digi-Key
1	U13	28-pin, 0.600-wide ZIF DIP socket (726-0010-0001)	DIP-28 wide ZIF 3M Textool: 2-0028-03345-000-006-0 Digi-Key: 3M2802-ND	Tactic Digi-Key
1	U14	20-pin ZIF DIP socket (726-0011-0001)	DIP-20 ZIF 3M Textool: 2-0020-03342-000-006-0 Digi-Key: 3M2002-ND	Tactic Digi-Key
1	U15	MC34164 micropower undervoltage sensing circuit (702-5137-0001)	SO-8 undervoltage sense, 5 V Motorola: MC34164D-5	Avnet Arrow/ Schweber Future Wyle
1	U16	28-pin ZIF SOP socket (726-0013-0001)	SOP-28 ZIF Enplas: OTS-28-1.27-04	Enplas
1	U17	20-pin ZIF SOP socket (726-0012-0001)	SOP-20 ZIF Enplas: OTS-20(28)-1.27-04	Enplas
1	U19	LT1086 1.5-A low-dropout positive regulator (702-5135-0001)	DD-3 voltage regulator, <25 V Digi-Key: LT1086CM-ND Linear Technology: LT1086CM	Digi-Key Arrow/ Schweber Marshall Digi-Key
2	U2, U10	7SZ125 buffer with 3-state output (701-5047-0001)	SOT23-5 tiny UHS, 3 V Digi-Key: NC7SZ125M5CT-ND Fairchild: NC7SZ125	Digi-Key

Table B-1. Parts List (Sheet 9 of 10)

Qty	Reference	Part Description (PII P/N)	Package Type	Disty
1	U3	LTC1383 low-power RS-232 transceiver (702-5136-0001)	SOIC-16 RS-232 transceiver, 5 V Digi-Key: LTC1383CS-ND Linear Technology: LTC1383CS	Digi-Key Arrow/ Schweber Marshall Digi-Key
1	U4	ADG774 wide-bandwidth quad 2:1 multiplexor (702-5138-0001)	SOIC-16 quad 2:1 multiplexor, 3 V/5 V Analog Devices: ADG774BR	Avnet Future
1	U5	7SZ32 single 2-input OR gate (701-5018-0001)	SOT23-5 tiny UHS, 2-input OR, 1.8 – 5.5 V Digi-Key: NC7SZ32M5CT-ND Fairchild: NC7SZ32	Digi-Key
3	U7, U9, U18	7SZ04 single inverter (701-5016-0001)	SOT23-5 tiny UHS, inverter, 1.8 – 5.5 V Digi-Key: NC7SZ04M5CT-ND Fairchild: NC7SZ04	Digi-Key
1	U8	7SZ38 single open-drain 2-input NAND gate (701-5023-0001)	SOT23-5 tiny UHS, open drain 2-input NAND, 1.8 – 5.5 V Digi-Key: NC7SZ38M5CT-ND Fairchild: NC7SZ38	Digi-Key
1	Y1	14-pin machine-pin DIP socket (726-0002-0003)	DIP-14 machine pin, open frame tin/gold Digi-Key: ED3314-ND Mill-Max: 110-93-314-41-001	Digi-Key
4	Z1, Z2, Z3, Z4	Rubber button bumper (301-0000-0006)	9/32-inch hole, rubber, non-adhesive, rounded Allied: 217-4076 Russell Industries: BUT-4076	Allied

Table B-1. Parts List (Sheet 10 of 10)

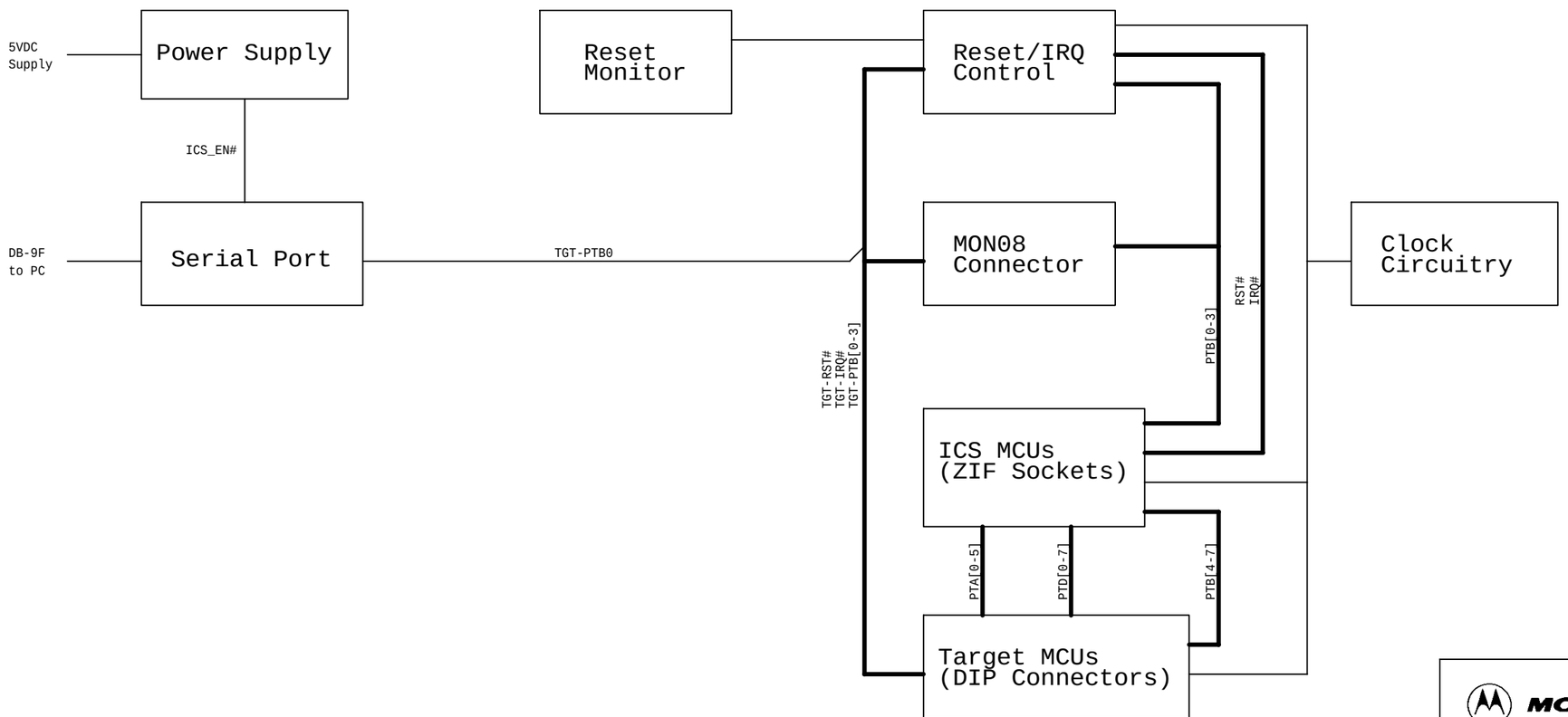
Qty	Reference	Part Description (PII P/N)	Package Type	Disty
1	Z10	5-V regulated universal power supply (730-9002-0001)	Wall-center-mount, c/w adapters, universal, regulated, 2.5 ID x 5.5 OD x 12 mm dc plug Golden Pacific Electronics: PS-5024A-PL06/ Sceptre: PS-5024A-PL06/S1	
7	Z5, Z6, Z7, Z8, Z9, Z11, Z12	SHUNT2 blue jumper shunt (705-9003-0001)	0.100 open top, blue, 10-µ–inch gold 3M: 929955-06 Digi-Key: 929955-06-ND	Richey-Cypress Milgray Norvell Digi-Key
1	ZY1	9.8304 MHz oscillator (713-0015-0003)	DIP-8/4 oscillator, 50 ppm, 9.8304 MHz, 5 V Digi-Key: XC266-ND ECS: OECS-2200B-098	Digi-Key Digi-Key

REVISIONS

ZONE	REV	DESCRIPTION	DATE	APPROVED
	0	Original Revision		

NOTES:

1. UNLESS OTHERWISE SPECIFIED:
   RESISTANCE VALUES ARE IN OHMS.
   RESISTORS ARE 1/10 WATT, 5%.
   CAPACITANCE VALUES ARE IN MICROFARADS.
2. INTERRUPTED LINES CODED 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 SIGNALS.
5. INTERPRET DIAGRAM IN ACCORDANCE WITH AMERICAN NATIONAL STANDARDS INSTITUTE SPECIFICATIONS, CURRENT REVISION, WITH THE EXCEPTION OF LOGIC BLOCK SYMBOLOGY.



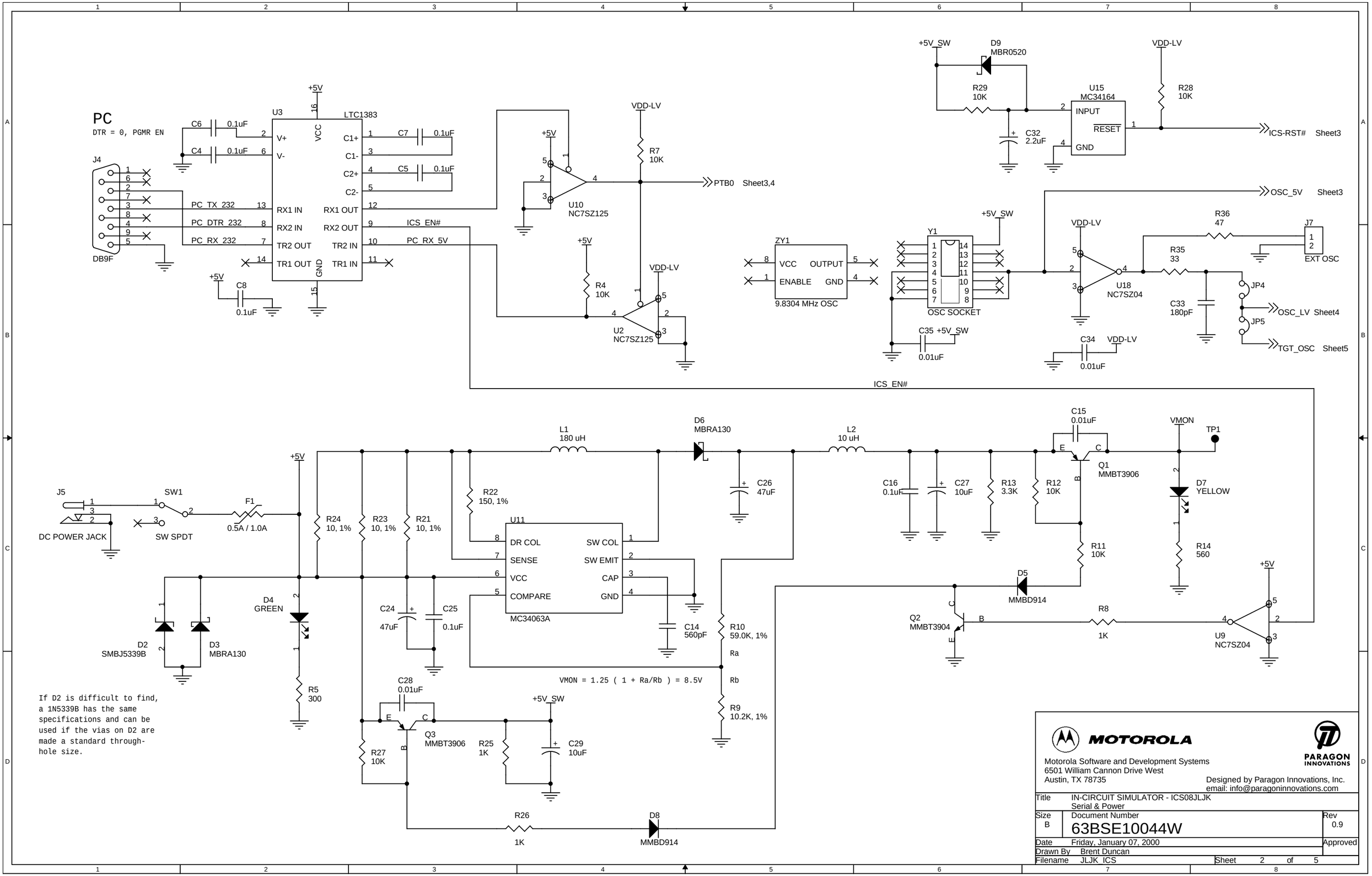
MOTOROLA

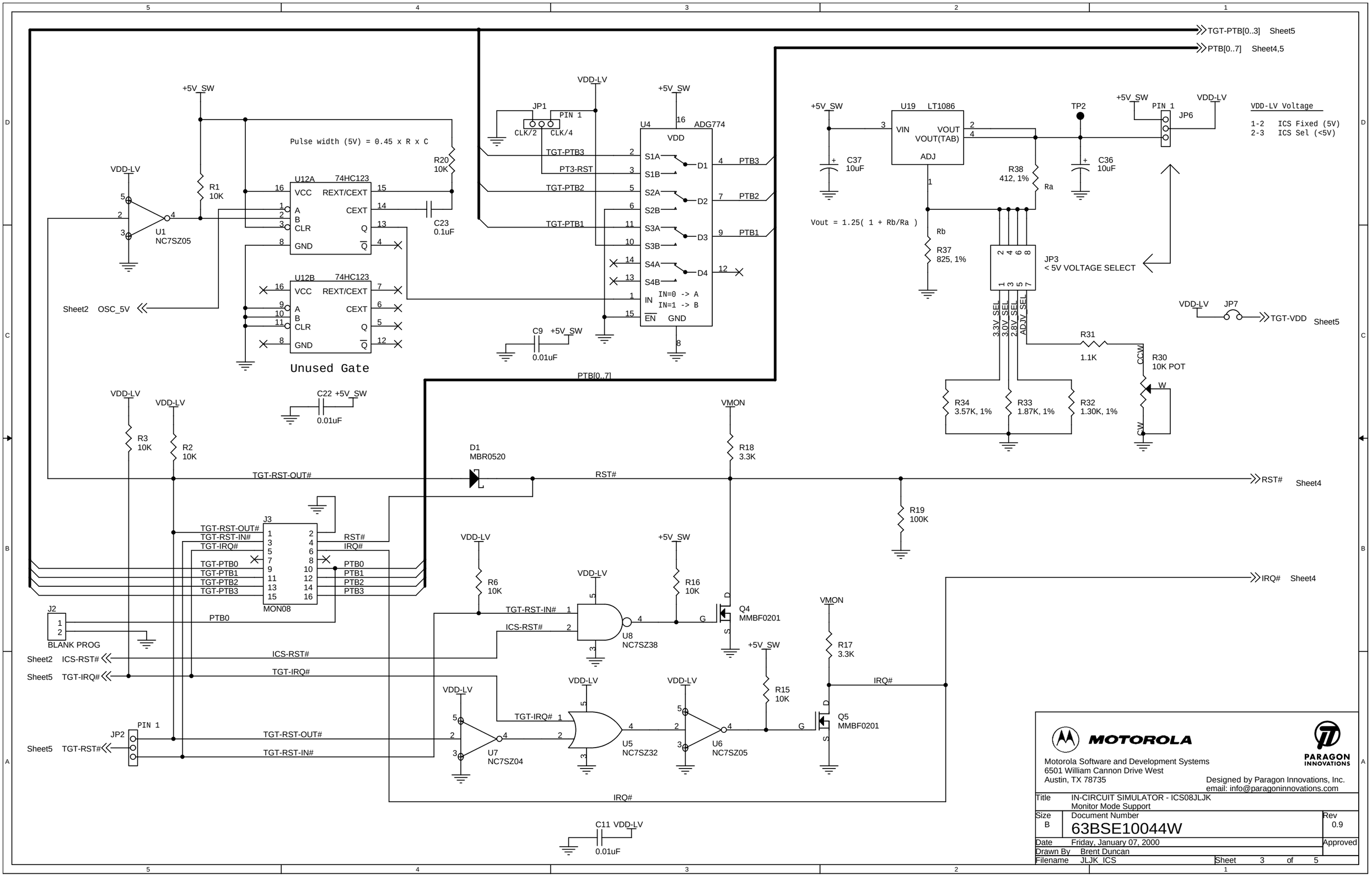
PARAGON INNOVATIONS

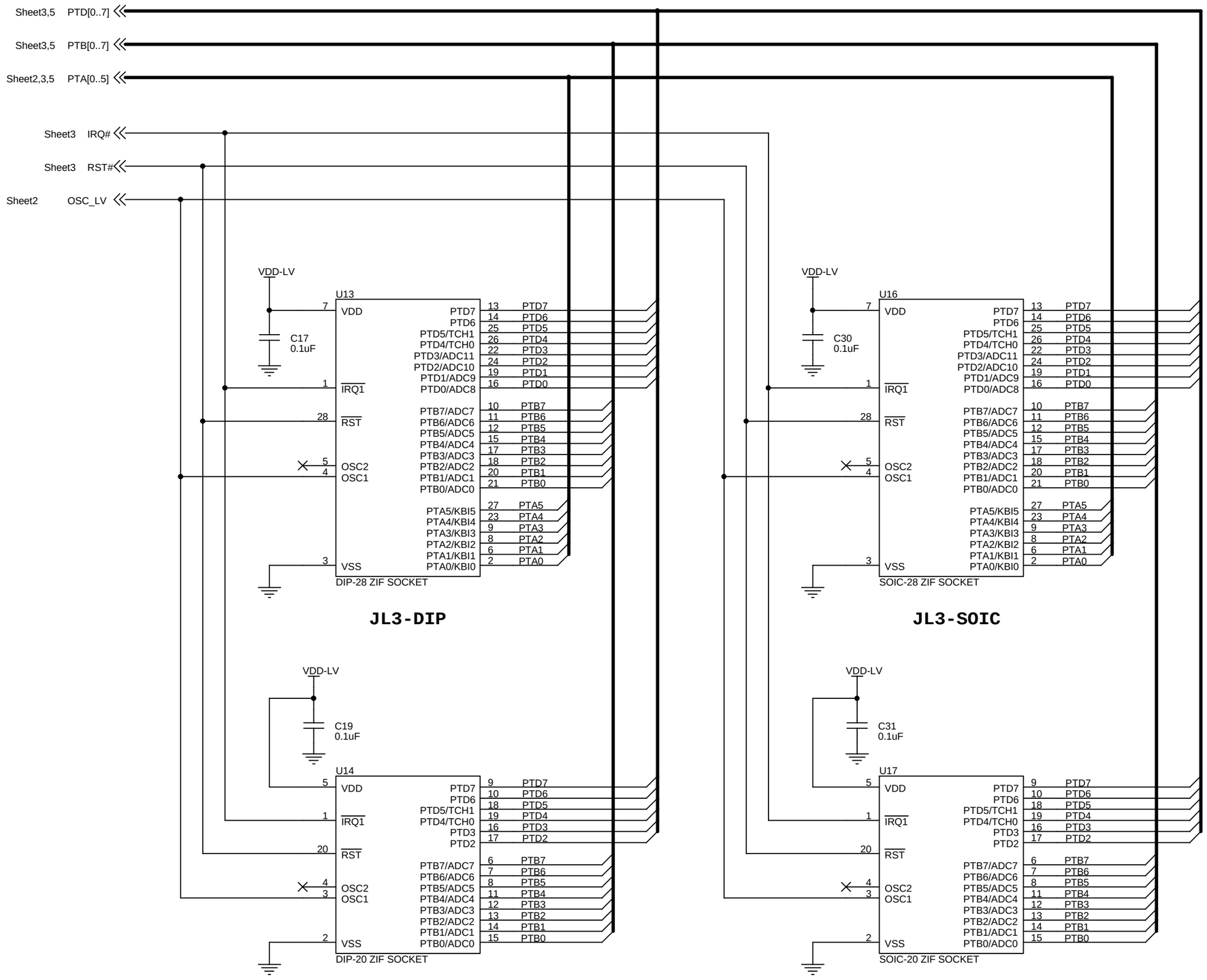
Motorola Software and Development Systems
6501 William Cannon Drive West
Austin, TX 78735

Designed by Paragon Innovations, Inc.
email: info@paragoninnovations.com

Title	IN-CIRCUIT SIMULATOR - ICS08JLJK Block Diagram	
Size B	Document Number 63BSE10044W	Rev 0.9
Date	Friday, January 07, 2000	Approved
Drawn By	Brent Duncan	
Filename	JLJK_ICS	Sheet 1 of 5


PC
DTR = 0, PGMR EN
J4
DB9F
PC_TX_232
PC_DTR_232
PC_RX_232
U3
LTC1383
+5V
VCC
V+
V-
C1+
C1-
C2+
C2-
RX1 IN
RX2 IN
TR2 OUT
TR1 OUT
GND
RX1 OUT
RX2 OUT
TR2 IN
TR1 IN
C6 0.1uF
C4 0.1uF
C7 0.1uF
C5 0.1uF
C8 0.1uF
ICS_EN#
PC_RX_5V
U10
NC7SZ125
VDD-LV
R7 10K
PTB0 Sheet3,4
R4 10K
U2
NC7SZ125
+5V_SW
D9
MBR0520
R29 10K
C32 2.2uF
U15
MC34164
INPUT
RESET
GND
R28 10K
ICS-RST# Sheet3
OSC_5V Sheet3
ZY1
VCC
OUTPUT
ENABLE
GND
9.8304 MHz OSC
Y1
OSC SOCKET
C35 +5V_SW
0.01uF
U18
NC7SZ04
R36 47
J7
EXT OSC
R35 33
C33 180pF
JP4
JP5
OSC_LV Sheet4
TGT_OSC Sheet5
C34 VDD-LV
0.01uF
ICS_EN#
L1
180 uH
D6
MBRA130
L2
10 uH
C15 0.01uF
VMON
TP1
Q1
MMBT3906
C16 0.1uF
C27 10uF
R13 3.3K
R12 10K
D7
YELLOW
R14 560
R11 10K
D5
MMBD914
Q2
MMBT3904
R8 1K
U9
NC7SZ04
J5
DC POWER JACK
SW1
SW SPDT
F1
0.5A / 1.0A
R24 10, 1%
R23 10, 1%
R21 10, 1%
R22 150, 1%
U11
DR COL
SENSE
VCC
COMPARE
SW COL
SW EMIT
CAP
GND
MC34063A
C26 47uF
C24 47uF
C25 0.1uF
C14 560pF
R10 59.0K, 1%
Ra
Rb
R9 10.2K, 1%
VMON = 1.25 (1 + Ra/Rb) = 8.5V
D2
SMBJ5339B
D3
MBRA130
D4
GREEN
R5 300
C28 0.01uF
Q3
MMBT3906
R27 10K
R25 1K
C29 10uF
R26 1K
D8
MMBD914
If D2 is difficult to find, a 1N5339B has the same specifications and can be used if the vias on D2 are made a standard through-hole size.
MOTOROLA
Motorola Software and Development Systems
6501 William Cannon Drive West
Austin, TX 78735
PARAGON INNOVATIONS
Designed by Paragon Innovations, Inc.
email: info@paragoninnovations.com
Title IN-CIRCUIT SIMULATOR - ICS08JLJK
Serial & Power
Size B
Document Number 63BSE10044W
Rev 0.9
Date Friday, January 07, 2000
Approved
Drawn By Brent Duncan
Filename JLJK_ICS
Sheet 2 of 5


TGT-PTB[0..3] Sheet5
PTB[0..7] Sheet4,5
+5V_SW
VDD-LV
Pulse width (5V) = 0.45 x R x C
R20 10K
R1 10K
U12A 74HC123
VCC REXT/CEXT
A CEXT
B Q
CLR
GND Q
C23 0.1uF
U1 NC7SZ05
U12B 74HC123
Sheet2 OSC_5V
Unused Gate
JP1
PIN 1
CLK/2 CLK/4
U4 ADG774
VDD
TGT-PTB3
PT3-RST
TGT-PTB2
TGT-PTB1
S1A S1B S2A S2B S3A S3B S4A S4B
D1 D2 D3 D4
PTB3 PTB2 PTB1
IN
IN=0 -> A
IN=1 -> B
EN GND
C9 0.01uF
PTB[0..7]
U19 LT1086
VIN VOUT
VOUT(TAB)
ADJ
C37 10uF
C36 10uF
TP2
JP6
VDD-LV Voltage
1-2 ICS Fixed (5V)
2-3 ICS Sel (<5V)
R38 412, 1%
Ra
Vout = 1.25(1 + Rb/Ra)
Rb
R37 825, 1%
JP3
< 5V VOLTAGE SELECT
3.3V SEL
3.0V SEL
2.8V SEL
ADJV SEL
JP7
TGT-VDD Sheet5
R31 1.1K
R30 10K POT
CCW
CW
W
R34 3.57K, 1%
R33 1.87K, 1%
R32 1.30K, 1%
C22 0.01uF
R3 10K
R2 10K
VMON
D1 MBR0520
R18 3.3K
TGT-RST-OUT#
RST#
RST# Sheet4
R19 100K
J3
MON08
TGT-RST-IN#
TGT-IRQ#
IRQ#
TGT-PTB0
PTB0
PTB1
PTB2
PTB3
IRQ# Sheet4
R6 10K
R16 10K
Q4 MMBF0201
U8 NC7SZ38
ICS-RST#
J2
BLANK PROG
Sheet2 ICS-RST#
Sheet5 TGT-IRQ#
R17 3.3K
R15 10K
Q5 MMBF0201
U7 NC7SZ04
U5 NC7SZ32
U6 NC7SZ05
JP2
Sheet5 TGT-RST#
C11 0.01uF
MOTOROLA
PARAGON INNOVATIONS
Motorola Software and Development Systems
6501 William Cannon Drive West
Austin, TX 78735
Designed by Paragon Innovations, Inc.
email: info@paragoninnovations.com
Title IN-CIRCUIT SIMULATOR - ICS08JLJK
Monitor Mode Support
Size B
Document Number 63BSE10044W
Rev 0.9
Date Friday, January 07, 2000
Approved
Drawn By Brent Duncan
Filename JLJK_ICS
Sheet 3 of 5



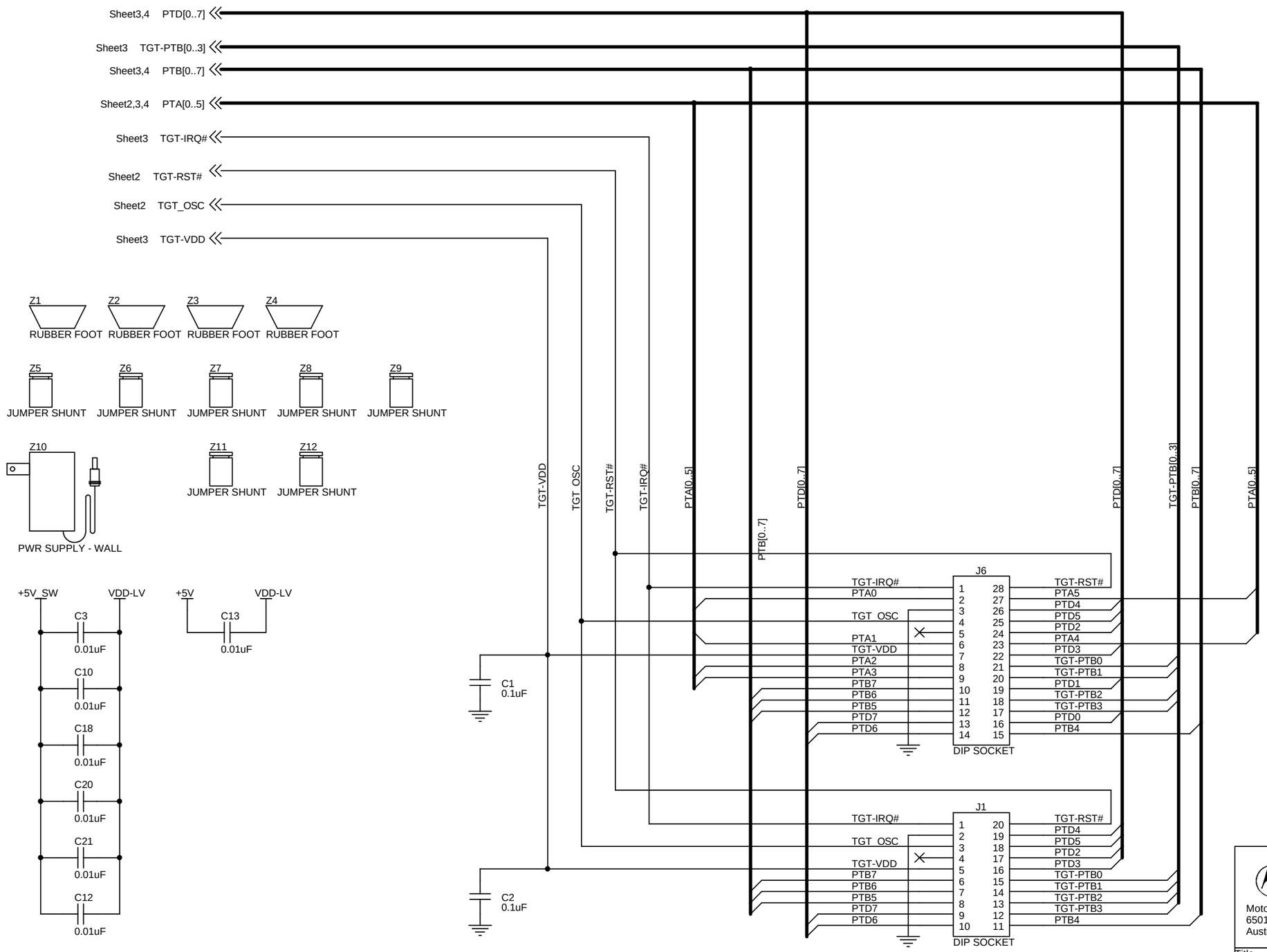
MOTOROLA

Motorola Software and Development Systems
6501 William Cannon Drive West
Austin, TX 78735

PARAGON INNOVATIONS

Designed by Paragon Innovations, Inc.
email: info@paragoninnovations.com

Title	IN-CIRCUIT SIMULATOR - ICS08JLJK MCU Sockets	
Size B	Document Number 63BSE10044W	Rev 0.9
Date	Friday, January 07, 2000	Approved
Drawn By	Brent Duncan	
Filename	JLJK_ICS	Sheet 4 of 5



MOTOROLA

PARAGON INNOVATIONS

Motorola Software and Development Systems
6501 William Cannon Drive West
Austin, TX 78735

Designed by Paragon Innovations, Inc.
email: info@paragoninnovations.com

Title	IN-CIRCUIT SIMULATOR - ICS08JLJK Target MCU Conn	
Size B	Document Number 63BSE10044W	Rev 0.9
Date	Friday, January 07, 2000	Approved
Drawn By	Brent Duncan	
Filename	JLJK_ICS	Sheet 5 of 5

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 HC08 MCU.

accumulator — An 8-bit register of the HC08 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 HC08. 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 HC08. 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 HC08, 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 HC08. 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 HC08.

index register — An 8-bit CPU register in the HC08 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 HC08. 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 HC08 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 HC08.

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 HC08 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 HC08 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.

$\mathbf{V}_{DD}$ — The positive power supply to a microcontroller (typically 5 volts dc).

$\mathbf{V}_{SS}$ — 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 HC08.

Z — Abbreviation for zero, a bit in the condition code register of the HC08. 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

A

ASCII
characters 54
assembler
description. 16

B

baud rates 22
blank programming header, J2 . . . . . 30, 47
board layout 20
bus frequency 22
bus frequency selection. 22

C

CASM08Z. 16
description. 16
checksum. 49
communications 22
communications, board-to-host. 17
connectors
target 37, 38

D

debuggers
ICD08SZ
description. 17
MON08 interface 43
drive space 17

E

electrostatic discharge 55
examples
    S records 52

F

features 16
FLASH memory 14, 16, 33, 63, 64
function keys 17

H

hardware
    installation 19
    requirements 17
    specifications 18
hotkeys. See function keys
humidity 18

I

I/O 14
ICD08SZ
    debugger 15
    description 16
ICS PWR LED 33
ICS08JLJK software 15
ICS08JLZ
    simulator 15
integrated development environment 16
interactive mode 20

J

J2 . . . . . 30, 47
JLJKICS
    description 16
    limitations 22
    software 15
JLJKICS limitations 22

L

LED
ICS power 33, 64
M68ICS08JLJK power 59
MCU power 59, 61, 64
limitations
JLJKICS 22

M

M68ICS08JLJK board 15, 19, 32, 33, 57, 59, 63, 67
board layout 20
components 56–57
M68ICS08JLJK in-circuit simulator
components 14
description. 13
features 16
hardware 16
hardware specifications. 18
MC68HC908JLJK MCU 15, 35
MCU 13, 35
MCU subsystem
clocks 25
DIP emulation cable 28
host system connector. 31
level translation. 36
memory
address. 49
system requirements 17
MON08 . . . . . 15, 56, 57, 63, 65, 66

O

operating systems 17

P

port B0 22
ports, serial 17
power
connector. 32

LEDs
ICS PWR LED 64
M68ICS08JLJK PWR LED 59
MCU PWR LED 59, 61, 64
requirements 18
power connector 32
PROG08SZ
description 16

R

RAM 17
record length 49
record type 49
relative humidity 18
requirements
hardware 17
host computer 17
operating system 17
software 17

S

S records 49–54
serial port
connector 32
use 17
software
ICS08JLJK (or JLJKICS software) 16
JLJKICS (or ICS08JLJK software) 16
requirements 17
specifications, hardware 18
S-record
content 49
creating 52
field contents 50
fields 50
overview 49
S0 record 52
S1 record 53
S9 record 54
termination record 54
types 51
stand-alone mode 20

T

target system
cables. 56
connecting to 14, 33, 43, 57
connectors 36–41, 43
description. 14
MCU 56, 57
MON08 interface 43, 63
power source 62
temperature
operating 18
storage 18

W

Windows 95 14
Windows 98 14
WinIDE 16
description. 16

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.

MOTOROLA

© Motorola, Inc., 2000

M68ICS08JLJKUM/D